Augmenting Deformable Part Models with Irregular-shaped Object Patches

Roozbeh Mottaghi
University of California, Los Angeles
roozbehm@cs.ucla.edu

Abstract

The performance of part-based object detectors generally degrades for highly flexible objects. The limited topological structure of models and pre-specified part shapes are two main factors preventing these detectors from fully capturing large deformations. To better capture the deformations, we propose a novel approach to integrate the detections from a family of part-based detectors with patches of objects that have irregular shape. This integration is formulated as MAP inference in a Conditional Random Field (CRF). The energy function defined over the CRF takes into account the information provided by an object patch classifier and the object detector, and the goal is to augment the partial detections with missing patches, and also to refine the detections that include background clutter.

The proposed method is evaluated on the object detection task of PASCAL VOC. Our experimental results show significant improvement over a base part-based detector (which is among the current state-of-the-art methods) especially for the deformable object classes.

1. Introduction

Part-based object detectors have shown remarkable performance in recent years [1, 2, 3, 5, 6, 7, 21, 25]. Among them, the Deformable Part Model (DPM) by Felzenszwalb et al. [5] and its variants such as [2] and [25] demonstrate the state-of-the-art performance on difficult object detection benchmarks. However, there are several limiting factors that prevent these types of part-based detectors from achieving the ideal performance. A major challenge is to define a topological structure for the object model. The structure should be flexible enough to capture the deformations of objects. On the other hand, learning and inference should be performed efficiently on these models, which often restricts the flexibility of the structure. Although impressive results are achieved using star structures [5] or hierarchies of deformable parts [25], these structures are not flexible enough to reliably capture variations in highly deformable objects such as cats or plants.

Additionally, specifying parts as rectangular blocks in DPMs imposes even more restriction on modeling the flexibility of objects. Rectangular blocks of features have been used as a common representation for parts in object detectors (e.g., [1, 3]). Although rectangular blocks capture the shape or context around the parts, they might not be the most appropriate choice for capturing object masks or part deformations.

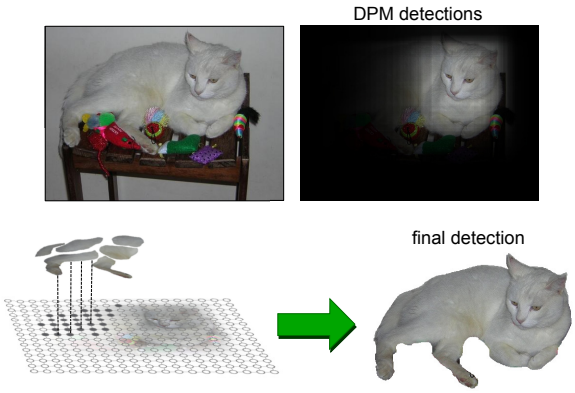

Figure 1. The Deformable Part Models (DPMs) are often unable to capture large deformations of objects. For example, the learned models for cats usually correspond to the cat head, which is a rigid part. As illustrated in the top-right image, the detections are concentrated on the head (the visibility is encoded by the value obtained by accumulating the score of detections over each pixel). Our goal is to append the missing HOG-bundles shown in the bottom-left panel (and remove the background HOG-bundles) through a CRF framework to better estimate the object location. Grabcut was applied on this picture to generate smooth boundaries. Grabcut is not used for the final evaluations.

In this paper, we aim to improve the family of Deformable Part Models ([5] and its variants) to better capture large deformations. Hence, we develop a framework to integrate the detections of DPMs with patches of irregular shape (HOG-bundles). The contribution of this paper is two-fold. First, we extend the idea of HOG-bundles [18] that are a representation for object parts. HOG-bundles are formed by unsupervised grouping of HOG cells, and approximated as rectangular blocks in [18]. The only properties used by [18] for describing bundles are width and height, which do not carry much information, especially for bundles with irreg-

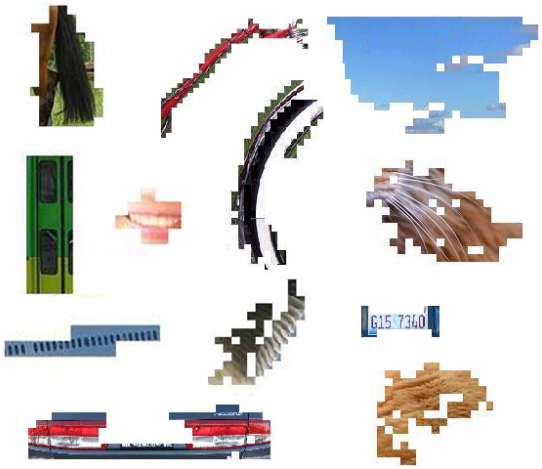

Figure 2. Examples of HOG-bundles. From top-left to bottom-right: horse tail, bicycle frame, sky, bus door, lips, bike wheel, cat whiskers, airplane windows, bird feather, license plate, car tail lights, and cat fur.

ular shape. We propose a set of descriptors for the bundles so we recognize the object category they belong to. There are advantages in using HOG-bundles in our framework: (i) they have irregular shape allowing better representation of the subparts of highly deformable objects, (ii) they are computed efficiently (less than 0.5s per image on a CPU), (iii) there are only a few hundred of them in each image. Example HOG-bundles are shown in Fig. 2.

The second contribution of this paper is formulating the problem of augmenting the detections of Deformable Part Models with HOG-bundles as the MAP estimation of a multi-label CRF. The goal is to find the labeling that minimizes a defined energy function, where the node label specifies to which object of the scene a pixel belongs. As a result, our method extends the detections with missing parts, shrinks the detections that include background clutter, and increases the confidence of the detections with low confidence (caused by partial occlusion or unusual pose). Fig. 1 shows an example, where the DPM is unable to detect the cat due to deformations, and the detections are concentrated on the head of the cat, which is a rigid part. Our approach distinguishes all regions corresponding to the cat by augmenting the detections with the missing object patches and eliminating the background clutter.

