Improving Part based Object Detection by Unsupervised, Online Boosting

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

Detection of objects of a given class is important for many applications. However it is difficult to learn a general detector with high detection rate as well as low false alarm rate. Especially, the labor needed for manually labeling a huge training sample set is usually not affordable. We propose an unsupervised, incremental learning approach based on online boosting to improve the performance on special applications of a set of general part detectors, which are learned from a small amount of labeled data and have moderate accuracy. Our oracle for unsupervised learning, which has high precision, is based on a combination of a set of shape based part detectors learned by off-line boosting. Our online boosting algorithm, which is designed for cascade structure detector, is able to adapt the simple features, the base classifiers, the cascade decision strategy, and the complexity of the cascade automatically to the special application. We integrate two noise restraining strategies in both the oracle and the online learner. The system is evaluated on two public video corpora.

1. Introduction

Detection of objects in images or videos is important for many applications, such as visual surveillance, content based image/video retrieval, *etc*. Recently, the boosting based framework, first proposed by Viola and Jones [13], has been successfully applied to detect some object categories, *e.g*. faces [13, 3] and pedestrians [2]. This framework uses a *supervised off-line* learning approach. In order to get a detector with good performance, tens of thousands samples could be needed [3]. Manually labeling such a huge amount of data is time-consuming. In some applications the environments considered are limited. For example, a surveillance system with stationary camera only watches a particular scene. In such a case, a specialized detector could be better than a general detector in terms of both accuracy and efficiency. With an off-line learning algorithm, to get a specialized detector we have to rerun the whole training procedure. Incremental learning, which adapts an existing general detector to a special task, is more desirable here.

In this work we propose an unsupervised, online learning approach to improve the performance of boosted object detectors learned from a small labeled training set, by using a large amount of unlabeled data. This online learning framework requires less manual labeling work and achieves better detection accuracy compared to the off-line learning. Our online learning algorithm "grows" a set of specialized part detectors from a set of general purpose seed part detectors. Unlike many common approaches, our method does not rely on motion segmentation techniques to detect the objects. We demonstrate this general framework on the object category of pedestrians.

1.1. Related work

The key components of unsupervised, online learning algorithms are 1) an automatic labeler, called an *oracle*, which segments and labels the objects from row data automatically, and 2) an online learning algorithm, which modifies the existing classifiers based on one sample at a time using the oracle's results.

The design of the oracle is not trivial, as it is an object detector itself. The difference of an oracle from a regular detector is that the precision of the oracle should be high while the detection rate could be low. Some existing work, *e.g*. [8, 4], use motion segmentation as oracle. However, motion based object detection is not robust due to many factors, such as shadows, reflections, merging and splitting of blobs, illumination change, *etc*. In order to improve the precision of the motion based oracle, some appearance based model can be used for verification, *e.g*. the PCA based representation by Roth *et al*. [4]. When the oracle is relatively weak, in order to get good precision, the labeling has to be very conservative, which results in very low detection rate.

Instead of making a separate oracle, some methods use the learning framework, called *co-training* [9], to improve the performance of a couple of classifiers by unlabeled data. The inputs of co-training are two classifiers and a set of unlabeled data. The confidently classified samples by the clas-

1-4244-1180-7/07/$25.00 ©2007 IEEE

sifier A/B are used to update the classifier B/A. It has been proved that when the two seed classifiers are not fully correlated, they can be improved by co-training on unlabeled data [9]. Levin *et al*. [12] use co-training to improve the performance of two vehicle detectors, which are learned from original images and foreground images. The correlation of these two types of inputs is relatively high, and the performance of the final detectors is not good. Javed *et al*. [5] apply co-training to boosted ensemble classifiers to classify moving blobs into vehicle and pedestrian. The samples confidently classified by a subset of the based classifiers in the ensemble are used to update the rest base classifiers. However, the base classifier, which is based on one dimension of a learned PCA model for vehicles and humans, is relatively weak, resulting in an ineffective oracle.

Given a sample collected by the oracle, some incremental learning algorithm is used to update the current classifier. Oza and Russell [14] propose an online version of boosting algorithm to learn ensemble classifier in an incremental way. Recently, some variations of this algorithm have been developed and applied to vision problems, including object detection [4, 5, 1]. For object detection, the boosted detectors [13] are very efficient because of their cascade structure. However, the existing online boosting algorithms are designed for standard ensemble classifiers, where the number of the base classifiers is fixed, and the decision is made only after all base classifiers are evaluated. An online learning algorithm for cascade detectors must be able to change the complexity of the cascade structure and to refine the cascade decision strategy adaptively.

