# Adaptive Region Pooling for Object Detection

Yi-Hsuan Tsai
UC Merced
ytsai2@ucmerced.edu

Onur C. Hamsici
Qualcomm Research, San Diego
ohamsici@qti.qualcomm.com

Ming-Hsuan Yang
UC Merced
mhyang@ucmerced.edu

## Abstract

*Learning models for object detection is a challenging problem due to the large intra-class variability of objects in appearance, viewpoints, and rigidity. We address this variability by a novel feature pooling method that is adaptive to segmented regions. The proposed detection algorithm automatically discovers a diverse set of exemplars and their distinctive parts which are used to encode the region structure by the proposed feature pooling method. Based on each exemplar and its parts, a regression model is learned with samples selected by a coarse region matching scheme. The proposed algorithm performs favorably on the PASCAL VOC 2007 dataset against existing algorithms. We demonstrate the benefits of our feature pooling method when compared to conventional spatial pyramid pooling features. We also show that object information can be transferred through exemplars for detected objects.*

## 1. Introduction

Objects appear with large appearance variations due to parts, features and imaging conditions such as viewpoints, scale, and background noise, to name a few. Such large intra-class variability poses a challenging problem for object detection. To cope with large appearance variation, regions or parts [17, 34, 36, 2, 27, 37, 9] are commonly used to encode the shape and scale information of objects, as well as to reduce the effect of background noise. Consider images shown in Figure 1 which illustrate segmented cars captured from three different viewpoints. Regions from similar viewpoints share more similar shapes, sizes and structures than the other ones. By observing this aspect, we can further relate the region structure to feature extraction. For instance, features obtained from regions of a side-view car should have a low similarity to features from one in the frontal view.

Automatically discovering parts of objects provides a useful mid-level feature representation for numerous vision tasks [5, 31, 12, 23]. However, these algorithms use rectangular patches to model object segments, which are less effective in describing non-rigid parts. Other approaches use simple representations such as spatial pyramid pooling of local features [26] that discards a significant amount of geometric information between regions.


Figure 1. Car images from three different viewpoints and their segmented regions are shown with white lines. Similar region structures are shared between cars from similar viewpoints.

In this paper, to address these limitations, we propose an object detection algorithm with a novel feature pooling method that utilizes the region structure information adaptively based on different exemplars, referred as *adaptive region pooling*. We automatically discover a set of representative exemplars in the training set that are segmented into parts, where the segmentation can be generated by region proposals [8, 11, 1, 32] for each image. After defining parts within the object bounding box, the region structure is encoded via feature extraction by our adaptive region pooling method. Our proposed algorithm is able to adjust the structure and the number of parts based on the segmentation of the training exemplars. We learn a regressor for each representative exemplar such that each model is able to deal with one region structure with part information.

Numerous approaches that use multi-models or subcategories for object detection have been proposed [14, 29, 20, 16, 5, 10, 18, 25, 21]. Felzenszwalb et al. [14] and Divvala et al. [10] learn mixture models including global and part components based on the aspect ratio of the bounding box. In this case, the number of parts and models are predefined and not inferred from the training examples, which requires careful tuning of model parameters for each cate-

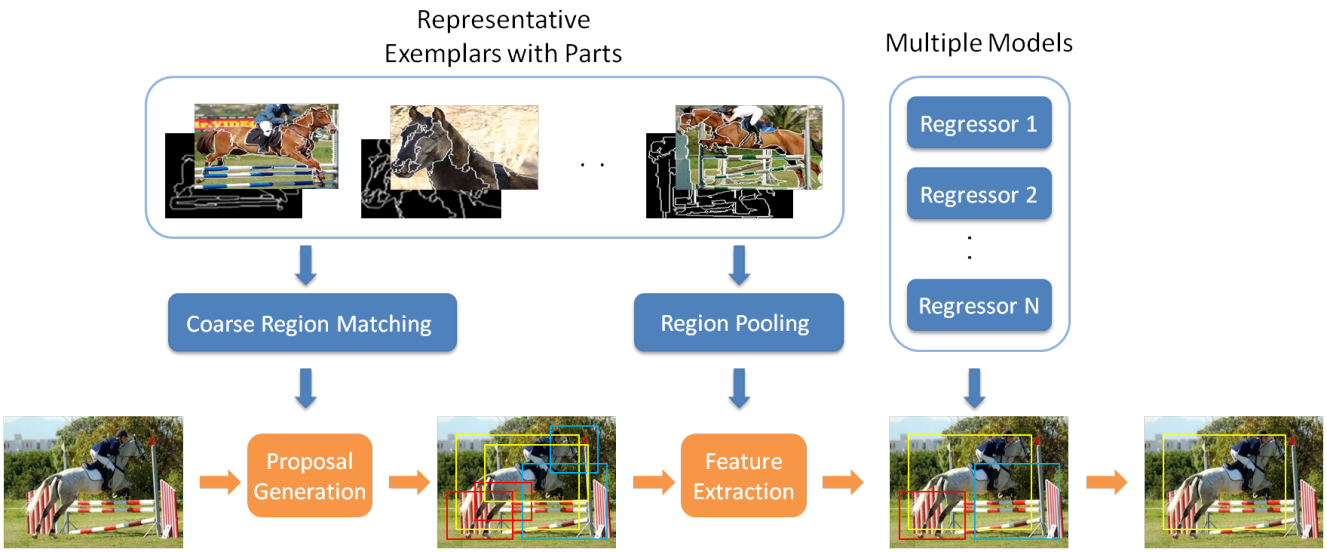



Figure 2. Main steps of the proposed algorithm. In training, we find a set of representative exemplars with parts and learn multiple regression models. Given a test image, we first generate region proposals. Proposals from each exemplar are represented by bounding boxes in different colors. We adopt the proposed region pooling method to extract features to regress testing scores (scores lower than a threshold are discarded). Finally, non-maximum suppression is applied to the sorted scores to generate detection results.

gory. Malisiewicz et al. [29] propose to train each positive exemplar as a model which limits the generalization capacity of each model. One possible way to address this limitation is to cluster or align exemplars into several groups, and learn a model for each group [5, 20, 16]. Bourdev et al. [20] use sliding windows in the testing phase that limit the use of more sophisticated features and classifiers. Other approaches [5, 16] require keypoint annotations and object masks to align objects in the training set. However, these additional annotations constrain the scalability of these algorithms to new datasets and problems.

