Simultaneous Object Detection and Segmentation by Boosting Local Shape Feature based Classifier

Bo Wu and Ram Nevatia
University of Southern California
Institute for Robotics and Intelligent Systems
Los Angeles, CA 90089-0273
{bowu|nevatia}@usc.edu

Abstract

This paper proposes an approach to simultaneously detect and segment objects of a known category. Edgelet features are used to capture the local shape of the objects. For each feature a pair of base classifiers for detection and segmentation is built. The base segmentor is designed to predict the per-pixel figure-ground assignment around a neighborhood of the edgelet based on the feature response. The neighborhood is represented as an effective field which is determined by the shape of the edgelet. A boosting algorithm is used to learn the ensemble classifier with cascade decision strategy from the base classifier pool. The simultaneousness is achieved for both training and testing. The system is evaluated on a number of public image sets and compared with several previous methods.

1. Introduction

Accurate delineation (*i.e*. segmentation) and detection of objects of one or more known classes is of fundamental interest in computer vision and needed for a number of higher level tasks. Traditionally, one would segment an image by one of the various region segmentation methods and then try to classify the regions as belonging to one of the desired classes. This approach works well when objects of interest have relatively homogeneous properties in some image attributes such as intensity, color or texture. However, for many common objects of interest, *e.g*. humans, the surfaces are not uniform and the texture can be arbitrarily complex. In such cases, no effective algorithms for bottom-up segmentation have been devised; existing methods tend to over or under segment an image. If objects of interest are moving, motion-based segmentation can be more reliable, but even here, merging of motion blobs with adjacent objects and with shadows and reflections can be problematic. We focus on static image analysis in this paper.

In recent years, methods for direct detection of objects have become popular. The best know example is perhaps that of face detection by Viola and Jones [24] where no prior segmentation is applied; rather, the image is scanned by windows of various size and a determination as to the presence of the desired object is made in this window. While such methods show good performance at the detection level, object delineation is not very precise; typically a bounding box which contains the object as well as some of the background is detected. A more accurate delineation process may then be applied inside the bounding box, as in [19]. We argue that better results can be obtained if the object models for segmentation and detection are built simultaneously; furthermore, the process is more efficient as the two may share many common feature computations.

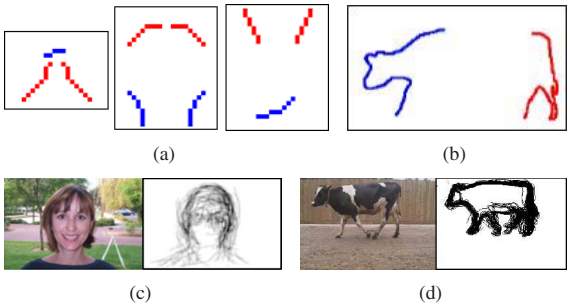


Figure 1. Local shape features in [9, 13, 14]: a) Edgelets selected for people [14]; b) Boundary fragments selected for cows [9]; c) Feature responses of a face [13]; d) Feature responses of a cow [9].

In this work, our objective is to simultaneous detect and delineate objects of a known category. The features used in the classifiers are shape features. In some existing work, discriminative local shape features, such as contour fragments in [9, 13], and *edgelet* in [14], are selected by boosting algorithm to model the appearance of the objects, see Fig.1. The selected features lie on the object boundary; their responses on the query image help to delineate the object. Based on this observation, we design the base classifiers for detection and segmentation from local shape features, and learn cascade structured classifier by boosting the base classifiers.

1-4244-1180-7/07/$25.00 ©2007 IEEE

1.1. Related Work

The main difference between general image segmentation methods, *e.g*. [25], and the segmentation of objects of a known class is the use of the prior knowledge, *i.e*. an object model, of the concerned class. In addition to guiding segmentation, the object models also function as discriminative models for recognition and detection, *e.g*. [1, 5, 9, 7, 6, 12, 22, 17], or generative models for pose estimation, *e.g*. [8]. In this work, we address the problem of simultaneous detection and segmentation.

Many of the recent efforts build object models based on some image features, other than color value or gray intensity. The use of features enables us to focus on informative and discriminative image cues. The features could be global, *e.g*. the edge template in [27], or local, *e.g*. the rectangle feature in [24]. Global features are relatively sensitive to partial occlusions compared to local ones. The properties that the features try to capture mainly include: 1) color, *e.g*. the mixtures of Gaussian color model in [7, 13], and the kernel density estimation of color distribution in [11]; 2) texture, *e.g*. the texton in [7]; and 3) shape, *e.g*. the part templates in [4], the boundary or contour fragments in [9, 13], the simplified SIFT descriptors in [6], and the edgelets in [14].

