Learning Collections of Part Models for Object Recognition

Ian Endres, Kevin J. Shih, Johnston Jiaa, Derek Hoiem
University of Illinois at Urbana-Champaign
{iendres2,kjshih2,jiaa1,dhoiem}@illinois.edu

Abstract

We propose a method to learn a diverse collection of discriminative parts from object bounding box annotations. Part detectors can be trained and applied individually, which simplifies learning and extension to new features or categories. We apply the parts to object category detection, pooling part detections within bottom-up proposed regions and using a boosted classifier with proposed sigmoid weak learners for scoring. On PASCAL VOC 2010, we evaluate the part detectors' ability to discriminate and localize annotated keypoints. Our detection system is competitive with the best-existing systems, outperforming other HOG-based detectors on the more deformable categories.

1. Introduction

One of the greatest challenges in object recognition is organizing and aligning images of objects from diverse categories. Objects within a semantic category, such as "dog" or "boat", have a diverse set of appearances due to variations in shape, pose, viewpoint, texture, and lighting. At its heart, the problem is one of correspondence. Given a collection of object examples, the learner must determine which examples or portions of examples should belong to the same appearance model. A detailed analysis by Zhu et al. [22] concludes that finding better methods to organize examples and parts into visual sub-categories is the most promising direction for future research.

In this paper, we focus on the problem of learning a *collection of part detectors* (Fig. 1) from a set of object examples with bounding box annotations. We define a good part collection to have the following properties:

1. Each part detector is discriminative. Relevant pieces of the object should score higher than the large majority of background patches.
2. Each part detector localizes a specific piece of the object or the whole object in a particular viewpoint. Parts should be predictive of pose.
3. The set of parts should cover the object examples. At least one part detector should confidently localize each object example.

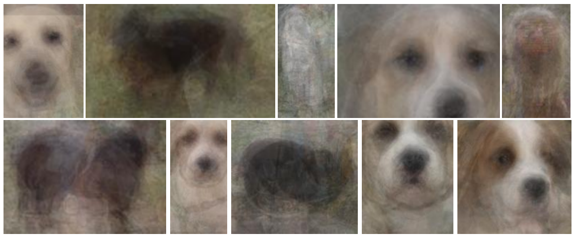

Figure 1: Averaged patches of top 15 detections on held-out set for a subset of "dog" part detectors that model different parts, poses, and shapes. See Fig 4 for more.

In the long run, we are interested in learning a large number of object category and attribute predictors using shared parts. Therefore, we also want to be able to add new part detectors incrementally without retraining existing models. To facilitate transfer learning, we want part detectors that can be applied individually and avoid structured models such as the Deformable Parts Model [10] that require joint inference.

Our main challenge is to simultaneously discover which pieces of examples belong together and to learn their appearance model. Our strategy (illustrated in Fig. 2) is to propose a large number of initial part models, each trained with a single positive example (Sec. 3.2). Based on measured discriminative power on validation examples, the system selects a subset of part models for refinement, aiming to maximize the discrimination and coverage of the collection of parts (Sec. 3.3). Since parts trained on one example tend to perform poorly, we improve them by searching for patches within the training object examples that are likely to correspond (Sec. 3.4). For example, after training an exemplar part model that corresponds to the right side of a particular dog's face, we search within other "dog" examples for the side of the face in the same pose. Finding such examples is difficult because many examples are not applicable (e.g., a side view of a running dog), and, even if the part is present, the detector may incorrectly localize. Including patches that do not correspond decreases localization and/or discrimination of the parts model. We experiment with criteria for selecting additional examples based on appearance score and spatial consistency and find that incrementally adding new

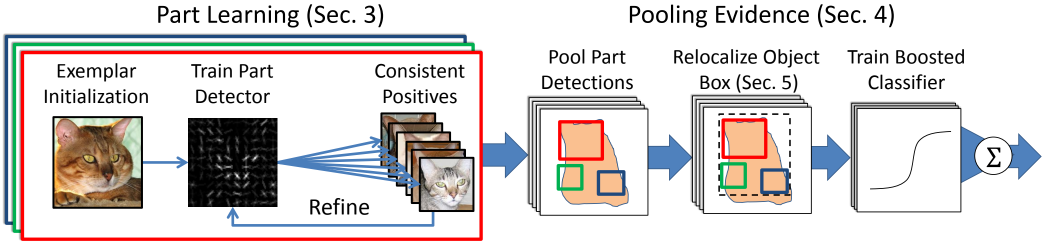

Figure 2: Overview of our part-based detection. Our approach is to train a large number of part detectors with a single positive exemplar (patch or whole object), select a subset of diverse and discriminative candidates, and refine models by incorporating additional consistent training examples. Parts are used to classify bottom-up region proposals into object categories using a boosting classifier, and part predictions are used to predict the the object bounding box.

example parts consistent with each criteria greatly improves localization accuracy.

