Layered Object Detection for Multi-Class Segmentation

Yi Yang Sam Hallman Deva Ramanan Charless Fowlkes
Dept. of Computer Science, University of California, Irvine
{yangyi,shallman,dramanan,fowlkes}@ics.uci.edu

Abstract

We formulate a layered model for object detection and multi-class segmentation. Our system uses the output of a bank of object detectors in order to define shape priors for support masks and then estimates appearance, depth ordering and labeling of pixels in the image. We train our system on the PASCAL segmentation challenge dataset and show good test results with state of the art performance in several categories including segmenting humans.

1. Introduction

Object detection is a fundamental task in computer vision. Most approaches formulate the problem as that of predicting a bounding box enclosing the object of interest - this is, for example, the evaluation criteria in the popular PASCAL Visual Object Recognition Challenge (VOC) [3]. However, a bounding-box output is clearly limited. For many objects, particularly those with complex, articulated shapes, a bounding box provides a poor description of the support of the object in the image. At the other extreme, one can attempt to produce an object class label for every pixel in the image. This is usually termed multi-class segmentation.

Object detection and multi-class segmentation have typically been approached as separate problems and tackled using substantially different techniques. Candidate bounding boxes are often generated using a scanning window approach and scored using a classifier trained on positive and negative examples[24, 2, 4]. In contrast, multi-class segmentation models have largely been built on top of Markov Random Field (MRF) models which enforce smoothness across pixel labels [6, 21, 13, 20, 22].

We posit that these two problems should be addressed jointly. Per-pixel labels in multi-class segmentation should benefit from highly discriminative template-based object detectors. Similarly, object detections should be consistent with some underlying segmentation of the image. In combining these approaches, two important issues arise:

1. Multi-class, multi-object pixel labeling benchmarks pose interesting challenges that are not captured by bounding-box detection benchmarks.

2. Producing per-pixel labels points to the need for a layered representation to reconcile overlapping detections.

Representing detections as a multi-class segmentation rather than a collection of bounding boxes has a significant impact on the way detection performance is measured. In some sense, multi-class segmentation benchmarks are more difficult than bounding box benchmarks since the latter allows for pixels to belong to multiple overlapping object instances, while the former forces pixels to have a single class or instance label. Suppose, for example, that cat and dog detectors tend to respond to similar image regions. One could artificially increase the recall rate on standard detection benchmarks by reporting overlapping cat and dog detections even though such a configuration may never exist. On the other hand, a segmentation output is necessarily *self-consistent* since each pixel may only take on a single label.

A simple approach for generating self-consistent detections is to paste box-shaped masks on top of one another to create a multi-class segmentation. This is the default procedure used by PASCAL to automatically generate segmentations from object detection systems [3]. Such a segmentation, though, depends on the details of how the detectors are calibrated. Even a good detector will produce poor segmentations if the detection threshold is set too low or too high.

Performance also depends crucially on what order detections are "pasted down". If boxes are composited in order of detector score, then it is essential that the detectors for different classes are calibrated relative to each other. Explicitly representing object detection with a layered representation not only captures this ordering but can also be advantageous in guiding more precise segmentation. Layering allows for one to link disjoint object segments separated by an occluder (Fig.1) based on estimating the layer appearance (e.g. color and texture).

In this paper, we describe a simple probabilistic model that captures the shape, appearance and depth ordering of a

Figure 1. Our framework. Multi-class object detections algorithms typically predict bounding box locations and class labels (**left**). Such reports only provide relatively coarse reports about objects. Multi-class segmentation algorithms provide class labels for every pixel (**center**). We propose to use object-specific bounding box representations to guide multi-class segmentation algorithms. To do so, we introduce layered representations (**right**) that reason about relative depth orderings of objects. Layered representations are helpful, for example, to link object segments separated by an occluder (e.g., the center person occluded by the horse).

collection of detections within an image. It explicitly represents the shape of a detected object in terms of a layered, per-pixel segmentation. This shape estimate is driven by a novel deformable spatial prior for object shape that adapts to particular instances based on the response of a mixture of part-based detectors. Given an ordering of layers, these object detections are composited to yield a generative "explanation" of the image in terms of a multi-class segmentation.