When global features are used, the object models are sometimes equal to the features, *e.g*. the edge template models in [27, 11]. When local features are used, we need some way to organize the features to form the object models. Many existing methods use random field (RF) approaches, *e.g*. the Layout Consistent Conditional Random Field in [1], the Located Hidden Random Field in [6], the texton based CRF in [7], the pose-specific MRF in [8], the Pictorial Structure enhanced MRF in [17]. The algorithms used to infer the RF models include loopy belief propagation, sequential tree-reweighted message passing, and graph cuts. These techniques are computationally expensive. Some other methods use star-shaped models to organize the local features, *e.g*. the Boundary-Fragment-Model in [9], and the Implicit Shape Model in [19, 22]. These models can be inferred by Hough Transformation much more efficiently. However, these methods usually assume a fixed object size so that the solution space is highly restricted. Some of these methods, *e.g*. [5, 1, 9, 6, 7, 12, 11, 17], result in simultaneous detection and segmentation.

Recently, the boosting based framework proposed by Viola and Jones [24] has been successfully applied to detect some object categories, *e.g*. faces [24, 15] and pedestrians [14]. The cascade decision strategy makes the boosted ensemble classifiers very efficient for multi-scale object detection. Unlike the random field approaches and the constellation approaches, boosting methods encode the shape of the objects by including a number of local features within the sample window. The relative positions of these local features represent the shape implicitly. Although some existing methods, *e.g*. [9, 7], use boosting as feature selector for segmentation, as far as we know, none of them learn the ensemble classifier as a direct segmentor. In this work, we propose a unified framework to learn cascade classifier for simultaneous detection and segmentation.

1.2. Outline of Our Approach

Following our previous work [14], we design our classifiers based on *edgelet* features. Compared to other local shape features, the main advantages of the edgelets are: 1) they consider both the magnitude and the direction of the edge; 2) they are parametric shapes which are not object class dependent; and 3) the computation of edgelets is efficient. Although color and texture are potentially useful cues, they are not used here, as we want to investigate the effect of shape features for object segmentation and segmentation in this work.

For training, first a large feature pool is built. For each feature in it, a pair of base classifiers for detection and segmentation is learned. A variation of the real AdaBoost algorithm [28] is used to select good features from the pool. We formulate the segmentation as binary classification problem. The input is an image sample and a pixel location within the sample window; the output is the figure-ground prediction. We define an effective neighborhood of each edgelet feature. The figure-ground distribution is learned within this neighborhood. The final boosted ensemble classifier with a cascade decision strategy works as a detector as well as a segmentor, *i.e*. given the input image, after one round of scanning, the outputs are the locations of the objects and the pixel level segmentation masks. The proposed approach is general and not restricted by object type. Our main contributions are: 1) the design of the base segmentors based on local shape features; and 2) a boosting algorithm for simultaneous learning of detector and segmentor.

We evaluate the detection and segmentation performance of our system on several public data sets, the UIUC car image set [21][1], the car images of Caltech101 set [10][2], the car images of TU Darmstadt set [22][3], the USC pedestrian set [14][4], and some data collected by us. The experimental results show that the detection performance of our method is comparable to the-state-of-art detection methods, while the segmentation outperforms the existing methods.

The rest of the paper are organized as follows: section 2 describes the design of the base classifiers for detection and segmentation; section 3 gives our boosting algorithm; section 4 shows experimental results; and some conclusions and discussions are given in the last section.

[1] http://l2r.cs.uiuc.edu/ cogcomp/Data/Car/

[2] http://www.vision.caltech.edu/Image_Datasets/Caltech101/

[3] http://www.pascal-network.org/challenges/VOC/databases.html#TUD

[4] http://iris.usc.edu/ bowu/DatasetWebpage/dataset.html

2. Design of the Base Classifiers

In the boosting framework, one base classifier is build for each simple feature. In our case, we use edgelet features [14]. An edgelet can be seen as a function from the image space to a real valued range, $f : \mathcal{X} \mapsto [0,1]$. Based on the feature f, we learn the base classifier for detection $h^{(d)}$ and the base classifier for segmentation $h^{(s)}$, *i.e.* a pair of classifiers sharing the same feature.

2.1. Base Classifier for Detection

The base detection classifier is a function from the image space $\mathcal{X}$ to a real valued object/non-object classification confidence space. Given a labeled sample set $S = \{(\mathbf{x}_i, y_i)\}$, where $\mathbf{x} \in \mathcal{X}$ is the image patch, and $y_i = \pm 1$ is the class label of $\mathbf{x}$, the base detection classifier $h^{(d)}$ is learned as a piecewise function:

$$\text{if} f(\mathbf{x}) \in \left[\frac{j-1}{n_d}, \frac{j}{n_d}\right), h^{(d)}(\mathbf{x}) = \frac{1}{2}\ln\left(\frac{W_+^j + \varepsilon}{W_-^j + \varepsilon}\right) \quad (1)$$

where $j = 1 \ldots n_d$, n_d is the bin number (in our experiments $n_d = 32$), ε is a smoothing factor [28], and $W_\pm$ is the probability distribution of the feature value for positive/negative samples, implemented as a histogram:

$$W_\pm^j = P\left(f(\mathbf{x}) \in \left[\frac{j-1}{n_d}, \frac{j}{n_d}\right), y = \pm 1\right) \quad (2)$$