We propose several criteria for evaluating a collection of parts in terms of the discrimination of parts individually, the coverage of object examples, the predictiveness of manually labeled keypoints on objects (these keypoints are not used in training), and the collective discrimination in terms of object detection performance (Sec. 6). We compare to Poselet-style part learning (using ground truth keypoint annotations) and deformable parts models. Our evaluation methods may be useful for other researchers attempting to develop and validate part learning.

To evaluate parts in terms of object detection performance, we need a method to localize and score an object region using the part detectors. Although not the focus of our paper, we show competitive performance on many categories using a simple method that pools part responses over proposed object regions with a boosting classifier (Sec. 4). We evaluate on PASCAL VOC2010 using the standard criteria and a criteria that ignores localization errors.

2. Related Work

The most related effort in discovering parts is the discriminative method by Singh et al. [19]. Their method is completely unsupervised and proceeds, in brief, by sampling a large number of patches, clustering them, and alternately training on one subset of images and applying to another to update the set of cluster members. Our method is supervised by object-level bounding boxes, enabling us to directly maximize measures of discrimination and coverage for a particular category. We are also able to explicitly evaluate the localization accuracy of the parts and to demonstrate competitive performance on the difficult VOC detection challenge.

Our work is also closely related to Poselets [3] in that we model category appearance with a large collection of part templates. However, our method does *not* require keypoint annotations to train parts. Despite reduced supervision, our method is able to outperform Poselets in many categories. We believe this reflects the difficulty in manually defining effective correspondences.

Other competitive object detection methods [5, 6, 10, 15, 21] that are supervised by bounding boxes differ primarily in how they automatically organize and align examples. Strategies include training one model per exemplar [15], discriminatively aligning and assigning whole-object examples into a moderate number of clusters [6], clustering and aligning with subtemplates [10], or implicitly aligning subtemplates using pyramid bag of words features [21].

Our method learns a moderate number of part templates which may correspond to whole objects or smaller pieces of objects, and applies them without a spatial model. Our method produces a diverse collection of part detectors for detection, pose prediction, and other recognition tasks that can be trained incrementally and applied individually. By avoiding the requirement for joint training (clustering or joint learning of appearance and spatial parameters), our system simplifies extension to additional parts, features, or categories. One motivation is to produce a flexible baseline system for studying spatial models, part sharing [17, 8], and large-scale learning.

3. Learning a Collection of Parts

A good collection of part detectors is discriminative, well-localized, and diverse, allowing easy distinction from other categories while accurately predicting pose and other attributes. Our method for part learning proposes a large number of exemplar-based part detectors, selects a discriminative subset with good coverage, then refines the detectors by finding matching part examples in the training set.

3.1. Modeling Part Appearance

We model the appearance of each part with a HOG template [4]. Each part's appearance is modeled as a linear classifier $\mathbf{w} \in \mathbb{R}^n$ over HOG features $\phi(l)$ for a given location l, which specifies the alignment in position, scale, and left/right flip. For a given candidate object box R, the goal of inference is to find the most likely location of each part within R: $\max_{l \in L(R)} \mathbf{w}^T \phi(l)$. The set $L(R)$ encodes the positions in the image that have sufficient overlap with the given candidate box subject to any transformation. The scores are computed efficiently using convolutions over a spatial pyramid of HOG cells. Our HOG templates range in size from 50-100 cells with maximum dimensions of 10x10.

3.2. Fast Candidate Proposal

To guide the search for high quality parts, we provide a strong yet simple initialization for each part. We randomly sample a patch from within the window of a positive training example and train a template to separate it from all background patches using the LDA accelerated version [12] of the exemplar-SVM [15]. This method precomputes a covariance matrix Σ_d and background mean μ_d of HOG features with dimensions d that captures the statistics across all positions and scales of natural images. Given exemplar features $\mathbf{x}_p$ for a candidate part, the template model $\mathbf{w}_p$ is very simply computed with $\mathbf{w}_p = \Sigma_d^{-1}(\mathbf{x}_p - \mu_d)$. Each initial template can then be used to find correspondences on other training examples that have consistent appearance.

We sample two types of candidate parts: (1) *Whole object templates* capture the global object appearance. Including a diverse set of whole object templates in our model allows us to capture multiple modes of appearance. We initialize one template for each positive training example. (2) *Sub-window templates* capture local appearance consistencies within an object. For each category, we train 2000 templates by sampling a random positive example, scale, aspect ratio, and location within the object bounding box.

3.3. Selecting a Diverse Set of Candidates

To avoid refining thousands of sampled parts candidates, we introduce a procedure to select a small subset of parts that are both discriminative and complementary. Our goal is to choose a set of high precision parts such that every positive example has a strong response from at least one part detector. We quantify these criteria with the *average max precision* measure. For a given collection of parts C and positive part score matrix S, where S_{ip} is the maximum response of the pth part on the ith example, we define

$$\text{AMP}(C, S) = \frac{1}{N}\sum_{i=1}^{N} \max_{p \in C} \text{Prec}_p(S_{ip}). \qquad (1)$$

For part p, $\text{Prec}_p(s)$ gives the precision from the PR curve of a positive example with score s. We use forward selection to iteratively choose the part that gives the greatest marginal AMP gain until no more progress can be made. The selected parts are then refined using the method in Section 3.4. For efficiency, we compute precision with all positive examples, but a subset of 200 negative training images. To compute PR curves, we use the highest scoring part detection with 80% overlap with each positive example and negative parts from images with no positive examples. For examples of the selected parts, see Fig. 4.