After a brief discussion of related work, we describe this layered representation in detail in Section 3, discuss how to perform inference in Section 4 and how parameters are learned from training data in Section 5. We then show experimental results on the 2009 PASCAL Segmentation challenge which demonstrate that the proposed method achieves state-of-the art results in many categories.

2. Related work

The reconciliation of recognition and segmentation has been an active area of research. Early approaches bias a segmentation engine using the output of object models [27, 17], while others iterate between bottom-up and top-down cues [14, 15, 23]. Our work is most similar to the ObjCut framework [11] that uses a part-based model to bias a bottom-up grouping process. Our work differs in that we focus on segmenting images of multiple instances of multiple types of objects.

Our work is also inspired by image representations that reason about occlusion through the use of "2.5D" models. Such approaches are typically explored in the video domain and include examples such as layered motion models [25], video sprites [10], and layered pictorial structures [12]. In the image domain, such layered approaches are less common but have been explored in e.g., [16, 5]. One approach of capturing rough depth ordering is to estimate occlusion boundaries, typically done in a Markov Random Field framework [26, 8, 9]. At the other extreme, one can attempt to recover full three-dimensional models from single images as in [9, 19]. Our goal is to explore more intermediate three-dimensional representations based on relative depth orderings of object segmentations.

3. Model

We now describe our layered generative model for object detection.

Detections: For a particular image, let d_n encode the class, score, and bounding box coordinates of the n^{th} detection, where $1 \leq n \leq N$. We assume that the detectors have been calibrated on training data so that detections across classes have comparable scores and thresholding scores at 0 yields an appropriate number of detections on average (we describe details of this calibration in the experimental results section).

Importantly, we model each detection in "2.5D" and order them from back to front with some permutation π so that $d_{\pi(N)}$ is the front-most detection, $d_{\pi(N-1)}$ is the second, etc. We define $d_{\pi(0)}$ to be a default background detection associated with a background layer that is included for all images. Let θ_n be the parameters of the appearance model associated with the n^{th} detection. We will model appearance with a color histogram.

Pixel Labels: Let x_i be the feature value associated with the i^{th} pixel. Because there is a one-to-one correspondence between a detection and a layer, we write $z_i \in \{0 \dots N\}$ for a label that simultaneously specify both the layer and detection associated with pixel i. Each layer also has its own binary segmentation mask denoted by $b_{in} \in \{0, 1\}$, where we define $b_{i0} = 1$. Note that a pixel i may belong to multiple segmentation masks but can only have one final object label (e.g., both b_{in} and b_{im} are 1 but due to occlusion, either $z_i = n$ or $z_i = m$)

Joint model: By convention, we use the lack of subscript to denote the set obtained by including all instances of the omitted subscript - e.g., $b_i = \{b_{i0} \dots b_{iN}\}$. Our first assumption is that, given the set of ordered detections d and

appearance models θ, the joint probability of pixel features x and labels z factors into a product over pixels:

$$P(z, x|\theta, d_\pi) = \prod_i P(z_i, x_i|\theta, d_\pi), \quad (1)$$

The model for each pixel can be further factored:

$$P(z_i = n, x_i|\theta, d_\pi) = P(z_i = n|d_\pi)P(x_i|\theta_n), \quad (2)$$

where $n \in \{0 \dots N\}$. The second term is a standard "likelihood" model that scores pixel x_i under the appearance model for detection n. The first term is a distribution over labels induced by the detections.

3.1. Layered Label Distributions

We obtain the distribution over labels by integrating over all layered binary segmentations:

$$P(z_i = m|d_\pi) = \sum_{b_i} P(z_i = m|b_i)p(b_i|d_\pi) \quad (3)$$

$$\text{where} \quad P(z_i = m|b_i) = b_{im} \prod_{n=m+1}^{N} (1 - b_{in}), \quad (4)$$

$$\text{and} \quad P(b_i|d_\pi) = \prod_{n=0}^{N} P(b_{in}|d_{\pi(n)}). \quad (5)$$

We define $b_{i0} = 1$. This defaults all pixels to be labeled as background when not explicitly covered by a detection. Combining the previous three equations, and noting that the occlusion model from (4) only requires us to integrate (3) over binary segmentations in layers in front of m, we obtain the simplified expression:

$$P(z_i = m|d_\pi) = \beta_{im} \prod_{n=m+1}^{N} (1 - \beta_{in}) \quad (6)$$

$$\text{where} \quad \beta_{in} = P(b_{in} = 1|d_{\pi(n)}) \quad (7)$$

3.2. Shape prior

In this section, we consider different models for the shape prior β_{in}. Arguably the simplest notion of a shape prior is to define a "soft" mask or alpha-matte which records the probability of a pixel at some location relative to the center of the detection belonging to the object.