The main issue of unsupervised learning is the oracle's errors, which can be categorized to two types for detection tasks, alignment error and labeling error. When a positive sample, *i.e*. a sub-window cut by the oracle, does include a object, but the position or size of the object is not accurate, we call this an alignment error. When the predicted label of a sample is wrong, we call this a labeling error. These errors make the learner over-fit, and must be restrained or eliminated in the oracle part, in the learning part, or in both. However none of the above efforts mention noise restraining strategies explicitly.

1.2. Outline of our approach

Our object detection system includes a set of cascade structured part detectors. Following the our previous work [2], we learn the seed part detectors by boosting edgelet feature based base classifiers. The size of the training set for the seed detectors is relatively small and the samples are for general purpose.

We design the oracle for unsupervised learning by combining the body part detection results, like in [2]. The confidence of samples are calculated from the part detection responses. Only the samples with high confidence are used for updating. To reduce the alignment errors, we learn a linear regressor to align the object samples. This oracle has high precision and unlike those in [8, 4, 5] it does not rely on motion segmentation.

We extend the online boosting algorithm in [14] to the case of cascade structured detectors. Our online boosting algorithm starts from the general seed detectors learned offline. Although theoretically online learning can start from scratch, Oza [15] has shown that the online boosting algorithm is likely to suffer a large loss initially when the seed model is too weak. Starting from some reasonable seed detectors learned with labeled data will make the online learning procedure more efficient.

Each base classifier in the detector is based on one edgelet feature. A shape affinity is defined to measure the similarity between two edgelet features. For each base classifier, a small neighborhood of it is constructed based on the shape affinity of edgelet. At each boosting iteration, the best base classifier in the neighborhood is selected. The decision strategy of the cascade is updated by looking at a short history of the collected samples. The sample passing rates of the base classifiers are estimated, based on which the number of the base classifiers is adapted.

We analyze the components of our method quantitatively on two public sets of surveillance videos. The experimental results show the efficiency of our system. Our main contributions are: 1) an oracle for unsupervised learning of object detection based on a set of part detectors; 2) an online learning framework for cascade structured detectors; and 3) the integration of noise restraining strategies in both the oracle and the learning components.

The rest of the paper are organized as follows: first in section 2, we introduce the data sets used for analysis and experiments; section 3 gives the off-line boosting algorithm, by which the seed detectors are learned; section 4 describes the oracle algorithm; section 5 describes the online boosting algorithm; the experimental results are given in section 6; and some conclusions and discussions in section 7.

2. Experimental Data Set

We have three data sets: a general positive sample set, a number of sequences from the CAVIAR video corpus [19], and a number of sequences from the CLEAR-VACE video corpus [20]. The general samples are collected from the MIT pedestrian set [18] and the Internet. There are 1,000 positive samples, and 1,000 negative images. The positive samples are normalized to 24×58 pixel. Both the positive samples and the negative images are for general purpose, without any bias for environment, illumination, *etc*. The size of the general set is relatively small and labeling it manually is affordable. We use this set to learn the seed part detectors.

We use the 26 sequences of the "shopping center corridor view", containing 36,292 frames, from the CAVIAR video corpus [19] to form our second data set. This set is captured with a stationary camera, mounted a few meters above the ground and looking down. We use six randomly selected sequences as a validation set for quantitative analysis; we use another ten sequences as the training set for online learning (we call this the *burn-in* set, in order to distinguish from the training set for off-line learning); the remaining ten sequences are used for testing.

The third data set consists of 10 sequences, containing 30,250 frames, from the CLEAR-VACE surveillance corpus [20]. The scene is an outdoor street. We use five sequences for burn-in and the other five for testing. Fig.1 shows some typical frames from the CAVIAR and the CLEAR-VACE sets.

(a) CAVIAR set (b) CLEAR-VACE set

Figure 1. Example frames of CAVIAR and CLEAR-VACE sets.

3. Learning of Seed Detectors

The original cascade in [13] has three levels: a base classifier, a strong classifier (or layer), and a cascade classifier. We modify the original structure to eliminate the concept of layers. Let h_t be the t-th base classifier, which is a mapping from the sample space $\mathcal{X}$ to a real valued range $[-1, 1]$. Let H_t be the partial sum of the first t base classifiers. Our modified cascade consists of T base classifiers, $\{h_t\}_{t=0}^{T-1}$, and T threshold $\{b_t\}_{t=0}^{T-1}$, a sample $\mathbf{x}$ is classified as positive iff

$$\forall t, H_t(\mathbf{x}) > b_t \tag{1}$$

This structure can be seen as a special case of the nested cascade proposed in [11] and the soft cascade proposed in [6]. One common advantage of these variations are the discriminative information obtained by the base classifiers are inherited along the cascade. Fig.2 gives our off-line learning algorithm, by which the part detectors are trained from the general sample set.

