Efficient Detector Adaptation for Object Detection in a Video

Pramod Sharma and Ram Nevatia
Institute for Robotics and Intelligent Systems, University of Southern California
Los Angeles, CA 90089, USA
{pksharma|nevatia}@usc.edu

Abstract

In this work, we present a novel and efficient detector adaptation method which improves the performance of an offline trained classifier (baseline classifier) by adapting it to new test datasets. We address two critical aspects of adaptation methods: generalizability and computational efficiency. We propose an adaptation method, which can be applied to various baseline classifiers and is computationally efficient also. For a given test video, we collect online samples in an unsupervised manner and train a random fern adaptive classifier . The adaptive classifier improves precision of the baseline classifier by validating the obtained detection responses from baseline classifier as correct detections or false alarms. Experiments demonstrate generalizability, computational efficiency and effectiveness of our method, as we compare our method with state of the art approaches for the problem of human detection and show good performance with high computational efficiency on two different baseline classifiers.

1. Introduction

Object detection is a challenging problem because of variations in different viewpoints, appearance, illumination etc . Common procedure for object detection is to train an object detector in an offline manner by using thousands of training examples. However, when applied on novel test data, performance of the offline trained classifier (baseline classifier) may not be high, as the examples in test data may be very different than the ones used for the training.

Several incremental/online learning based detector adaptation methods [7, 24, 11, 15, 26, 17, 19, 23, 21] have been proposed to address this issue. Most of these approaches are either boosting based [7, 24, 17] or SVM based [11, 21], which limits applicability of these approaches to a specific type of baseline classifier. We propose a detector adaptation method, which is independent of the baseline classifier used, hence is applicable to various baseline classifiers.

With increasing size of new test video datasets, computational efficiency is another important issue to be addressed. [21, 11, 24] use manually labeled offline training samples for adaptation, which can make the adaptation process computationally expensive, because the size of the training data could be large after combining offline and online samples. Some approaches [7, 17] have been proposed to address this issue, as they do not use any offline sample during the training of the adaptive classifier. However, these approaches optimize the baseline classifier using gradient descent methods, which are inherently slow in nature.

Figure 1. Some examples from Mind's Eye dataset [1]. This is a challenging dataset, as it has many different human pose variations.

Detector adaptation methods need online samples for training. Supervised [7] and semi-supervised [6, 26] methods require manual labeling for online sample collection, which is difficult for new test videos. Hence, unsupervised sample collection is important for adaptation methods.

Background subtraction based approaches [12, 13, 10, 15] have been used for unsupervised online sample collection. However background subtraction may not be reliable for complex backgrounds. Tracking based methods [17, 4] have also been used. However, existing state of the art tracking methods [9, 18, 25] work well for pedestrian category only. Therefore, these methods may not be applicable for different kinds of objects with many pose variations and articulations (see Figure 1).

We propose a novel generalized and computationally efficient approach for adapting a baseline classifier for a specific test video. Our approach is generalized because it is independent of the type of baseline classifiers used and does not depend on specific features or kind of training algorithm used for creating the baseline classifier.

For a given test video, we apply the baseline classifier at a high precision setting, and track obtained detection responses using a simple position, size and appearance based tracking method. Short tracks are obtained as tracking output, which are sufficient for our method, as we do not seek long tracks to collect online samples. By using tracks and detection responses, positive and negative online samples are collected in an unsupervised manner. Positive online samples are further divided into different categories for variations in object poses. Then a computationally efficient multi-category random fern [14] classifier is trained as the adaptive classifier using online samples only. The adaptive classifier improves the precision of baseline classifier by validating the detection responses obtained from the baseline classifier as correct detections or false alarms

Rest of this paper is divided as follows: Related work is presented in section 2. Overview of our approach is provided in section 3. Our unsupervised detector adaptation approach is described in section 4. Experiments are shown in section 5, which is followed by conclusion.

2. Related Work

In recent years, significant work has been published for detector adaptation methods. Supervised [7] and semi supervised [6, 26] approaches, which require manual labeling, have been proposed for incremental/online learning but manual labeling is not feasible for the large number of videos. Background subtraction based methods [12, 13, 10, 15] have been proposed for unsupervised online sample collection, but these methods are not applicable for datasets with complex backgrounds. Many approaches [24, 4, 17, 23] have used detection output from the baseline classifier or tracking information for unsupervised online sample collection. Unsupervised detector adaptation methods can be broadly categorized into three different categories: Boosting based methods, SVM based approaches and generic adaption methods.

Boosting based: Roth et al. [15] described a detector adaptation method in which they divide the image into several grids and train an adaptive classifier separately for each grid. Training several classifiers separately, could be computationally expensive. Wu and Nevatia [24] proposed an online Real Adaboost [16] method. They collect online samples in an unsupervised manner by applying the combination of different part detectors.