Let k_n as the class label of the n^{th} detection and $i' = \mathcal{T}_n(i)$ be the index of a pixel i which has been mapped by some transformation $\mathcal{T}_n$ (e.g., translation and scaling) into the coordinate system of the n^{th} detection. We can then specify a per-class shape prior by:

$$\beta_{in} = \alpha_{i',k_n} \quad (8)$$

We visualize examples of such priors in in Fig.2.

Object Pose: Local detectors based on mixture models return a mixture component label l_n for each detection. This label often captures the pose of an object - e.g., side versus frontal cars. It is natural to define a shape prior for each discrete pose as:

$$\beta_{in} = \alpha_{i',k_n,l_n} \quad (9)$$

Part Pose: Finally, part-based detectors also return a vector of part locations $\{p_1 \dots p_T\}$ for each detection. Assuming that parts are layered in depth, we can derive a model similar to Sec.3.1 that composites the contributions of overlapping parts onto one another. Part t will contribute the labeling of a pixel so long as the $T - t$ parts in front do not account for that pixel:

$$\beta_{in} = \sum_{t=1}^{T} \alpha_{i',k_n,p_{tn}} \prod_{s=t+1}^{T} (1 - \alpha_{i',k_n,p_{sn}}) \quad (10)$$

where $i' = \mathcal{T}_{tn}(i)$, the location of pixel i in the coordinate system of the t^{th} part in detection n. One can also define a shape prior from a mixture of part models by adding an additional mixture index l_n to (10).

3.3. Order prior

The previous model is conditioned on the ordering $d = \{d_{\pi(0)} \dots d_{\pi(N)}\}$. To examine different orderings, it will be useful to model π as a random variable by writing:

$$P(x, z, \pi|d, \theta) = P(x, z|\pi, d, \theta)P(\pi|d) \quad (11)$$

The first term is on the right equivalent to (1). The second term is a prior over orderings of detections. One choice would be an uninformative prior – we may not favor one depth ordering over another. However, it is reasonable to assume that most local object models produce higher scores on unoccluded instances compared to occluded instances. This assumption suggests that one should favor depth orderings that place high scoring objects in front of lower scoring objects. A second feature which is useful in ordering detections is that when multiple objects rest on a ground-plane, the object whose bottom edge is lower in the image is typically closer to the camera.

Writing the score for detection n as s_n and the coordinate as y_n, we define a conditional Markov Random Field (MRF) prior on permutations by:

$$P(\pi|d) = \frac{1}{Z(d)} \prod_{m<n} e^{-\phi(d_{\pi(m)}, d_{\pi(n)})}$$

$$\text{with} \quad \phi(d_{\pi(m)}, d_{\pi(n)}) = w_s \mathbf{1}_{[s_{\pi(m)} < s_{\pi(n)}]} + w_y \mathbf{1}_{[y_{\pi(m)} < y_{\pi(n)}]}$$

where $\mathbf{1}$ is the indicator function, w_s and w_y are model parameters, and Z is a normalizing constant.

