Recognition using Regions [*]

Chunhui Gu, Joseph J. Lim, Pablo Arbeláez, Jitendra Malik
University of California at Berkeley
Berkeley, CA 94720
{chunhui,lim,arbelaez,malik}@eecs.berkeley.edu

Abstract

This paper presents a unified framework for object detection, segmentation, and classification using regions. Region features are appealing in this context because: (1) *they encode shape and scale information of objects naturally;* (2) *they are only mildly affected by background clutter.*

Regions have not been popular as features due to their sensitivity to segmentation errors. In this paper, we start by producing a robust bag of overlaid regions for each image using Arbeláez et al.*, CVPR* 2009*. Each region is represented by a rich set of image cues (shape, color and texture). We then learn region weights using a max-margin framework. In detection and segmentation, we apply a generalized Hough voting scheme to generate hypotheses of object locations, scales and support, followed by a verification classifier and a constrained segmenter on each hypothesis.*

*The proposed approach significantly outperforms the state of the art on the ETHZ shape database (*87.1% *average detection rate compared to Ferrari* et al.*'s* 67.2%*), and achieves competitive performance on the Caltech* 101 *database.*

1. Introduction

Ever since the early work on face detection in the late 90s ([28], [32]), the dominant strategy for object detection in a scene has been multi-scale scanning. A fixed size and shape window is swept across the image, and the contents of the window are input to a classifier which gives an answer to the question: is there an instance of object category C (face, car, pedestrian, etc.) in the window? To find objects of different sizes, the image is sub-sampled in a pyramid, typically with neighboring levels being a quarter octave ($\sqrt[4]{2}$) apart. This strategy continues to hold in recent papers, such as [7] on pedestrian detection and [10] on the PASCAL challenge. Various speed-ups have been offered over time, ranging from cascades [32], branch and bound strategies [18] to more efficient classifier evaluation [23].

Yet, there is something profoundly unsatisfying about this approach. First of all, classification of a window as containing, say, a horse, is not the same as segmenting out the pixels corresponding to a horse from the background. Hence, some post-process relying on quite different cues would be required to achieve that goal. Secondly, the brute-force nature of window classification is not particularly appealing. Its computational complexity is proportional to the product of the number of scales, locations, and categories. Thirdly (and this may matter more to some than to others), it differs significantly from the nature of human visual detection, where attention is directed to certain locations based on low-level salience as well as high-level contextual cues, rather than uniformly to all locations.

So what is the alternative? The default answer going back to the Gestalt school of visual perception, is in "perceptual organization". Low and middle level vision furnishes the entities on which recognition processes can operate. We then have a choice of what these entities should be: points, curves or regions? Over the last decade, low-level interest point-based features, as proposed by [30] and [21], have tended to dominate the discourse. The computer vision community, by and large, didn't have faith in the ability of generic grouping processes to deliver contours or regions of sufficiently high accuracy for recognition.

Our belief is that recent advances in contour [22] and region detection [2] make this a propitious time to build an approach to recognition using these more spatially extended and perceptually meaningful entities. This paper focuses on using regions, which have some pleasant properties (1) they encode shape and scale information of objects naturally; (2) they specify the domains on which to compute various features, without being affected by clutter from outside the region.

While definitely a minority trend, there has been some relevant work in the last decade using regions/segments which we review briefly. [16] estimates the 3D geometric context of a single image by learning local appearance and

[*] This work is funded by ARO MURI W911NF-06-1-0076, ONR MURI N00014-06-1-0734, and the Haas Scholars Fellowship.

geometric cues on super-pixels. [29] uses a normalized cut-based multi-layer segmentation algorithm to identify segmented objects. This line of work suffers initially from unreliable regions produced by their segmentation methods. The work from [25] and [31] is most similar to our approach. However, in addition to the problem of unstable regions, [25] takes regions as whole bodies of objects and ignores local parts, while [31] represents objects as region trees but also exploits structural cues of the trees for matching and such cues may not be reliable.

Starting with regions as the basic elements of our approach, we use a generalized Hough-like voting strategy for generating hypotheses of object location, scale and support. Here, we are working in a long-standing tradition in computer vision [8, 3, 21, 20, 27, 24].

The rest of this paper is organized as follows. Section 2 overviews our method and describes the use of regions as elementary units. Section 3 describes a discriminative learning framework for region weighting. Section 4 describes our main recognition algorithm which has three stages: (1) voting, (2) verification, and (3) segmentation. We show our experimental results in Section 5, and conclude in Section 6. Figure 1 shows some of our final detection and segmentation results.

2. Overview of the Approach

The pipeline of our region-based recognition framework is as follows: first, each image is represented by a bag of regions derived from a region tree as shown in Figure 2. Regions are described by a rich set of cues (shape, color and texture) inside them. Next, region weights are learned using a discriminative max-margin framework. After that, a generalized Hough voting scheme is applied to cast hypotheses of object locations, scales, and support, followed by a refinement stage on these hypotheses which deals with detection and segmentation separately.

2.1. Region Extraction

We start by constructing a region tree using the hierarchical segmentation engine of [2]. The regions we consider are the nodes of that tree, including the root which is the entire image. We use them as the basic entities for our approach.

Figure 2 presents an example of our region trees, as well as a bag of regions representing the input image.

