Detection by Detections: Non-parametric Detector Adaptation for a Video

Xiaoyu Wang* Gang Hua† Tony X. Han*
*Electrical and Computer Engineering Dept. †Computer Science Dept.
University of Missouri Stevens Institute of Technology
Columbia, MO 65211 Hoboken, NJ 07030
xw9x9@mail.missouri.edu ghua@stevens.edu hantx@missouri.edu

Abstract

We propose an approach to improving the detection results of a generic offline trained detector on a specific video. Our method does not leverage visual tracking as most detection by tracking methods do. Instead, the proposed detection by detections approach can serve as a more confident initialization for detection by tracking methods. Different from other supervised detector adaptation methods, we constrain the task to videos and no supervised labels for the target video are required for the adaptation; we intend to fill the gap between detection by tracking and pure detection by frames. As a non-parametric detector adaptation method, confident detections are collected to re-rank and to group other detections. We focus on methods with high precision detection results since it is necessitated in real application. Extensive experiments with two state-of-the-art detectors demonstrate the efficacy of our approach.

1. Introduction

We have observed significant advancement of the research on object detection in the past decade [3, 6, 1, 11, 15, 13, 31, 17, 27, 4, 22, 30]. Most previous methods perform detection from static images. To detect objects in videos, a large amount of work takes a tracking by detection and/or detection by tracking approach [40, 1, 37, 9, 33, 7, 36, 2, 38, 8], where visual tracking is performed to further validate the detection hypothesis, or detection results are leveraged for robust tracking.

In tracking-by-detection or detection-by-tracking, most often the detection results will serve as a cue to build the matching method for tracking. The detection component may be further improved with the result of a tracker through online learning. But the improvement may be heavily downgraded if we directly use the noisy detections to initiate the tracker.

Since almost all state-of-the-art detectors are trained from a large set of labeled examples, the performance of a detector is inevitably degraded when the detector is applied to a video taken under a visual condition which is very different from those of the training examples. Therefore, how to adapt a learned generic detector to the specific visual condition of a video becomes a very important problem to be explored.

Many researchers have devoted their efforts in developing online learning/adaptation algorithms for detectors [39, 23, 24, 25, 35, 10, 12, 18, 26]. For most of online adaptation methods, an initial detector is firstly trained from a small set of labeled examples. The initial detector is then enhanced with newly available labeled examples, which can be obtained from either background modeling/subtraction [25, 10, 18, 26], or semi-supervised learning, such as self-training [23, 35], or co-training [24, 12]

However, background modeling/subtraction may not always be feasible especially when we are dealing with unconstrained web videos. On the other hand, semi-supervised learning is prone to introducing label noise, which deteriorates the adaptation of a parametric model for detection. Taking these facts into consideration, we propose a non-parametric detector adaptation algorithm, which can adapt an offline-trained frame-based object detector to the visual characteristic of a specific video clip.

Unlike the tracking-by-detection and detection-by-tracking approach, where the final trajectory is fulfilled by tracking, our adapted detector is still performing detection instead of tracking. It achieves higher precision than the original detector on each target video. Therefore, it can serve as a more reliable initialization for any tracking-by-detection and detection-by-tracking methods.

To achieve this, the original detector with a low detection threshold setting is firstly applied to the target video. All detected visual examples are collected to form the candidate detection pools with both *positive* and *negative* examples pertaining to the target video. Dense features are extracted from these visual examples to form a vocabulary tree, from which a set of detected examples with high detection scores are sparsely encoded to form the transfer classifier.