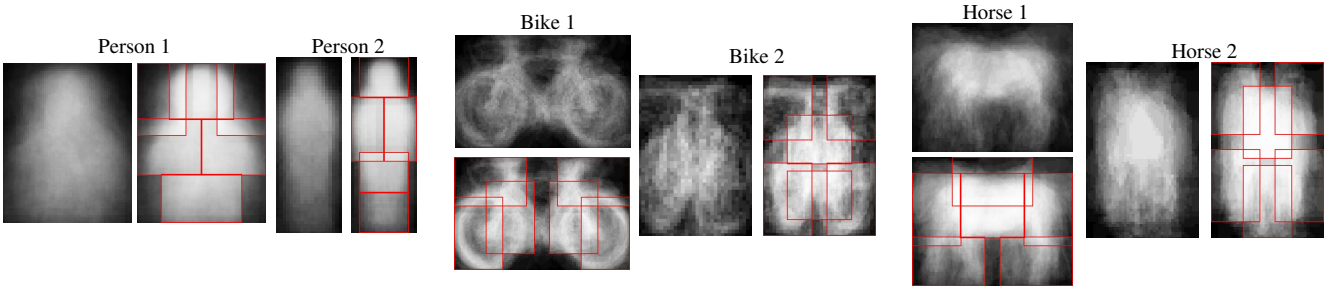

Figure 2. We show examples of our shape prior α. Shape priors are represented as soft segmentation masks. We show priors derived from a mixture model of deformable parts including both "root" and "part" templates. Note that the mixture models capture shapes corresponding to different aspects, and part-based shape models tend to be more peaked than the root. For example, the horse's legs are blurred out in the first mixture component, but are visible in the composited part model. This is because part models are learned from deformable part annotations, while root shape models are learned from rigid bounding boxes.

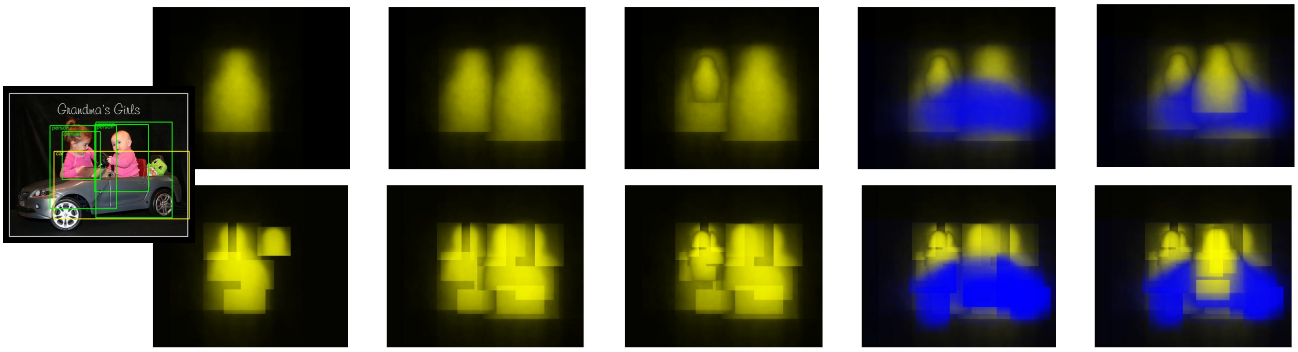

Figure 3. We show examples of our order-dependant layered label distribution $P(z|d_\pi)$. We show the original image with overlaid candidate detections on the **left**. In the **top** row, we show the composited layered distribution iteratively built from detections ordered back to front. For visualization purposes, we color distributions according to the object type. In the **bottom** row, we show the distribution built from part-based detections which deform to better match the shape of the detected instance. In general the part composites are more accurate. For example, the front car wheel is better modeled with parts. One notable exception is the mislocalized head in the first instanced person.

3.4. Exploiting Bottom-up Grouping

The model as described makes no use direct use of bottom-up grouping constraints such as the presence of contours separating object boundaries. A simple way to incorporate such information is to utilize a segmentation engine which generates superpixels (we use [1]) and assign superpixels to layers instead of pixels. In this case, we can use the same notation but let i index a superpixel instead of a pixel. For example, z_i will indicate the label of superpixel i and x_i a feature vector (e.g. color distribution) extracted from i. Since superpixels are image dependent, we still maintain a per-pixel alpha-matte which we use to define a distribution over z_i:

$$P(z_i = m|d_\pi) \propto \prod_{j \in \mathcal{S}_i} \beta_{jm} \prod_{n=m+1}^{N} (1 - \beta_{jn}) \qquad (12)$$

The superpixel-constrained label distribution is equivalent to the label distribution from Sec.3.1 conditioned on the fact that groups of pixels in the same superpixel must share the same label. This conditioning requires the use of a proportionality sign in (12) to ensure that the left-hand side is a proper probability distribution.

Figure 4. We show an example superpixel grouping from [1] tuned to return roughly 200 superpixels. We use this bottom up information in our probabilistic model by conditioning on the fact that all pixel labels from within a superpixel must share be identical.