4. Oracle Design by Combining Part Detectors

We design our oracle based on our previous work [2], which combines the responses of part detectors to form object hypotheses. A part response is represented by a 4-tuple $\mathbf{rp} = \{l, \mathbf{p}, s, f\}$, where l is the part type, $\mathbf{p}$ is the image position, s is the size, and f is a classification confidence. For positive responses, f is defined by

$$f \triangleq 1 - \exp\left(-\frac{\sum_t h_t(\mathbf{x})}{\sum_t \max|h_t|}\right) \tag{5}$$

where $\mathbf{x}$ is the image patch of the part response, and $\max|h_t|$ is the maximum absolute value of h_t. For negative responses, f is defined by

$$f \triangleq \frac{1}{e-1}\left[1 - \exp\left(1 - \frac{T_{pass}}{T}\right)\right] \tag{6}$$

where T is the overall number of base classifiers in the cascade, and T_{pass} is the number of base classifiers the sample has passed. The negative confidence is designed based on the filtering property of cascade classifiers. Intuitively, the later in the cascade a sample is rejected, the more similar it is to real objects.

A combined response is represented by the set of its part responses and their visibility scores, v, $\mathbf{rc} = \{\mathbf{rp}_i, v_i\}_{i \in Prt}$, where Prt is the set of part labels. For humans, $Prt = \{FB, HS, TS, L\}$, where FB, HS, TS, L represent full-body, head-shoulder, torso, and legs respectively. The visibility score v is obtained from the combined detection algorithm as in [2].

4.1. Positive Sample Collection

Suppose we want to collected positive samples for part P_1, we define the *panel confidence* of a part response $\mathbf{rp}_{P_1}$ in a combined response $\mathbf{rc}$ by

$$\tilde{f}_{P_1} \triangleq \sum_{i \in Prt - \{P_1\} \wedge v_i > \theta_v} f_i \tag{7}$$

where θ_v is a visibility threshold (set to 0.7 in our experiments). The above confidence is called panel confidence, as it makes use of information from a set of part detectors; oppositely, we call f the *self confidence*. The panel confidence of P_1 does not include the self confidence of P_1, as we want to see the sample from different "views". When the panel confidence $\tilde{f}$ is larger than a threshold, θ_{pos}, we consider the sample confidently positive. We use two metrics to measure the performance of the oracle, *precision* and *utility ratio*. Suppose, there are N positive responses in total, after thresholding N_u are kept for online learning, in which N_c are good ones, then the precision and utility ratio are respectively defined by

$$Pr \triangleq \frac{N_c}{N_u}, \quad Ur \triangleq \frac{N_u}{N} \tag{8}$$

Fig.3 shows the curves of the two metrics with different θ_{pos} for the full-body detection on the CAVIAR validation set. In our experiments, we set $\theta_{pos} = 0.073$, which results in a precision of 98% and a utility ratio of 20%. Fig.4 shows examples of good and bad positive samples.

The positive samples not only need to be labeled correctly, but also must be aligned spatially. However, the spatial accuracy of the samples cut by the oracle is very good. We developed an automatic alignment method based on linear regression to improve the spatial accuracy. The input of

```
• Given the initial sample set $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^{m}$, where $\mathbf{x}_i \in \mathcal{X}, y_i = \pm 1$, 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 $F$, and the threshold for bootstrapping $\theta_B$;
• Construct the base classifier pool, $\mathcal{H}$, from the edgelet features;
• Initialize the sample weights $D_0(i) = 1/m$, the current false alarm rate $F_0 = 1$, and $t = 0$;
• while $t < T$ and $F_t > F$ do
    1. For each base classifier $h$ in $\mathcal{H}$, calculate $h$ as a histogram of its edgelet feature value $f_h$:
       $\forall f_h(\mathbf{x}) \in \left[\frac{j-1}{n}, \frac{j}{n}\right), h(\mathbf{x}) = \frac{1}{2} \ln \left( \frac{W_+^j + \varepsilon}{W_-^j + \varepsilon} \right)$    (2)
       where $n$ is the bin number of the histogram, $W_{\pm}^{j} = \sum_{f_h(\mathbf{x}_i) \in [\frac{j-1}{n}, \frac{j}{n}) \wedge y = \pm 1} D_t(i)$, and $\varepsilon$ is a smoothing factor [17];
    2. Select $h_t$ by
       $h_t = \arg\min_{h \in \mathcal{H}} \left\{ 2 \sum_{j=1}^{n} \sqrt{W_+^j W_-^j} \right\}$    (3)
    3. Update sample weights by
       $D_{t+1}(i) = D_t(i) \exp\left[-y_i h_t(\mathbf{x}_i)\right]$    (4)
       and normalize $D_{t+1}$ to a p.d.f.
    4. Select the threshold $b_t$ for the partial sum $H_t$, 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 the sample set. If the remaining negative samples are less than $\theta_B$ percent of the original, recollect the negative samples by bootstrapping on the negative image set;
    6. $t{+}{+}$
• Output $\{\{h_t\}, \{b_t\}\}$ as the cascade classifier.
```