2.2. Region Description

We describe a region by subdividing evenly its bounding box into an $n \times n$ grid, as illustrated in Figure 3. In the experiments reported, we use $n = 4$. Each cell encodes information only inside the region. We capture different region cues from the cells, and each type of cue is encoded by concatenating cell signals into a histogram. In this paper, we consider the following region cues:

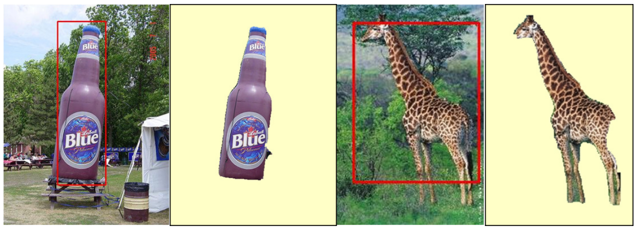

Figure 1. Detection and segmentation results on two examples in the ETHZ shape database using our unified approach.

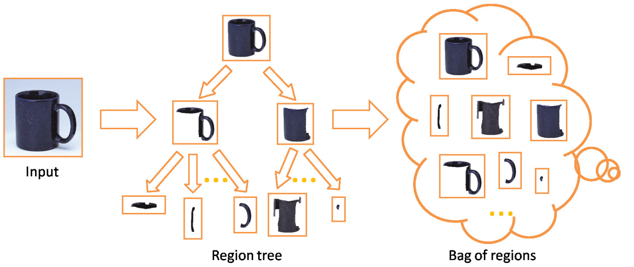

Figure 2. The “bag of regions” representation of a mug example. Regions are collected from all nodes of a region tree generated by [2]. Therefore, these regions range in scale from super pixels to the whole image. Note that here “bag” implies discarding tree structure.

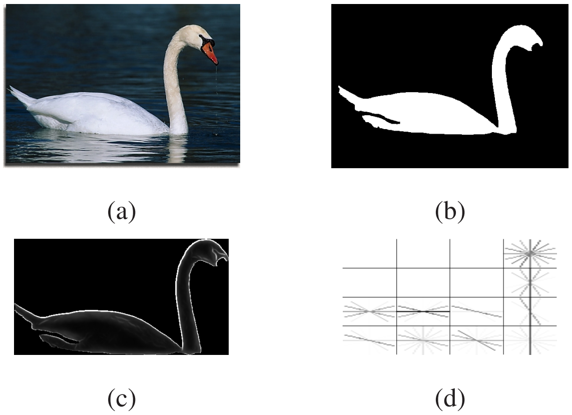

Figure 3. The “contour shape” region descriptor. (a) Original image, (b) A region from the image, (c) gPb representation of the region in (b), (d) Our contour shape descriptor based on (c). Descriptors using other image cues are computed in the same manner.

- Contour shape, given by the histogram of oriented responses of the contour detector gPb [22]
- Edge shape, where orientation is given by local image gradient (computed by convolution with a $[-1\ 0\ 1]$ filter along x- and y-axes). This captures high frequency information (*e.g.* texture), while gPb is designed to suppress it.
- Color, represented by the L^*, a and b histograms in the

CIELAB color space

- Texture, described by texton histograms

Distances between histograms of region cues are characterized using χ^2 measure.

Our region representation has several appealing properties. Firstly, the scale invariant nature of region descriptors enables us to compare regions regardless of their relative sizes. Secondly, background clutter interferes with region representations only mildly compared to interest point descriptors. Thirdly, our region descriptor inherits insights from recent popular image representations such as GIST [26], HOG [7] and SIFT [21]. At the coarsest scale, where the region is the root of the tree, our descriptor is similar to GIST. At the finest scale, when the regions are the leaves of the tree, our representation resembles the SIFT descriptor.

3. Discriminative Weight Learning

Not all regions are equally significant for discriminating an object from another. For example, wheel regions are more important than uniform patches to distinguish a bicycle from a mug. Here, we adapt the framework of [13] for learning region weights. Given an exemplar $\mathcal{I}$ containing one object instance and a query $\mathcal{J}$, denote $f_i^{\mathcal{I}}, i = 1, 2, \ldots, M$ and $f_j^{\mathcal{J}}, j = 1, 2, \ldots, N$ their bags of region features.

The distance from $\mathcal{I}$ to $\mathcal{J}$ is defined as:

$$\mathcal{D}(\mathcal{I} \to \mathcal{J}) = \sum_{i=1}^{M} w_i^{\mathcal{I}} d_i^{\mathcal{I}\mathcal{J}} = \langle w^{\mathcal{I}}, d^{\mathcal{I}\mathcal{J}} \rangle, \tag{1}$$

where $w_i^{\mathcal{I}}$ is the weight for feature $f_i^{\mathcal{I}}$, and

$$d_i^{\mathcal{I}\mathcal{J}} = \min_j d(f_i^{\mathcal{I}}, f_j^{\mathcal{J}}) \tag{2}$$

is the elementary distance between $f_i^{\mathcal{I}}$ and the closest feature in $\mathcal{J}$. Note that the exemplar-to-query distance is asymmetric, *i.e.*, $\mathcal{D}(\mathcal{I} \to \mathcal{J}) \neq \mathcal{D}(\mathcal{J} \to \mathcal{I})$.