4. Inference

Given an image and a set of detections, we would like to infer the class labels for each pixel z_i. Ideally, one would like to estimate the binary labels z by marginalizing out over the color models θ. Exact inference is difficult because θ is continuous and the induced joint potential between θ and z

is non-Gaussian. Furthermore, there may be large cliques in the associated junction tree due to multiple overlapping detections, making it difficult to explicitly discretize and search over θ. Instead, it is natural to pursue approximate the distribution over θ by its MAP value using coordinate descent or the Expectation-Maximization (EM) algorithm. We first consider inference for the simpler case of a fixed ordering of detections.

4.1. Coordinate ascent

We outline here a coordinate ascent algorithm for maximizing (1) by iterating between updates for z and c:

$$1.\ \max_{z} P(x, z|\theta, d_\pi) \quad \text{by computing} \quad \max_{z_i} P(x_i, z_i|\theta, d_\pi)$$

$$2.\ \max_{\theta} P(x, z|\theta, d_\pi) \quad \text{by computing} \quad \max_{\theta_n} \prod_{i:z_i=n} P(x_i|\theta_n)$$

Step 1 is performed by computing $P(z_i = n, x_i|\theta, d)$ for each pixel i and possible label n. Step 2 corresponds to standard maximum likelihood estimation (MLE). In our case, we use color histogram models and so use frequency counts to estimate θ_n. One could also define a EM algorithm that learns histograms using weighted MLE, where the weight of pixel i for histogram θ_n is given by $P(z_i = n|x, \theta, d)$. We found simple coordinate ascent to work well.

4.2. Orderings

The previous sections assumed that the ordering was fixed. We would now also like to optimize over the ordering as well:

$$\max_{z,\theta,\pi} P(x, z|\theta, d, \pi)P(\pi|d) = \max_{\pi} P(\pi|d) \max_{z,\theta} P(x, z|\theta, d, \pi)$$

For each ordering π, we can compute the inner maximization by coordinate ascent as previously described or utilize a soft formulation that replaces the inner maximization with EM.

Because the number of detections in an image is often small, it is often practical to perform a brute force search over orderings. The search space for the maximization over π can be further restricted by noting it is only necessary to enumerate those orderings that generate distinct label priors $P(z|d)$. If an image only contains two detections that do not overlap, then either order generates the same label prior. A simple method for exploiting this observation is to construct a $N \times N$ adjacency graph of overlapping detections, and ignore the relative ordering between different connected components.

5. Learning

In Section 3.2 we describe a MLE procedure for learning the shape priors α from labeled training data. For simplicity, we include equations for estimating $\alpha_{i'k}$ from a single training image but the extension to multiple images and pose/part-based shape priors are straightforward.

Bernoulli models: First consider the fully observed case in which layered segmentation masks b_{in} are given. Learning corresponds to standard Bernoulli MLE:

$$\alpha_{i',k} = \operatorname{argmax} \sum_n \log P(b_{in}|d) \quad \text{where} \quad i = T_n^{-1}(i')$$

$$= \operatorname*{argmax}_{\gamma} \sum_{n:c_n=k} b_{in} \log \gamma + (1 - b_{in}) \log(1 - \gamma)$$

In this case, $\alpha_{i',k}$ is set to the fraction of times the i^{th} pixel for class k is 'on'.

Layered Bernoulli models: In practice, it is easier to label z rather than b because one does not need to estimate the spatial extent of occluded objects. Fortunately, one can still compute MLE estimate of α by marginalizing out labels for occluded regions:

$$\alpha_{i',k} = \operatorname{argmax} \sum_n \log P(z_i|d) \quad \text{where} \quad i = T_n^{-1}(i')$$

$$= \operatorname*{argmax}_{\gamma} \sum_{n:c_n=k} \mathbf{1}_{[z_i=n]} \log \gamma + \mathbf{1}_{[z_i<n]} \log(1 - \gamma)$$

The above formulation is very similar to standard Bernoulli MLE except that occluded pixels are ignored.

6. Experimental Results

In this section, we present results on the PASCAL VOC 2009 segmentation competition [3]. PASCAL is widely-acknowledged as the most difficult available testbed for both object detection and multi-class segmentation. The competition contains 1500 training and validation images along with ground-truth labelings which give per-pixel labelings for 3200 instances of 20 object categories. Test annotations for the dataset are not released. Instead, benchmarking algorithm performance is done on a held-back test set through a web interface.