3.4. Refining Part Models by Mining New Examples

Finding other positive examples that correspond to the same part as the exemplar significantly improves the reliability of the part detector. Including irrelevant examples can cause the detector to drift from the exemplar and become incoherent, hurting the localization and detection performance of the final model. Given a set of detections on the training set, we show how to automatically decide which correspond to the same part and how to use them to improve the appearance model. We incrementally add examples that are consistent with two criteria based on appearance and location. This process is closely related to self-paced learning from [14], in that we both train on automatically selected subsets of examples to improve appearance models. However, our objectives are quite different: while their method aims to find better local optima for explaining all training examples, we encourage the model to specialize to get the best fit to a subset of the training examples.

Appearance Consistency Estimate. Given the current model for part p and set S_p of consistent examples (initially S_p is just the initial exemplar), we compute the probability that an example is correctly detected given the appearance score of its best-aligned location l^*: $P\left(\text{Correct}|\mathbf{w}_p^T\phi(l^*)\right)$. We first estimate the probability of being correct by splitting the space of scores into 20 bins and counting the number of elements in S and the negative set. We then fit a sigmoid to the scores to minimize the least squared error between the sigmoid's predicted probability and the binned estimate of the probability. In practice this estimate is more stable than Platt's method [18] when there are few positive examples. Then, we update the set S_p with any examples whose new probability of being correct is greater than a threshold τ. This thresholding prunes out examples with low appearance scores, leading to more consistent models.

Spatial Consistency Estimate. We further prune spatially inconsistent examples with a simple spatial constraint. This constraint selects examples that are detected in the same location relative to the object bounding box, which acts as a rough proxy for physical location. The location of the detection within the initial examplar's object bounding box gives a relative offset in scale and location for the expected position of the part. After appropriate scaling, translation, and flipping, we transfer this expected part location to each positive example. Part detections with insufficient overlap with the expected position are removed from the set S_p. We find that this additional spatial constraint is helpful for rigid objects, but may be too selective for highly deformable objects like cats.

Learning the Appearance. Next, we use the set S_p of consistent examples, the set S_n of negative examples and the initial appearance model $\mathbf{w}_p$ to update the appearance parameters and best location of each example. We optimize the parameters with a latent SVM coordinate descent approach [10] that iteratively infers the most likely alignments and uses the corresponding patches to retrain appearance weights, optimizing max-margin objective:

$$\min_{\mathbf{w}_p} \frac{\lambda}{2}||\mathbf{w}_p||^2 + \sum_{i \in S_p \cup S_n} H\left(y_i, \max_{l_i \in L(R_i)} \mathbf{w}_p^T \phi(l_i)\right). \quad (2)$$

Each training example is defined by three variables: A label $y_i \in \{-1, 1\}$, a region of support R_i, and a vector of latent variables l_i encoding position, scale, and left/right orientation. For positive examples ($y_i = 1$), R_i corresponds to the ground truth bounding box. For negative examples ($y_i = -1$), R_i corresponds to a candidate object region proposed by a method such as [7]. $H(\cdot)$ is the hinge loss. The highest scoring latent variable l_i is chosen from the set of valid locations and latent configurations $L(R_i)$.

4. Object Detection Using Parts

Once the collection of part detectors are trained, we pool the responses into a final object hypothesis. We use a "bag of parts" model scored over proposed regions. To score a region, we propose a sigmoid weak learner for boosting part detections that outperforms the more common stubs.

Pooling with Candidate Object Regions. We use the category independent object proposals of [7] to generate 500 candidate object windows for each image. This method generates the candidates using a set of binary segmentations from different seed locations, then ranks them based on a their likelihood of containing an object. For each object candidate, we infer the highest scoring alignment for each part, providing a feature vector of part responses. These responses are used to train a boosted model for each category to classify regions as explained in the next section. To avoid over-fitting, we compute leave-one-out (LOO) scores for each positive training example by retraining the classifier on all but the current image.

Scoring Object Regions Based on Part Scores. Once the intermediate part detectors are learned, boosting is used to learn a comprehensive classifier over their collective responses for each region. Boosting fits our problem characteristics. Although part detectors are individually effective, a linear classifier is not suitable because, while a high-scoring response is strong evidence for an object, a low-scoring response is only weak evidence for a non-object. Further, boosting selects a sparse set of parts, improving detection speed.