In the weight learning stage, supposing $\mathcal{I}$ is an object of category $\mathcal{C}$, we find a pair of $\mathcal{J}$ and $\mathcal{K}$ such that $\mathcal{J}$ is an object of the same category $\mathcal{C}$ and $\mathcal{K}$ is an object of a different category. The learning algorithm enforces the following condition:

$$\begin{aligned} \mathcal{D}(\mathcal{I} \to \mathcal{K}) &> \mathcal{D}(\mathcal{I} \to \mathcal{J}) & (3)\\ \Longrightarrow \langle w^{\mathcal{I}}, d^{\mathcal{I}\mathcal{K}} \rangle &> \langle w^{\mathcal{I}}, d^{\mathcal{I}\mathcal{J}} \rangle & (4)\\ \Longrightarrow \langle w^{\mathcal{I}}, x^{\mathcal{I}\mathcal{J}\mathcal{K}} \rangle &> 0, & (5) \end{aligned}$$

where $x^{\mathcal{I}\mathcal{J}\mathcal{K}} = d^{\mathcal{I}\mathcal{K}} - d^{\mathcal{I}\mathcal{J}}$. Supposing we construct T such pairs for $\mathcal{I}$ from the training set, thus $x_1, x_2, \ldots, x_T$ (we dropped the superscripts for clarity). The large-margin optimization is formulated as follows:

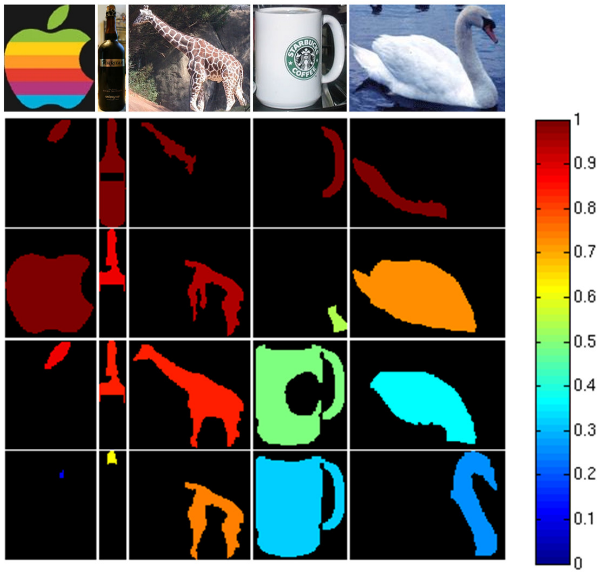

Figure 4. Weight learning on regions. For each column, the top image is the exemplar, and the bottom four are regions in order of highest learned weight. Note that the most discriminative regions (leaf and body of the apple logo, handle of the mug) have the highest weights from learning. (best viewed in color)

$$\begin{aligned} \min_{w,\xi} \quad & \frac{1}{2} w^T w + C \sum_{i=1}^{T} \xi_i & (6)\\ s.t. : \quad & w^T x_i \geq 1 - \xi_i, \xi_i \geq 0, \forall i = 1, 2, \ldots, T & (7)\\ & w \succeq 0. & (8) \end{aligned}$$

When integrating multiple cues for a single region, we learn one weight for each cue. Figure 4 shows some examples of learned weights on regions when contour shape cue is used.

As in [13], we model the probability of query $\mathcal{J}$ being in the the same category as exemplar $\mathcal{I}$ by a logistic function:

$$p(\mathcal{I}, \mathcal{J}) = \frac{1}{1 + \exp[-\alpha_{\mathcal{I}} \mathcal{D}(\mathcal{I} \to \mathcal{J}) - \beta_{\mathcal{I}}]} \tag{9}$$

where $\alpha_{\mathcal{I}}$ and $\beta_{\mathcal{I}}$ are parameters learned in training.

4. Detection and Segmentation Algorithms

Our unified object recognition framework contains three components: voting, verification and segmentation. For a given query image, the voting stage casts initial hypotheses of object positions, scales and support based on region matching. These hypotheses are then refined through a verification classifier and a constrained segmenter, respectively, to obtain final detection and segmentation results. Figure

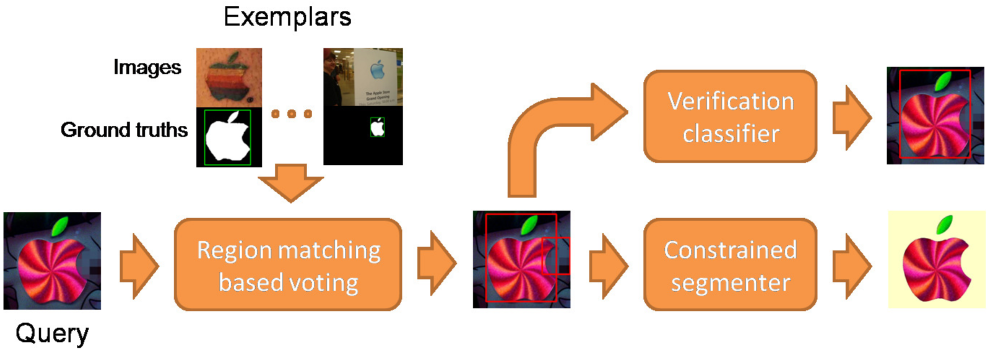

Figure 5. The pipeline of our object recognition algorithm consist of three stages. For an input query image, the voting stage casts initial hypotheses of object positions, scales and support based on matched regions from exemplars. These hypotheses are the inputs of the next stages and are refined through a verification classifier and a constrained segmenter, respectively, to obtain final detection and segmentation results. Figure 6 describes details of the voting stage, and Figure 7 illustrates the segmentation pathway.