6.1. Implementation

To generate detections, we used the part-based detector of [4] which was trained using the PASCAL VOC training dataset. These detectors, which are trained independently for each object class using a support vector machine (SVM), produce scores which are not directly comparable. In order to calibrate the detectors with respect to segmentation, we estimated an optimal threshold for each detector by evaluating the segmentation benchmark at different threshold settings. To perform this search over thresholds, we considered only detections for a single class at a time and used a simple segmentation model which labeled all pixels inside the object bounding box.

We found that the optimal threshold varies widely across different classes (as does the maximal detector performance). The inability of the SVM to learn consistent bias terms for each detector presumably relates to the disconnect between the segmentation benchmark and the detection benchmark. We utilized the per class threshold by simply subtracting the optimal threshold from the detector score and only utilizing detections which scored greater than 0. The offset detector scores were also used in the layering model.

Given an image x and a set of calibrated detections d, our final algorithm is as follows:

For each ordering π, iterate until convergence:

1. $z_{S_i} := \underset{m}{\text{argmax}} \prod_{j \in S_i} P(z_j = m|d_\pi)P(x_j|\theta_m)P(\pi|d)$

2. $\theta_m(j) := \frac{\sum \mathbf{1}_{[x_i=j \quad \text{and} \quad z_i=m]}}{\sum \mathbf{1}_{[z_i=m]}}$

Output superpixel labels z_{S_i} with most probable ordering π

6.2. Benchmark Results

Figure 5 shows the quantitative performance of our system on the 2009 PASCAL segmentation challenge. We compare our results to other top results reported at the PASCAL 2009 workshop [3], ignoring our own previous entry that was a preliminary version of the system described here. Our system performs quite well compared to the average performance across entries into the competition. Specifically, our system performed ranks first over all other entries in the "person", "bicycle", and "car" categories. Because people are the overwhelmingly common object in the PASCAL dataset, our system tends to produce quite reasonable segmentations for many images. We present example image segmentation results in Fig.7.

Figure 5 documents experiments where we analyzed the contribution of different model components to the overall performance. These performance results were computed on the set of "trainval" segmentation images (rather than the official test protocol). To avoid testing on data used to train the local detectors, we removed the subset of images that were also present in the detection training set.

Bottom-up grouping: Overall, the bottom-up grouping constraints tend to provide an improvement for instances of objects with strong boundaries. We observe this phenomena for many bicycles detections. Bicycles have a wiry shape that is hard to model using our pixel-based shape prior. However, boundary edges tend to provide a strong bottom-up signal that greatly constraints the grouping process. This is evident by looking at the performance of our bicycle pixel classifier with bottom-up grouping removed (Fig.5).

Instance-specific appearance model: Our instance-specific appearance model estimation tends to provide an improvement for instances whose color differs from the background. This is true of people, whose appearance varies due to clothing. By estimating an instance-specific color model, our system is able to use clothing-specific cues to help segment out the person. Because only a single model is estimated, our system oftentimes will segment out regions associated with one dominant color. This suggests a useful extension is learning a part-specific color model that can capture the difference in appearance between the torso and legs, for example.

Mixture-of-deformable-parts shape prior: The mixture-of-deformable part spatial prior also tends to help, but to a smaller degree. We hypothesize this is the case because part locations can be inaccurate, as shown in the first detection in Fig.3. We hypothesize the integrating the instance-specific appearance model into the dynamic programming search over part deformations [4] would improve part location estimates, and in turn, the improvement due to part-based shape priors.

Layering: While our layered model provides an elegant approach to dealing with overlapping detections, we found that in practice there seems to be relatively little benefit to searching over depth orderings in the PASCAL dataset. We hypothesize this holds for two reasons. Firstly, because images are sparsely labeled with 20 object categories, it is relatively rare for two objects of different classes to overlap. Only 40% of images had overlaps in the ground-truth bounding boxes, of which half only had a single overlap. Secondly, our local detectors often fail to detect partially occluded objects. Both these facts suggest there are relatively few "interesting" cases where occlusion reasoning might help.

Can ordering help? To verify our hypothesis, we conducted a further experiment using a set of true positive detections culled from the subset of "interesting" images where ordering affected the final segmentation. On this set, we ran our iterative coordinate ascent algorithm for each possible ordering. On average, the worst-scoring ordering produced a score 54.8%, the best possible scored 61.2%, and our system scored 57.7%. This indicates that depth-order reasoning can be useful (given accurate detectors and densely-labeled images), and that our model captures some, but not all, of the gain to be had.