To address the above-mentioned problems, we propose a region matching method to search and group training samples without using additional annotations. Regions that have similar appearance and size to an exemplar are grouped together as samples to learn a model. Our coarse region matching step constrains the samples that are similar to an exemplar, which facilitates the learning procedure. In addition, the same matching strategy is adopted in the testing stage to greatly reduce the number of proposals needed for evaluation (See Table 1 for comparisons). Since not all the training samples are equally useful, we measure the quality of regions by the overlap ratio between the region and the ground truth bounding box. Then we learn a linear Support Vector Regression (SVR) model [27, 7] with training samples obtained from the matching step and features extracted by our region pooling method. Figure 2 shows the main steps of our algorithm.

We carry out experiments on the PASCAL VOC 2007 benchmark dataset. First, we compare the proposed algorithm with other exemplar-based methods [29, 20]. In addition, we show that our adaptive region pooling method achieves better performance than the conventional spatial pyramid pooling method. Second, we show that our approach accommodates the convolutional neural networks (CNN) features [39, 30, 15] to achieve state-of-the-art results. Finally, we evaluate the performance of transferred object information to the detected objects using the proposed algorithm quantitatively and qualitatively.

Table 1. Comparisons of related algorithms. Our approach generates a small number of proposals for evaluation.

| | # of models | # of windows per model | # of proposals |
|---|---|---|---|
| ESVM [29] | all exemplars | sliding windows | $> 10^6$ |
| LDA [20] | $< 100$ | sliding windows | $> 10^6$ |
| Our method | $< 50$ | 20 | $< 10^3$ |

In summary, we present a unified algorithm using multiple exemplar-based models and a novel adaptive region pooling approach. The contributions of the paper are as follows: 1) We develop an algorithm to automatically find a set of diverse exemplars and regions as parts without using any additional annotations for learning. 2) We propose a feature pooling method which adapts to the local region structure of an object. 3) We present a coarse region matching scheme to efficiently select candidates for learning and testing. 4) We show that our algorithm can transfer object information with state-of-the-art detection performance.

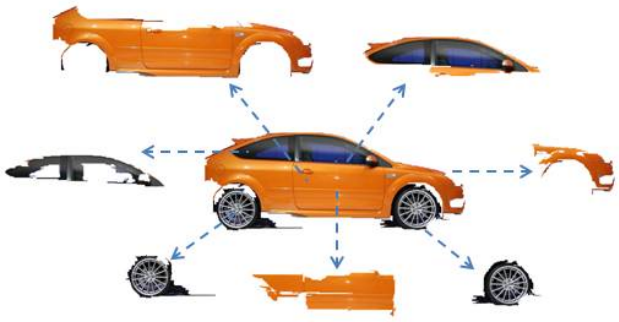


Figure 3. The parts that are obtained from an exemplar. Parts can be non-rigid regions which overlap with other segments. The center image is the object mask obtained by the union of all the parts.

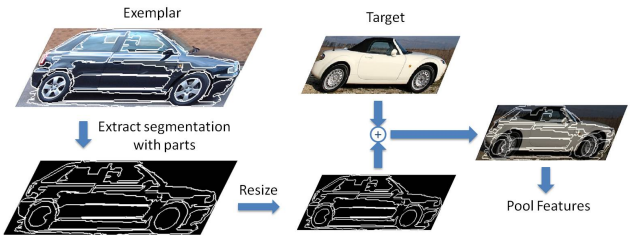



Figure 4. Our feature pooling procedure. Given an exemplar with parts, we resize the region structure to the same size as the target region. The resized part mask is then applied to the target region for pooling features on each part. Note that the exemplar and target are actually sets of regions. Here we present them as entire bounding boxes.

## 2. Adaptive Region Pooling

**Selecting Representative Exemplars.** One way to learn multiple exemplar-based models is to cluster the training data, and use the exemplars within a cluster as positive samples [20]. However, large appearance variations of training examples lead to less desirable clustering results, where exemplars that are less common can be easily absorbed by the dominant clusters. In [16, 5], this problem is addressed with additional annotations of keypoints and object masks that are used to align and cluster the training examples, thereby limiting the application domains.

Instead of relying on every exemplar in the cluster, we propose to find a diverse subset of the exemplars and their similar region proposals. Toward this end, we use the Spectral Clustering method [4] that utilizes pairwise similarity between exemplars. In this stage, we use spatial pyramid pooling with two layers of the SIFT histograms as appearance features and compute the Laplacian matrix using the inner product between features. We select the $k$ eigenvectors of the Laplacian matrix that has the smallest eigenvalues, and use K-means algorithm to cluster all the exemplars in this subspace to different groups. The parameter $k$ is selected with the heuristic approach to find the largest eigenvalue drop in the sorted eigenvalues. In each cluster, the exemplar that is closest to the center of a cluster is selected as the representative exemplar.