Related work. A number of methods have been proposed to overcome the deficiencies of DPMs in modeling the flexibility of objects. For example, Girshick et al. [9] have proposed a method based on object grammars to allow more flexibility in describing objects. Their current result is limited only to people detection. Schnitzspan et al. [21] describe a method for automatic discovery of parts and their topological structure, but their results do not compare favorably with the results of the current state-of-the-art DPMs. Parkhi et al. [19] improve the detection of cats and dogs by training a DPM for a distinctive part (head detector) and extending the detected regions using color cues. The advantage of our method to theirs is that we do not require a distinctive part detector. Also, our method generalizes to other categories as we do not rely on color cues only.

Several works, for example [10, 15, 17] to mention a few, have used CRFs with or without object detection priors to address the problem of object class segmentation. These approaches specify the object class that a pixel belongs to but they cannot determine how many instances of an object category are present.

OBJ CUT, proposed by Kumar et al. [14], combines CRFs with Pictorial Structures to solve a joint segmentation-detection problem. LayoutCRF [24] addresses the problem of detecting and segmenting partially occluded objects using CRFs. These approaches have not been applied to highly articulated objects and they have not been designed to handle viewpoint and scale changes.

The method of Gould et al. [11] integrates multi-class segmentation with object detection using a region-based approach. Work by Ladicky et al. [16] proposes another CRF-based framework for estimating the class category, location, and segmentation of objects in a scene. The goal of these methods is different from ours as they find a labeling for scene contents while we try to improve object detectors. Therefore, they have not provided results on challenging *object detection* datasets.

2. HOG-bundle Description and Classification

Our goal in this section is to develop a binary classifier to distinguish the HOG-bundles of a certain object category from the others. HOG-bundles are formed by unsupervised grouping of neighboring HOG cells that have similar features and dominant orientations. Each HOG-bundle has been described by its height and width in [18], which is not adequate for this classification task. Therefore, we propose a set of descriptors for HOG-bundles and use them as inputs to the classifiers. In addition to SIFT and color, we propose two other descriptors more specific to HOG-bundles.

2.1. HOG-bundle Descriptors

Gradient Descriptor: is defined based on the histograms of the gradients of pixels in four regions of a HOG-bundle. This descriptor is somewhat similar to SIFT, where the main difference is that the pixels that do not belong to the bundle are not used in building the histograms. Also, the gradient histograms are computed over irregular-shaped regions instead of rectangular patches. A gradient orientation histogram is built for each region separately, where each bin represents the sum of the magnitude of the gradients whose orientation corresponds to that bin. The regions intersect at the center of the bundle and are rotated according to the dominant orientation of the HOG-bundle (see Fig. 3(a)). As

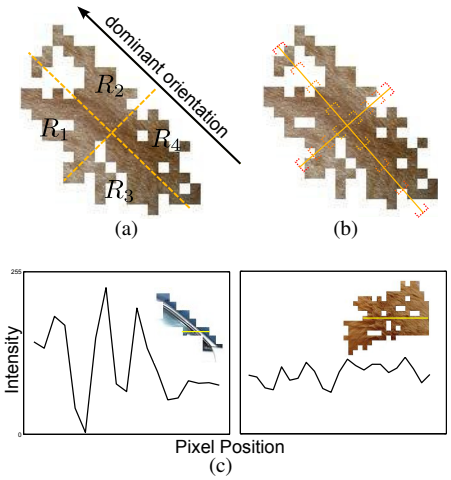

Figure 3. (a) Regions used for computing the gradient descriptor. The regions are rotated according to the bundle dominant orientation. (b) Line segments used for computing the jaggedness descriptor. The start and end of each segment are represented by the same color. (c) Intensity profiles of a car and a cat bundle computed over the yellow line. The profile is almost periodic with small changes for a cat bundle, while there are sharp peaks in the profile of a car bundle.

in [18], the dominant orientation of a HOG-bundle is defined as the mean of the orientations with maximum magnitude in the constituent HOG cells.

The gradient descriptor $\mathbf{g}$ is constructed by concatenation of the vectors $\frac{1}{|R_1|}\mathbf{g}_1$ through $\frac{1}{|R_4|}\mathbf{g}_4$, where $\mathbf{g}_i$ is the histogram corresponding to region R_i and $|R_i|$ is the number of pixels that belong to region R_i. We quantize the orientations into 18 bins so this descriptor has $72 = 4 \times 18$ dimensions.

SIFT Descriptor: is defined based on the histogram of SIFT features computed over the bundles. First, a vocabulary of 4096 SIFT words corresponding to four different SIFT scales is generated (1024 words for each scale) using the standard K-means clustering. Then, for each cell of a bundle, the SIFT descriptor is computed at four scales (8×8, 12×12, 16×16, 20×20 patches). Using the Soft-binning technique [20], we find a 4096 dimensional histogram for each cell. We accumulate all of the histograms corresponding to different cells by summing over the corresponding bins and normalizing them according to the number of the HOG cells in a HOG-bundle. It should be noted that the patch for the SIFT descriptors is centered at the center of the HOG cells. We denote the SIFT descriptor by $\mathbf{s}$.

Jaggedness Descriptor: provides a rough estimate for the changes of intensity over a HOG-bundle. For example, as shown in Fig. 3(c), a cat fur bundle is usually composed of parallel lines, and the intensity over the bundle changes smoothly and almost periodically, while there is a sharp peak in bundles that correspond to edges of car parts.