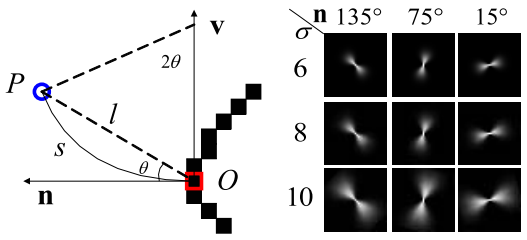
2.2. Base Classifier for Segmentation

The base segmentation classifier is a function from the space $\mathcal{X} \times \mathcal{U}$ to a real valued figure-ground classification confidence space, where $\mathcal{U}$ is the 2D image coordinate space, *i.e.* $\mathcal{U} = \{0, 1, \ldots\}^2$. In general, a local feature only contributes to the shape around its neighborhood. It is not efficient to predict the state of the feet from a edgelet falling on the head-top. Based on this observation, we define the *effective field* of an edgelet based on a saliency decay function. This is motivated by the tensor voting method for shape grouping [26]. As shown in Fig.2(a), O is a point on an edgelet feature, whose normal $\mathbf{n}$ and tangent $\mathbf{v}$ are known, P is a neighbor of O, and $\overset{\frown}{OP}$ is the arc of the *osculating circle* at O that goes through P. Let l be the Euclidean distance between O and P, and θ be the angle between $\mathbf{n}$ and $\overrightarrow{OP}$. The effect of O on P is defined by

$$DF(s, \kappa, \sigma) = \exp\left(-\frac{s^2 + c\kappa^2}{\sigma^2}\right) \quad (3)$$

where $s = \frac{l\theta}{2\sin\theta}$ is the length of the arc $\overset{\frown}{OP}$, and $\kappa = \frac{2\sin\theta}{l}$ is the curvature, c is a constant which controls the decay with high curvature, and σ is the scale of analysis, which determines the size of the effective field. Note that σ is the only free parameter. In practice, σ is quantized to five values, 2, 4, 6, 8, 10, according to the size of our training samples, and the normal orientation of the edgelet point is quantized to six bins, $[\frac{\pi}{6}(i-1), \frac{\pi}{6}i), i = 1 \ldots 6$. So there are overall 30 bases of the effective field, see Fig.2(b) for some examples. For a k point edgelet, denote by $\mathbf{F}_i$ the effective field of the i-th point, then the effective field of the whole feature is defined by

$$\mathbf{F}(\mathbf{u}) = \max\{\mathbf{F}_1(\mathbf{u}), \ldots, \mathbf{F}_k(\mathbf{u})\}, \mathbf{u} \in \mathcal{U} \quad (4)$$



(a) (b)

Figure 2. Effective field of edgelet point: a) definition of decay function; b) examples of bases of effective field.

The learning of the base segmentation classifier is similar to that of detection. For the positive training samples, their segmentation ground-truth are given as binary masks. Let $S_+ = \{(\mathbf{x}_i, y_i = 1, \mathbf{m}_i)\}$, where $\mathbf{m}$ is the segmentation mask that has the same dimension as $\mathbf{x}$. Assume that the effective field $\mathbf{F}$ of a feature f has been determined (how to optimize the shape of the effective field will be described in section 3), the base segmentation classifier $h^{(s)}$ is learned as a piecewise function:

$$\text{if} f(\mathbf{x}) \in \left[\frac{j-1}{n_s}, \frac{j}{n_s}\right), h^{(s)}(\mathbf{x}; \mathbf{u}) = \frac{1}{2}\ln\left(\frac{\mathbf{W}_+^j(\mathbf{u}) + \varepsilon}{\mathbf{W}_-^j(\mathbf{u}) + \varepsilon}\right) \quad (5)$$

where $j = 1 \ldots n_s$, n_s is the bin number for segmentation (in our experiments $n_s = 8$), and $\mathbf{W}_\pm(\mathbf{u})$ is the feature value histogram of figure/ground pixels weighted by the effective field:

$$\mathbf{W}_\pm^j(\mathbf{u}) = \mathbf{F}(\mathbf{u}) \cdot P\left(f(\mathbf{x}) \in \left[\frac{j-1}{n_s}, \frac{j}{n_s}\right), \mathbf{m}(\mathbf{u}) = \pm 1\right) \quad (6)$$

In practice, both $h^{(d)}$ and $h^{(s)}$ are implemented as look-up-table. The difference is that each bin of $h^{(d)}$ is a real valued scalar, while each bin of $h^{(s)}$ is a real valued matrix.