The collection of these exemplars generates the set of the representative exemplars. In the training phase, we use each exemplar from the subset to search for similar regions as training samples. Hence we need a feature extraction method that preserves the discriminative structures of the exemplars. We propose a novel feature pooling algorithm that accounts for part information with region-based exemplar models.

**Discovering Parts.** For each representative exemplar found in the training set, we aim to discover parts within the object bounding box based on the segmentation. Unlike the conventional approaches that define the parts as a set of rectangular regions, we present a method that allows to properly find non-rigid deformable regions. We apply several rules that determine if a segment can be an object part:

1. Regions that connect to pixels outside of the ground truth bounding box are removed to minimize the effect of background noise.

2. Regions overlapping with each other in the hierarchical segmentation structure are removed based on an overlap threshold with a preference for larger segments.

3. Small regions that cover less than 100 pixels are eliminated due to lack of distinctive information.

Finally, the parts of an object are selected as at most $L$ largest regions from the remaining ones after applying the above rules. Figure 3 illustrates an example of the parts that are extracted from an exemplar. Note that the object representation is flexible since the parts can overlap with each other. In addition, the number of parts for each exemplar can be different according to the object structure obtained from the segmentation algorithm. For instance, objects with complicated structure may have several parts, while other simpler objects are represented with only a few parts.

**Feature Pooling.** We define our feature pooling algorithm according to the parts for each exemplar in the previous steps. Unlike spatial pyramid pooling that is defined over a pre-defined grid, our pooling method aims to match meaningful segments from the exemplar to the target regions. We illustrate the procedure in Figure 4. First, an exemplar is segmented into $L$ parts as $\mathbf{p}^e = \{p_1^e, p_2^e, \ldots, p_L^e\}$. Second, given a target region $R$, we resize parts of the exemplar to match the bounding box size of the target region. This allows $R$ to be partitioned into the same structure as $\mathbf{p}^e$ to obtain $\mathbf{p}^r = \{p_1^r, p_2^r, \ldots, p_L^r\}$. Then features are pooled based

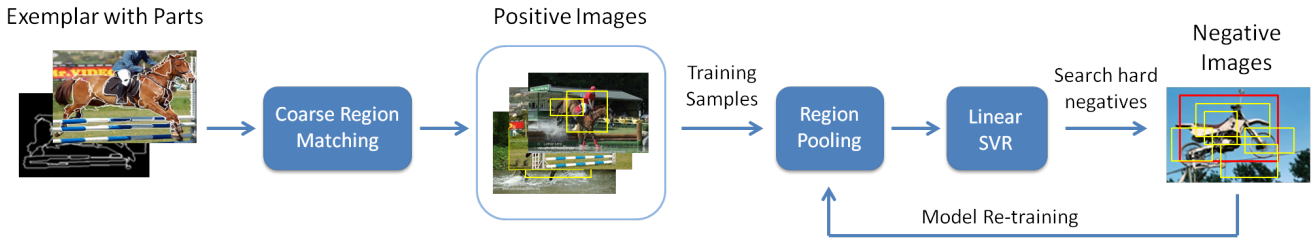



Figure 5. Main steps to learn a linear SVR model. A set of samples are first selected by coarse region matching in positive images. Features are then extracted with region pooling to learn an initial model. Hard negatives with regression scores higher than a threshold are added to retain the model.

on $\mathbf{p}^r$ as $\mathbf{x}^r = [\mathbf{x}_1^r; \mathbf{x}_2^r; \ldots; \mathbf{x}_L^r]$, where $\mathbf{x}_i^r$ is a feature vector for part $i$ and $\mathbf{x}^r$ is the concatenated feature vector from all the $\mathbf{x}_i^r$. Note that each pair of $p_i^e$ and $p_i^r$ targets the same part.

## 3. Multiple Exemplar-based Models

In this work, we learn a linear SVR model for each representative exemplar. A set of training samples that are similar to the exemplar are obtained by a coarse region matching procedure in positive images. We extract features with the proposed adaptive region pooling method for the training samples to learn an initial model. Using the initial model, we re-train the model by searching for hard negatives in other negative images. Note that the regression score is computed based on the union-over-intersection overlap between the bounding box of the ground truth annotation and region proposals. The learning procedure is illustrated in Figure 5.

**Coarse Region Matching.** We adopt an efficient region matching strategy for selecting both training samples and testing proposals. Given an exemplar with the object mask $M^e$, which is the union regions of parts $\mathbf{p}^e$ (See Figure 3), we compute the similarity score between $M^e$ and a target region $R$ based on the appearance and the size of the region:

$$S(M^e, R) = \langle \mathbf{z}^e, \mathbf{z}^r \rangle \cdot \left( \frac{\min(|M^e|, |R|)}{\max(|M^e|, |R|)} \right), \qquad (1)$$

where $\mathbf{z}^e, \mathbf{z}^r$ are feature vectors, and $|M^e|$ and $|R|$ denote the size of an exemplar mask and a target region, respectively. For each $\mathbf{z}$, we use the global pooling feature of the SIFT histograms for efficiency to describe the appearance similarity. The first term of (1) takes the inner product between features of an exemplar and a region. The second term of (1) measures the similarity for the sizes of the regions and is a value between 0 and 1. This term encourages that regions with similar sizes are selected. We consider both terms since regions are sensitive to different sizes, and is dissimilar to the exemplar if only considering features. For instance, a large background region might have similar features to a small object region.

We use the same coarse region matching scheme in the training and testing stage to ensure consistency in the sample space. In training, the coarse region matching allows us to select samples that are similar to one exemplar and enables us to learn a discriminative linear model. In testing, it eliminates a large set of easy negatives. We evaluate the recall rate for localizing objects of our coarse region matching approach in Section 4.