Recently, Sharma et al. [17] proposed an unsupervised incremental learning approach for Real Adaboost framework by using tracking information to collect the online samples automatically and extending the Real Adaboost exponential loss function to handle multiple instances of the online samples. They collect missed detections and false alarms as online samples, therefore their method relies on tracking methods which can interpolate object instances missed by the baseline classifier. Our proposed approach uses a simple position, size and appearance based tracking method in order to collect online samples. This simplistic tracking method produces short tracks without interpolating missed detections, which is sufficient for our approach.

SVM based: Kembhavi et al. [11] proposed an incremental learning method for multi kernel SVM. Wang et al [21] proposed a method for adapting the detector for a specific scene. They used motion, scene structure and geometry information to collect the online samples in unsupervised manner and combine all this information in confidence encoded SVM. Their method uses offline training samples for adaptation, which may increase the computation time for training the adaptive classifier.

Both boosting and SVM based adaptation methods are limited to a specific kind of algorithm of baseline classifier, hence are not applicable for various baseline classifiers.

Generic: In [23], Wang et al. proposed a detector adaptation method in which they apply the baseline classifier at low precision and collect the online samples automatically. Dense features are extracted from collected online samples to train a vocabulary tree based transfer classifier. They showed the results on two types of baseline classifiers for pedestrian category, whereas our proposed method show the performance with different articulations in human pose in addition to the pedestrian category.

3. Overview

The objective of our work is to improve the performance of a baseline classifier by adapting it to a specific test video. An overview of our approach is shown in Figure 2. Our approach has the following advantages over the existing detector adaptation methods:

1. **Generalizability:** Our approach is widely applicable, as it is not limited to a specific baseline classifier or any specific features used for the training of the baseline classifiers.

2. **Computationally Efficient:** Training of the random fern based adaptive classifier is computationally efficient. Even with thousands of online samples, adaptive classifier training takes only couple of seconds .

3. **Pose variations:** It can handle different pose variations and articulations in object pose.

For online sample collection, we apply baseline detector at a high precision (high threshold) setting. Obtained detection responses, are tracked by applying a simple tracking-by-detection method, which only considers the association of detection responses in consecutive frames based on the size, position and appearance of the object. For each frame, overlap between bounding boxes of the output tracks and

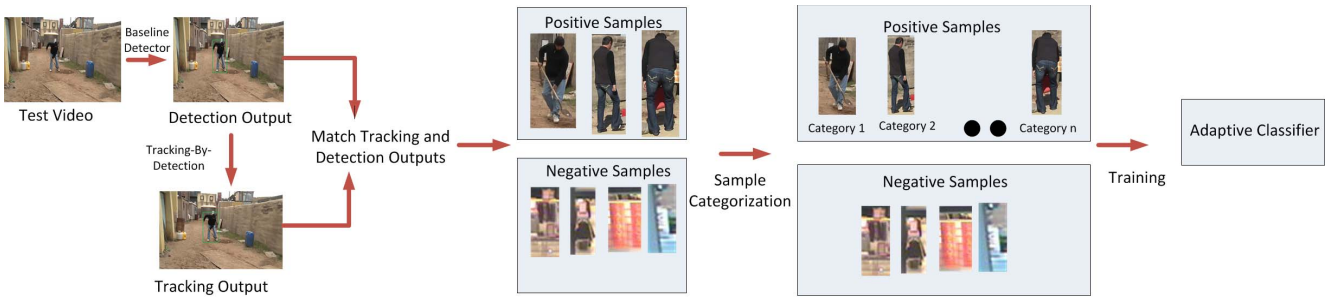


Figure 2. Overview of our detector adaptation method

detection responses is computed. Those detection responses which match with the track responses and have a high detection confidence are collected as positive online samples. False alarms are collected as negative online samples. Positive online samples are further divided into different categories for variations in the poses for the target object and then a random fern classifier is trained as adaptive classifier.

Testing is done in two stages: First we apply the baseline classifier at a high recall setting (low threshold). In this way, baseline classifier produces many correct detection responses in addition to many false alarms. In the next stage, these detection responses from baseline classifier are provided to the learned random fern adaptive classifier, which classifies the obtained detection responses as the correct detections or the false alarms. In this way our adaptation method improves the precision of the baseline classifier.

We demonstrate the performance of our method on two datasets: CAVIAR [2] and Mind's Eye [1]. We show the generalizability of our method by applying it on two different baseline classifiers: boosting based [8] and SVM based [5] classifier. Experiments also show that the method is highly computationally efficient and outperforms the baseline classifier and other state of the art adaptation methods.