To compute the descriptor, we consider 8 line segments over a HOG-bundle, 4 in the direction of the dominant orientation, and 4 in the perpendicular direction. All of the line segments are centered around the bundle center, and their length is $w/4$, $w/2$, $3w/4$, and w for the segments along the width of the bundle and $h/4$, $h/2$, $3h/4$, and h for the segments along the height of the bundle. Fig. 3(b) shows the line segments for an example bundle. If p and q are neighboring pixels along a line segment where p precedes q and both of them belong to the HOG-bundle, the feature vector for the i^{th} line segment, $\mathbf{j}_i$, is defined as: $\mathbf{j}_i = \frac{1}{L_i}\left[\sum_{p:I_p>I_q}\nabla I_{pq}, \sum_{p:I_p<I_q}\nabla I_{qp}, \sum_{p:I_p=I_q}\mathbb{1}(I_p = I_q)\right]$, where $\nabla I_{pq} = I_p - I_q$, and I_p and I_q are the intensities of pixels p and q, respectively. L_i is the number of pixels of segment i that belong to the bundle, and $\mathbb{1}(.)$ is the indicator function. The first two components are intensity differences while the third component is a normalized count. We leave it to the classifier to find the proper weights for combining these different quantities. Since we have 8 line segments, the bundle descriptor, $\mathbf{j} = [\mathbf{j}_1, \ldots, \mathbf{j}_8]$, has 24 dimensions.

Color Descriptor: is the normalized color histogram of pixels of a bundle. We use HSV color space and compute a histogram with 16 bins for each channel. The color descriptor, $\mathbf{c}$, is defined by the concatenation of the histograms for each channel and has $48 = 3 \times 16$ dimensions. The histograms are normalized by the number of pixels in a bundle.

2.2. Feature Combination

One of the unary terms of our CRF formulation in Section 3.1 is the likelihood of a HOG-bundle being part of a certain object category. This likelihood can be computed by a classifier. Our goal in this section is to classify HOG-bundles according to the descriptors mentioned previously. For this purpose, we develop a two-stage binary SVM classifier to combine our features.

In the first stage, we learn a binary SVM classifier with RBF kernel for each descriptor separately (for the SIFT descriptor we use a linear kernel due to its high dimensionality). As the result, we learn the parameters $\{\alpha_i^d, b^d\}$ that are used in the SVM discriminant function for bundle $\mathbf{z}$: $o(\mathbf{z}^d) = \sum_{\mathbf{v}_i^d \in \mathcal{S}} \alpha_i^d y_i K(\mathbf{z}^d, \mathbf{v}_i^d) + b^d$, where $\mathbf{z}^d$ is one of the bundle descriptors (d indexes the descriptors $\mathbf{g}$, $\mathbf{s}$, $\mathbf{j}$, and $\mathbf{c}$ defined above), and $\mathbf{v}_i^d$ is a member of the support vectors set $\mathcal{S}$. The bundle label y_i is binary-valued and determines if the bundle belongs to a certain category or not. Also, the kernel is defined as $K(\mathbf{z}^{(.)}, \mathbf{v}_i^{(.)}) = \exp(-\gamma \left\|\mathbf{z}^{(.)} - \mathbf{v}_i^{(.)}\right\|^2)$ for the Gradient, Jaggedness, and Color descriptors, and a linear kernel is used for the SIFT descriptor $K(\mathbf{z}^{\mathbf{s}}, \mathbf{v}_i^{\mathbf{s}}) = (\mathbf{z}^{\mathbf{s}})^T \mathbf{v}_i^{\mathbf{s}}$.

To combine the features, we use a linear SVM classifier that uses the output of the first stage as its input. Our approach is similar to [8] with the difference that we use a binary linear SVM classifier in the second stage instead of

their LP-B method. As the result of the first stage of classification, we obtain four real valued functions, $o(\mathbf{z}^{\mathbf{g}})$, $o(\mathbf{z}^{\mathbf{s}})$, $o(\mathbf{z}^{\mathbf{j}})$, and $o(\mathbf{z}^{\mathbf{c}})$, corresponding to the four different features we defined. The linear SVM classifier in the second stage finds the discriminant hyperplane parametrized by β_d and c: $O(\mathbf{z}) = \sum_{d\in\{\mathbf{g},\mathbf{s},\mathbf{j},\mathbf{c}\}} \beta_d o(\mathbf{z}^d) + c$. Therefore, a real value is assigned to each bundle $\mathbf{z}$ in an image. The positive training examples for these classifiers are the bundles completely contained inside the bounding boxes of a particular category in the training images. The rest of the bundles are considered as negative.

3. Integration of Deformable Part Models with HOG-bundles

Our object detection framework fits into the paradigm of image labeling, where the labels correspond to object instances or background. In this section, we explain the Conditional Random Field (CRF) model that we use to integrate the information provided by the Deformable Part Models and HOG-bundles.

Our CRF model has a random variable for each image pixel, where the variables take a value from a discrete set of labels $\mathcal{L} = \{l_0, l_1, \ldots, l_M\}$. l_0 corresponds to the background and M denotes the number of objects of a *particular class* in the image (l_1 = car$_1$ and l_2 = car$_2$, for instance). We define a Gibbs distribution for the posterior pixel labeling $\mathbf{x}$ given an observed image $\mathcal{I}$: $P(\mathbf{x}|\mathcal{I}) = \frac{1}{Z}\exp\{-E(\mathbf{x})\}$, where Z is the partition function and $E(\mathbf{x})$ is the energy function. To determine the most probable assignment of pixels to objects, we compute the maximum-a-posteriori (MAP) estimate labeling, $\mathbf{x}^* = \arg\max_{\mathbf{x}} P(\mathbf{x}|\mathcal{I}) = \arg\min_{\mathbf{x}} E(\mathbf{x})$. The energy function is computed according to a set of unary and pairwise potential functions that we describe next.