3. Boosted Ensemble Classifier

Let $\mathcal{H}$ be the base classifier pool that consists of the base classifier pairs built from all possible edgelets. Each element in the pool is a pair of base detector and segmentator, *i.e.* $(h^{(d)}, h^{(s)})$. We use a variation of boosting algorithm to learn an ensemble classifier from $\mathcal{H}$ as a strong detector and segmentor. The original cascade structure by Viola and Jones [24] has three levels of classifiers: the base classifier

(or weak classifier), the strong classifier (or layer), and the cascade classifier. The base classifiers give weak predictions. The strong classifier combines the predictions of the base classifiers to make better decision. The cascade consists of a series of strong classifiers, each of which accepts most positive samples and reject as many negative samples as possible. It is the cascade structure that makes the detector efficient.

We modify the original cascade structure to eliminate the concept of layers, so that the learning becomes one integral boosting procedure. The function of the layers is to reject non-object samples, which can be implemented at the base classifier level. Let $h_t^{(d)}$ be the t-th base detection classifier, and $H_t^{(d)}$ be the partial sum of the first t base detection classifiers, *i.e*.

$$H_t^{(d)}(\mathbf{x}) \triangleq \sum_{i=1}^{t} h_i^{(d)}(\mathbf{x}), \mathbf{x} \in \mathcal{X} \tag{7}$$

Our modified cascade consists T base detection classifiers, $\{h_t^{(d)}\}_{t=1}^T$, and T threshold $\{b_t\}_{t=1}^T$, a sample $\mathbf{x}$ is classified as object iff

$$\forall t, H_t^{(d)}(\mathbf{x}) > b_t \tag{8}$$

This structure can be seen as a special case of the nested cascade proposed in [20] and the soft cascade in [18]. One common advantage of these variations is the discriminative information obtained by the base classifiers are inherited along the cascade.

One important feature of a boosting algorithm is the evolution of weights. For traditional detection problems, each sample is assigned a real valued weight $D^{(d)}$ representing its importance or difficulty. During the boosting procedure, the weights of the misclassified samples are increased while those of the correctly classified samples are decreased, so that more and more attention is given to the difficult part of the sample space. For segmentation, not only do the difficulties of different samples vary, but also the difficulties of different positions of the same sample are different. Hence, for a positive sample, we assign a weight $\mathbf{D}^{(s)}(\mathbf{u})$ to each of its pixels. During the boosting procedure, the weights for segmentation are evolved as the same way of the weights for detection.

At each boosting round, the best base classifier pair is selected from $\mathcal{H}$, where two components need to be optimized, the edgelet feature and the effective field. The edgelet features are enumerated in the feature pool and the effective field is defined by the shape of its edgelet and the parameter σ. As we allow different σ's for different points in one edgelet, for a k point edgelet there are 5^k possible field shapes. When the sample size is 24×58, there are overall 857,604 possible edgelets [14]. It is too expensive to do brute force search in this Cartesian space. We separate the optimization in two steps: first search for the best edgelet with a default σ value, and then search for the best σ configuration. The optimization of the σ configuration is done in a greedy way. At one time, the σ of one edgelet point is modified, while the others are fixed. Fig.3 gives the full algorithm of simultaneous learning of detector and segmentor. The output of this algorithm is an ensemble classifier with a cascade decision strategy for detection. As segmentation is a symmetric classification problem, we take the default threshold to be zero, *i.e*. the pixel $\mathbf{u}$ of $\mathbf{x}$ is classified as figure, iff

$$H_T^{(s)}(\mathbf{x};\mathbf{u}) = \sum_{t=0}^{T} h_t^{(s)}(\mathbf{x};\mathbf{u}) > 0 \tag{9}$$

In practice, we use the prior figure-ground distribution as the first base segmentation classifier $h_0^{(s)}$.

To cover multi-scale objects, the classifier is applied to the input images at different scales and positions, and resulting responses are clustered so that multiple responses corresponding to one object are merged.

4. Experimental Results

We applied this approach to three object categories, frontal/rear view pedestrians, side view pedestrians, and side view cars. These are important types of objects for many applications, such as visual surveillance, driving assistance system, and human computer interaction. Quantitative results on both detection and segmentation are reported.

4.1. Results on Frontal/Rear View Pedestrian

First we evaluate our method on walking or standing people of frontal or rear viewpoint. We collected 2,000 positive samples and 6,000 negative images from the MIT pedestrian set [29] and the Internet. The positive samples are resized to 24×58 pixels. We randomly select 600 positive samples and label their segmentation ground-truth manually. We use a polygon to delineate the object, however, the boundary pixels are sometimes ambiguous and can not be classified clearly. Hence we mark a two pixel width do-not-care (DNC) boundary, see Fig.4.

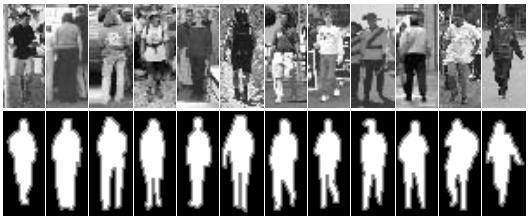


Figure 4. Samples for frontal/rear view pedestrian: the first row is the image samples; the second row is the segmentation ground-truth (The grey pixels are do-not-care).