4. Unsupervised Detector Adaptation

In the following subsections, we describe the two different modules of our detector adaptation method : Online sample collection and training of the random fern based adaptive classifier.

4.1. Unsupervised Training Samples Collection

To collect the online samples, we first apply the baseline classifier at high precision setting for each frame in the video and obtain the detection responses $\mathcal{D} = \{\mathbf{d_i}\}$. These detection responses are then tracked by using a simple low level association [9] based tracking-by-detection method. A detection response $\mathbf{d_i}$ is represented as $\mathbf{d_i} = \{x_i, y_i, s_i, a_i, t_i, l_i\}$. (x_i, y_i) represents the position of the detection response, s_i its size, a_i its appearance, t_i its frame index in the video and l_i the confidence of the detection response. The link probability between two detection responses $\mathbf{d_i}$ and $\mathbf{d_j}$ is defined as :

$$P_l(\mathbf{d_j}|\mathbf{d_i}) = \mathcal{A}_p(\mathbf{d_j}|\mathbf{d_i})\mathcal{A}_s(\mathbf{d_j}|\mathbf{d_i})\mathcal{A}_a(\mathbf{d_j}|\mathbf{d_i}) \quad (1)$$

where $\mathcal{A}_p$ is the position affinity, $\mathcal{A}_s$ is size affinity and $\mathcal{A}_a$ is the appearance affinity. If the frame difference between two detection responses is not equal to 1, the link probability is zero. In other words, the link probability is only defined for detection responses in consecutive frames.

$\mathbf{d_i}$ and $\mathbf{d_j}$ are only associated with each other, if $P_l(\mathbf{d_j}|\mathbf{d_i})$ is high:

$$P_l(\mathbf{d_j}|\mathbf{d_i}) > \max(P_l(\mathbf{d_j}|\mathbf{d_k}), P_l(\mathbf{d_l}|\mathbf{d_i})) + \lambda, \forall\ (k \neq i,\ l \neq j) \quad (2)$$

where λ is an adjustment parameter. Obtained track responses $\mathcal{T} = \{\mathbf{T_i}\}$ are further filtered and tracks of length 1 are removed from $\mathcal{T}$.

4.1.1 Online Samples

For each frame in the video, the overlap between the bounding boxes of $\mathcal{D}$ and $\mathcal{T}$ is computed. A detection response $\mathbf{d_i}$ is considered as positive online sample if:

$$\mathcal{O}(\mathbf{d_i} \cap \mathbf{T_k}) > \theta_1\ and\ l_i > \theta_2 \quad (3)$$

Where $\mathcal{O}$ is the overlap of the bounding boxes of $\mathbf{d_i}$ and $\mathbf{T_k}$. θ_1 and θ_2 are the threshold values. Also one track response can match with one detection response only.

On the other hand, a detection response is considered as negative online sample if:

$$\mathcal{O}(\mathbf{d_i} \cap \mathbf{T_k}) < \theta_1\ \forall k = 1, ...M,\ and\ l_i < \theta_3 \quad (4)$$

where M is the total number of track responses in a particular frame.

High confidence for detection response increases the likelihood that the obtained response is a positive sample. Similarly low confidence for detection response, would lead to high false alarm probability. Some of the collected positive and negative online samples are shown in Figure 3.

4.1.2 Pose Categorization

We consider different pose variations in the target object (e.g. standing, sitting, bending for human) as different categories, as the appearance of the target object varies considerably with the articulation in the pose. Hence, we divide the positive online samples into different categories. For this purpose, we use the poselet [5] detector as the baseline classifier. A detection response $\mathbf{d_i}$ obtained from the poselet detector is represented as $\mathbf{d_i} = \{x_i, y_i, s_i, a_i, t_i, l_i, h_i\}$, where h_i is the distribution of the poselets. We model this distribution with 150 bin histogram, each bin depicting one of the 150 trained poselets.

We train a pose classifier offline, in order to divide the positive online samples into different categories. We collect the training images for different variations in the human pose and compute the poselet histograms for these training images, by applying the poselet detector. The poselet histogram set $\mathbb{H} = \{\widehat{h}_i\}$ is utilized for dividing the samples into different categories.

For a given test video, collected positive online samples are represented as, $\mathcal{P} = \{\mathbf{P_i}\}$, where $\mathbf{P_i} = \{x_i, y_i, s_i, a_i, h_i, l_i, v_i\}$, v_i is the target category, which is determined as:

$$v_i = \arg\min_{l}(\mathbf{B}(h_i, \widehat{h}_l)) : h_i \in \mathbf{P_i}, \widehat{h}_l \in \mathbb{H} \qquad (5)$$

where $\mathbf{B}$ is the Bhattacharya distance [20]. In this manner we divide the positive online samples into different categories. Each of these categories are considered as a separate class for adaptive classifier training.