3.1. Unary Terms

Superposition term (P_s)**.** This term is defined based on the detections from a DPM and measures the likelihood of presence of an object of the category of interest at a certain location in an image. The discriminant function for region $\mathcal{X}$ with root part $\mathbf{p}_1$ (see [2] for instance for further details) is defined as:

$$F(\mathcal{X}, \mathbf{p}_1) = \max_{\mathbf{p}} \ \mathbf{w}.\Phi(\mathcal{X}, \mathbf{p}), \tag{1}$$

where $\mathbf{w}$ represents the learned parameters, and $\Phi(.,.)$ is a vector of shape and appearance features defined over region $\mathcal{X}$ and parts $\mathbf{p}$. The set of detections, $\mathcal{D}$, is defined as: $\mathcal{D} = \{\mathcal{X} : F(\mathcal{X}, \mathbf{p}_1) > 0\}$.

Since a DPM is usually not able to capture large deformations, it generates multiple imperfect detections corresponding to the deformed object. The idea here is that we superimpose all of the detections to obtain a more robust estimate for the location of objects. Let x_i denote the label for the i^{th} node (pixel). The foreground likelihood function is written as:

$$P_s(x_i \neq l_0|\mathcal{D}) = \frac{1}{C} \sum_{\substack{\mathcal{X}\in\mathcal{D} \\ \text{s.t.:} i\in\mathcal{X}}} F(\mathcal{X}, \mathbf{p}_1), \tag{2}$$

where by $i \in \mathcal{X}$, we mean $\mathcal{X}$ includes the pixel that corresponds to node i. Also, C is a normalizing constant that is equal to the maximum superposition value that we obtain from the images of the category of interest in the validation dataset. The definition of this term is along the lines of the voting strategies used for object detection (e.g., [4]).

It should be noted that we do not prune out the detections in $\mathcal{D}$ by non-maximal suppression or any other heuristics at this stage. Pruning the overlapping detections discards the information that the superposition term relies on. Column (c) of Fig. 7 visualizes this term for some example images.

HOG-bundle term (P_h)**.** This term corresponds to the output of the HOG-bundle classifier (O) that we defined in Section 2.2. We use a logistic function to convert the real-valued predictions to probabilities. So we obtain the probability of a HOG-bundle belonging to a particular object category. This probability is uniform across a bundle i.e. the same probability is assigned to the constituent pixels of a bundle. The HOG-bundle term is defined as:

$$P_h(x_i \neq l_0|\mathbf{z}) = \max\big(P_h(x_i \neq l_0|\mathbf{z}_+), P_h(x_i \neq l_0|\mathbf{z}_-)\big), \tag{3}$$

where $\mathbf{z} = \{\mathbf{z}_+, \mathbf{z}_-\}$ is the set of HOG-bundles that cover the i^{th} node. According to [18], images have two sets of overlapping HOG-bundles constructed according to two different parts of the HOG feature vector. We refer to these sets as positive and negative bundles. $\mathbf{z}_+$ and $\mathbf{z}_-$ are members of the positive and negative set, respectively. If a pixel belongs to two overlapping bundles, the higher probability is assigned to that pixel. This term is visualized in column (b) of Fig. 7.

The Superposition and HOG-bundle terms are combined linearly: $P_{lm}(x_i) = \eta P_s(x_i|\mathcal{D}) + (1-\eta)P_h(x_i|\mathbf{z})$, where $\eta \in [0, 1]$. We refer to this linear combination as the *likelihood map* in the rest of the paper. Hence, the unary term of the energy function, $\psi(x_i)$, is given by:

$$\psi(x_i) = \begin{cases} P_{lm}(x_i) & x_i \in \{l_1, \ldots, l_M\} \\ P_b & x_i = l_0, \end{cases} \tag{4}$$

where P_b is an adaptive background likelihood that varies for different images.

The procedure to compute the background likelihood (P_b) is as follows. There are sharp discontinuities in the likelihood map (P_{lm}) at the object boundaries, where one side of the discontinuity belongs to an object and the other side belongs to the background. Our idea is to find the sharp discontinuities and estimate the background likelihood from

the pixels around the discontinuities that potentially belong to the background. The discontinuities can be found by thresholding the magnitude of the difference of a pixel in the likelihood map from its neighbors. We compute the difference map, $|\nabla P_{lm}|$, by accumulating $|\nabla^{t} P_{lm}|$, $|\nabla^{tr} P_{lm}|$, $|\nabla^{r} P_{lm}|$, and $|\nabla^{br} P_{lm}|$ corresponding to difference with the top, top right, right, and bottom right neighbors, respectively. The sharp discontinuities happen at pixels with $|\nabla P_{lm}|$ greater than a threshold T (the values less than T usually correspond to the internal discontinuities of an object). For each pixel at a discontinuity, we find the neighboring pixel that has the minimum likelihood in the likelihood map P_{lm}. The background likelihood P_b for each image is estimated as the median of the likelihood of these neighboring pixels with the lowest likelihood.