The DNC pixels are ignored in both training and testing. This strategy is similar to that in [7]. To evaluate segmentation, four fifths of the 600 samples are used as training data, the rest one fifth is used as testing data. So our data set for

- Given the initial sample set $S = S_+ \cup S_- = \{(\mathbf{x}_i, +1, \mathbf{m}_i)\} \cup \{(\mathbf{x}_i, -1)\}$, and a negative images set;
- Set the algorithm parameters: the maximum base classifier number T, the positive passing rates $\{P_t\}_{t=1}^{T}$, the target false alarm rate F, the relative importance of detection to segmentation λ, and the threshold for bootstrapping θ_B;
- Initialize the sample detection weights $D_0^{(d)}(\mathbf{x}) = \frac{1}{||S||}$ for all samples, the sample segmentation weight fields $\mathbf{D}_0^{(s)}(\mathbf{x};\mathbf{u}) = \frac{1}{||S_+||||\mathbf{x}||}$ for all positive samples, the current false alarm rate $F_0 = 1$, and $t = 0$;
- Construct the base classifier pool, $\mathcal{H}$, from the edgelet features;
- while $t < T$ and $F_t > F$ do
  1. Search for the best edgelet
     (a) For each pair $(h^{(d)}, h^{(s)})$ in $\mathcal{H}$, generate the effective field for segmentation with a default value of σ(=4), calculate $h^{(d)}$ and $h^{(s)}$ by Equ.1 and Equ.5 respectively. $W_\pm$ and $\mathbf{W}_\pm$ are calculated under the weight distributions $D_t^{(d)}$ and $\mathbf{D}_t^{(s)}$ respectively;
     (b) Select $(h_t^{(d)}, h_t^{(s)})$ by

$$\left(h_t^{(d)}, h_t^{(s)}\right) = \underset{\left(h_t^{(d)}, h_t^{(s)}\right) \in \mathcal{H}}{\arg\min} \left\{ \lambda 2 \sum_{j=1}^{n} \sqrt{W_+^j W_-^j} + (1-\lambda) \frac{1}{Z} \sum_{j=1}^{n} \sum_{\mathbf{u} \in \mathcal{U}} \sqrt{\mathbf{W}_+^j(\mathbf{u}) \mathbf{W}_-^j(\mathbf{u})} \right\} \tag{10}$$

     where $Z = \sqrt{\sum_j \sum_{\mathbf{u}} \mathbf{W}_+^j(\mathbf{u}) \sum_j \sum_{\mathbf{u}} \mathbf{W}_-^j(\mathbf{u})}$
  2. Search for the best shape of the effective field. For each point of the edgelet, set $\sigma = 2, 4, 6, 8, 10$, find the best value that minimizes the criteria in Equ.10.
  3. Update sample weights by

$$\begin{cases} D_{t+1}^{(d)}(\mathbf{x}) = D_t^{(d)}(\mathbf{x}) \exp\left[-y h_t^{(d)}(\mathbf{x})\right], \forall \mathbf{x} \in S \\ \mathbf{D}_{t+1}^{(s)}(\mathbf{x};\mathbf{u}) = \mathbf{D}_t^{(s)}(\mathbf{x};\mathbf{u}) \exp\left[-\mathbf{m}(\mathbf{u}) h_t^{(s)}(\mathbf{x};\mathbf{u})\right], \forall \mathbf{x} \in S_+, \mathbf{u} \in \mathcal{U} \end{cases} \tag{11}$$

     and normalize $D_{t+1}^{(d)}$ and $\mathbf{D}_{t+1}^{(s)}$ to p.d.f.
  4. Select the threshold b_t for the partial sum $H_t^{(d)}$, so that a portion of P_t positive samples are accepted; and reject as many negative samples as possible;
  5. Remove the rejected samples from S. If the remaining negative samples are less than θ_B percent of the original, recollect S_- by bootstrapping on the negative image set.
  6. $t++$
- Output $\{(h_t^{(d)}, h_t^{(s)}), b_t\}$ as the cascade classifier for detection and segmentation.

Figure 3. Algorithm of simultaneously learning of detector and segmentor. In our experiments, $T = 1{,}000$, $F = 10^{-6}$, $\lambda = 0.7$, and $\theta_B = 75\%$. The setting of $\{P_t\}$ is similar to the original cascade's layer acceptance rates. The cascade is divided into 20 segments, the lengthes of which grow gradually. The base classifiers at the end of the segments have positive passing rate of 99.8%, and the other base classifiers have passing rate of 100.0%.

frontal/rear pedestrian contains 1,880 positive training samples, 480 of which have segmentation ground-truth, and 120 test samples with segmentation ground-truth. The learned classifier consists of 360 features. Fig.5 shows the first few selected features and their learned segmentors. They are evenly distributed and correspond to natural body parts.