4.2. Adaptive Classifier Training

Ozuysal et al. proposed an efficient random fern [14] classifier, which uses binary features to classify a test sample. These binary features are defined as a pair of points chosen randomly for a given reference window size of the input training samples and based on the intensity values of the points in the pair, the feature output is determined. For a given test sample, let $\{C_1, C_2, ..., C_K\}$ be the K target classes and $\{f_1, f_2, ..., f_N\}$ are N binary features. The target category $c_i^{'}$ is determined as:

$$c_i^{'} = \arg\max_{c_i} P(f_1, f_2, ..., f_N \mid C = c_i) \qquad (6)$$

In order to classify an image with binary features, many of such features are needed, which makes the computation of joint distribution of features $P(f_1, f_2, ..., f_N)$ infeasible. On the other hand, if we assume all the features are independent, it will completely ignore the correlation among features. Hence, these features are divided into independent groups, called ferns. If we have total M ferns, each fern will have $\frac{N}{M}$ features, and conditional probability

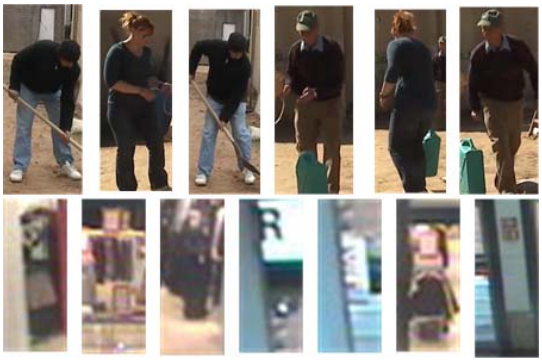


Figure 3. Examples of some of the positive (first row) and negative (second row) online samples collected in unsupervised manner from Mind's Eye [1] and CAVIAR [2] datasets.

$P(f_1, f_2, ..., f_N \mid C = c_i)$ can be written as:

$$P(f_1, f_2, ..., f_N \mid C = c_i) = \prod_{k=1}^{M} P(F_k \mid C = c_i) \qquad (7)$$

where F_k is the set of binary features for the k^{th} fern.

During the training, distribution of each category is computed for each fern independently. This distribution is modeled as a histogram, where each category distribution has $L = 2^{\frac{N}{M}}$ bins, as the output of the $\frac{N}{M}$ binary features will have $2^{\frac{N}{M}}$ possible values. For a test sample, we compute $P(F_k \mid C = c_i)$ as:

$$P(F_k \mid C = c_i) = \frac{n_{k,i,j}}{\sum_{j=1}^{L} n_{k,i,j} + \beta} \qquad (8)$$

where $n_{k,i,j}$ is the value of the j^{th} bin from the distribution of i^{th} category for k^{th} fern, β is a constant. Learning algorithm of random fern based adaptive classifier is described in algorithm 1.

For the training of the adaptive classifier, we only use online samples collected in an unsupervised manner, no manually labeled offline samples are used for the training. We train a multi-class random fern adaptive classifier by considering different categories of the positive samples as different target classes, all negative online samples are considered as a single target class. For a test video, first online samples are collected from all the frames and then random fern classifier is trained. Training procedure is performed only once for a given test video.

5. Experiments

We performed experiments for the problem of human detection and evaluated our method for generalizability, computation time performance and detection performance. We performed all experiments on a 3.16 GHz, Xeon computer.

Algorithm 1 Unsupervised Detector Adaptation

• **Training**:
• **Given**: $\mathbb{D}$, $\mathbb{T}$, $\mathbb{H}$, Test Video V, with total F frames
• **Init**: Positive Online Samples, $\mathbb{P} = \{\}$, Negative online samples, $\mathbb{N} = \{\}$
for $i = 1$ to F **do**
  - Match $\mathbb{D}$ with $\mathbb{T}$ and collect positive (S^+) and negative (S^-) samples for this frame
  - $\mathbb{P} = \mathbb{P} \cup S^+$, $\mathbb{N} = \mathbb{N} \cup S^-$
for $i = 1$ to $|\mathbb{P}|$ **do**
  • **Init**: $v_i = -1$, $d_{min} = \infty$
  for $j = 1$ to $|\mathbb{H}|$ **do**
    - $\gamma = \mathbf{B}(\mathbf{h_i}, \widehat{\mathbf{h_j}})$
    if $\gamma < d_{min}$ **then**
      $d_{min} = \gamma$, $v_i = j$
- Train Random fern classifier using online samples $\mathbb{P}$ and $\mathbb{N}$.
• **Test**:
for $i = 1$ to F **do**
  - Apply baseline classifier at low threshold δ to obtain detection responses $\mathbb{D}_f$
  for $j = 1$ to $|\mathbb{D}_f|$ **do**
    - Apply Random fern classifier to validate the detection responses as the true detections and false alarms