**SVR Models.** The samples that are collected by coarse region matching are used to train a linear SVR, defined formally as:

$$\begin{aligned} \min_{\mathbf{w}, \xi_i, \xi_i^*} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i^*) \\ \text{subject to} \quad O(y_i, y) - \langle \mathbf{w}, \mathbf{x}_i \rangle - b \leq \epsilon + \xi_i \\ \langle \mathbf{w}, \mathbf{x}_i \rangle + b - O(y_i, y) \leq \epsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0, \end{aligned} \qquad (2)$$

where $\mathbf{x}_i$ denotes the feature vector of a region proposal extracted from our region pooling method, and $O(y_i, y)$ denotes the regression score computed by the overlap ratio between the bounding box of the ground truth $y$ and the region proposal $y_i$, and $\epsilon$ is a small constant that controls the error tolerance. The overlap ratio $O(y_i, y)$ of (2) is defined by the PASCAL VOC evaluation criteria [13] that guides the quality of the proposals. Given an image $I$ with a set of ground truth bounding boxes $\{G_i^I\}$ and a region proposal $R$, the overlap ratio is computed by the maximal overlap between $R$ and the ground truth set $\{G_i^I\}$: $O(R, G_i^I) = \max \frac{|R \cap G_i^I|}{|R \cup G_i^I|}$.

For the initial model, we use the top $N$ samples by coarse region matching in each positive image, where the overlap ratio can be any number from 0 to 1. To refine the model, we run one iteration for negative mining by adding samples with regression scores larger than 0.3 among the top $N$ samples in negative images. The overlap ratios for these negative samples are set to 0 to re-train the model.

Our regression models are able to predict the score of a region being an object part. Since the model is trained with the overlap ratio, scores from different models are compa-

Table 2. Average recall rate for coarse region matching on the validation set.

| Top $N$ proposals | 10 | 20 | 30 | 40 | 50 |
|---|---|---|---|---|---|
| Ave. recall rate | 82.3% | 86.8% | 88.4% | 89.6% | 90.4% |

rable to each other. Hence, unlike the commonly used SVM models that require a calibration step or a second level classifier to combine scores, outputs of our model do not require to be adjusted by the mapping from classifier scores through a logistic function using a validation set. In addition, it is usually critical to determine which samples belong to positives or negatives when using SVM models. In our case, we use overlap ratios to measure the quality of region proposals, which reduces the ambiguity in selecting positive or negative samples.

Similarly, in the testing stage, we search the top $N$ candidates in each image by coarse region matching, and use our region pooling method to obtain feature vectors. After all of the region proposals are scored by our regression models, we apply non-maximum suppression on those bounding boxes to generate the final detection result. First, bounding boxes are sorted by regression scores, and a greedy method is used to find the one with the highest score while removing others that overlap with the previously selected bounding box by more than 30%. The main steps of the testing phase are summarized in Figure 2.

## 4. Experimental Results

**Setup and Implementations.** We conduct experiments on the object detection task of PASCAL VOC 2007 dataset [13]. Training and validation sets are used for learning and performance is evaluated on the test set. We obtain the region proposals (around 2000 proposals for each image) that represent object parts by fast selective search [32], and extract appearance features of dense SIFT descriptors using the VLFeat toolbox [33]. We learn a 8192 dimensional codebook, and the SIFT histogram is then built by locality constrained linear coding [35] with 5 nearest neighbors and maximum pooling.[1] For extracting CNN features, we use the output of the seventh layer [22], where the CNN models are pre-trained as described in Krizhevsky's framework [24]. In the test stage, it takes 1 to 3 seconds (depending on the number of parts) to extract CNN features on a PC with 3.4GHz Core i7 CPU, and the rest takes around 0.55 seconds for testing each image on one model.

We first evaluate the recall rate of localizing objects with our coarse region matching approach. Second, the detection performance of the proposed algorithm is compared to several state-of-the-art methods. The experimental results demonstrate the advantages of using adaptive region pooling. We also show that our algorithm can be used to transfer object information to detected objects.

[1]DPM [14] with HOG features performs well against selective search [32] with SIFT features, and using HOG or SIFT features has pros and cons on different categories [32]. We use SIFT features to compare our region pooling method with standard SPM features.

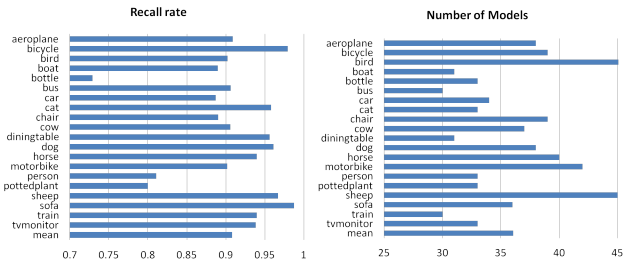



Figure 6. The left figure shows the recall rate for top 20 proposals selected by coarse region matching in the testing set; The right figure shows the number of models for each category.

**Object Localization.** Coarse region matching is one of the key parts of our algorithm, and it is used to restrict the training and testing samples that are similar to the exemplars. This step is specifically important to find useful region proposals while maintaining a high recall rate. We evaluate the quality of region proposals by calculating the recall rate for top $N$ regions selected by coarse region matching. We localize an object if the overlap between the ground truth bounding box and the selected region proposal is more than $50\%$.