3.2. Pairwise Term

We define a pairwise potential function based on the color histogram of HOG-bundles to represent the relationship between the neighboring nodes. It is more robust to define this term based on the color difference of HOG-bundles rather than the individual pixels. Since an internal edge of an object and its surrounding region are usually included in a single HOG-bundle, assigning two different labels to the two sides of internal intensity or color discontinuities typically has a large penalty. The pairwise term $\phi(x_i, x_j, \{\mathbf{z^c}\})$ for two neighbor nodes i and j is given by:

$$\phi(x_i, x_j, \{\mathbf{z^c}\}) = \begin{cases} \exp(-\left\|\mathbf{z^c}(i) - \mathbf{z^c}(j)\right\| / \kappa) & x_i \neq x_j \\ 0 & x_i = x_j, \end{cases} \quad (5)$$

where $\mathbf{z^c}(i)$ is the color histogram for the bundle that node i belongs to (refer to Section 2.1 for the definition of this color histogram) and $\|.\|$ is the ℓ_2 norm. The parameter κ is set to the average of the distances between the color histogram of neighboring bundles. Note that the pixels that belong to a single HOG-bundle have the same histogram, and the changes happen only at the boundary pixels of each HOG-bundle.

Recall that there are two sets of overlapping bundles in each image. Therefore, a pairwise term is defined over each set separately, and we denote them by $\phi_+(x_i, x_j, \{\mathbf{z^c}\})$ and $\phi_-(x_i, x_j, \{\mathbf{z^c}\})$ corresponding to the positive and negative bundles, respectively. The histogram differences are shown for some example images in Fig. 4.

3.3. Object Detection

Our goal is to determine which pixels belong to the object category of interest, and also to find the regions corresponding to the different instances of that category in the image. To find the most probable labels, we minimize an energy function that is computed according to the unary and pairwise potentials we defined above. Given the

Figure 4. The color histogram difference of the positive and negative HOG-bundles. Darker lines represent sharper discontinuities.

random field defined over a standard 8-neighborhood grid $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with nodes $\mathcal{V}$ and edges $\mathcal{E}$, the energy function is defined as:

$$E_s(\mathbf{x}) = -\sum_{i \in \mathcal{V}} \log \psi(x_i) + \lambda \sum_{(i,j) \in \mathcal{E}} \phi_s(x_i, x_j, \{\mathbf{z^c}\}), \quad (6)$$

where $s \in \{+, -\}$ specifies whether the pairwise term corresponds to the positive bundles or negative bundles, and λ is a weighting coefficient that is estimated empirically from validation data. We could combine these two energy functions corresponding to positive and negative bundles to a single energy function but we obtain better results by decoupling these two terms. We find the MAP estimates $\mathbf{x}^*_+$ and $\mathbf{x}^*_-$ corresponding to $E_+(\mathbf{x})$ and $E_-(\mathbf{x})$, respectively, by the sequential tree-reweighted message passing (TRW-S) [13]. The final labeling $\mathbf{x}^*$ is computed as follows. If $\mathbf{x}^*_+$ and $\mathbf{x}^*_-$ agree on a label for a node, that label is assigned to that node. If they disagree, we assign the background label (l_0) to the node.

3.4. Multiple Instance Detection

So far we have treated all of the foreground objects similarly. Now we explain how we distinguish multiple instances of the object category of interest. We use the information from the bounding boxes generated by the DPM for this purpose. The idea is to assign a label to a small set of pixels in each bounding box, and then let the energy minimization find the best labeling for the other pixels in the image. We choose a small set of pixels from each box since a bounding box might include background clutter that should be removed. A fixed label is assigned to the pixels in each bounding box whose value in the likelihood map (P_{lm}) is greater than a certain fraction f of the maximum value in that box. Hence, the threshold varies for each box. We set f to 0.7 in our experiments. We denote the nodes with a fixed label by $\mathbf{x}_k$ and find the MAP estimate conditioned on these known pixels.

The procedure for assigning fixed labels is as follows. First a non-maximal suppression similar to [5] is performed to prune out the overlapping detections. Then, we start from the top-scoring bounding box and assign label l_1 to the

	plane	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	motor	person	plant	sheep	sofa	train	tv
w/o Grad.	81.4	59.2	63.2	69.5	61.5	62.4	60.5	58.7	43.9	67.8	68.8	58.7	67.7	59.8	46.5	59.5	72.3	66.8	66.5	68.6
w/o SIFT	77.5	54.9	60.7	64.5	59.6	56.0	55.2	56.4	60.8	61.8	62.3	56.0	65.1	54.5	46.4	54.1	68.1	60.2	62.6	60.5
w/o Jagged.	82.1	60.3	64.7	70.0	61.6	62.3	60.6	59.5	44.0	68.3	69.5	58.9	68.7	59.8	46.7	59.8	72.6	66.7	67.4	69.1
w/o Color	81.5	59.3	61.6	68.2	61.4	62.2	60.6	59.2	55.3	68.8	67.1	58.9	64.7	59.9	46.9	59.0	72.5	65.8	66.4	69.8
all	82.5	60.6	64.8	70.4	61.6	62.3	60.6	59.9	44.9	69.3	69.8	58.9	69.5	59.8	47.6	60.5	73.4	67.8	68.1	69.9

Table 1. Classification results of the bundles of PASCAL VOC 2007 `test` dataset. The AUC of the classifier ROC curve is shown. Each row shows the results where one of the features is excluded. The shown AUCs correspond to the average result of five different random selections of the negative set during training.

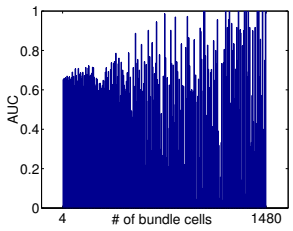

Figure 5. The effect of bundle size on the classification of Horse bundles.