Figure 2. Off-line learning algorithm of cascade classifier. In our experiments, $T = 4,000$, $F = 10^{-6}$, and $\theta_B = 75\%$. The setting of $\{P_t\}$ is similar to the original cascade's layer acceptance rates. The cascade is divided into 30 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%.

the regressor is a vector of the first 200 feature values of the detector, and the output is the positions of the head and feet. We learn the regressor from 500 labeled samples. Before alignment, the standard deviation of the head/feet positions is 1.73 pixels; after alignment, it reduces to 0.65 pixels.

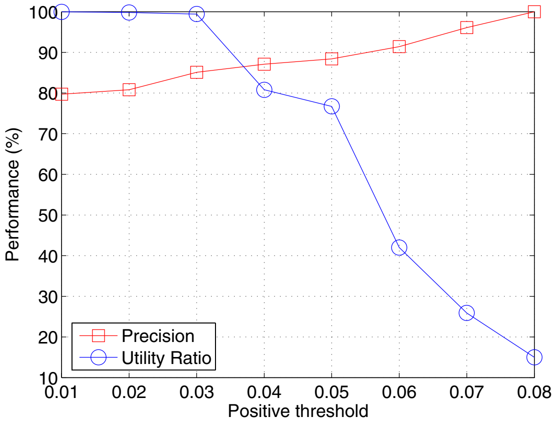


Figure 3. Performance of positive sample collection.

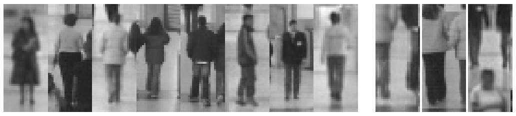
4.2. Negative Sample Collection

The collection of negative samples is similar to that of positive. Suppose we have a part response $\mathbf{rp}_{P_1}$ which does not correspond to any combined response. We calculate the panel confidence of $\mathbf{rp}_{P_1}$ by Equ.7. If $\tilde{f}_{P_1}$ is smaller than a threshold θ_{neg}, we consider the sample confidently negative. We set $\theta_{neg} = -2.88$, which results in a precision of 96% and a utility ratio of 18%.

As no quantitative analysis of the oracle is reported in previous work, it is difficult to compare our oracle with the existing methods directly. But one advantage of our method is that we do not rely on motion segmentation. Our oracle can be seen as an extension of the co-training framework. Instead of two classifiers dancing together, we have a group dance of multiple classifiers, which is much more reliable.

5. Online Boosting for Cascade Classifier

With video sequences as input, a series of samples are collected by the oracle and then fed into an online learning algorithm. We integrate the real valued classification function of real AdaBoost [17], the noise restraining techniques of ORBoost [7] and AveBoost2 [10], and the learning of cascade decision strategy into the online boosting [14].



(a) Good positive (b) Bad positive

Figure 4. Examples of positive samples collected.

5.1. Online Updating Base Classifiers

From the off-line training, we record the base classifier pool $\mathcal{H}$, and the weight distribution $W_{\pm}$. Given a new sample $(\mathbf{x}^{\star}, y^{\star})$ and its current weight w (the weight computation is described in section 5.2), we update $W_{\pm}$ of a base classifier h and then recompute h by Equ.2. As $W_{\pm}$ is a histogram, online updating of it is straightforward; formally,

$$\text{if } f_h(\mathbf{x}^{\star}) \in \left[\frac{j-1}{n}, \frac{j}{n}\right), W_{y^{\star}}^{j} = W_{y^{\star}}^{j} + w\alpha_{y^{\star}} \tag{9}$$

where $\alpha_{y^{\star}}$ is the weight updating rate. In our experiments, we set $\alpha_{+} = 10^{-3}$ and $\alpha_{-} = 10^{-5}$ as negative samples are more redundant.