In this section, we provide the experiment details and show the performance of our method:

Datasets: Two different datasets are used for experiments: CAVIAR [2] and Mind's Eye [1]. Two sequences: OneShopLeave2Enter (CAVIAR1) and WalkByShop1front (CAVIAR2) are used from CAVIAR dataset. These sequences have 1200 and 2360 frames respectively of size 384 X 288. Ground-truth (GT) is available at [2]. CAVIAR1 has 290 GT instances of the human, whereas CAVIAR2 has 1012 GT instances of the human.

Two video clip sequences (ME1 and ME2) are used from Mind's Eye dataset. These sequences have 425 and 300 frames respectively, each of size 1280 X 760. We manually annotated the ground-truth for these sequences. ME1 has 641 GT instances of the human, whereas ME2 has 300 GT instances of the human. ME1 has two different pose variations: standing/walking, bending, whereas ME2 has the pose variations for digging and standing/walking.

Baseline classifiers: To demonstrate the generalizability of our approach, we performed experiments with two different baseline classifiers: For CAVIAR, boosting based classifier is used as described in [8]. For Mind's Eye dataset, we used publicly available trained poselets and Matlab implementation available at [3].

In the following subsections, we present computation time and detection performance experiments.

5.1. Computation Time Performance

We evaluated computational efficiency of our approach for the training of the adaptive classifier after collection of online samples. We performed this experiment for online samples collected from CAVIAR dataset and trained the adaptive classifier for two target categories. For the adaptive classifier training, we only use the online samples collected in unsupervised manner, no offline samples are used for the training.

We compare the performance of our method with [17], which also does not use any of the offline samples for incremental learning. [17] uses bags of instances, instead of single instance, hence we count all the training samples in the bag in order to count the total number of samples used for the training.

In Figure 4, we show the run time performance for different number of ferns and number of binary features. We can see that random fern based adaptive classifier training outperforms [17] in run time performance. [17] optimizes parameters of baseline detector using gradient descent method, hence training time of incremental detector is high. Whereas our random fern adaptive classifier is independent of the parameters of baseline classifier and uses simple binary features for the training, hence is computationally efficient.

For CAVIAR dataset, we use 30 random ferns with 10 binary features, which takes only 1.2 seconds for training of 1000 online samples, whereas the method described in [17] takes approximately 35 seconds, which makes our method approximately 30 times faster than [17]. Total training time of random fern classifier for CAVIAR1 sequence takes only 8 seconds for approximately 16000 online samples, whereas for CAVIAR2 it takes only 19 seconds with approximately 43000 online samples.

5.2. Detection Performance

We evaluated the detection performance for the CAVIAR and Mind's Eye datasets. We do not use any prior on the size and the location of the object for either detection or tracking. Tracking parameters are used as described in [9]. β is set to 0.1 for all the experiments. For detection evaluation, we used the same criteria as used in [8]. This metric considers a detection output as correct detection only if it has more than 50% overlap with ground truth.

5.2.1 CAVIAR Dataset

For this dataset, we use Real Adaboost based baseline classifier [8] and train it for 16 cascade layers for full body of human. 30 random ferns are trained for 10 binary features, for two target categories (positive and negative classes).

Division of positive samples into different categories is not required for this dataset, as all the humans in the se-

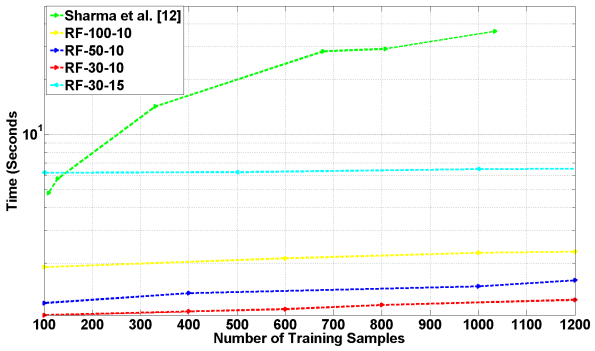


Figure 4. Run time performance of our approach. X-axis represents the number of online samples used for the classifier training, Y-axis is shown in log scale and represents runtime in seconds. RF-I-K : I random ferns with K binary features.

quence belong to the pedestrian category. θ_1, θ_2 and θ_3 were empirically set to 0.3, 20 and 10 respectively. These parameters remain same for all the experiments on this dataset.