5 depicts the pipeline of our recognition algorithms for the apple logo category. The query image is matched to each apple logo exemplar in the training set, whose ground truth bounding boxes and support masks are both given as inputs. All region weights are determined as in Section 3.

4.1. Voting

The goal here, given a query image and an object category, is to generate hypotheses of bounding boxes and (partial) support of objects of that category in the image. To achieve it, we use a generalized Hough voting scheme based on the transformation between matched regions as well as the associated objects in the exemplars.

Specifically, given exemplar $\mathcal{I}$, its ground truth bounding box $B^{\mathcal{I}}$ and support mask $M^{\mathcal{I}}$, we match a region $R^{\mathcal{I}}$ in $\mathcal{I}$ to another region $R^{\mathcal{J}}$ in query $\mathcal{J}$. Then the vote for the bounding box $\hat{B}$ of the object in $\mathcal{J}$ is characterized by:

$$\theta_{\hat{B}} = \mathcal{T}(\theta_{B^{\mathcal{I}}} | \, \theta_{R^{\mathcal{I}}}, \theta_{R^{\mathcal{J}}}) \tag{10}$$

where $\theta = [x, y, s_x, s_y]$ characterizes the center coordinates $[x, y]$ and the scales $[s_x, s_y]$ of a region or bounding box, and $\mathcal{T}$ is some pre-defined transformation function with its parameters derived by the matched regions $\theta_{R^{\mathcal{I}}}$ and $\theta_{R^{\mathcal{J}}}$.

A voting score is also assigned to each box by combining multiple terms:

$$S_{vot}(\hat{B}) = \tilde{w}_{R^{\mathcal{I}}} \cdot g(d_{R^{\mathcal{I}}}, d_{R^{\mathcal{J}}}) \cdot h(R^{\mathcal{I}}, R^{\mathcal{J}}) \tag{11}$$

where $\tilde{w}_{R^{\mathcal{I}}}$ is the learned weight of $R^{\mathcal{I}}$ after normalization, $g(d_{R^{\mathcal{I}}}, d_{R^{\mathcal{J}}})$ characterizes similarity between descriptors $d_{R^{\mathcal{I}}}$ and $d_{R^{\mathcal{J}}}$, and $h(R^{\mathcal{I}}, R^{\mathcal{J}})$ penalizes region shape differences between two regions.

In general, $\mathcal{T}$ in Eqn.10 can be any given transformation function. In our experiments, we restrict our transformation model to allow only translation and scaling in both x- and y-axes. Thus, in the x-direction:

$$x^{\hat{B}} = x^{R^{\mathcal{J}}} + (x^{B^{\mathcal{I}}} - x^{R^{\mathcal{I}}}) \cdot s_x^{R^{\mathcal{J}}} / s_x^{R^{\mathcal{I}}} \tag{12}$$

$$s_x^{\hat{B}} = s_x^{B^{\mathcal{I}}} \cdot s_x^{R^{\mathcal{J}}} / s_x^{R^{\mathcal{I}}} \tag{13}$$

and same equations apply to the y-direction. Figure 6 illustrates such generalized Hough voting based on a pair of matched regions.

Eqn.11, 12 and 13 summarizes bounding box voting between one pair of matched regions. An early rejection is applied to the voted box either if its voting score is too low or if the box is (partially) outside the image. For all matched regions between a query $\mathcal{J}$ and all exemplars of one category, we generate a set of bounding boxes accordingly for objects of that category in $\mathcal{J}$ for each pair of regions. Finally, we cluster these bounding boxes by a mean-shift [6] algorithm in the feature space θ_B. Here, we favor mean-shift over other clustering methods because it allows adaptive bandwidth setting for different clusters. Thus, two large bounding boxes are more likely to merge than two small boxes if they differ in the same amount in the feature space.

One main advantage of this voting algorithm based on region matching is that it can recover the full support of an object if only a small fraction of that object (*e.g.*, the leaf of the apple logo or the handle of the mug) is matched. It gives not only position but also reliable scale estimation of the bounding boxes. It also allows for aspect ratio deformation of bounding boxes during transformation.

4.2. Verification

A verification classifier is applied to each bounding box hypothesis from voting. In general, any object model, *e.g.*, [10] and [23], can be applied to each hypothesis. However, in order to fully exploit the use of region representation, we follow the method of [13] using the region weights derived in Section 3.

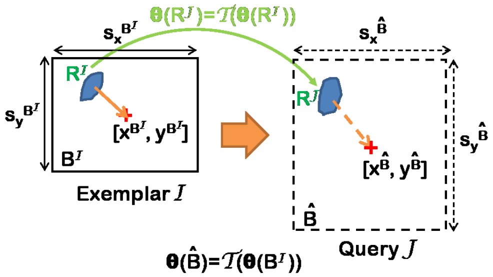
$\theta(R^{\mathcal{J}})=\mathcal{T}(\theta(R^{\mathcal{I}}))$
$s_x^{B^{\mathcal{I}}}$
$s_x^{\hat{B}}$
$R^{\mathcal{I}}$
$s_y^{B^{\mathcal{I}}}$
$[x^{B^{\mathcal{I}}}, y^{B^{\mathcal{I}}}]$
$B^{\mathcal{I}}$
Exemplar $\mathcal{I}$
$R^{\mathcal{J}}$
$s_y^{\hat{B}}$
$[x^{\hat{B}}, y^{\hat{B}}]$
$\hat{B}$
Query $\mathcal{J}$
$\theta(\hat{B})=\mathcal{T}(\theta(B^{\mathcal{I}}))$