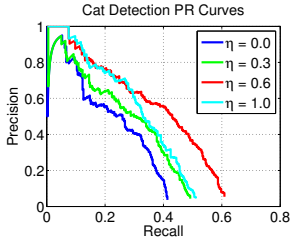

Figure 6. The effect of varying η on the PR curve of Cat detection.

nodes inside the box that satisfy the above threshold criteria. We continue by checking the second top-scoring detection and assign label l_2 to the nodes satisfying the criteria and so on. Since the bounding boxes might overlap, we do not change the label for a node that has already been assigned a label.

We are interested in finding the most probable labeling for the nodes with unknown label $\mathbf{x}_u$ conditioned on the known labels $\mathbf{x}_k$ (specified above): $\mathbf{x}_u^* = \arg\max_{\mathbf{x}_u} P(\mathbf{x}_u|\mathbf{x}_k)$. Since $P(\mathbf{x}_u|\mathbf{x}_k) = P(\mathbf{x}_u, \mathbf{x}_k)/P(\mathbf{x}_k)$ and $P(\mathbf{x}_k)$ is a constant and $\mathbf{x} = \mathbf{x}_k \cup \mathbf{x}_u$, we can write $P(\mathbf{x}_u|\mathbf{x}_k) \propto \exp(-E(\mathbf{x}))$ (we have dropped the positive and negative subscripts for simplicity of the notation). Hence, the energy function is similar to the previously defined energy, and the procedure for finding the optimal labeling does not change.

For bounding box-based evaluation methods, we fit a bounding box around all of the pixels with the same label and output those bounding boxes as our new detections. So there is a bounding box corresponding to each label. The score of a detection is obtained by averaging the value of the likelihood map over the pixels inside the bounding box whose label is the same as the bounding box label. We also add the score of the original detection from the DPM to the computed score.

4. Experiments

We evaluate our method on the PASCAL VOC 2007 and 2010 datasets, which contain 20 object classes. The evaluation details are described below.

4.1. Evaluation of HOG-Bundle Classification

First, we evaluate the discrimination power of the HOG-bundle descriptors by learning bundle classifiers. The positive training examples are the bundles completely contained inside the bounding boxes corresponding to a particular category in `train` images of PASCAL 2007 dataset. The rest of the bundles form the negative example set. Since the number of negative examples exceeds the number of positive examples, we use a randomly sampled subset of the negative set that has the same size as the positive set for training. For testing, we use the `test` set of the dataset that includes around 900K HOG-bundles.

Table 1 shows the area under the ROC curve (AUC) of the classifiers for different combinations of features. Fig. 5 also shows for an example category that the AUC of the final classifier becomes larger as the bundle size increases. It confirms the intuition that it is difficult to recognize the object category based on a small set of pixels.

4.2. Detection Results

To evaluate the performance of our method, we use the DPM of the second winner of PASCAL Challenge 2010 (similar to [2], but excluding the shape masks) as our base object detector. As we show later, our method is independent of the base detector, and other DPMs can be used as well. We prune out most of the irrelevant detections and keep the ones whose score is greater than a certain fraction of the score of the top-scoring detection in the corresponding image. To estimate this fraction for each category, we compute the ratio of the score of the correct detection with the lowest score to the score of the top-scoring detection for all of the validation images containing objects of the category of interest. The fraction is estimated by averaging all of these ratios. On average we keep the detections whose score is above 50% of the score of the top detection in each image. The number of foreground labels M in each image (defined in the beginning of Section 3) is equal to the number of the remaining detections in that image.

Table 2 shows that our method provides significant improvement (up to 8.0%) over the base detector on PASCAL 2007 dataset, especially for the highly deformable classes such as cat, bird, dog, plant, and also for diningtable that has a highly variable appearance. We also show how the performance changes when we use the superposition potential only as our unary term. We combine the superposition and HOG-bundle terms linearly but more sophisticated methods can be used. Fig. 6 shows the effect of choosing different weights for the linear combination on the detection of an example category. We set η to 0.6 in all of our experiments.

We also provide comparisons to some of the state-of-the-art methods [2, 5, 22, 23]. Unlike [5] and [22], we do not

	plane	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	motor	person	plant	sheep	sofa	train	tv	H. D. Avg.
Base Detector	33.5	54.8	12.3	15.6	29.3	51.4	52.8	31.2	21.8	25.4	34.9	18.1	52.9	43.6	40.2	17.5	22.3	31.9	42.7	44.2	28.3
SP only (ours)	30.5	51.7	18.3	11.1	25.4	38.7	53.9	33.8	19.8	22.9	40.1	21.1	41.7	41.6	26.5	13.1	18.9	33.9	44.5	45.4	26.3
SP + HB (ours)	34.5	55.7	**18.6**	17.9	28.4	53.1	52.9	**39.2**	22.1	26.7	**41.0**	**24.3**	55.8	47.3	38.7	**21.1**	**23.9**	34.0	47.3	46.0	**32.1**
gain	+1.0	+0.9	+6.3	+2.3	-0.9	+1.7	+0.1	+8.0	+0.3	+1.3	+6.1	+6.2	+2.9	+3.7	-1.5	+3.6	+1.6	+2.1	+4.6	+1.8	+3.8
MKL [23]	37.6	47.8	15.3	15.3	21.9	50.7	50.6	30.0	17.3	**33.0**	22.5	21.5	51.2	45.5	23.3	12.4	**23.9**	28.5	45.3	**48.5**	25.9
UoCTTI [5]	31.2	**61.5**	11.9	17.4	27.0	49.1	**59.6**	23.1	23.0	26.3	24.9	12.9	60.1	**51.0**	**43.2**	13.4	18.8	36.2	49.1	43.0	26.1
Active Masks [2]	34.8	54.4	15.5	14.6	24.4	50.9	54.0	33.5	20.6	22.8	34.4	24.1	55.6	47.3	34.9	18.1	20.2	30.3	41.3	43.3	28.8
NUS-Context [22]	**38.6**	58.7	18.0	**18.7**	**31.8**	**53.6**	56.0	30.6	**23.5**	31.1	36.6	20.9	**62.6**	47.9	41.2	18.8	23.5	**41.8**	**53.6**	45.3	31.5