We compare the performance of our method with two state of the art approaches [23, 17]. From Figure 5, we can see that our method significantly outperforms both HOG-LBP [22] and [23]. Also from Table 1, we can see that for CAVIAR1, at a recall of 0.65, Sharma et al's method improves the precision over baseline by 14%, whereas our method improves the precision by 22%. For CAVIAR2, our method improves the precision over baseline by 59% at recall of 0.65, whereas Sharma et al.'s method improves the precision by 19%.

Both our approach and Sharma et al's method outperforms baseline detector [8], however for CAVIAR2 sequence, long tracks are not available for some of the humans, hence not enough missed detections are collected by Sharma et al's approach, due to which its performance is not as high. Our approach does not require long tracks, hence gives better performance as compared to [17].

5.2.2 Mind's Eye Dataset

We use trained poselets available at [3] for experiments on Mind's Eye dataset. We train 15 random ferns with 8 binary features for the adaptive classifier training. Adaptive classifier is trained for four target categories (standing/walking, bending, digging and negative). θ_1 is set to 0.3, whereas θ_2 and θ_3 are set to 40 and 20 respectively. These parameter settings remain same for all the experiments on this dataset.

During online sample collection, not many negative samples are obtained, hence we add approximately 1100 negative online samples collected in unsupervised manner from the CAVIAR dataset in the online negative samples set for both the ME1 and ME2 sequences. Three training images

Table 1. Precision improvement performance for CAVIAR dataset at recall 0.65

Sequence	**Sharma** [17]	**baseline** [8]	**Ours**
CAVIAR1	0.56	0.42	**0.64**
CAVIAR2	0.4	0.21	**0.8**

are used to learn pose categorization histograms. These images have standing/walking, bending and digging poses respectively. None of these training images are from either ME1 or ME2 sequence.

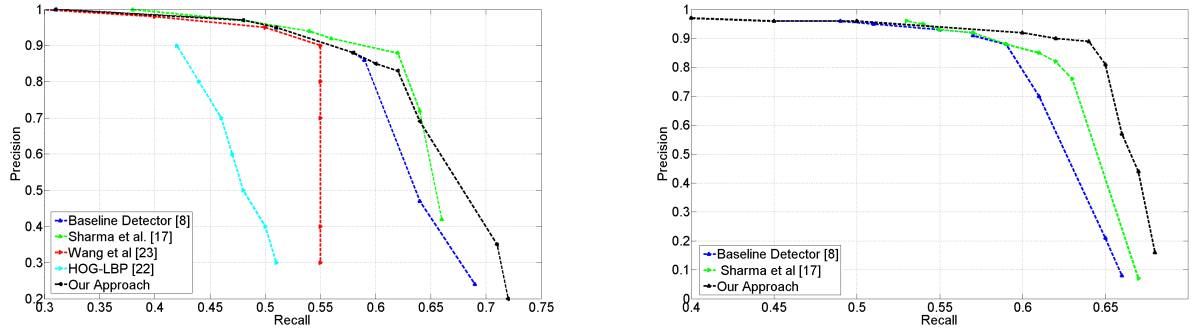
We compare the performance of our approach with the baseline classifier (poselet detector [5]), and show that by dividing the positive samples into different categories, we get better performance as compared to the case where we do not divide the positive samples into different categories. Precision-Recall curves for both ME1 and ME2 sequences are shown in Figure 6.

For both ME1 and ME2 sequences, our method gives better performance than poselet detector. The best performance is obtained when we divide the positive samples into different categories. From Table 2, we can see that our method improves the precision for ME1 by 5% at recall 0.96, when we use sample categorization module, whereas without sample categorization, improvement in precision is 2%. For ME2 sequence, at recall 0.6, we improve the precision for poselet detector by 12% with sample categorization, whereas without sample categorization improvement is 7%.

Some of the detection results are shown in Figure 7. Our approach can be utilized as an efficient pre-processing step to improve the detection results, before applying tracking-by-detection method on baseline classifiers. Also trained multi-category adaptive classifier can be used as pose identification such as standing, bending, digging etc.