We select $N$ with an experiment on the validation set. Table 2 shows the average recall rates for top $N = \{10, 20, \ldots, 50\}$ proposals. Although larger $N$ gives higher recall, it results in an increase in computational complexity. We select $N = 20$ for a good balance between accuracy and efficiency. This value achieves higher than $85\%$ recall rate for 17 out of 20 categories, with an average recall rate of $90.8\%$ on the testing set. Figure 6 shows the number of exemplars we use for training and the recall rate for each category. In average, only 36 exemplars from each category are used for learning models (less than $6\%$ of all the exemplars), which means that the total number of proposals that are evaluated for each image is approximately $36 * 20 = 720$. This is significantly smaller than the other approaches in the literature shown in Table 1. Specifically, the ESVM and LDA methods both use more than $10^6$ proposals. In the training phase, we use $N = 50$ to make a richer set of samples among positive images for the initial regression model.

**Object Detection.** Table 3 provides comparison of the detection mean Average Precisions (mAP) obtained by a set of algorithms for each category. For the spatial pyramid pooling (SPM), we extract SIFT features with two-layer grids ($3 \times 3$ in the lower level and 1 for the global one), resulting in a feature vector of 81920 dimensions. For a fair com-

Table 3. Detection mAP on the PASCAL VOC 2007 test set for each category. We compare the performance of our adaptive region pooling and the SPM features by using our algorithm. The first two rows show state-of-the-art exemplar-based approaches.

| | aeroplane | bicycle | bird | boat | bottle | bus | car | cat | chair | cow | dtable | dog | horse | motorbike | person | plant | sheep | sofa | train | tvmonitor | mean |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ESVM [29] | 20.4 | **40.7** | 9.3 | 10 | 10.3 | 31 | 40.1 | 9.6 | **10.4** | 14.7 | 2.3 | 9.7 | 38.4 | 32 | 19.2 | **9.6** | **16.7** | 11 | 29.1 | 31.5 | 19.8 |
| LDA [20] | 17.4 | 35.5 | **9.7** | **10.9** | **15.4** | 17.2 | **40.3** | 10.6 | 10.3 | 14.3 | 4.1 | 1.8 | **39.7** | 26 | **23.1** | 4.9 | 14.1 | 8.7 | 22.1 | 15.2 | 17.1 |
| Ours (SPM) | **35.5** | 36.5 | 6 | 7.3 | 3.4 | 30.1 | 38.8 | 25.1 | 5.9 | **16.4** | 19.6 | 14.7 | 25.9 | **32** | 15.4 | 2.3 | 15.7 | 24.4 | 32.5 | 28.2 | 20.8 |
| Ours (Region) | 31.5 | 37.7 | 5.5 | 7.9 | 5 | **33.5** | 37.3 | **32** | 5 | 13.8 | **27.2** | **15.4** | 25.6 | 31.7 | 13.8 | 1.3 | 16.2 | **28.3** | **34** | **31.7** | **21.7** |

parison, we use the adaptive region pooling with at most $L = 10$ parts for each exemplar, which results in a SIFT histogram of dimensionality that varies from $8192$ to $81920$. Table 3 shows that our pooling method performs favorably against SPM on most of the categories. In some categories such as dining table and sofa, the proposed pooling method achieves significant improvement. This is because these objects have better segmentation to represent parts or have strong region structure to encode the part information of objects. This also indicates that the performance of adaptive region pooling algorithm can be significantly improved if there is a good segmentation algorithm for other categories.

We also compare our method to state-of-the-art exemplar-based approaches where context cues are not used for any of the algorithms. Both of our results that use different feature representations provide a higher mean mAP than Exemplar SVMs [29] and LDA models [20]. Although the LDA models perform well on some categories, its mAP is the lowest. Our proposed approach achieves the best result in $10$ categories, and outperforms the other two methods with a large margin in several categories.

As shown in Table 3, our algorithm performs well on categories with rigid objects such as train, sofa and aeroplane. This is not surprising as our region-based models rely on the segmentation, and usually it is a simpler task to segment rigid objects. However, our method does not perform very well on categories such as bottle, person and plant. For these categories, we find that either the recall rate for region proposals is much lower than the others (see Figure 6) or the number of positive images is small. In addition, our method also performs well on non-rigid objects such as cat and dog. It indicates that our region pooling approach can handle deformable objects well by utilizing part information.

**Object Detection with CNN Features.** Our algorithm is capable of using any powerful representation such as CNN features [39, 30, 15] to achieve better detection results. To accomplish this, we only replace the features in the region pooling stage and keep all the other steps and parameters the same as in the previous experiment in Table 3. Instead of pooling SIFT features in each part, we use the bounding box of each part as the input to CNN models to obtain features. Then we concatenate these part features into one feature vector. Note that parts are still obtained in the same way as the procedure described in Figure 4.

Table 4 shows the results compared with other state-of-the-art methods.[2] Our method performs favorably against methods that utilize CNN features (we compare the best results of [39, 30] without bounding box regression). We show that our method obtains better performance in $10$ categories and achieves better mean mAP. Note that the recent work of [15] provides better precision, but this method is not exemplar-based and does not exploit the object structure. Our method is the only one that has the ability to transfer the object information in Table 4.