Table 2. Detection performance on PASCAL VOC 2007 `test` set. The evaluation metric is Average Precision (%). The colored columns represent the highly deformable classes. "SP only" refers to the case that we use only the Superposition potential as our unary term. "SP+HB" refers to the case that we use the combination of Superposition and HOG-bundle potentials. The row labeled by "gain" shows the improvement of "SP+HB" over the "Base Detector". The last column is the mean AP for the highly deformable classes.

	plane	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	motor	person	plant	sheep	sofa	train	tv	Avg.
Baseline [5]	17.3	35.2	7.1	8.2	21.4	49.3	38.9	22.9	9.9	9.2	**12.7**	10.5	16.0	26.8	31.9	14.0	10.8	13.4	25.2	27.1	20.4
Ours	**21.7**	**47.3**	**8.9**	**15.0**	**30.9**	**59.9**	**40.6**	**32.4**	**10.5**	**18.2**	10.9	**19.01**	**16.2**	**38.2**	**32.1**	**15.7**	**17.2**	**31.5**	**32.4**	**41.8**	**27.0**

Table 3. Detection performance on PASCAL VOC 2010. The evaluation metric is Average Precision obtained using the instance-based pixel-wise scoring.

use contextual cues. Nevertheless, our method outperforms these methods on highly deformable object classes (colored columns in Table 2) in terms of mean AP. It should be noted that our performance is dependent on the underlying detector. There are several cases that the HOG-bundle classifier has a high response on the object but we cannot detect the object because of no response from the base detector.

Fig. 7 shows the bounding boxes generated by the base detector and our detections for some example images. For the cat image, the base detector finds only the head since the DPMs cannot reliably model the deformation of cats. Our method extends the detection to include the whole body. The dog image shows a case that the background clutter is included in the detection since the base detector does not have a reliable model for dog deformations either. Our method shrinks the detection window to exclude the background clutter. Also, the base detector's confidence is low for the chair case (it can be observed in the superposition image). We improve the confidence by incorporating the information from the HOG-bundles.

Instance-based Pixel-wise Scoring: Our detection method provides object masks in addition to the bounding boxes and is able to specify the instance label for each pixel. To justify these, we require a more accurate measure than PASCAL's that is based on the overlap only between bounding boxes (not the object masks). The evaluation strategy should also be more accurate than segmentation evaluation methods that only consider the category label (not the object instance label). Therefore, we propose a new evaluation method. First, we sort the detections by their score. We start from the top-scoring detection and find the groundtruth object instance with the most overlap with the detection. Only the detections with a score above a threshold contribute to the overall precision and recall, which are computed based on matched and unmatched pixels. Precision-recall curves are obtained by varying this threshold [1].

[1] Refer to the supplementary material for more details.

Recently, [12] provided the pixel-wise instance labels for PASCAL VOC detection dataset, which makes our evaluation possible. Table 3 shows the average precisions obtained by using the new evaluation method on PASCAL 2010 subset of the dataset, which includes about 10,000 images. On average, our method provides 6.6 gain over the baseline in terms of AP. To show that our method is applicable to other DPMs, we use [5] as our base for this experiment. It should be noted that our learning is performed on PASCAL 2007 `trainval` set. In all of our experiments, we use only a single scale of HOG features (8×8 cells) to construct HOG-bundles.

5. Conclusion

We propose a novel approach to augment the detections of Deformable Part Models with patches of irregular shape (HOG-bundles) to better capture the object masks. We develop a CRF framework to find the most probable assignment of pixels to each object instance in images. Our method provides significant improvement over the base DPMs on bounding box-based and mask-based benchmarks.

Acknowledgments

This work was supported by ONR-MURI grant N000141010933 and by DARPA MSEE award FA8650-11-1-7149 sponsored by the U.S. AFRL. The author would like to thank Alan Yuille, George Papandreou, and Yuanhao Chen for their help at various stages of the project.

References

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

[2] Y. Chen, L. Zhu, and A. Yuille. Active mask hierarchies for object detection. In *ECCV*, 2010. 1, 4, 6, 7

[3] D. Crandall and D. Huttenlocher. Composite models of objects and scenes for category recognition. In *CVPR*, 2007. 1

[4] A. Farhadi, I. Endres, and D. Hoiem. Attribute-centric recognition for cross-category generalization. In *CVPR*, 2010. 4

[5] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *PAMI*, 2010. 1, 5, 6, 7

[6] P. Felzenszwalb and D. Huttenlocher. Pictorial structures for object recognition. *IJCV*, 2005. 1

[7] S. Fidler and A. Leonardis. Towards scalable representations of object categories: Learning a hierarchy of parts. In *CVPR*, 2007. 1

[8] P. Gehler and S. Nowozin. On feature combination for multiclass object classification. In *ICCV*, 2009. 3