Figure 6. Voting stage. This shows a Hough voting scheme based on region matching using a specific transformation function. $\theta = [x, y, s_x, s_y]$ includes the center coordinates $[x, y]$ and the scales $[s_x, s_y]$ of a bounding box. $\mathcal{T}$ transforms a ground truth bounding box $B^{\mathcal{I}}$ of $R^{\mathcal{I}}$ to a new bounding box $\hat{B}$ of $R^{\mathcal{J}}$ based on matching between $R^{\mathcal{I}}$ and $R^{\mathcal{J}}$. This transformation provides not only position but also scale estimation of the object. It also allows for aspect ratio deformation of bounding boxes.

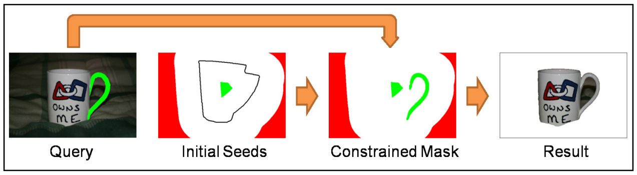

Figure 7. Segmentation stage. The initial seeds (green for object and red for background) are derived from transformation of the exemplar mask (with black boundary). The constrained mask is a combination of the seeds and the matched part (mug handle in this case). Note that our method is able to recover the complete object support from one of its parts.

The verification score of a bounding box $\hat{B}$ with respect to category C is defined as the average of the probabilities of $\hat{B}$ to all exemplars of category C:

$$S_{ver}(\hat{B}) = \frac{1}{N}\sum_{i=1}^{N} p(\mathcal{I}_{c_i}, \hat{B}) \tag{14}$$

where $\mathcal{I}_{c_1}, \mathcal{I}_{c_2}, \ldots, \mathcal{I}_{c_N}$ are all exemplars of category C, and $p(\mathcal{I}_{c_i}, \hat{B})$ are computed using Eqn.9. The overall detection score $S_{det}(\hat{B})$ of $\hat{B}$ for category C is a combination of the voting score $S_{vot}(\hat{B})$ and the verification score $S_{ver}(\hat{B})$, for instance, the product of the two:

$$S_{det}(\hat{B}) = S_{vot}(\hat{B}) \cdot S_{ver}(\hat{B}) \tag{15}$$

4.3. Segmentation

The segmentation task we consider is that of precisely extracting the support of the object. It has been addressed in the past by techniques such as OBJ CUT [17]. In our framework, the region tree is the result of bottom-up processing; top-down knowledge derived from the matched exemplar is used to mark some of the leaves of the region tree as definitely belonging to the object, and some others as definitely background. We propagate these labels to the rest of the leaves using the method of [1], thus getting the benefit of both top-down and bottom-up processing.

More precisely, let $\mathcal{I}$, $M^{\mathcal{I}}$ and $B^{\mathcal{I}}$ be the exemplar, its ground truth support mask and bounding box, respectively. Then, for a region $R^{\mathcal{I}}$ in $\mathcal{I}$ and one of its matching region $R^{\mathcal{J}}$ in the query image $\mathcal{J}$, we compute $\mathcal{T}(M^{\mathcal{I}})$, the transformation of the ground truth mask $M^{\mathcal{I}}$ on $\mathcal{J}$. $\mathcal{T}(M^{\mathcal{I}})$ provides an initial top-down guess for the location, scale and shape of the object in $\mathcal{J}$. Its complement provides the top-down guess for the background. Since we do not want to have the segmentation be completely determined by these top-down guesses, we allow for a zone of “don’t know” pixels in a fixed neighborhood of the boundary of the transformed exemplar mask, and consider as the priors for object and background only pixels greater than a given Euclidean distance from the boundary of the projected ground truth mask $\mathcal{T}(M^{\mathcal{I}})$. Since we have the constraint that the whole matched region $R^{\mathcal{J}}$ must be part of the object, we union this with the object mask to produce the “constrained mask”.

Thus, we construct a segment $\mathcal{M}$ on the query by using both the exemplar mask and the low-level information of the query image, as illustrated in Figure 7. As an early rejection test, we compute the overlap between $\mathcal{M}$ and the transformed mask $\mathcal{T}(M^{\mathcal{I}})$, and discard it if the score is low.

We also assign a score $S_{seg}(\mathcal{M})$ to $\mathcal{M}$ based on matched regions $R^{\mathcal{I}}$ and $R^{\mathcal{J}}$:

$$S_{seg}(\mathcal{M}) = \tilde{w}_{R^{\mathcal{I}}} \cdot g(d_{R^{\mathcal{I}}}, d_{R^{\mathcal{J}}}) \tag{16}$$

where $\tilde{w}_{R^{\mathcal{I}}}$ and $g(d_{R^{\mathcal{I}}}, d_{R^{\mathcal{J}}})$ are defined in Section 4.1. Thus, we define the confidence map of $\mathcal{J}$ to $\mathcal{I}$ based on $R^{\mathcal{I}}$ as the maximal response of each region in $\mathcal{J}$. The final confidence map for $\mathcal{J}$ for a given category is the double summation of these confidence maps over all regions in $\mathcal{J}$, and over all exemplars of that category.