All detections from the original detector will be either

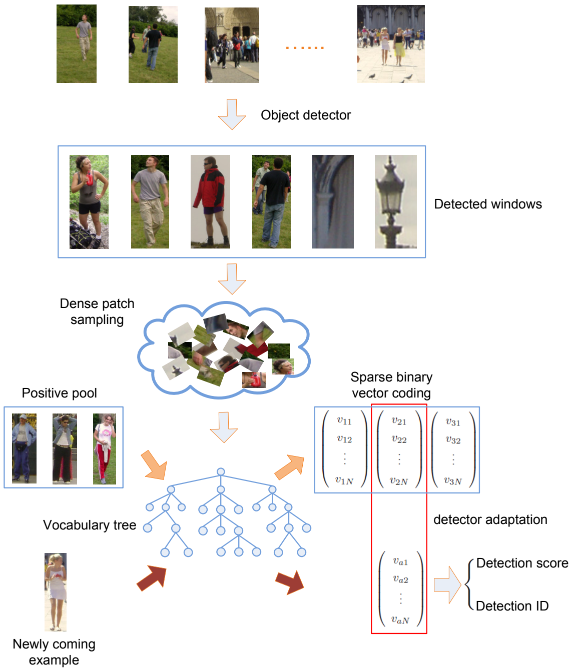


Figure 1: The flow chart of our non-parametric transfer learning method for video object detection.

validated or suppressed by the new classifier. Validated detections which receive the highest support from the same positive example can be further clustered into a group. An interesting fact is that our vocabulary tree based classifier can also serve as a recognition system to recognize or cluster the identity of the detected objects. We sketch the flow of our detection by detections approach in Figure 1.

The efficacy of the proposed method resides in our observation that a detectable video object should be confidently detected at least once among all frames using the original frame-based detector. Unlike the detection-by-tracking approach, we do not leverage dense temporal smoothing. In this sense, it serves as an intermediate approach, which bridges the gap between the detection-by-tracking and the pure frame-based detection. Hence our main contribution is a simple and effective non-parametric *detection by detections* method to extend any static-image-based object detector to video object detection, which needs neither the original training data, nor manually labeled online examples.

The reminder of the paper is organized as follows. Related work is discussed in Section 2. Section 3 introduces the details of the non-parametric detector adaptation technique. Experimental results are presented and discussed in Section 4. We finally conclude in Section 5.

2. Related work

There is a large set of successful and enlightening work in the area of object detection. Mainly popularized by the seminal Viola-Jones detector [28], Haar wave-let [20] has been widely used as the feature for general object detection. Lowe's SIFT feature [14] made a breakthrough for object recognition [19]. Viola *et al*. [29] leverage 3D Harr features in spatiotemporal domain for pedestrian detection from videos. The histogram of oriented gradient (HOG) [3] has been demonstrated to be very effective for object detection in PASCAL VOC challenge [5].

Sliding window based holistic classification methods have achieved many state-of-the-art performances [3, 6, 11, 27, 15, 17]. Various part-based models have been proposed to deal with different visual complications such as partial occlusions [34, 31], and pose variations [6]. Lin *et al*. [13] employs a multiple instance learning method to achieve part-based object detection which is robust to feature misalignment.

Many previous works resort to vocabulary tree for scalable object recognition [16, 32, 19]. In this paper, we leverage a vocabulary tree to encode a visual example as a binary vector for non-parametric detector adaptation. The proposed non-parametric detector with adaptation can achieve better detection accuracy in each target video when compared with the original detector.

Our *detection by detections* algorithm can be formed as a nonparametric transfer learning algorithm. According to a recent survey [21], there is few work dealing with unsupervised parameter-transfer learning and relation knowledge transfer learning, mainly due to the complexity and the sensitivity in parameter adaptation/settings. In contrast, the proposed nonparametric transfer learning method is simple and effective as validated in the experiments section.

3. Non-parametric detector adaptation

As illustrated in Figure 1, we set a generic human detector to work on high recall and consequently low precision point. All detections from each of the individual video frames are gathered to build a vocabulary tree using hierarchical k-means [19]. The vocabulary tree is then applied to efficiently encode the set of most confident visual detections as a set of binary vectors. A classifier is built based upon these encoded positive examples. For any candidate detections, we measure its similarity to the positive pool to determine the detection confidence. We proceed to present the key technical components of our detector adaptation algorithm, i.e., the *vocabulary tree encoding scheme* (Section 3.1), the *matching algorithm* on the resulting binary vector codes (Section 3.2), the *transfer classification algorithm* (Section 3.3), and the *nearest neighbor identity grouping algorithm* (Section 3.4).