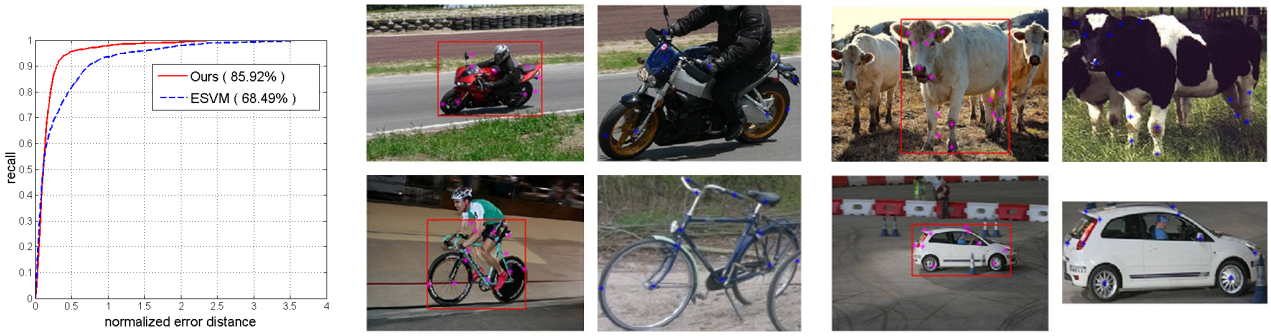
Here we present the first exemplar-based method that achieves state-of-the-art results. A possible reason that previous exemplar-based approaches cannot perform well against state-of-the-art methods is due to the combination of several weak models. One way to improve the performance is to use more powerful features. Our algorithm allows for this and is designed to flexibly adopt any kind of feature type. This is shown in Table 3 and 4, where the mean mAP for DPM improves less than $10\%$ with CNN features, while our algorithm improves more than $20\%$ using CNN features.

**Object Transfer via Exemplars.** Like other exemplar-based approaches [29, 20, 16], our algorithm provides an application to transfer similar exemplars to the detected object. In addition, since each of the exemplars is segmented into parts, both the object mask and the part information can be transferred. For each test image, we select the exemplar whose model assigns the highest score, so that this exemplar includes the most similar part information to the detected object. Then the same approach for adaptive region pooling is used to resize the part mask and to apply it

[2] We obtain the selective search performance by reading the figure in [32].

Table 4. Detection mAP on the PASCAL VOC 2007 test set for each category. We compare our adaptive region pooling method with CNN features to state-of-the-art methods. Note that only our algorithm is an exemplar-based approach that can transfer object information.

| | aeroplane | bicycle | bird | boat | bottle | bus | car | cat | chair | cow | dtable | dog | horse | motorbike | person | plant | sheep | sofa | train | tvmonitor | mean |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DPMv5 [14] | 33.2 | 60.3 | 10.2 | 16.1 | 27.3 | 54.3 | **58.2** | 23 | 20 | 24.1 | 26.7 | 12.7 | 58.1 | 48.2 | 43.2 | 12 | 21.1 | 36.1 | 46 | 43.5 | 33.7 |
| Selective Search [32] | 43.5 | 46.5 | 10.4 | 12 | 9.3 | 49.4 | 53.7 | 39.4 | 12.5 | 36.9 | 42.2 | 26.4 | 47 | 52.4 | 23.5 | 12.1 | 29.9 | 36.3 | 42.2 | 48.8 | 33.7 |
| Regionlets (CNN) [39] | 44.6 | 55.6 | 24.7 | 23.5 | 6.3 | 49.4 | 51 | **57.5** | 14.3 | 35.9 | 45.9 | 41.3 | **61.9** | 54.7 | **44.1** | 16 | 28.6 | 41.7 | **63.2** | 44.2 | 40.2 |
| DPM (CNN) [30] | 39.7 | 59.5 | **35.8** | 24.8 | **35.5** | 53.7 | 48.6 | 46 | **29.2** | 36.8 | 45.5 | 42 | 57.7 | 56 | 37.4 | **30.1** | 31.1 | **50.4** | 56.1 | 51.6 | 43.4 |
| Ours (CNN) | **58.1** | **60.6** | 31 | **29.3** | 17.8 | **61** | 56.1 | 55.9 | 18.1 | **42.3** | **52.9** | **46.9** | 52 | **58** | 32.7 | 20.3 | **43.7** | 46.6 | 53.2 | **57.6** | **44.7** |




(a) (b) (c)

Figure 7. Keypoints transfer results on the PASCAL VOC 2007 dataset. Figure (a) shows the recall-error curve comparing to the ESVM method. The number in the legend indicates the recall rate when the error distance is 0.25. Figure (b) and (c) visualize keypoint transfer results. For each pair of the result, the right figure is the exemplar (keypoints marked in blue) that transfers keypoints to the detected object (keypoints marked in pink) in the left figure. Best viewed in color with enlarged images.

on the target region.