7. Conclusion

We have proposed a simple model which performs this pixel labeling based on the output of scanning window classifiers. There is clearly room for improvement, particularly in better models for determining depth ordering and occlusion (e.g., figure-ground cues[18] or geometric context[7]). It also seems that integration of detection could be taken further by re-scoring detections based on segmentation and by

	¬ordering	¬color	¬superpixel	¬parts	all
background	79.37	78.93	78.65	**79.62**	79.36
aeroplane	35.26	32.39	30.61	**37.22**	35.26
bicycle	**25.46**	23.12	20.7	24.58	25.45
bird	**2.81**	2.78	2.68	2.79	**2.81**
boat	**9.87**	9.16	9.64	9.14	**9.87**
bottle	41.44	39.73	**41.76**	40.19	41.29
bus	49.83	48.52	48.54	48.72	**49.87**
car	46.88	45.66	44.25	46.14	**47.03**
cat	**18.4**	17.68	16.81	15.06	**18.4**
chair	**10.05**	9.06	9.57	8.37	10
cow	17.74	16.83	**18.1**	15.91	17.77
diningtable	6.94	6.8	6.79	6.85	**7.27**
dog	**11.53**	10.55	11.18	10.91	**11.53**
horse	16.07	14.6	15.33	15.19	**16.21**
motorbike	**25.72**	24.38	24.46	24.88	25.62
person	**36.88**	34.98	35.3	32.4	36.81
pottedplant	**15.55**	14.92	15.17	14.32	**15.55**
sheep	21.09	18.77	20.33	17.77	**21.1**
sofa	**12.63**	12.2	12.14	12.05	**12.63**
train	**28.6**	27.43	27.88	27.86	**28.6**
tvmonitor	**46.41**	46.01	46.36	43.67	46.28
average	**26.6**	25.45	25.53	25.41	**26.6**

Figure 5. We analyze different components of our system on the 2009 segmentation training and validation data. The rightmost column is our full system, while the middle four represent our full system minus particular components, such as ordering, instance-specific color estimation, bottom-up grouping, and part-based priors. Our system performs quite for segmenting bicycles and people. Because people represent the overwhelmingly common object in PASCAL, our system tends to produce quite reasonable segmentations overall. For bicycle, bottom-up grouping provides a clear improvement. For people, the color estimation and deformable part-based prior provides a strong improvement. This is likely because people tend to vary in appearance due to clothing, and our instance-specific color model is able to guide the final pixel labeling to more accurate configurations. Similarly people articulate their limbs, and so our part-based prior is able to better bias the grouping process. Overall, we see that each component improves our average score.

	Mean	Max	Us	Our rank
background	41.2	83.5	78.0	8
aeroplane	18.8	56.3	32.8	7
bicycle	10.4	26.6	29.4	1
bird	11.0	40.6	3.2	17
boat	11.5	36.1	5.0	16
bottle	18.2	46.1	33.1	3
bus	25.5	50.5	43.4	3
car	20.6	42.3	43.8	1
cat	12.6	35.3	8.3	12
chair	4.2	9.1	5.1	9
cow	11.7	33.1	11.9	9
diningtable	9.1	27.0	8.2	11
dog	9.1	24.5	5.6	14
horse	17.5	42.7	21.0	7
motorbike	23.4	56.4	24.4	9
person	20.9	37.5	38.6	1
pottedplant	9.7	37.1	14.6	6
sheep	19.7	43.6	14.8	13
sofa	8.5	21.9	3.5	17
train	19.2	41.0	27.5	7
tv/monitor	22.3	47.8	45.7	2
average	16.4	36.2	23.7	7

Figure 6. We show results of our system on the held-out testset of the 2009 PASCAL Segmentation Challenge [3] using the public web interface. We compare to all the original systems, omitting our own entry that was a preliminary version of the system described here. We perform quite well compared to the average performance across all entries. For "people", "bicycles", and "cars" we obtain the best performance. Since "people" are the overwhelmingly common object in the PASCAL dataset, our system tends to produce quite reasonable segmentations for many images. We show examples in Fig.7

.

utilizing richer bottom-up segmentation information (e.g., richer appearance models, hierarchical superpixel segmentations).