3.1. Binary codes with a vocabulary tree

Given the set of detections $\mathcal{D} = \{\mathbf{d}_i | i = 1, 2, \ldots, N\}$ extracted by a generic static frame based detector, we ap-

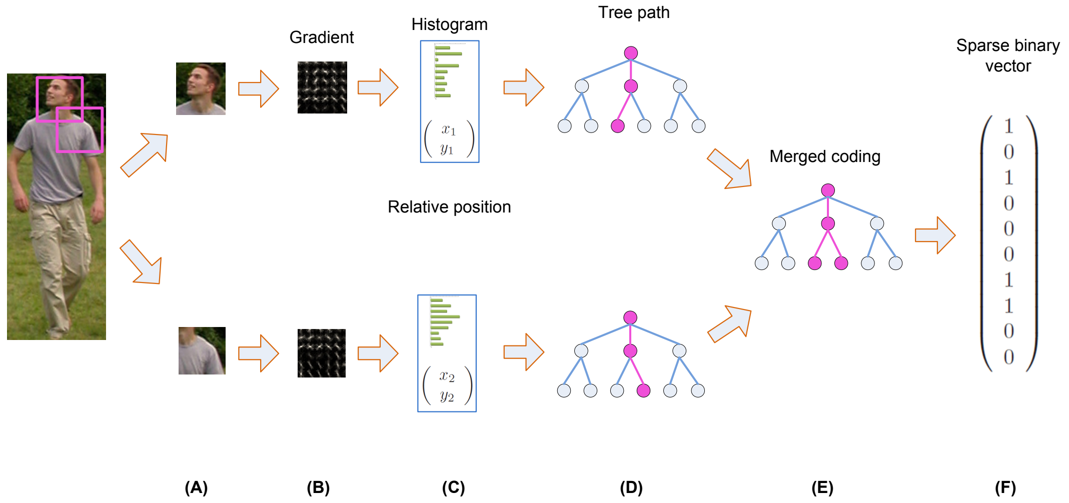


Figure 2: Feature extraction and tree coding illustration:(A) Extract densely overlapped patches;(B)Compute gradients, orientations and local binary patterns;(C) Build orientation histogram, combined with horizontal and vertical distance to the anchor position; (D) Pass the patch feature through the tree to get a tree path; (E) Merge the tree path (F) Get the binary vector.

ply the hierarchical k-means algorithm presented in [19] due to its scalability and demonstrated performance. Often the case, there is a confidence score associated with each $\mathbf{d}_i \in \mathcal{D}$ from the detector, which we denote as $o(\mathbf{d}_i)$ or in short o_i.

For each visual detection $\mathbf{d}_i \in \mathcal{D}$, we densely sample a set of 16×16 image patches from the detection window, denote as $\mathbf{P}_i = \{p_{ij} | j = 1, 2, \ldots, n\}$. For each image patch, we extract HOG feature which is widely used to capture contour information of an object. The HOG feature extraction will produce the feature vector $\mathbf{f}_{ij} \in \mathbb{R}^m$.

Taking a scheme similar to [32], we augment the visual feature vector with the position (x_{ij}, y_{ij}) of the patch relative to its anchor position inside the detection window. This results in the final augmented visual descriptor $\mathbf{g}_{ij} = [\mathbf{f}_{ij}, x_{ij}, y_{ij}]$, a $p = m + 2$ dimensional vector. We denote the set of all features extracted from $\mathbf{d}_i$ as $\mathbf{G}_i = \{\mathbf{g}_{ij} | j = 1, 2, \ldots, n\}$. The set of all augmented feature vector, $\mathcal{G} = \{\mathbf{G}_i | i = 1, 2, \ldots, N\}$ is put into a hierarchical k-means algorithm to induce a vocabulary tree T of height l, with each of the $r = \frac{k^{l+1}-1}{k-1}$ nodes of the tree labeled in level-order.