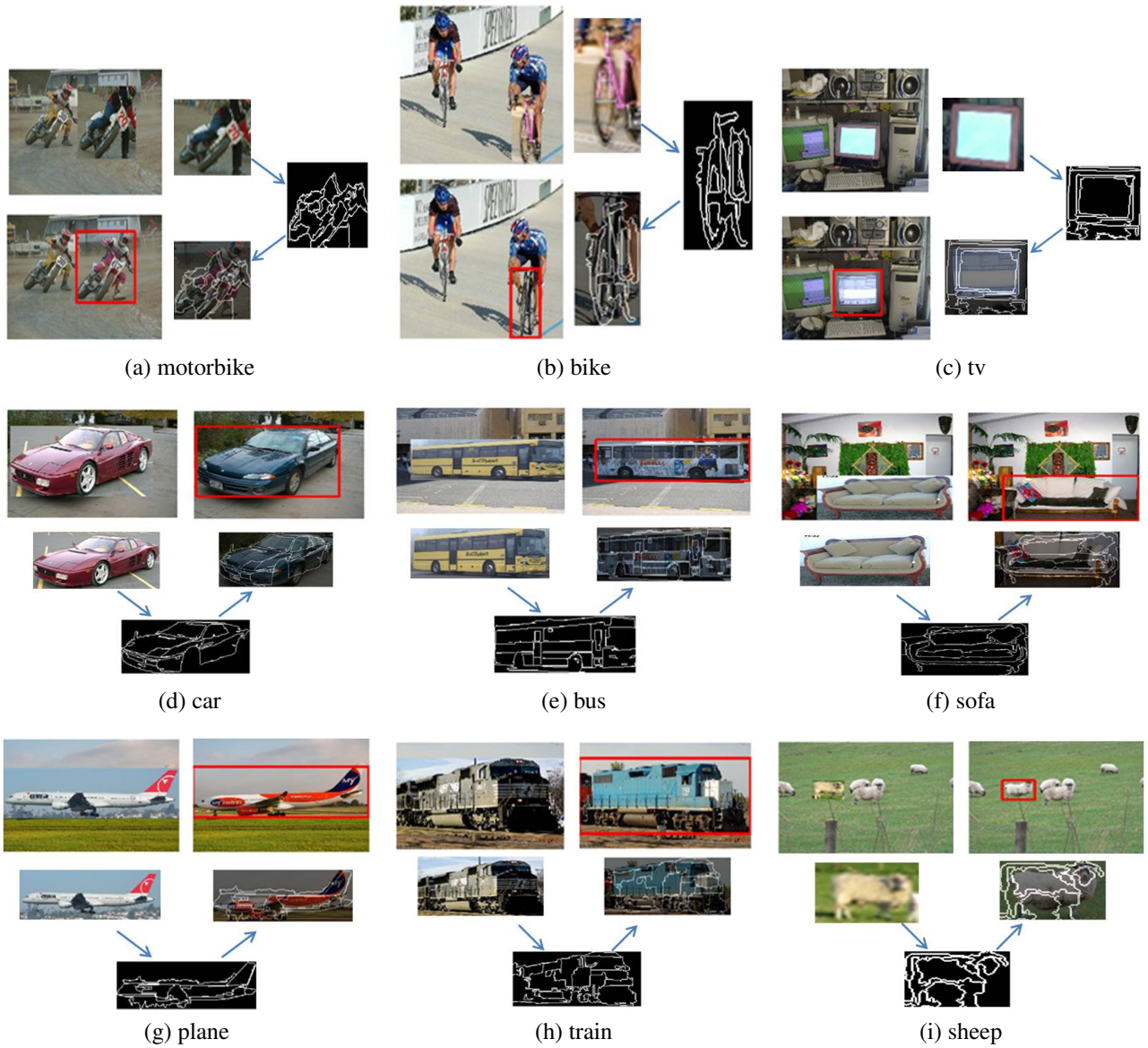
To evaluate the quality of transferring object information, we provide both quantitative and qualitative results. First, we evaluate on the keypoint annotation dataset of [6]. We use the keypoint annotations in the dataset as test ground truths, and manually annotate keypoints for training images (both images are in the PASCAL VOC 2007). Note that these keypoint annotations are only used for evaluation rather than in the training stage. Figure 7 shows the recall rate versus the normalized error distance [38] in average for all the test images. Since keypoint correspondence between ground truths and transferred keypoints is not one-to-one, error distances are computed between the transferred keypoint and its nearest ground truth annotation.

As shown in Figure 7, ESVM [29] and our method perform competitively when the error distance is small, while our method achieves significantly better recall rates for larger error distances. It indicates that our method can handle more difficult cases when transferring object information. For instance, when the error distance is 0.25, our method achieves 85.9% recall rate, while ESVM obtains only 68.5%. We also apply another metric where we only consider correctly detected objects. It gives us a similar recall rate when the error distance is 0.25, where ours is 87.9% against ESVM's 73.2%.

The recall-error curve shows that our algorithm that only uses a very small subset of training exemplars can achieve better keypoint transfer results than ESVM that uses all the exemplars. Figure 7 also shows some results of transferred keypoints on detected objects. In addition, Figure 8 presents the visualized results of transferred object mask and parts. Part information are well fit in the detected object, indicating that pooling features via parts helps the matching between regions with similar structure. More results are in the supplementary material.

## 5. Conclusions and Future Work

In this paper, we propose a novel object detection algorithm which utilizes non-rectangular regions as parts and multiple region-based exemplar SVRs. The adaptive region pooling method extracts features that accounts for the structure of object parts, which facilitates handling the large variation of objects. We develop a coarse region matching that efficiently selects samples, ensures the model generalization



(a) motorbike (b) bike (c) tv

(d) car (e) bus (f) sofa

(g) plane (h) train (i) sheep

Figure 8. Our region-based models enable the application to transfer object masks and parts to detected objects via exemplars. From (a) to (i), the red bounding box is the detected object and the top-left figure shows the transferred mask. In addition, part information of the exemplar is transferred to the detected object, which has a similar region structure with the exemplar. Best viewed in color with enlarged images.

for learning, and rejects easy false positives for testing. Our algorithm performs favorably on the PASCAL VOC 2007 dataset for object detection. The results show that our pooling approach achieves better performance than the conventional SPM method. We also illustrate that our method is flexible to use any features such as CNN features to achieve state-of-the-art results. Finally, we present the application to transfer object keypoints and parts from the exemplars to the detected objects. Both the quantitative and qualitative results demonstrate the benefits of our algorithm in transferring object information.

Our method explores a new area between part-based and exemplar-based models with region proposals. It is of great interest to apply our adaptive region pooling method on other vision problems to see how region or part information can help recognition. Moreover, adding limited supervised information for finding representative exemplars or learning better parts should boost the performance, which is still scalable to extended datasets. Non-rigid object models [19] and 3D CAD models [3, 28] can also be used to generalize the application of transferring rigid and non-rigid object information. Additional geometric information, such as object poses or parts in 3D, can be aligned with detected objects.