We construct the final classifier by boosting over binary decision stubs using a logistic loss [11] as seen in Algorthim 1. Training data X is an N by D matrix for N examples and D part features plus any auxiliary features. Each weak learner added by the boosting selects one feature and maps its values to an object score.

Algorithm 1 Boosted Decision Sigmoids

```
Input: Training data $X$, Training Labels $Y \in \{-1, 1\}$, Max Iterations $M$, Set of weak learners $\mathcal{H}$
Output: Region classifier $C(x)$
Initialize, balance, and normalize weights $\omega_i$ for each example such that:
    $\sum_{i^+|_{y_{i^+}=1}} \omega_{i^+} = \sum_{i^-|_{y_{i^-}=-1}} \omega_{i^-} = \frac{1}{2}$
for $m = 1, 2, ...M$ do
    for all weak learners $c_j(x) \leftarrow f_j(x, y; \omega) \in \mathcal{H}$ do
        compute the weighted logistic loss:
        $L(c_j) = \sum_i \omega_i \log\left(1 + e^{-y_i c_j(x_i)}\right)$
    end for
    Select $c^m(x) = \operatorname{argmin}_{c_j} L(c_j)$ based on (3)
    Update weights: $\omega_i = \frac{1}{(1+e^{y_i c^m(x_i)})}$, $\forall i = 1, 2, ..., N$
    Normalize weights to sum to 1
end for
return Final classifier $C(x) = \sum_m c^m(x)$
```

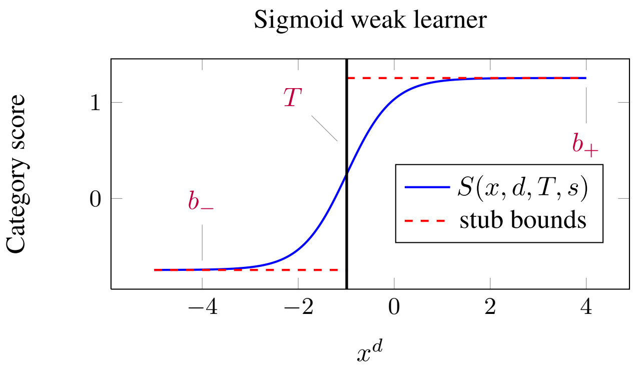

Figure 3: Illustration of our sigmoid learner

Our weak learners are sigmoid-smoothed stub (1-level decision tree) functions. In each round of boosting, we generate a set of candidate weak learners by setting thresholds T for each feature to be evenly spaced between the least positive example and the greatest negative example. The sigmoid function is specified as:

$$c(x) = S(x, d, T, s) = b_- + \frac{b_+ - b_-}{1 + e^{-s(x^d - T)}} \quad (3)$$

$$b_+ = \frac{1}{2} \log \frac{\sum_i \omega_i \mathbb{1}[y_i = 1 \wedge x_i^d \geq T]}{\sum_i \omega_i \mathbb{1}[y_i = -1 \wedge x_i^d \geq T]} \quad (4)$$

$$b_- = \frac{1}{2} \log \frac{\sum_i \omega_i \mathbb{1}[y_i = 1 \wedge x_i^d < T]}{\sum_i \omega_i \mathbb{1}[y_i = -1 \wedge x_i^d < T]}, \quad (5)$$

where b_- and b_+ are the bounds on the classifier confidence computed on the weighted distribution, x^d is the dth dimension of a single example in X, and smoothness weight s is set to the inverse standard deviation of the features values in column d ($s = \sigma_d^{-1}$). Fig. 3 provides an illustration of our sigmoid weak learner. By smoothing the stub's sharp transition boundary with the sigmoid, we aim to avoid overfitting.

A part detector may not have a valid response on an object candidate that is too small or has an incompatible aspect ratio. To handle these cases, feature values corresponding to these failed cases are assigned a do not care value $\mathbf{DNC}$. If $x^d = \mathbf{DNC}$ in training, the corresponding example is ignored any weak learner assigned to column d. In testing, the weak learners will return a confidence score of zero for the example.

Latent Learning. When learning, our method must select the best region for each positive example from the set of 500 pre-computed candidate proposals. We initialize learning with the highest overlapping positive region for each positive example and 30,000 random negative regions. We alternate between retraining the boosted classifier and a resampling phase where we use the current model to mine hard negatives and to reselect the highest scoring positives.

5. Improving Localization

Our part detections are inferred without a spatial model, so nested or overlapping candidate object regions that contain the same strong part detections are likely to receive the same object score. We add a weak learner based on HOG features over the region silhouette to improve region selection and then repredict the bounding box based on part locations for better localization.

Capturing Object Shape. To capture the rough shape of the contents of each region, we compute HOG features on an 8x8 cell grid over the region mask. We then collect the features for each of the positive examples (greater than 50% overlap with ground truth) and a random sampling of negative regions (less than 35% overlap) and train a linear SVM classifier. Including this classifier's prediction in boosting successfully corrects localization errors without resorting to deformation models, allowing us to avoid more complex training and additional optimization during inference.