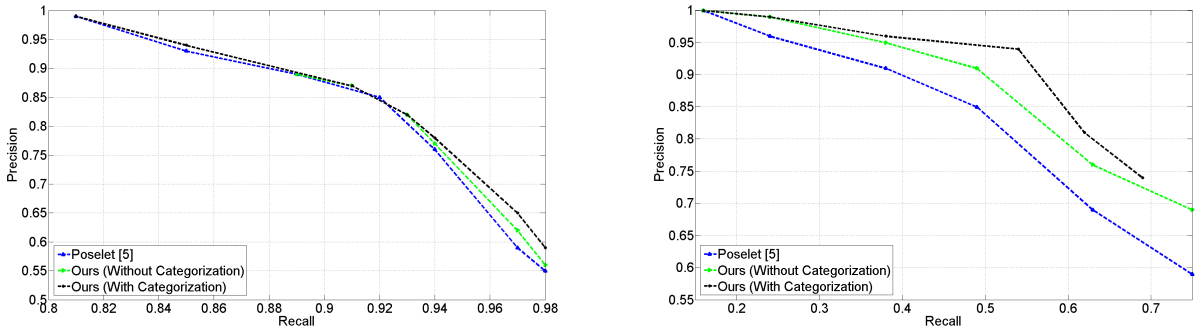
6. Conclusion

We proposed a novel detector adaptation approach, which efficiently adapts a baseline classifier for a test video. Online samples are collected in an unsupervised manner and random fern classifier is trained as the adaptive classifier. Our approach is generalized, hence can easily be applied with various baseline classifiers. Our approach can also handle pose variations of the target object. Experiments demonstrate that our method is computationally efficient as compared to the other state of the art approaches. We show the detection performance on two challenging datasets for the problem of human detection. In future, we plan to apply our adaptation method on other categories of objects and other baseline classifiers.



(a) CAVIAR1 (b) CAVIAR2

Figure 5. Recall-Precision curves for Detection Results on CAVIAR Dataset



(a) ME1 (b) ME2

Figure 6. Recall-Precision curves for Detection Results on Mind's Eye Dataset.

Table 2. Best precision improvement performance on Mind's Eye Dataset. For ME1, precision values are shown at recall 0.97, whereas for ME2 recall is 0.6. Ours-1: Without sample categorization, Ours-2: With Sample Categorization

Sequence	**Poselet** [5]	**Ours-1**	**Ours-2**
ME1	0.65	0.67	**0.7**
ME2	0.72	0.79	**0.84**

7. Acknowledgment

This research was sponsored, in part, by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-10-2-0063. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

References

[1] http://www.visint.org/.

[2] http://groups.inf.ed.ac.uk/vision/CAVIAR/CAVIARDATA1/.

[3] http://www.cs.berkeley.edu/~lbourdev/poselets/.

[4] M. Andriluka, S. Roth, and B. Schiele. People-tracking-by-detection and people-detection-by-tracking. In *CVPR*, 2008.

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

[6] H. Grabner, C. Leistner, and H. Bischof. Semi-supervised on-line boosting for robust tracking. In *ECCV*, 2008.

[7] C. Huang, H. Ai, T. Yamashita, S. Lao, and M. Kawade. Incremental learning of boosted face detector. In *ICCV*, 2007.

[8] C. Huang and R. Nevatia. High performance object detection by collaborative learning of joint ranking of granules features. In *CVPR*, 2010.

[9] C. Huang, B. Wu, and R. Nevatia. Robust object tracking by hierarchical association of detection responses. In *ECCV*, 2008.

[10] O. Javed, S. Ali, and M. Shah. Online detection and classification of moving objects using progressively improving detectors. In *CVPR*, 2005.

[11] A. Kembhavi, B. Siddiquie, R. Miezianko, S. McCloskey, and L. Davis. Incremental multiple kernel learning for object detection.