The vocabulary tree T naturally defines a mapping $\mathbf{T}$ from $\mathbf{g}_{ij} \in \mathbb{R}^p$ to $\mathbf{c}_{ij} \in \mathbb{B}^r$, where $\mathbb{B}^r$ is an r dimensional binary vector space. More specifically, if any $\mathbf{g}$ traverses node q of the hierarchical k-means tree, then the q^{th} bit of $\mathbf{c}_{ij}$ is set to 1, otherwise it will be set to 0. We further define the group mapping $\mathbf{c}(\mathbf{G}_i)$ from $\mathcal{G}$ to $\mathbf{c} \in \mathbb{B}^r$, i.e.,

$$\mathbf{c}(\mathbf{G}_i) = \cup_{j=1}^{n} \mathbf{T}(\mathbf{g}_{ij}) = \cup_{j=1}^{n} \mathbf{c}_{ij} \qquad (1)$$

where $\cup$ indicates the bitwise OR operation.

To encode a new visual example, *i.e.*, a new candidate detection window $\mathbf{x}$, we also extract a set of augmented visual descriptors of the densely sampled image patches from the window. This set of visual descriptors are then encoded with Equation 1. This encoding process is illustrated in Figure 2; each visual example is encoded by an r dimensional binary vector in the end. For convenience, we tolerate a slight abuse of notation: we also use $\mathbf{c}(\mathbf{x})$ to denote the binary vector encoding of a visual example $\mathbf{x}$, which shall follow all the patch feature extraction and vocabulary tree encoding protocol outlined above. The meaning of the notation is clear within the context.

The design of our binary codes is largely motivated by the experimental evaluation of Moosmann et al. [16]. The reason that binary encoding achieves better recognition results may be due to its robustness against noise by eliminating the perturbation caused by the frequency counting.

3.2. Similarity measure of the binary codes

Given the binary vector codes $\mathbf{c}_i = \mathbf{c}(\mathbf{x}_i)$ and $\mathbf{c}_j = \mathbf{c}(\mathbf{x}_j)$ of two visual examples $\mathbf{x}_i$ and $\mathbf{x}_j$, we adopt the following similarity measure to compute the similarity between the two visual examples,

$$s(\mathbf{c}_i, \mathbf{c}_j) = exp\left\{-\frac{\|\mathbf{c}_i - \mathbf{c}_j\|^2}{\|\mathbf{c}_i\| \cdot \|\mathbf{c}_j\|}\right\}, \qquad (2)$$

where $\|\mathbf{c}\|$ indicates the number of non-zero bits in the binary vector $\mathbf{c}$. Because $\mathbf{c}_i$ and $\mathbf{c}_j$ are binary vectors, $\|\mathbf{c}_i - \mathbf{c}_j\|^2$ is actually the hamming distance between them. We further normalize the hamming distance using the L1 norm of the binary code. The normalized hamming distance is then transformed to a similarity measure by taking the negative exponential. This normalization process is important because the binary vectors of large L1 norm will not be over-penalized in the matching process. Certainly other type of normalization such as L2 can also be used, and we empirically evaluate the effects of different normalization schemes in our experiments.

3.3. Transfer classification

To build our final transferred adaptive detector, we firstly collect a set of positive visual detections of high confidences which are higher than a threshold t,

$$\mathcal{E} = \{\mathbf{d}_k | \mathbf{d}_k \in \mathcal{D}, o(\mathbf{d}_k) > t\}, \tag{3}$$

where $\mathcal{E} \subset \mathcal{D}$, composes our positive example pool. Each $\mathbf{d}_k \in \mathcal{E}$ is further encoded by the induced vocabulary tree T. This forms a set of binary codes to encode the positive examples, i.e.,

$$\mathcal{C} = \{\mathbf{c}(\mathbf{d}_k) | \mathbf{d}_k \in \mathcal{E}\} = \{\mathbf{c}_k | \mathbf{d}_k \in \mathcal{E}\}\}. \tag{4}$$