Repredicting Object Boxes. We use the predicted part locations to vote for a refined object bounding box. Each part votes independently, and we combine with a weighted average based on the probability of the detection being correct and a learned weighting. This weighting is based on how well each part can predict each of the four sides of the bounding box.

First, for each part type p, we learn to predict the object box. We use the calibration procedure outlined in section 3.4 to learn to predict the probability of being correctly localized given score s_p: $P(\text{Cor.}|s_p)$. Then we select the highest scoring location $\mathbf{b}_p$ for each positive ground truth box $\mathbf{g}_i$. We encode the offset $\mathbf{o}_{p,i}$ between the ground truth and its detection by subtracting part's center location $\mathbf{c}_{p,i}$ from the four sides of the box and normalize by the length in pixels of the part diagonal, indicated by $||\mathbf{b}_{p,i}||$. We then collect the offsets for all of the examples and compute a weighted average using their probabilities of being correct:

$$\text{Example Offset :} \quad \mathbf{o}_{p,i} = \frac{\mathbf{g}_i - \mathbf{c}_{p,i}}{||\mathbf{b}_{p,i}||} \tag{6}$$

$$\text{Average Offset :} \quad \mathbf{o}_p = \frac{\sum_i P(\text{Cor.}|s_{p,i})\mathbf{o}_{p,i}}{\sum_i P(\text{Cor.}|s_{p,i})} \tag{7}$$

$$\text{Predicted Box :} \quad \mathbf{e}_{p,i} = \mathbf{o}_p \cdot ||\mathbf{b}_{p,i}|| + c_{p,i}. \tag{8}$$

To account for the predicted left/right orientation, we flip the left and right sides of the box accordingly. During inference, we reverse this procedure and predict the expected object box for each part by accounting for the flip, then scaling the box offset and adding it to the predicted box center.

Next, we find a relative weighting over all of the parts' predicted boxes that encodes how well each part tends to predict the box. We learn four weights $A_{p,d}$ ($d = 1..4$) for each part corresponding to the four sides of the bounding box. Given the part weights $A_{p,d}$ for each part p and side of the box d, we compute the final predicted box $\hat{\mathbf{b}}$:

$$\hat{\mathbf{b}}_{i,d}(A) = \frac{\sum_p A_{p,d} \cdot P(\text{Cor.}|s_{p,i}) \cdot \mathbf{e}_{p,d}}{\sum_p A_{p,d} \cdot P(\text{Cor.}|s_{p,i})}. \tag{9}$$

To learn the weights, we want to minimize the squared error between the predicted box and the the ground truth box $\mathbf{g}_i$ for each example. We normalize the prediction error by the length of the ground truth box's diagonal to account for different object sizes:

$$\min_A \sum_i \sum_{d \in [1,4]} \left(\frac{\hat{\mathbf{b}}_{i,d}(A) - \mathbf{g}_{i,d}}{||\mathbf{g}_i||} \right)^2. \tag{10}$$

6. Experiments

In this section we validate each of our design decisions and compare our final Boosted Collection of Parts model to two successful part-based models.

Dataset. We use the standard PASCAL 2010 VOC detection dataset [9] to evaluate our method. To validate the individual components our method, we use a diverse subset of categories from the train/val split: “aeroplane”, “bicycle”, “boat”, “cat”, “dog”, and “sofa”. We evaluate the spatial consistency of our parts on the poselet keypoint annotations [1]. We compare our overall detection performance to other part-based methods on all 20 categories of the test set.

6.1. Part Validation

We validate our refined parts' detection performance and spatial consistency for the first 40 parts chosen by our part selection procedure. Fig. 4 visualizes a subset of the refined parts for each of the validation categories. We selectively refine parts with (1) appearance criteria only and (2) the intersection of appearance and spatial criteria.

Baselines. We compare our part refinement procedure to three baselines: (1) exemplar models trained on the initial

	Aeroplane			Bicycle			Boat			Cat			Dog			Sofa		
	mAP	3KP	xKP	mAP	3KP	xKP	mAP	3KP	xKP	mAP	3KP	xKP	mAP	3KP	xKP	mAP	3KP	xKP
Initial Exemplar	15.2	10.1	14.1	17.4	23.5	34.6	3.5	6.0	12.4	23.6	14.1	12.8	18.1	6.2	8.9	6.6	4.0	9.1
Refined: All-In	36.5	17.8	21.3	39.7	32.6	41.3	4.0	5.2	9.6	42.3	33.2	22.0	25.8	11.8	12.9	8.0	3.6	7.2
Refined: App	**38.1**	22.2	23.9	**39.9**	33.4	41.6	**5.7**	**8.1**	13.9	**46.5**	**39.5**	**22.5**	**29.5**	**16.5**	**14.7**	8.3	4.4	**11.1**
Refined: App+Spat.	37.3	**31.4**	**27.3**	37.2	**38.3**	**42.4**	4.6	7.6	**14.8**	39.5	33.7	22.2	24.4	13.0	13.3	**8.7**	**5.6**	10.8
Refined: Keypt.	28.0	24.2	$\underline{28.3}$	37.6	$\underline{45.4}$	$\underline{44.2}$	4.0	6.8	$\underline{15.6}$	40.9	$\underline{39.7}$	$\underline{23.6}$	23.5	$\underline{18.0}$	$\underline{17.4}$	7.0	5.2	$\underline{12.9}$
DPM	–	–	$\underline{27.8}$	–	–	$\underline{43.7}$	–	–	13.5	–	–	14.3	–	–	11.9	–	–	$\underline{13.3}$