5. Experimental Results

We evaluate our object recognition method on the ETHZ shape and the Caltech 101 databases.

5.1. ETHZ Shape

The ETH Zurich shape database (collected by V. Ferrari *et al.* [12]) consists of five distinctive shape categories (applelogos, bottles, giraffes, mugs and swans) in a total of 255 images. It is a challenging database because target objects appear over a wide range of scales and locations (see Figure 10). In particular, we mark object support in the images as ground truth masks for our segmentation task.

Initially, we construct region trees for images. This gives on average $\sim$ 100 regions per image. Since color and tex-

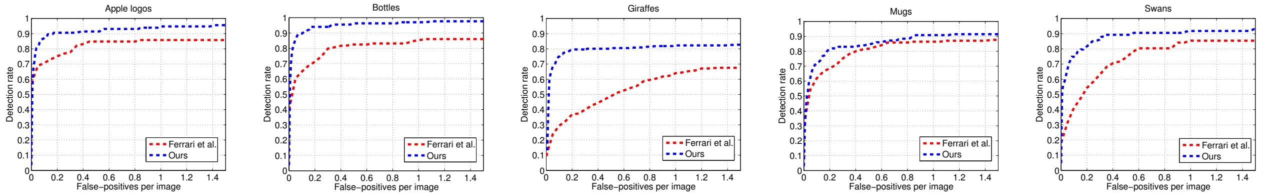

Figure 8. Comparison of detection performance with Ferrari *et al*. [11] on the ETHZ shape database. Each plot shows the detection rate as a function of false positives per image (FPPI) under the PASCAL criterion (a detected bounding box is considered correct if it overlaps $\geq 50\%$ ”intersection over union” with the ground truth bounding box). Our method significantly outperforms theirs over all five categories at every FPPI point between $[0, 1.5]$.

ture cues are not very useful in this database, we only use gPb-based contour shape cues as region features. In the weight learning stage, we construct exemplar images and their similar/dissimilar pairs in the following way: we take the bounding boxes of objects in training as exemplars. For each exemplar, similar instances are the bounding boxes containing objects of the same category as the exemplar, and dissimilar instances are the ones containing objects of different categories as well as a collection of background regions, all in the training set.

In the voting stage, we choose the functions in Eqn.11 as:

$$g(d_{R^{\mathcal{I}}}, d_{R^{\mathcal{J}}}) = \max\{0, 1 - \sigma \cdot \chi^2(d_{R^{\mathcal{I}}}, d_{R^{\mathcal{J}}})\} \qquad (17)$$

$$h(R^{\mathcal{I}}, R^{\mathcal{J}}) = \mathbf{1}[\alpha \leq \text{Asp}(R^{\mathcal{I}})/\text{Asp}(R^{\mathcal{J}}) \leq 1/\alpha] \qquad (18)$$

where $\chi^2(\cdot)$ specifies the chi-square distance, and $\text{Asp}(R)$ is the aspect ratio of the bounding box of R. The last equation enforces aspect ratio consistency between matched regions. In the experiment, we use $\sigma = 2$ and $\alpha = 0.6$.

We split the entire set into half training and half test for each category, and the average performance from 5 random splits is reported. This is consistent with the implementation in [11] which reported the state-of-the-art detection performance on this database. Figure 8 shows our comparison to [11] on each of the categories. Our method significantly outperforms [11] on all five categories, and the average detection rate increases by 20% ($87.1 \pm 2.8\%$ with respect to their 67.2%) at false positive per image (FPPI) rate of 0.3 under the PASCAL criterion. Detection rates on individual categories are listed in Table 1.

We also evaluate segmentation performance on each of the 5 categories using mean average precision (AP) of pixel-wise classification. AP is defined by the area underneath the recall-precision curve. Table 2 shows the precision accuracies. The overall mean AP on the object segments using our constrained segmentation algorithm achieves $75.7 \pm 3.2\%$, significantly higher than on the bounding boxes from voting. Examples of object detection and segmentation results are shown in Figure 10.

Table 3 compares the number of sliding windows, regions, and bounding boxes that need to be considered for

Categories	Voting only	Verify only	Combined
Applelogos	87.2 ± 9.0	85.4 ± 5.3	90.6 ± 6.2
Bottles	93.0 ± 3.0	93.2 ± 5.4	94.8 ± 3.6
Giraffes	79.4 ± 1.3	73.6 ± 5.5	79.8 ± 1.8
Mugs	72.6 ± 12.0	81.4 ± 5.4	83.2 ± 5.5
Swans	82.2 ± 10.0	80.8 ± 9.7	86.8 ± 8.9
Average	82.9 ± 4.3	82.9 ± 2.8	$\mathbf{87.1 \pm 2.8}$

Table 1. Object detection results in ETHZ shape. Detection rates (%) at 0.3 FPPI based on only voting scores, only verification scores, and products of the two are reported, for each individual category and the overall average over 5 trials.