We evaluated the segmentation accuracy for frontal/rear pedestrians on the 120 testing samples. An precision-recall (PR) curve is generated by changing the threshold for segmentation, see Fig.6. As the articulation effect is not very strong from this viewpoint, we achieve the highest accuracy on this class, the equal error rate (EER) is about 96.8%. (The segmentation accuracy is calculated at the pixel level.) We evaluate the detection performance on another test set, USC pedestrian set A [14], which has 205 images with 313 humans. Fig.7 shows the PR curves. It can be seen that our detector is comparable to that in [14]. Some example results are shown in Fig.10.

4.2. Results on Side View Pedestrian

We also evaluated our method on walking or standing people of left profile viewpoint. We treat the side view pedestrians and frontal/rear view pedestrians as two categories, as their appearance are too different to included in one cascade detector. Similar to the case of frontal/rear view pedestrians, we collected 2,000 positive samples of left profile view pedestrians and 6,000 negative images from the Internet. The positive samples are resized to 24×58 pixels. The segmentation ground-truth are labeled manually for 600 randomly selected positive samples, four fifths of which are used for training, and one fifth for testing. The

learned classifier consists of 560 features. The PR curve of segmentation for left profile view pedestrians is shown in Fig.6. The articulation effect from this viewpoint is significant. This category has the largest intra-class variation. Our method achieves a EER of about 95.1%.

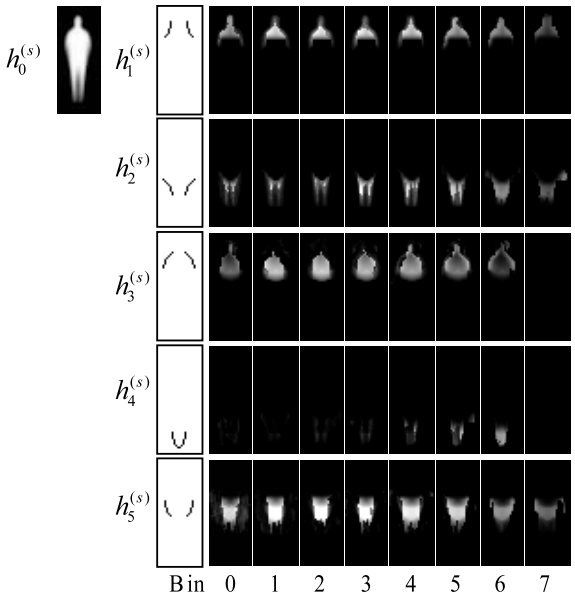


Figure 5. The first five features selected and their learned segmentors. (the 0-th segmentor is the prior distribution.)

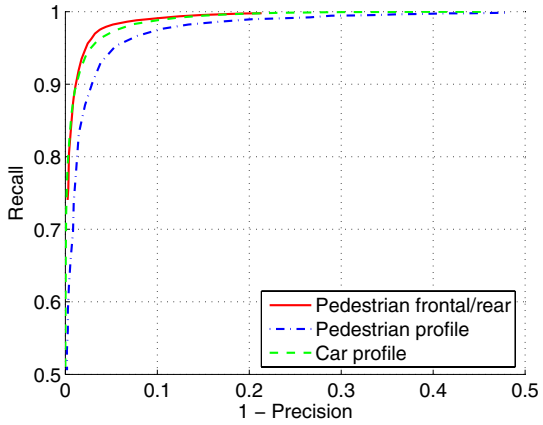


Figure 6. Segmentation performance on the normalized samples.

Although there are many existing methods, *e.g.* [2], report quantitative detection performance on side view pedestrians, as far as we know there is no appropriate public test set for this task. Hence we collected our own test set, which contains 136 images with 195 humans. Fig.8 shows the detection PR curve of our method on this test set. Although the test sets are not same, it can be seen that our method is comparable to that in [2]. Some example results are shown in Fig.10.

4.3. Results on Side View Car

Last, we evaluated our method on cars of left profile viewpoint. We collected 1,800 positive samples of side view cars and 6,000 negative images from the training set of the UIUC car set [21] and the Internet. The car models included in this set are mainly small and mid-size cars including sedans, pickups, vans, and SUVs. Most of the intra-class variation is due to the variety of car models. The positive samples are resized to 75×30 pixels. The segmentation ground-truth are labeled manually for 600 randomly selected positive samples, four fifths of which are used for training, and one fifth for testing. The learned classifier consists of 480 features. The PR curve of segmentation for side view cars is shown in Fig.6. Our method achieves a EER of about 96.0%.

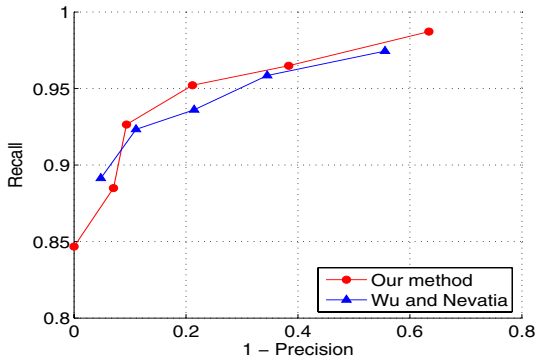


Figure 7. Detection PR curves for frontal/rear view pedestrian.