Table 1: Evaluation of part detection and spatial consistency for each refinement method using three criteria: Mean AP over all parts of a category (**mAP**), the mean AP for detecting the top three keypoints for each part type (**3KP**), and the maximum AP for each keypoint over all parts (**xKP**). **App** and **Spatial** indicate selective refinement with appearance and spatial constraints. Underlines indicate cases where the DPM or models trained on keypoint annotations outperform selective refinement.

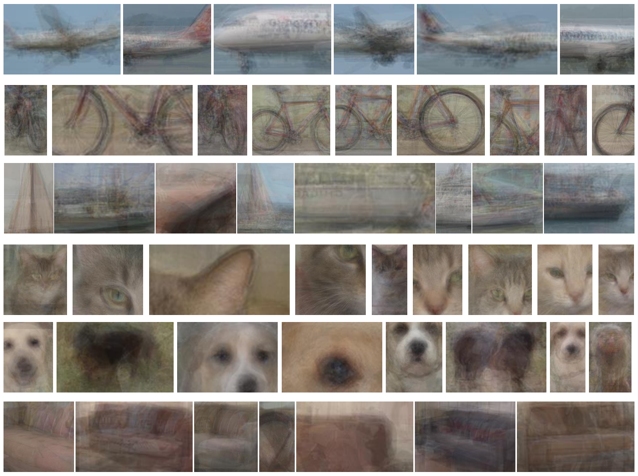

Figure 4: Averages of patches from the top 15 detections on the held-out validation set for a sampling of parts trained for each category on the PASCAL training set. Note the diversity and spatial consistency of most parts. For dogs, different parts on similar portions of the face seem to account for differences across breeds. Some parts correspond to the face (left), others to the whole object (next to left), and others to a small detail, such as the eye or nose.

sampled patches. For fair comparison we retrain with the full exemplar-SVM method rather than LDA-SVM; (2) an "all-in" latent-SVM where every example is used to train a part; (3) parts trained using the poselet annotations where each example is aligned by minimizing the mean squared distance to the annotated keypoints of the initial exemplar, using our implementation of the part learning outlined in [2]. For localization, we also compare to DPM [10].

Detection Performance. To evaluate the discriminative performance of our parts while ignoring localization, detections that are 80% within a positive bounding box are true positives and any detections in images without positive objects are false positives.

Spatial Consistency. To evaluate spatial consistency, we measure each part's ability to predict the keypoint annotations of [1]. Since these keypoints were not used to train our detectors, we compute the offset of each keypoint relative to a part as the median x, y offset values of the 15 highest scoring detections on the training set. Then for each part, we collect the highest scoring detection that overlaps with the positive ground truth example, predict the keypoints using the offsets, and measure the error as the euclidean distance to the ground truth annotation. We count a ground truth keypoint as recalled if the error is less than 10% of the object diagonal. Finally, we compute the average precision of correctly detecting each keypoint. We repeat this process for each part, and summarize the results in two ways: (1) We take the mean average precision of the top three keypoint types for each part and then average over all parts (called **3KP**). This gives a measure of the average spatially consistency of the parts. (2) For each keypoint type, we select the maximum AP over all of the parts and average over keypoints (**maxKP**). This gives a summary of how well a collection of parts can correctly localize all keypoints.

Discussion of Results. The results are summarized in Table 1. First, we confirm that models trained with a single exemplar are unable to generalize to many examples, leading to poor detection performance and consistency. Second, we find that the baseline "all-in" refinement procedure has lower spatial consistency, often by a significant margin. Forcing the "all-in" model to simultaneously capture multiple modes of appearance leads to a less coherent model. In contrast, our selective refinement procedure is more finely tuned because it is allowed to choose examples from a single mode of appearance.

Next, we compare the strengths of our two consistency criteria. The spatial consistency measure takes advantage of the physical regularity of rigid objects like aeroplanes and bicycles, leading to significant gains in keypoint prediction accuracy. However for the more deformable objects, or cases where part performance is less reliable, these constraints become too restrictive and hurt performance. In these cases, selecting examples based on appearance alone works extremely well.