Accurately explaining a scene in terms of multiple objects of different classes requires resolving conflicting detector outputs into a single coherent explanation. The multi-class segmentation problem forces this issue since each pixel must be assigned a single discrete label. This make it a much more challenging and rewarding problem than returning a ranked list of possible detections.

Acknowledgements Funding for this research was provided by a gift from Microsoft Research, a UC Labs research program grant and NSF Grant IIS-0812428.

References

[1] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. From contours to regions: An empirical evaluation. In *CVPR*, 2009. 4

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

[3] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2009 (VOC2009) Results. 1, 5, 6, 7

[4] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *IEEE PAMI*, 2009. 1, 5, 6

[5] R. Gao, T. Wu, S. Zhu, and N. Sang. Bayesian inference for layer representation with mixed markov random field. In *Energy Minimization Methods in CVPR*, pages 213–224. 2

[6] X. He, R. Zemel, and M. Carreira-Perpinan. Multiscale conditional random fields for image labeling. In *CVPR*, volume 2, 2004. 1

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

[8] D. Hoiem, C. Rother, and J. Winn. 3d layout crf for multi-view object class recognition and segmentation. In *CVPR*, 2007. 2

[9] D. Hoiem, A. Stein, A. Efros, and M. Hebert. Recovering occlusion boundaries from a single image. In *ICCV*, 2007. 2

[10] N. Jojic and B. Frey. Learning flexible sprites in video layers. In *CVPR*, volume 1, 2001. 2

[11] M. Kumar, P. Ton, and A. Zisserman. Obj cut. In *CVPR*, volume 1, 2005. 2

Figure 7. We show example results of our detection-based layered segmentation algorithm. Because many images contain people, our system tends to produce quite reasonable segmentations. The bottom-up grouping helps constrain the segmentation when the shape prior is too blurred. This tends to be the case for wiry objects such as the bicycle on the **bottom right**. Our instance-based color model helps segment out objects with region properties that differ from their backgrounds, such as the person riding the horse on the **top right**. Failures occur when the color model only captures the dominant color of the object, as in the case of the **lower center** image of the person whose jacket matches the background.

[12] M. Kumar, P. Torr, and A. Zisserman. Learning layered pictorial structures from video. In *Indian Conf on Comp Vis, Graphics and Image Proc*, pages 158–163, 2004. 2

[13] S. Kumar and M. Hebert. A hierarchical field framework for unified context-based classification. In *ICCV*, volume 2, 2005. 1

[14] B. Leibe, A. Leonardis, and B. Schiele. Combined object categorization and segmentation with an implicit shape model. In *ECCV 04 workshop on statistical learning in computer vision*, pages 17–32, 2004. 2

[15] A. Levin and Y. Weiss. Learning to combine bottom-up and top-down segmentation. *International Journal of Computer Vision*, 81(1):105–118, 2009. 2

[16] M. Nitzberg, D. Mumford, and T. Shiota. *Filtering, Segmentation and Depth*. Springer-Verlag, 1993. 2

[17] D. Ramanan. Using segmentation to verify object hypotheses. *CVPR*, 2006. 2

[18] X. Ren, C. Fowlkes, and J. Malik. Figure/ground assignment in natural images. In *ECCV*, 2006. 6

[19] A. Saxena, M. Sun, and A. Ng. Make3D: Learning 3D Scene Structure from a Single Still Image. *IEEE TPAMI*, pages 824–840, 2009. 2

[20] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. *ECCV*, 3951:1, 2006. 1

[21] A. Torralba, K. Murphy, and W. Freeman. Contextual models for object detection using boosted random fields. *NIPS*, 2004. 1

[22] Z. Tu. Auto-context and its application to high-level vision tasks. In *IEEE CVPR*, 2008. 1

[23] Z. Tu, X. Chen, A. Yuille, and S. Zhu. Image parsing: Unifying segmentation, detection, and recognition. *IJCV*, 63(2):113–140, 2005. 2

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

[25] J. Wang and E. Adelson. Representing moving images with layers. *IEEE Trans on Image Processing*, 3(5):625–638, 1994. 2

[26] J. Winn and J. Shotton. The layout consistent random field for recognizing and segmenting partially occluded objects. In *CVPR*, 2006. 2

[27] S. Yu, R. Gross, and J. Shi. Concurrent object recognition and segmentation by graph partitioning. *NIPS*, pages 1407–1414, 2003. 2