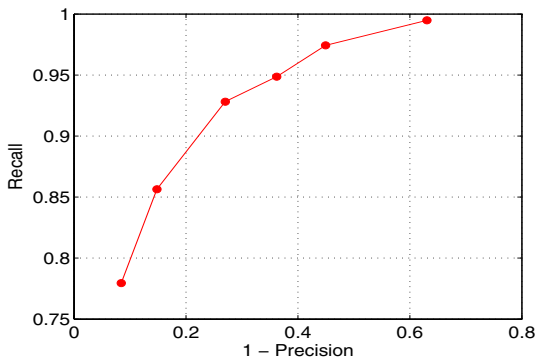


Figure 8. Detection PR curves for left profile view pedestrian.

All the curves in Fig.6 are performance on the normalized samples, where the size and position of the objects are aligned. However, when searching in a real image, the sliding window is usually not well aligned with the object. To evaluate our method under this situation and compare with others, we ran our system on two public test sets, the side view car set of Caltech101 image set [10] with 123 images and 123 cars, and the side view car set of TU Darmstadt image set [22] with 50 images and 50 cars. Both of the two sets are provided with segmentation ground-truth. These sets are designed for object recognition task. Although the

object positions are not aligned, the object sizes are roughly the same. This reduce the search space greatly. Our detector achieves 100% detection accuracy, *i.e.* no missed object and no false alarm, on these two sets. Another public set for car is the UIUC car set [21], but no segmentation ground-truth are provided for it. Although some existing methods report segmentation results on this set, we do not test our segmentor on it. Table.1 gives the comparison of our segmentation method with some previous works. Note that all the previous methods listed in Table.1 use a part of the sets as training and test on the rest; while our method trains on totally independent samples and uses the whole sets as testing. It can be seen that our method outperforms the others.

Method	TU Darmstadt	Caltech101	UIUC
Winn & Shotton [1]	-	-	96.5%*
Kapoor & Winn [6]	95.0%*	-	-
Winn & Jojic [12]	94.0%*	-	-
Our method	**97.6%**	**98.3%**	-

Table 1. The per-pixel figure-ground segmentation accuracy of side view car. (The number with a $\star$ means the testing is done on a subset of the image set.)

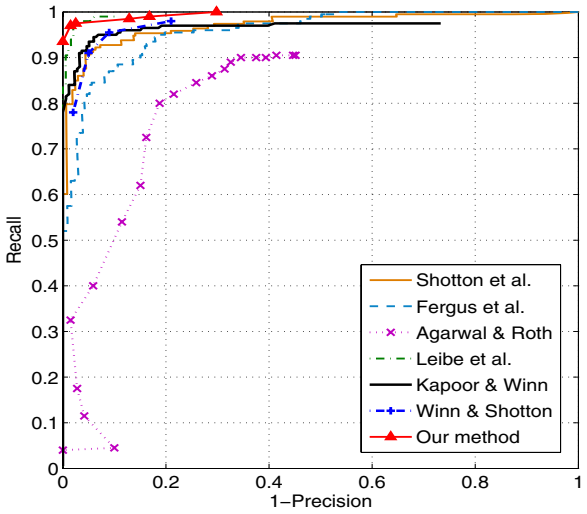


Figure 9. Detection PR curves on UIUC single-scale car test set.

We evaluate our car detector on the UIUC image set [21], which contains two test sets: a single-scale set with 170 images and 200 cars which are roughly the same size, and a multi-scale set with 108 images and 139 cars. This set contains cars of both left and right profile views. We mirror our left profile detector to get a right profile one, and apply these two detectors on the images. Fig.9 shows the PR curves of our method and some previous works on the single-scale test set. Table.2 lists the comparison of EER on both the single-scale and the multi-scale sets. It can be seen that our method is comparable to the state-of-the-art methods, and less affected by the multi-scale situation than the others. Some example results are shown in Fig.10.

Our program is coded in C++ using OpenCV functions. The experiments are done with a 2.8G Hz 32-bit Pentium PC. The training procedure needs about 72 hours. For single-scale detection and segmentation, the speed is about 200ms per image. For multi-scale, it is about 600ms per image.

Method	Single-scale	Multi-scale
Agarwal *et al*. [21]	$\sim$ 77%	$\sim$ 40%
Garg *et al*. [23]	$\sim$ 88.5%	-
Leibe *et al*. [22]	97.5%	-
Shotton *et al*. [13]	92.1%	-
Fritz *et al*. [16]	-	87.8%
Kapoor & Winn [6]	94.0%	-
Mutch & Lowe [3]	99.94%	90.6%
Our method	**97.5%**	**93.5%**

Table 2. Detection equal-error rates on the UIUC car image set.

5. Conclusion and Discussion

We developed a method to simultaneously detect and segment objects of a known category. Base detectors and segmentors are designed based on the edgelet features. Boosting algorithm is used to construct the cascade structured ensemble classifiers. Experimental results show that our method is comparable to the state-of-the-art methods for detection and outperforms the others for segmentation.