To achieve some variability at the feature level, we construct a small neighborhood C_h of h, based on its edgelet feature. Denote an edgelet feature by $\mathbf{E} = \{\mathbf{u}_i, \mathbf{n}_i\}_{i=1}^{K}$, where K is the number of points, $\mathbf{u}_i$ and $\mathbf{n}_i$ are the image position and normal of the i-th point. The shape affinity between two edgelets $\mathbf{E}_1$ and $\mathbf{E}_2$ is defined by

$$A(\mathbf{E}_1, \mathbf{E}_2) \triangleq \frac{1}{K}\sum_{i=1}^{K} \langle \mathbf{u}_{1,i} - \bar{\mathbf{u}}_1, \mathbf{u}_{2,i} - \bar{\mathbf{u}}_2 \rangle \cdot e^{-\frac{1}{2}\|\bar{\mathbf{u}}_1 - \bar{\mathbf{u}}_2\|^2} \tag{10}$$

where $\bar{\mathbf{u}}$ is the mean of $\{\mathbf{u}_i\}$. We only consider the edgelets with the same length. The size of the neighborhood is set to 10. Fig.5 shows the features in the neighborhood of the first base classifier of the full-body detector. It can be seen that they cover a good variety. Given a new sample, we update all the base classifier in C_h, and select the best one according to Equ.3. This optimization strategy is similar to feature selection in [1]; however, our neighborhood is local and much smaller, because it is constructed not randomly, but based on the shape affinity of edgelets.

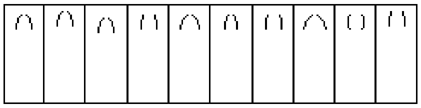


Figure 5. Features in the first neighborhood for full-body.

5.2. Weight Updating

The on-line boosting algorithm [14] imitates the weight evolution procedure of the off-line boosting. The weight updating strategy makes the learning procedure focus on the difficult instances, but, this also makes the boosting algorithms susceptible to labeling errors [16]; this is inevitable for un-supervised learning. We integrate the noise restraining strategies of AveBoost2 in [10] and ORBoost in [7] into our online boosting algorithm.

In the original real AdaBoost [17], the weights are updated by Equ.4. The exponential increase makes the learner over-fit on noises very fast. Oza [10] developed a boosting algorithm, called AveBoost2, in which the weight updating is smoothed by averaging the current weight with the previous one. It has been shown that AveBoost2 outperforms AdaBoost with noisy input [10]. We modify their off-line smoothing strategy so that

$$D_{t+1} = \frac{\gamma_w D_t \exp\left[-y^{\star} h_t(\mathbf{x}^{\star})\right] + D_t}{\gamma_w + 1} \tag{11}$$

where γ_w is a constant smoothing factor. In our experiments, we set $\gamma_w = 10$.

Although a smoothing technique is used, the weights of mislabeled samples tend to keep growing during boosting. Karmaker and Kwek [7] developed an off-line boosting approach, ORBoost, in which a cut-off threshold θ_c, is used as a ceiling of the weights. A sample is considered to be an outlier, if its weight grows larger than θ_c (set to 10 in our experiments). We integrate this technique into our online boosting algorithm. When the weight of a new sample hits the threshold, we stop updating, and take a "rollback" action. Fig.6 shows a comparison between online learning with and without noise restraining on the CAVIAR burn-in set. It can be seen that the tendencies of the two curves are similar, but the curve with noise restraining is more smooth and outperforms the one without noise restraining.

5.3. Updating Cascade Thresholds

The cascade decision strategy, $\{b_t\}$, learned from the general training set may not be optimal for a particular application. Online updating of the thresholds is necessary. We keep a short history of the positive samples collected, S_{+}. For the positive passing rates $\{P_t\}$ that are less than 100%, we sort the values, $\{H_t(\mathbf{x}_i)|\mathbf{x}_i \in S_{+}\}$, and then choose the threshold. For the $\{P_t\}$ that are 100%, we maintain the minimum value of $\{H_t(\mathbf{x}_i)|\mathbf{x}_i \in S_{+}\}$.

5.4. Adaptation of Cascade Complexity

In the previous online boosting algorithm [14, 1] the complexity of the classifier is fixed. However, similar to the situation of decision strategy, the complexity need to be adapted to the particular problem. We use the sample passing rate to measure the discriminative power of the cascade detector. When scanning an image, suppose there are $N_{pass,t}$ sub-windows passing the t-th partial sum H_t, then the sample passing rate of H_t is defined by

$$r_t \triangleq \frac{N_{pass,t}}{N_{pass,t-1}} \tag{12}$$

This passing rate reflects the contribution of the t-th base classifier h_t. The later the base classifier is in the cascade, the closer its r_t is to 1. Suppose at the beginning, there are T base classifiers in total. Denote by $r_T(0)$ the original sample passing rate of the whole cascade. During online learning, we keep updating all the sample passing rates r_t. If after learning with i samples, there exists a $r_t(i)$, such that $r_t(i) > r_T(0)$, we consider the base classifiers, $h_{t+1}, \ldots, h_T$, unnecessary, and remove them from the cascade. If after learning with i samples, $r_T(i) < r_T(0)$, we consider the current cascade to be relatively weak, and add more base classifiers to its end.