Comparing to the parts trained directly on the keypoint annotations, we find that our spatial consistency is as good or better in many cases. However, it is clear that some categories such as bicycles and dogs could benefit from keypoint annotation. Note that in every case, our parts are more discriminative than the poselets-style parts. Finally, we compare to the DPM. Since our individual parts are not directly comparable, we only compare the coverage of the keypoints. Again, we have extremely competitive performance even though our parts are localized independently whereas the DPM jointly localizes with a spatial model.

6.2. Detection Validation

We summarize the detection performance of our Boosted Collections of Parts in Tables 2, 3. Parts are trained using both appearance and spatial selection criteria.

Classifier Comparison. We compare our boosted sigmoid classifier to several baseline classifiers using average precision at 50% overlap. Each classifier is trained on the full set of parts with shape features and box relocalization. We train two versions of our boosted sigmoids: (1) trained directly on the outputs of our part models and (2) on the leave-one-out (LOO) predictions. We find that these LOO predictions help reduce overfitting to the training set, which is a common problem for classifiers trained on the outputs of other classifiers. When compared to the other baselines, we see that both sigmoid-based classifiers outperform the SVM and boosted stub classifiers. We found that the linear SVM's decision boundary is too simple, causing it to underperform on the training and test sets. In contrast, the boosted stub's sharp threshold transitions hurt generalization. By smoothing the transition boundary with the sigmoid, we find a good balance between expressiveness and generalization.

Localization Comparison. To highlight localization errors, we evaluate with the standard 50% bounding box overlap as well as 10% overlap (as in [13]) which ignores localization errors. We see that the parts alone do well with the 10% criteria, but localize poorly at 50% overlap. Adding the shape features to specifically target localization errors boosts accuracy. Further adding the repredicted bounding boxes gives even greater gains, with comparable performance with the DPM on many categories.

Analysis of Overall Performance. We compare our BCP model to two state of the art part-based methods on PASCAL VOC's test set: the DPM [10] and Poselets [3] (Table 3). Our BCP model achieves competitive performance to both methods for many categories and performs especially well for deformable objects like cat and dog. This highlights its ability to handle rich variation in spatial layout. Our method falls short for some of the more rigid categories (bike, bottle, etc.), where the other methods are known to excel. It should be noted that the inital region proposals have low recall for many of the categories that underperform, such as bottle, car, and sheep.

False Positive Analysis. In Fig. 5 we compare the sources of the highest scoring false positives to the DPM using the analysis code from [13]. Both systems have a comparable number of false positives, but we find that our system makes more sensible mistakes. While the DPM makes random confusions with background, our model instead commonly confuses cats and dogs. Similarly, our model more frequently confuses boats more with other vehicles.

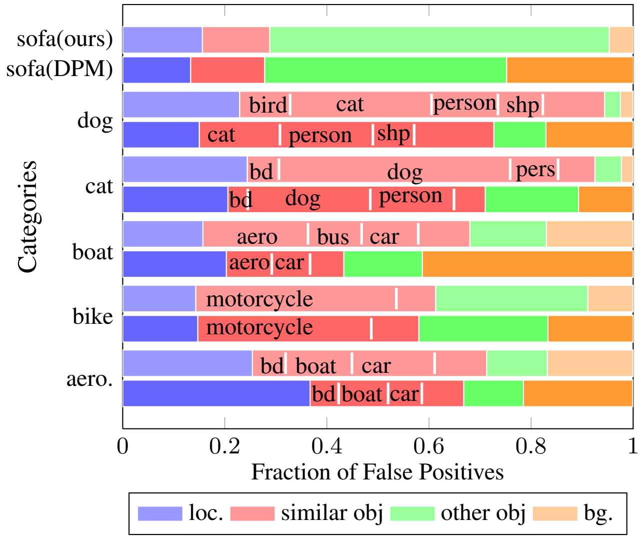

Figure 5: Fraction of top false positives due to localization error (blue), similar categories (red), dissimilar categories (green), and background (orange) using analysis code from [13]. For each category, the first row is our method; the second row, DPM [10]. The most confused categories among similar objects are separated out with white lines and labeled (bd=bird, shp=sheep). Our method consistently has less confusion with background and more confusion with similar objects.

7. Conclusions and Future Work

We present a framework to learn a diverse collection of discriminative parts that have high spatial consistency. To detect objects, we pool part detections within a small set of candidate object regions with loose spatial constraints and training a novel boosted-sigmoid classifier. Our method outperforms DPM on 5 of 20 categories and 8 of 18 for Poselets. The complementary nature of our approach can be seen in the significantly different error patterns from DPM with less confusion with background and more confusion with similar categories. Our method is an important step in building more general object recognition systems. Our boosted collections of parts can extend naturally to the multi-class feature-sharing methods of [20, 16], allowing us to revisit these large-scale learning problems with stronger HOG-based appearance models. Further, an existing collection of parts could be used to guide the search for the structure and layout of novel categories, allowing quick bootstrapping of new category models. Our limited supervision requirements allow scaling to many categories, and our la-