## Acknowledgments

This work is supported in part by the NSF CAREER Grant #1149783 and NSF IIS Grant #1152576.

## References

[1] B. Alexe, T. Deselaers, and V. Ferrari. What is an object? In *CVPR*, 2010. 1

[2] P. Arbelaez, B. Hariharan, C. Gu, S. Gupta, L. Bourdev, and J. Malik. Semantic segmentation using regions and parts. In *CVPR*, 2012. 1

[3] M. Aubry, D. Maturana, A. Efros, B. Russell, and J. Sivic. Seeing 3d chairs: exemplar part-based 2d-3d alignment using a large dataset of cad models. In *CVPR*, 2014. 8

[4] M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *NIPS*, 2001. 3

[5] L. Bourdev and J. Malik. Poselets: Body part detectors trained using 3d human pose annotations. In *ICCV*, 2009. 1, 2, 3

[6] T. Brox, L. Bourdev, S. Maji, and J. Malik. Object segmentation by alignment of poselet activations to image contours. In *CVPR*, 2011. 7

[7] J. Carreira, R. Caseiro, J. Batista, and C. Sminchisescu. Semantic segmentation with second-order pooling. In *ECCV*, 2012. 2

[8] J. Carreira and C. Sminchisescu. Constrained parametric min-cuts for automatic object segmentation. In *CVPR*, 2010. 1

[9] X. Chen, R. Mottaghi, X. Liu, S. Fidler, R. Urtasun, and A. Yuille. Detect what you can: Detecting and representing objects using holistic models and body parts. In *CVPR*, 2014. 1

[10] S. K. Divvala, A. A. Efros, and M. Hebert. How important are deformable parts in the deformable parts model? In *ECCV*, 2012. 1

[11] I. Endres and D. Hoiem. Category independent object proposals. In *ECCV*, 2010. 1

[12] I. Endres, K. J. Shih, J. Jiaa, and D. Hoiem. Learning collections of part models for object recognition. In *CVPR*, 2013. 1

[13] M. Everingham, L. J. V. Gool, C. K. I. Williams, J. M. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 88(2):303–338, 2010. 4, 5

[14] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramana. Object detection with discriminatively trained part based models. *PAMI*, 32(9):1627–1645, 2010. 1, 5, 7

[15] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2013. 2, 6

[16] C. Gu, P. Arbeláez, Y. Lin, K. Yu, and J. Malik. Multi-component models for object detection. In *ECCV*, 2012. 1, 2, 3, 6

[17] C. Gu, J. Lim, P. Arbelaez, and J. Malik. Recognition using regions. In *CVPR*, 2009. 1

[18] C. Gu and X. Ren. Discriminative mixture-of-templates for viewpoint classification. In *ECCV*, 2010. 1

[19] O. C. Hamsici, P. F. Gotardo, and A. M. Martinez. Learning spatially-smooth mappings in non-rigid structure from motion. In *ECCV*, 2012. 8

[20] B. Hariharan, J. Malik, and D. Ramanan. Discriminative decorrelation for clustering and classification. In *ECCV*, 2012. 1, 2, 3, 6

[21] M. Hoai and A. Zisserman. Discriminative sub-categorization. In *CVPR*, 2013. 1

[22] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014. 5

[23] M. Juneja, A. Vedaldi, C. V. Jawahar, and A. Zisserman. Blocks that shout: Distinctive parts for scene classification. In *CVPR*, 2013. 1

[24] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 5

[25] T. Lan, M. Raptis, L. Sigal, and G. Mori. From subcategories to visual composites: A multi-level framework for object detection. In *ICCV*, 2013. 1

[26] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006. 1

[27] F. Li, J. Carreira, and C. Sminchisescu. Object recognition as ranking holistic figure-ground hypotheses. In *CVPR*, 2010. 1, 2

[28] J. Lim, H. Pirsiavash, and A. Torralba. Parsing ikea objects: Fine pose estimation. In *ICCV*, 2013. 8

[29] T. Malisiewicz, A. Gupta, and A. Efros. Ensemble of exemplar-svms for object detection and beyond. In *ICCV*, 2011. 1, 2, 6, 7

[30] P.-A. Savalle, S. Tsogkas, G. Papandreou, and I. Kokkinos. Deformable part models with cnn features. In *ECCV workshop*, 2014. 2, 6, 7

[31] S. Singh, A. Gupta, and A. A. Efros. Unsupervised discovery of mid-level discriminative patches. In *ECCV*, 2012. 1

[32] K. van de Sande, J. Uijlingsy, T. Gevers, and A. Smeulders. Segmentation as selective search for object recognition 23:59:59 utc. In *ICCV*, 2011. 1, 5, 6, 7

[33] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms. `http://www.vlfeat.org/`, 2008. 5

[34] S. Vijayanarasimhan and K. Grauman. Efficient region search for object detection. In *CVPR*, 2011. 1

[35] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Locality-constrained linear coding for image classification. In *CVPR*, 2010. 5

[36] X. Wang, M. Yang, S. Zhu, and Y. Lin. Regionlets for generic object detection. In *ICCV*, 2013. 1

[37] J. Yang, Y.-H. Tsai, and M.-H. Yang. Exemplar cut. In *ICCV*, 2013. 1

[38] X. Zhu and D. Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *CVPR*, 2012. 7

[39] W. Y. Zou, X. Wang, M. Sun, and Y. Lin. Generic object detection with dense neural patterns and regionlets. In *BMVC*, 2014. 2, 6, 7