Categories	Bounding Box	Segments
Applelogos	50.2 ± 7.7	77.2 ± 11.1
Bottles	73.0 ± 2.6	90.6 ± 1.5
Giraffes	34.0 ± 0.7	74.2 ± 2.5
Mugs	72.2 ± 5.1	76.0 ± 4.4
Swans	28.8 ± 4.2	60.6 ± 1.3
Average	51.6 ± 2.5	$\mathbf{75.7 \pm 3.2}$

Table 2. Object segmentation results in ETHZ shape. Performance (%) is evaluated by pixel-wise mean Average Precision (AP) over 5 trials. The mean APs are computed both on the bounding boxes obtained in Section 4.1, and the segments obtained in Section 4.3.

Categories	Sld. Windows	Regions	Bnd. Boxes
Applelogos	$\sim 30,000$	115	3.1
Bottles	$\sim 1,500$	168	1.1
Giraffes	$\sim 14,000$	156	6.9
Mugs	$\sim 16,000$	189	5.3
Swans	$\sim 10,000$	132	2.3

Table 3. A comparison of the number of sliding windows, regions, and bounding boxes that need to be considered for different categories in ETHZ shape. The number of regions for each category is the average number of regions from images of that category. The number of bounding boxes is the average number of votes from Section 4.1 that need to obtain full recall of objects. The number of sliding windows is estimated in the Appendix.

Image cues	5 train	15 train	30 train
(R) Contour shape	41.5	55.1	60.4
(R) Edge shape	30.0	42.9	48.0
(R) Color	19.3	27.1	27.2
(R) Texture	23.9	31.4	32.7
(R) All	40.9	59.0	65.2
(P) GB	42.6	58.4	63.2
(R) Contour shape+(P) GB	44.1	**65.0**	**73.1**
(R) All + (P) GB	45.7	64.4	72.5

Table 4. Mean classification rate (%) in Caltech 101 using individual and combinations of image cues. (R) stands for region-based, and (P) stands for point-based. (R)All means combining all region cues (Contour shape+Edge shape+Color+Texture). We notice that cue combination boosts the overall performance significantly.

different categories. We show that our voting scheme obtains 3-4 orders of magnitude reduction on the number of windows compared to the standard sliding window approach.

5.2. Caltech-101

The Caltech-101 database (collected by L. Fei-Fei *et al.* [9]) consists of images from 101 object categories (excluding the background class). The significant variation in intraclass pose, color and lighting makes this database challenging. However, since each image contains only a single object, usually large and aligned to the center, we bypass the voting step and consider the entire image as the bounding box of the object. Thus, we use this database to benchmark only our verification step.

We follow the standard approach for evaluation. For each category, we randomly pick 5, 15 or 30 images for training and up to 15 images in a disjoint set for test. Each test image is assigned a predicted label, and mean classification rate is the average of the diagonal elements of the confusion matrix.

To exploit multiple image cues, we extract four types of region descriptors (two types of shape, color and texture, all described in Section 2.2), as well as one point descriptor (Geometric Blur or GB [4]). Table 4 lists the mean classification rates with different combinations of these image cues. We observe a performance gain (from 55.1% to 59.0% under 15 training) by combining different region cues in our method. In addition, a second and significant boost in performance is obtained by combining region contour shape with point GB cues (from 58.4% to 65.0% under 15 training). This boost illustrates that region based descriptors complements conventional point based descriptors (*e.g.* SIFT [21]) in recognition. Our method achieves competitive performance in this database in comparison with other recently published approaches in Figure 9.

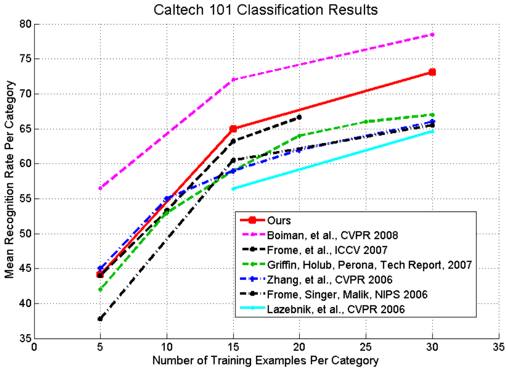

Figure 9. Mean recognition rate (%) over number of training images per category in Caltech 101. With 15 and 30 training images per category, our method outperforms [14], [15], [33], [13] and [19] but not [5].

6. Conclusion

In this paper, we have presented a unified framework for object detection, segmentation, and classification using regions. Building on a novel region segmentation algorithm which produces robust overlaid regions, we have reported state-of-the-art detection performance on the ETHZ shape database, and competitive classification performance on the Caltech 101 database. We have further shown that (1) cue combination significantly boosts recognition performance; (2) our region-based voting scheme reduces the number of candidate bounding boxes by orders of magnitude over standard sliding window scheme due to robust estimation of object scales from region matching.

Appendix

We compute the optimal sliding window parameter choices with respect to the ground truth labeling of the test set in ETHZ shape. This gives us an estimate of the total number of candidates a sliding window classifier would need to examine in order to achieve full recall. To this end, we first compute relative scales of objects with respect to image sizes in the test set. We denote the minimum and maximum scales as S_{min}, and S_{max}. So $0 < S_{min} < S_{max} < 1$. Next, we assume that the minimum span between neighboring windows in each image axis is a quarter of the minimum scale. Then for each level of window scale, we have roughly $1/(S_{min}/4)^2$ candidate locations. As for searching over scales, we make a second assumption that the neighboring levels are $1/8$ octave apart. Then the number of scales needed to cover the range of $[S_{min}, S_{max}]$ is $8\log_2(S_{max}/S_{min})$. So if we ignore aspect ratio change of objects, the estimate of the number of windows N becomes

$$\begin{aligned} N &= 1/(S_{min}/4)^2 \cdot 8\log_2(S_{max}/S_{min}) && (19) \\ &= 128\log_2(S_{max}/S_{min})/S_{min}^2 && (20) \end{aligned}$$