Now we put all the components together. Given a new sample collected by the oracle, it is sent through the current cascade. The base classifiers are updated and the sample weights are modified accordingly. For efficiency, the thresholds of the cascade are updated every 100 samples and the complexity of the cascade is adjusted every 1,000 samples. Fig.7 gives the full online boosting algorithm for the cascade classifier.

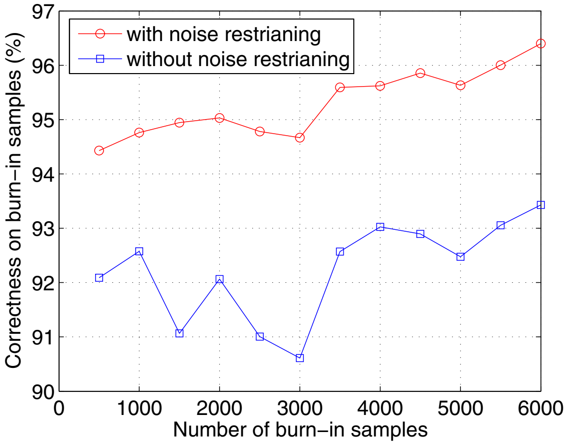


Figure 6. Performance of full-body detector learning on burn-in set with and without noise restraining strategies.

6. Experimental Results

We apply our method on the problem of pedestrian detection and evaluate our system on the CAVIAR set [19] and the CLEAR-VACE [20] set. The parameters of the oracle and the learner are determined based on the analysis on the validation set from CAVIAR corpus. The burin-in set, the test set, and the validation set have no overlap.

6.1. Results on CAVIAR set

We update the four seed part detectors online with the 10 burn-in sequences of CAVIAR set. For comparison we manually label $4,000$ samples from the burn-in set and collect 800 negative images for in-door environments from the Internet to form a specialized and "clean" training set, from which four specialized part detectors are learned by off-line boosting. The performance of the general seed detectors, the online updated detectors, and the specialized detectors are evaluated on the 10 test sequences of CAVIAR set. Table 1 gives the comparisons on performance and complexities of the detectors. It can be seen that by online learning both the part detectors and the combined detector, which serves as the oracle, are improved greatly. The average detection rate of the individual part detectors is increased by 20.0% while the false alarm rate is reduced by 2.0 per frame. The combined detector's detection rate is increased by 16.0% and its false alarm rate is reduced by 0.24 per frame.

The only previous online learning approach, which reports quantitative results on CAVIAR set is that in [4]. Our updated combined detector has a recall rate of 94.2% and a precision of 97.2%, which is much better than the 60% recall and 85% precision rates in [4]. In the four part detectors, only the complexity of the legs detector increases; this may be because the appearance variation of legs is larger than the other parts. Also the other three part detectors have better performance than the legs detector. Fig.8(a) gives some detection results before and after online learning. The specialized detectors can be seen as the upper limit of the online learning algorithm. Although the accuracy of our online updated detectors are comparable to the specialized ones, the specialized detectors use many fewer base classifiers.

		FB	HS	Ts	L	Com
Original	DR	74.9	68.7	69.1	53.5	78.2
	FA	0.7	3.9	3.9	3.7	0.3
	# of BC	1800	2900	1000	1800	
Updated	DR	91.2	85.7	87.5	80.9	94.2
	FA	0.2	0.8	1.1	1.9	0.06
	# of BC	800	1400	800	2100	
Specialized	DR	91.5	84.7	89.4	81.1	94.9
	FA	0.4	0.3	0.5	0.4	0.03
	# of BC	200	400	300	500	

Table 1. Comparison on the CAVIAR set (DR: detection rate in percentage; FA: false alarm per frame; BC: base classifier; Com: combined).