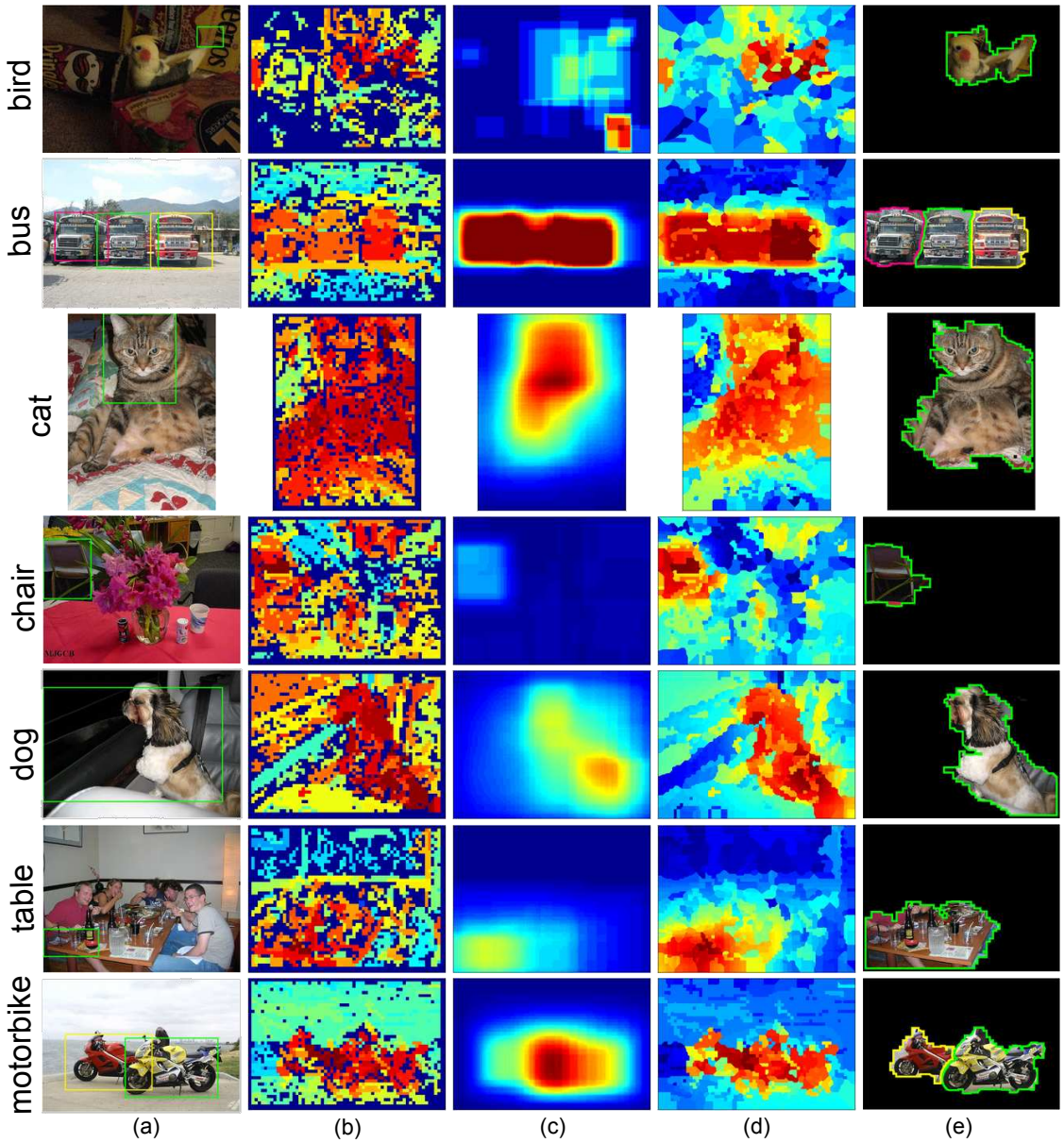

Figure 7. (a) Original detections of the base detector, (b) Heat map of the HOG-bundle term (P_h), (c) Heat map of the Superposition term (P_s), (d) Smoothed heat map of the likelihood map (P_{lm}), (e) Final output of our method. The boundary for each object instance is illustrated by a different color (corresponding to the color of the detections in column (a)).

[9] R. Girshick, P. Felzenszwalb, and D. McAllester. Object detection with grammar models. In *NIPS*, 2011. 2

[10] J. Gonfaus, X. Boix, J. van de Weijer, A. Bagdanov, J. Serrat, and J. Gonzalez. Harmony potentials for joint classification and segmetation. In *CVPR*, 2010. 2

[11] S. Gould, T. Gao, and D. Koller. Region-based segmentation and object detection. In *NIPS*, 2009. 2

[12] B. Hariharan, P. Arbelaez, L. Bourdev, S. Maji, and J. Malik. Semantic contours from inverse detectors. In *ICCV*, 2011. 7

[13] V. Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *PAMI*, 2006. 5

[14] M. P. Kumar, P. H. S. Torr, and A. Zisserman. OBJ CUT. In *CVPR*, 2005. 2

[15] L. Ladicky, C. Russell, P. Kohli, and P. H. S. Torr. Associative hierarchical crfs for object class image segmentation. In *ICCV*, 2009. 2

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

[17] D. Larlus and F. Jurie. Combining appearance models and markov random fields for category level object segmentation. In *CVPR*, 2008. 2

[18] R. Mottaghi, A. Ranganathan, and A. Yuille. A compositional approach to learning part-based models of objects. In *ICCV Workshop on 3D Representation and Recognition*, 2011. 1, 2, 3, 4

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

[20] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *CVPR*, 2008. 3

[21] P. Schnitzspan, S. Roth, and B. Schiele. Automatic discovery of meaningful object parts with latent CRFs. In *CVPR*, 2010. 1, 2

[22] Z. Song, Q. Chen, Z. Huang, Y. Hua, and S. Yan. Contextualizing object detection and classification. In *CVPR*, 2011. 6, 7

[23] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman. Multiple kernels for object detection. In *ICCV*, 2009. 6, 7

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

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