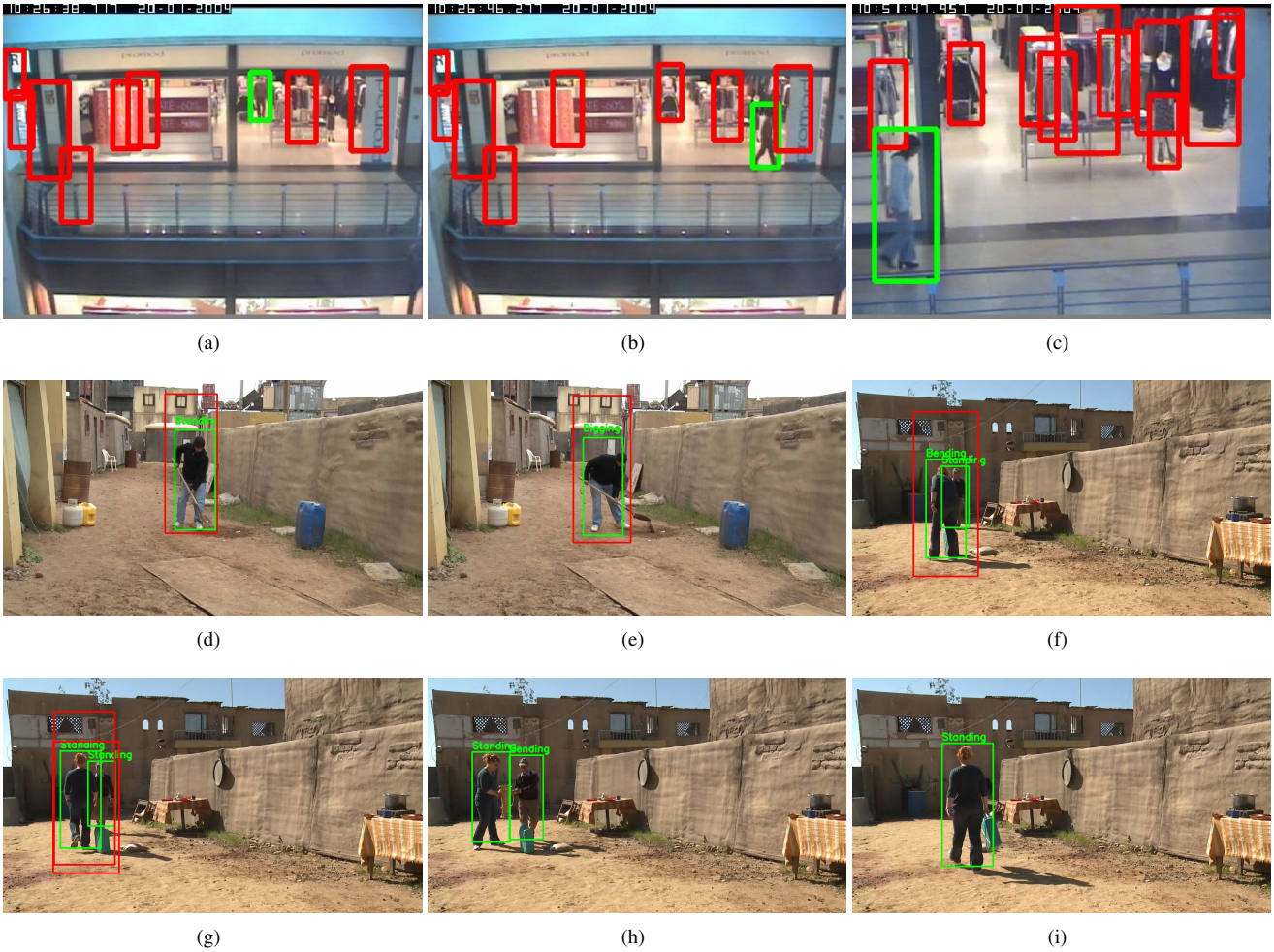


Figure 7. Examples of some of the detection results when applied baseline detector at low threshold (best viewed in color). Red: Detection result classified as false alarm by our method. Green: Detection result classified as correct detection by our method. Identified category name is also specified for Mind's Eye dataset (second row).

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

[13] V. Nair and J. J. Clark. An unsupervised, online learning framework for moving object detection. In *CVPR*, 2004.

[14] M. Ozuysal, M. Calonder, V. Lepetit, and P. Fua. Fast keypoint recognition using random ferns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2010.

[15] P. M. Roth, S. Sternig, H. Grabner, and H. Bischof. Classifier grids for robust adaptive object detection. In *CVPR*, 2009.

[16] R. E. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. *Mach. Learn.*, 37:297–336, December 1999.

[17] P. Sharma, C. Huang, and R. Nevatia. Unsupervised incremental learning for improved object detection in a video. In *CVPR*, 2012.

[18] G. Shu, A. Dehghan, O. Oreifej, E. Hand, and M. Shah. Part-based multiple-person tracking with partial occlusion handling. In *CVPR*, 2012.

[19] S. Stalder, H. Grabner, and L. Van Gool. Cascaded confidence filtering for improved tracking-by-detection. In *ECCV*, Berlin, Heidelberg, 2010. Springer-Verlag.

[20] S. Theodoridis and K. Koutroumbas. Pattern recognition, elsevier science, 2003.

[21] M. Wang, W. Li, and X. Wang. Transferring a generic pedestrian detector towards specific scenes. In *CVPR*, 2012.

[22] X. Wang, T. X. Han, and S. Yan. An hog-lbp human detector with partial occlusion handling. In *ICCV*, 2009.

[23] X. Wang, G. Hua, and T. X. Han. Detection by detections: Non-parametric detector adaptation for a video. In *CVPR*, 2012.

[24] B. Wu and R. Nevatia. Improving part based object detection by unsupervised, online boosting. In *CVPR*, 2007.

[25] B. Yang and R. Nevatia. An online learned crf model for multi-target tracking. In *CVPR*, 2012.

[26] B. Zeisl, C. Leistner, A. Saffari, and H. Bischof. On-line semi-supervised multiple-instance boosting. In *CVPR*, 2010.