6.2. Results on CLEAR-VACE set

For the second set from CLEAR-VACE corpus [20], we update the seed detectors online with the 5 burn-in sequences, and evaluate on the 5 test ones. Table 2 gives the comparisons. This set is harder than the CAVIAR set, as the scene is more cluttered. It can be seen that although we achieve similar improvements on accuracy, more base classifiers are needed for this set than for the CAVIAR set. Both the complexities of the legs detector and the torso detector increase after online learning. The average detection rate of the part detectors is increased by 19.7% and the false alarm rate is reduced by 2.4 per frame. The combined detector's detection rate is increased by 14.7% and its false alarm rate

```
• Inherit from the off-line boosting procedure: the cascade detector $\{\{h_t\},\{b_t\}\}$, the base classifier pool $\mathcal{H}$, the weight distribution $W_{\pm}$, the neighborhood $\{C_t\}$, the positive passing rate $\{P_t\}$, and the training set $S$;
• Set the algorithm parameters: the updating rate for positive/negative samples $\alpha_{\pm}$, the smoothing rate $\gamma_w$ and the cut-off threshold $\theta_c$ of weight updating ;
• Compute the sample passing rate $\{r_t\}$ from the first $50$ frames of the burn-in set;
• Initialization: populate the sample history $S_+$ and $S_-$ by $S$, and $i = 0$.
• For all frames in the burn-in set do
    – Get a new frame, from which use the oracle to obtain a number of samples, $(\mathbf{x}^\star_{i+1}, y^\star_{i+1}), \ldots, (\mathbf{x}^\star_{i+m}, y^\star_{i+m})$;
    – Update the sample passing rate $\{r_t\}$;
    – For all the samples collected from this frame do
        ∗ $i$++;
        ∗ Initialize sample weight $D_0 = 1$;
        ∗ For $t = 0$ to $T - 1$ do, where $T$ is the size of the current cascade
            1. Update the weight distribution of every $h \in C_t$ by Equ.9, and recompute $h$;
            2. Find the best base classifier in $C_t$ by minimizing the criterion in Equ.3
            3. Compute sample weight $D_{t+1}$ by Equ.11
            4. If $D_{t+1} > \theta_c$ break updating and rollback;
            5. Add $\mathbf{x}^\star_i$ to $S_{y^\star}$. If $i \mod 100 = 0$, update $\{b_t\}$ according to $S_{+1}$ and $\{P_t\}$;
            6. If $i \mod 1000 = 0$, adapt the complexity of the cascade according to $\{r_t\}$, and update $T$.
        ∗ Update the oracle.
• Output $\{\{h_t\},\{b_t\}\}$ as the cascade classifier.
```

Figure 7. Online learning algorithm of cascade classifier.

is reduced by 0.22 per frame. Fig.8(b) gives some detection results before and after online learning.

7. Conclusion and Discussion

We proposed an unsupervised, online learning approach to improve the performance of a set of part detectors for objects of a known category. The oracle in our system, which is based on the combination of part detection responses, has very high precision and does not rely on motion segmentation. Our online learner, which is based on an online boosting algorithm, adapts the local shape features, the base classifiers, the cascade decision strategy, and the complexity of the classifier automatically. The experimental results show that our method can greatly improve the performance on a particular application of the seed detectors by learning from a large amount of unlabeled data.

		FB	HS	Ts	L	Com
Original	DR	73.1	65.0	67.8	58.7	78.4
	FA	0.7	4.0	3.6	7.1	0.3
	# of BC	1800	2900	1000	1800	
Updated	DR	92.4	80.5	88.3	82.2	93.1
	FA	0.5	1.5	1.7	2.0	0.08
	# of BC	1000	1800	1600	2400	

Table 2. Comparison on the CLEAR-VACE set (See Table 1 for abbreviation).

In our experiments, we learned one cascade detector for all view points and all poses. This enables us to focus our analysis on the online learning algorithm. In practice, a view based detector [11] or a tree structured detector [3] could have better performance; our online learning algorithm is easy to extend to these more complicated classifiers.

Although the experimental videos used are captured by stationary cameras, our system is able to work on moving/zooming camera as it does not rely on motion segmentation. Also, even though we have shown results for one class of objects (pedestrians) only, our methodology applies to any type of objects for which reasonably component part detectors can be constructed.

Besides detection, online learning can also be used to improve the performance of object tracking methods, *e.g.* the tracker in [1]. We plan to extend our current framework to tracking problems in the future work.