We will use interchangeably $\mathcal{E}$ and $\mathcal{C}$ to represent the positive pool when there is no confusion. Given any new candidate detection or visual instance $\mathbf{x}$, we decide whether it is a real detection based on the following similarity scoring, i.e.,

$$\mathbf{h}(\mathbf{x}) = \frac{1}{\|\mathcal{C}\|} \sum_{\mathbf{c}_k \in \mathcal{C}} s(\mathbf{c}(\mathbf{x}), \mathbf{c}_k). \tag{5}$$

According to Equation 2, the similarity is achieved by averaging the similarity between the detection and each of the example in the positive pool. The averaging operation is very helpful in dealing with noisy false positives in the positive pool. We make the final classification decision by assigning a confidence threshold h_t, i.e.,

$$a(\mathbf{x}) = \begin{cases} 1 & \mathbf{h}(\mathbf{x}) \geq h_t \\ 0 & \mathbf{h}(\mathbf{x}) < h_t \end{cases}. \tag{6}$$

That is, if the average distance of the candidate $\mathbf{x}$ to examples in the positive pool $\mathcal{E}$ is sufficiently small, we declare it to be a detection, otherwise, we reject it. For this adaptive detector to work in each target video, we have made an assumption that *a video object would be detected by the original generic detector in at least some of the video frames*. In our experiments, we manifest that this assumption is indeed held in a variety of different videos if a state-of-the-art detector is used.

3.4. Identity grouping of detections

The proposed encoding scheme enables us not only to perform detector adaptation, but also to design an effective grouping algorithm to group the video object detections based on their identities. This functionality is important to a wide range of applications including video browsing, video annotation, and people search from videos.

Our grouping algorithm clusters all detections by measuring their similarities to a set of k representative examples. To select the k representative examples, we firstly run k-means in our positive example pool $\mathcal{C}$ to cluster it into k groups. For each cluster, we select the detection with the highest detection confidence as its representative example. We denote this set of representative examples to be $\mathcal{R} = \{\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_k\}$.

The group ID $g(\mathbf{u})$ of any other example $\mathbf{u}$, either a new one or an one from the positive pools, is assigned to be

$$g(\mathbf{u}) = \arg\max_{i \in \mathcal{R}} (s(\mathbf{u}, \mathbf{r}_i)). \tag{7}$$

This is a simple nearest neighbor clustering algorithm. It shall be noted that the number of groups or the number of representative examples is not necessarily the number of the objects appeared in the target video. It is possible that an object have multiple representative examples, corresponding to different views and poses across the videos. This implies that an object may correspond to multiple groups. This is acceptable because we care more about the identity clustering accuracy within each group. Certainly, some post merging may be performed with appropriate human supervision and interaction.

4. Experiments

We use five video sequences from CAVIAR [1], i.e., *OneLeaveShop1front.mpg* (ols1), *OneLeaveShop2front.mpg* (ols2), *OneShopOneWait1front.mpg* (ols2), *OneLeaveShopReenter2font.mpg* (olsr2), *OneStopEnter2front.mpg* (ose2) for evaluation. We adopt the evaluation criterion of PASCAL VOC challenge. A detection is treated as a true positive if it has more than 0.5 overlap with the ground truth. We compute the detection average precision (AP) to compare the performances. All detection windows are normalized to 48 by 96.

Inside each detection window, we use 16 by 16 pixels blocks and the shift stride is set to be 4 pixels both horizontally and vertically. Hence, for each window, 189 blocks are extracted. We strictly follow the procedure of [3] to build the HOG feature, in which each block contains four block normalized cells. A 59 dimension LBP histogram is also extracted from each block. The horizontal and vertical distances between an anchor position and each image patch are encoded in the patch feature representation as well. The distance is multiplied by a coefficient of 0.01 for range normalization. We use a branch factor $k = 10$ to train the vocabulary tree.