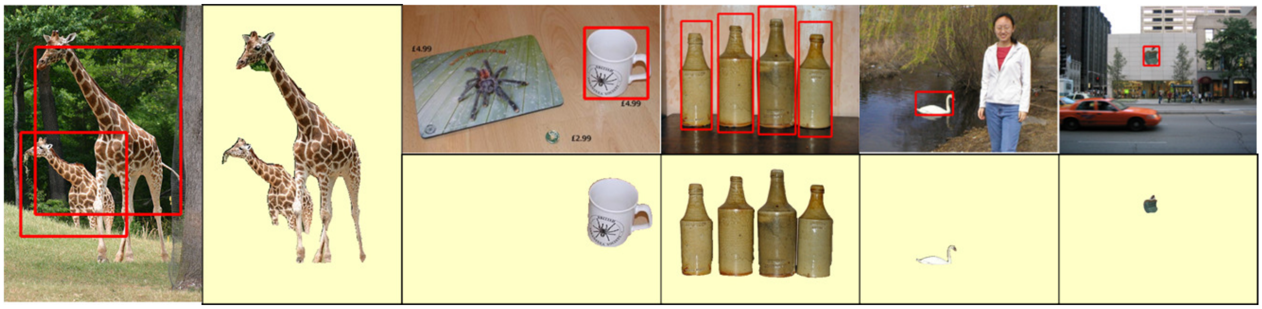

Figure 10. Detection and segmentation results in the ETHZ shape database.

References

[1] P. Arbeláez and L. Cohen. Constrained image segmentation from hierarchical boundaries. In *CVPR*, 2008.

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

[3] D. H. Ballard. Generalizing the hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2):111–122, 1981.

[4] A. Berg and J. Malik. Geometric blur and template matching. In *CVPR*, 2001.

[5] O. Boiman, E. Shechtman, and M. Irani. In defense of nearest-neighbor based image classification. In *CVPR*, 2008.

[6] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *TPAMI*, 24(5):603–619, 2002.

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

[8] R. O. Duda and P. E. Hart. Use of the hough transformation to detect lines and curves in pictures. *Commun. ACM*, 15(1):11–15, 1972.

[9] L. Fei-Fei, F. Fergus, and P. Perona. Learning generative visual models from few training examples: an incremental bayesian approach testing on 101 object categories. In *Workshop on Generative-Model Based Vision, CVPR*, 2004.

[10] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *CVPR*, 2008.

[11] V. Ferrari, F. Jurie, and C. Schmid. Accurate object detection with deformable shape models learnt from images. In *CVPR*, 2007.

[12] V. Ferrari, T. Tuytelaars, and L. V. Gool. Object detection by contour segment networks. In *ECCV*, 2006.

[13] A. Frome, Y. Singer, and J. Malik. Image retrieval and classification using local distance functions. In *NIPS*, 2006.

[14] A. Frome, Y. Singer, F. Sha, and J. Malik. Learning globally-consistent local distance functions for shape-based image retrieval and classification. In *ICCV*, 2007.

[15] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. Technical Report 7694, California Institute of Technology, 2007.

[16] D. Hoiem, A. Efros, and M. Hebert. Geometric context from a single image. In *ICCV*, pages I: 654–661, 2005.

[17] M. Kumar, P. Torr, and A. Zisserman. Obj cut. In *CVPR*, pages I: 18–25, 2005.

[18] C. H. Lampert, M. B. Blaschko, and T. Hofmann. Beyond sliding windows: Object localization by efficient subwindow search. In *CVPR*, 2008.

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

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

[21] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, November 2004.

[22] M. Maire, P. Arbeláez, C. Fowlkes, and M. Malik. Using contours to detect and localize junctions in natural images. In *CVPR*, 2008.

[23] S. Maji, A. Berg, and J. Malik. Classification using intersection kernel support vector machine is efficient. In *CVPR*, 2008.

[24] S. Maji and J. Malik. Object detection using a max-margin hough transform. In *CVPR*, 2009.

[25] T. Malisiewicz and A. Efros. Recognition by association via learning per-exemplar distances. In *CVPR*, pages 1–8, 2008.

[26] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *IJCV*, 42(3):145–175, May 2001.

[27] A. Opelt, A. Pinz, and A. Zisserman. A boundary-fragment-model for object detection. In *ECCV*, pages 575–588, 2006.

[28] H. A. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *TPAMI*, 20:23–38, 1998.

[29] B. C. Russell, A. A. Efros, J. Sivic, W. T. Freeman, and A. Zisserman. Using multiple segmentations to discover objects and their extent in image collections. In *CVPR*, 2006.

[30] C. Schmid and R. Mohr. Local grayvalue invariants for image retrieval. *TPAMI*, 19(5):530–535, May 1997.

[31] S. Todorovic and N. Ahuja. Learning subcategory relevances for category recognition. In *CVPR*.

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

[33] H. Zhang, A. C. Berg, M. Maire, and J. Malik. Svm-knn: Discriminative nearest neighbor classification for visual category recognition. In *CVPR*, 2006.