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

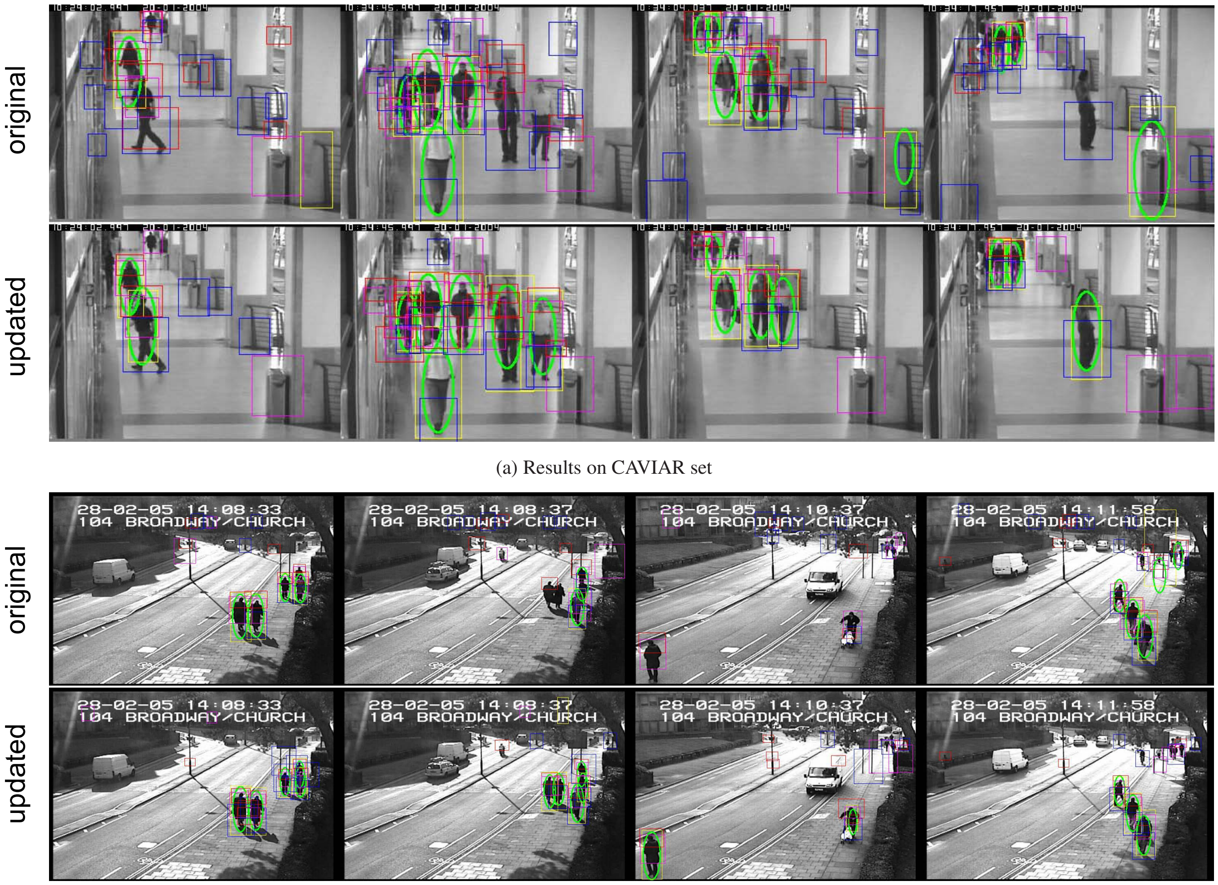
References

[1] H. Grabner and H. Bischof. Online Boosting and Vision. CVPR 2006.

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

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

[4] P. Roth, H. Grabner, D. Skocaj, H. Bischof, and A. Leonardis. Online Conservative Learning for Person Detection. VS-PETS 2005.



(a) Results on CAVIAR set

(b) Results on CLEAR-VACE set

Figure 8. Examples of detection results before and after online learning. (Yellow for full-body; red for head-shoulder; purple for torso; blue for legs; and green for combined)

[5] O. Javed, S. Ali, and M. Shah. Online Detection and Classification of Moving Objects Using Progressively Improving Detectors. CVPR 2005.

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

[7] A. Karmaker, and S. Kwek. A Boosting Approach to Remove Class Label Noise. In the Fifth International Conference on Hybrid Intelligent System, 2005.

[8] V. Nair and J. Clark. An Unsupervised, Online Learning Framework for Moving Object Detection. CVPR 2004.

[9] M.-F. Balcan, A. Blum, and K. Yang. Co-Training and Expansion: Towards Bridging Theory and Practice. NISP 2004.

[10] N. Oza. AveBoost2: Boosting for Noisy Data. In the Fifth International Workshop on. Multiple Classifier Systems. 2004

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

[12] A. Levin, P. Viola, and Y. Freund. Unsupervised improvement of visual detectors using co-training. ICCV, 2003.

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

[14] N. Oza and S. Russell. Online Bagging and Boosting. In the Eighth International Workshop on Artificial Intelligence and Statistics, 2001.

[15] N. Oza. Online Ensemble Learning. PhD thesis, University of California, Berkeley, 2001.

[16] T. Dietterich. An Experimental Comparision of Three Methods for Constructing Ensembles of Decision Trees: Bagging, Boosting and Randomization. Machine Learning: 40, 139-158, 2000.

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

[18] C. Papageorgiou, T. Evgeniou, and T. Poggio A Trainable Pedestrian Detection System. Intelligent Vehicles, 1998. pp. 241-246

[19] http://homepages.inf.ed.ac.uk/rbf/CAVIAR/

[20] http://isl.ira.uka.de/clear06/?Evaluation_Tasks