[1] The data is from the EC Funded CAVIAR project/IST 2001 37540, found at URL: http://homepages.inf.ed.ac.uk/rbf/CAVIAR/

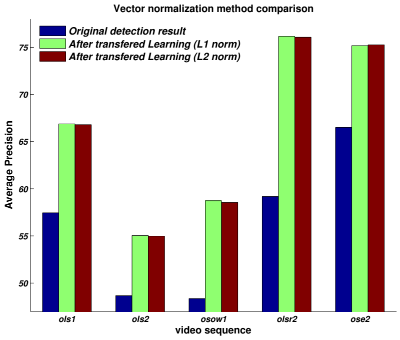


Figure 3: Performance comparison between different normalization methods

4.1. Two human detectors

We use the deformable-model-based human detector [6] and the HOG-LBP human detector [31] as the original frame-based detectors. Both detectors are trained on the INRIA pedestrian data-set. For most of the five videos, the recalls of the deformable model detector [6] are smaller than 0.1 due to its disadvantage in detecting small objects. To circumvent this disadvantage and better evaluate the detector adaptation performance, we upsampled all the video frames by a factor of 2 before the detection. Both detectors are applied with the proposed adaptation algorithm for a comprehensive evaluation.

4.2. The performance of detection by detections

There are several settings and parameters to be explored in our detector adaptation algorithm, including the depth of the vocabulary trees, the threshold t to select the positive pools, and the various normalization schemes for the similarity measure of the binary vectors. We first apply our adaptation algorithm on the more efficient HOG-LBP detector to explore different parameter settings. We then fix the selected parameters to evaluate the adaptation performance for both detectors.

4.2.1 The normalization scheme evaluation

We compare two normalization methods for the similarity measure defined in Equation 2, including the adopted L1 norm, and the L2 norm. We present the AP measure in all 5 videos using these two normalization schemes along with the AP score of the original HOG-LBP detector, as shown in Figure 3. The two normalization methods have similar performance over all five videos.

4.2.2 The evaluation on threshold for positive pool

Since the threshold t determines the positive example pool $\mathcal{E}$, we had expected that it would impact the performance of the adapted detector. From our experiments, it turns out that the proposed approach is not very sensitive to the threshold t. We believe it benefited from the similarity averaging operation in our transfer classifier. Due to the space limit, we only present how the performance varies according to the change of the threshold t for the video sequence “ols2”. Figure 4f shows the result. We can see that the curve is very flat from $t = 0$ to $t = 1$. The curves of the other four videos are similar. The insensitivity of the threshold t makes our algorithm more generalizable which is very important in real application.

Table 1: Performance comparison of different tree depth(Average Precision %). “Org” shows the original detection results obtained from the HOG-LBP detector

Tree Depth	Ols1	Ols2	Osow1	Olsr2	Ose2	Avg
Org	57.45	48.67	48.36	59.17	66.49	56.03
2	65.60	51.36	51.03	72.58	70.9	62.29
3	**67.02**	54.51	58.50	75.54	74.30	65.97
4	66.72	54.96	58.79	76.07	74.93	66.29
5	66.88	**55.04**	58.73	**76.15**	**75.18**	**66.40**
6	66.64	54.96	**58.83**	75.62	75.16	66.24

Table 2: Adaptation based on deformable model [6]. Original detection results VS adaptation results. (Average Precision %)