In this work, only the shape information is used. However, color and texture are very useful cues for image segmentation and object detection. We plan to investigate to use multiple complementary types of features to improve the performance in our future work.

Acknowledgements: This research was funded, in part, by the U.S. Government VACE program.

References

[1] J. Winn, and J. Shotton. The Layout Consistent Random Field for Recognition and Segmentation Partially Occluded Objects. CVPR 2006.

[2] E. Seemann, and B. Leibe, and B. Schiele. Multi-Aspect Detection of Articulated Objects. CVPR 2006.

[3] J. Mutch, and D. Lowe. Multiclass Object Recognition with Sparse, Localized Features. CVPR 2006.

[4] E. Borenstein, and J. Malik. Shape Guided Object Segmentation. CVPR 2006.

[5] S. Todorovic, and N. Ahuja. Extracting Subimages of an Unknown Category from a Set of Images. CVPR 2006.

[6] A. Kapoor, and J. Winn. Located Hidden Random Fields: Learning Discriminative Parts for Object Detection. ECCV 2006.

[7] J. Shotton, J. Winn, C. Rother, and A. Criminisi. TextonBoost: Joint Appearance, Shape and Context Modeling for Multi-Class Object Recognition and Segmentation. ECCV 2006.

[8] M. Bray, P. Kohli, and P. Torr. POSECUT: Simultaneous Segmentation and 3D Pose Estimation of Humans using Dynamic Graph-Cuts. ECCV 2006.

[9] A. Opelt, A. Pinz, and A. Zisserman. A Boundary-Fragment-Model for Object Detection. ECCV 2006.

Figure 10. Examples of detection and segmentation results: for each pair left is the detection result and right is the segmentation result. When there are multiple objects, different colors represent different objects. When there are overlap between two objects, we assume the object with larger y-coordinate (assume the left-top corner of the image is (0,0)) is closer to the camera.

[10] L. Fei-Fei, R. Fergus and P. Perona. One-Shot learning of object categories. IEEE Trans. Pattern Recognition and Machine Intelligence. 28(4): 594-611, April 2006.

[11] L. Zhao, and L. Davis. Closely Coupled Object Detection and Segmentation. ICCV 2005.

[12] J. Winn, and N. Jojic. LOCUS: Learning Object Class with Unsupervised Segmentation. ICCV 2005.

[13] J. Shotton, A. Blake, and R. Cipolla. Contour-Based Learning for Object Detection. ICCV 2005.

[14] B. Wu, and R. Nevatia. Detection of Multiple, Partially Occluded Humans in a Single Image by Bayesian Combination of Edgelet Part Detectors. ICCV 2005.

[15] C. Huang, H. Ai, Y. Li, and S. Lao. Vector Boosting for Rotation Invariant Multi-View Face Detection. ICCV 2005.

[16] M. Fritz, B. Leibe, B. Caputo, and B. Schiele. Integrating Representative and Discriminative Models for Object Category Detection. ICCV 2005.

[17] M. Pawan Kumar, P. Torr, and A. Zisserman. OBJ CUT. CVPR 2005.

[18] L. Bourdev, and J. Brandt. Robust Object Detection via Soft Cascade. CVPR 2005.

[19] B. Leibe, E. Seemann, and B. Schiele. Pedestrian Detection in Crowded Scenes. CVPR 2005.

[20] C. Huang, H. Ai, B. Wu, and S. Lao. Boosting Nested Cascade Detector for Multi-View Face Detection. ICPR 2004.

[21] S. Agarwal, A. Awan and D. Roth. Learning to detect objects in images via a sparse, part-based representation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 26(11):1475–1490, November 2004.

[22] B. Leibe, A. Leonardis, and B. Schiele. Combined Object Categorization and Segmentation with an Implicit Shape Model. Workshop on Statistical Learning in Computer Vision, in conjunction with ECCV 2004.

[23] A. Garg, S. Agarwal, T.S. Huang. Fusion of Global and Local Information for Object Detection. ICPR 2002.

[24] P. Viola, and M. Jones. Rapid Object Detection Using a Boosted Cascade of Simple Features. CVPR 2001.

[25] Z. Tu, S.-C. Zhu, and H.-Y. Shum. Image Segmentation by Data Driven Markov Chain Monte Carlo. ICCV 2001.

[26] G. Medioni and M.S. Lee and C.K. Tang. A Computational Framework for Segmentation and Grouping. Elsevier Science, 2000.

[27] D. M. Gavrila, and V. Philomin. Real-Time Object Detection for Smart Vehicles. ICCV 1999.

[28] R. E. Schapire and Y. Singer. Improved Boosting Algorithms Using Confidence-rated Predictions. Machine Learning, 37: 297-336, 1999.

[29] C. Papageorgiou, T. Evgeniou, and T. Poggio. A Trainable Pedestrian Detection System. In: Proc. of Intelligent Vehicles, 1998. pp. 241-246