	Classifier Comparison				**Localization Comparison**				
	Baselines		Sigmoids		BCP				DPM
	SVM	Stubs	Direct	**LOO**	Part Only	Part+Reloc.	Part+Shape	**Part+Reloc.+Shape**	[10]
Aeroplane	41.6	47.3	46.9	**48.4**	50.2/13.3	54.7/33.7	57.5/40.3	**61.8/48.4**	58.3/45.0
Bicycle	37.7	36.8	40.8	**43.0**	45.2/14.8	47.9/36.9	47.7/34.9	**50.6/43.0**	56.9/52.7
Boat	1.7	4.2	**5.7**	5.0	14.0/3.1	**17.7**/4.7	14.8/**5.2**	16.0/5.0	17.5/6.4
Cat	30.7	34.4	34.2	**36.9**	55.3/28.5	56.4/34.5	54.0/34.1	**59.1/36.9**	41.8/24.4
Dog	19.3	18.8	**22.1**	20.9	35.7/16.6	39.1/17. 6	38.2/**21.2**	**42.0**/20.9	21.0/8.5
Sofa	7.1	6.6	9.5	**14.1**	26.6/6.1	27.1/11.3	24.6/9.4	**28.6/14.1**	25.5/17.6

Table 2: Detection validation with different classifiers and localization methods. Single numbers indicate AP evaluation. For comparing localization performance, the first number reports AP without localization errors by using 10% bounding box overlap, and the second with 50% overlap. Underlined numbers indicate cases where DPM outperforms.

	Aero	Bike	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow	Table	Dog	Horse	MBike	Person	Plant	Sheep	Sofa	Train	TV
DPM	48.7	52.0	8.9	12.9	32.9	51.5	47.1	29.0	13.8	23.0	11.1	17.6	42.1	49.3	45.2	7.4	30.8	17.1	40.6	35.1
Poselet	33.2	51.9	8.5	8.2	34.8	39.0	48.8	22.2	-	20.6	-	18.5	48.2	44.1	48.5	9.1	28.0	13.0	22.5	33.0
BCP	44.3	35.2	9.7	10.1	15.1	44.6	32.0	35.3	4.4	17.5	15.0	27.6	36.2	42.1	30.0	5.0	13.7	18.8	34.4	28.6

Table 3: Detection comparison on PASCAL VOC 2010 test set.

tent search could allow hybrid methods that use a mixture of detailed supervision and bounding box annotations.

Acknowledgements

This research is supported in part by ONR MURI grant N000141010934, NSF CAREER award 10-53768, and NSF award IIS 09-04209.

References

[1] L. Bourdev. Poselets and their applications in high-level computer vision. http://www.cs.berkeley.edu/l̃bourdev/poselets/.

[2] L. Bourdev, S. Maji, T. Brox, and J. Malik. Detecting people using mutually consistent poselet activations. In *ECCV*, 2010.

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

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

[5] S. K. Divvala, A. A. Efros, and M. Hebert. How important are deformable parts in the deformable parts model? In *Parts and Attributes Workshop , ECCV*, 2012.

[6] S. K. Divvala, A. A. Efros, and M. Hebert. Object instance sharing by enhanced bounding box correspondence. In *BMVC*, 2012.

[7] I. Endres and D. Hoiem. Category independent object proposals. In *ECCV*, 2010.

[8] I. Endres, V. Srikumar, M.-W. Chang, and D. Hoiem. Learning shared body plans. In *CVPR*, 2012.

[9] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2010 Results. http://www.pascal-network.org/challenges/VOC/voc2010/workshop/index.html.

[10] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *PAMI*, 2010.

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

[12] B. Hariharan, J. Malik, and D. Ramanan. Discriminative decorrelation for clustering and classification. In *ECCV*, 2012.

[13] D. Hoiem, Y. Chodpathumwan, and Q. Dai. Diagnosing error in object detectors. In *ECCV*, 2012.

[14] P. Kumar, B. Packer, and D. Koller. Self-paced learning for latent variable models. In *NIPS*, 2010.

[15] T. Malisiewicz, A. Gupta, and A. A. Efros. Ensemble of exemplar-svms for object detection and beyond. In *ICCV*, 2011.

[16] A. Opelt, A. Pinz, and A. Zisserman. Incremental learning of object detectors using a visual shape alphabet. In *CVPR*, 2006.

[17] P. Ott and M. Everingham. Shared parts for deformable part-based models. In *CVPR*, 2011.

[18] J. C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in Large Margin Classifiers*, pages 61–74. MIT Press, 2000.

[19] S. Singh, A. Gupta, and A. A. Efros. Unsupervised discovery of mid-level discriminative patches. In *ECCV*, 2012.

[20] A. Torralba, K. P. Murphy, and W. T. Freeman. Sharing features: efficient boosting procedures for multiclass object detection. In *CVPR*. 2004.

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

[22] X. Zhu, C. Vondrick, D. Ramanan, and C. Fowlkes. Do we need more training data or better models for object detection? In *BMVC*, 2012.