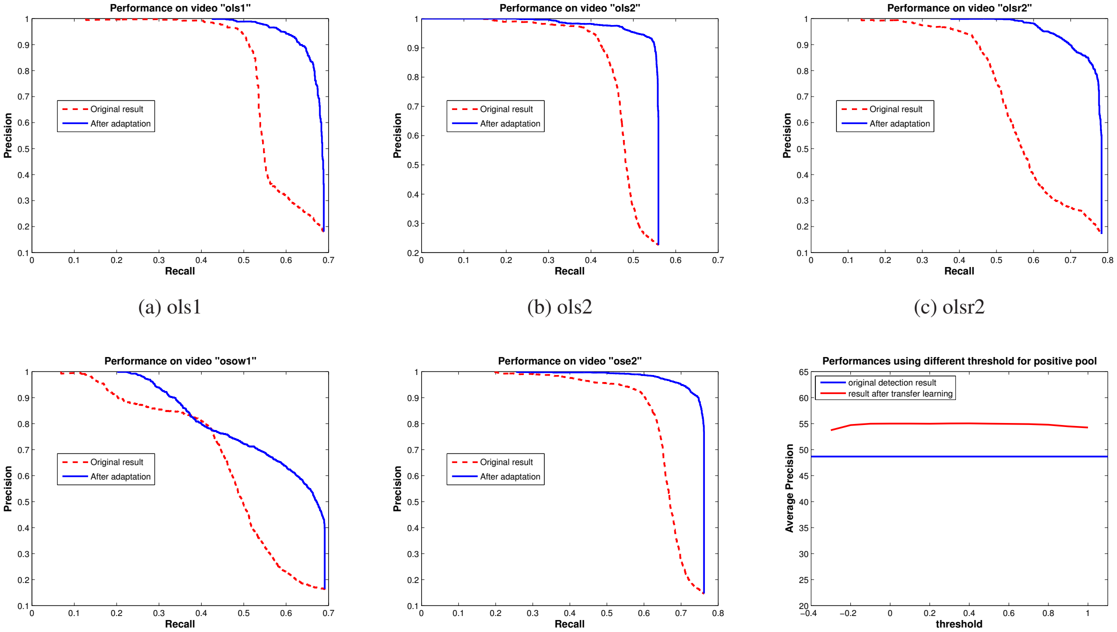
Method	Ols1	Ols2	Osow1	Olsr2	Ose2
Original	0.537	0.429	0.467	0.559	0.759
Adaptation	**0.553**	**0.461**	**0.501**	**0.591**	**0.790**

4.2.3 The tree depth exploration

Another parameter to explore is the depth of the hierarchical k-means tree. Table 1 shows how the tree depth d affects the performance of our transferred detector. It is easy to observe that the detection results are similar when the tree depth changes from 3 to 6, which contrasts the big leap in accuracy when we change the depth from 2 to 3. Although we can go further for the tree depth exploration, deep tree is less preferable in real-world applications due to the memory constrains and computing efficiency. Balancing the performance and the efficiency, we set the depth of our vocabulary tree as 5.

4.2.4 Performance of video object detection

Based on the above parameter exploration, we use L1 normalization, set $t = 0$, and select 5 as the tree depth. From Figure 4a to Figure 4e, we report performance of the proposed non-parametric transfer learning approach for video



(a) ols1

(b) ols2

(c) olsr2

(d) osow1

(e) ose2

(f) Clip threshold Analysis

Figure 4: Adaptation performance: Figure (a)-(e) show the adaptation performance based on the HOG-LBP detector; Figure (f) shows how the performance varies against the clip threshold when composing the positive example pool.

objects detection on each of the 5 testing videos based on the HOG-LBP detector. In all these 5 figures, "Original result" indicates the detection performance from the original HOG-LBP detector; "After adaptation" corresponds to the transfer learning with HOG feature.

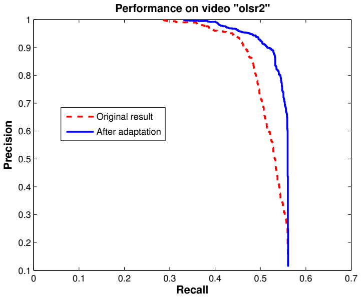


Figure 5: Adaptation using deformable model [6] on video "olsr2".

The ROC curves show that our approach achieves significantly better results than the original detector. For the HOG-LBP detector, we improve the AP by 10.37% over all 5 videos. The steep drops of precision close to the maximum recall rate indicate that almost all of the positive examples have been re-scored with a high confidence. It should be mentioned that we only use the HOG feature for adaptation in order to make sure the big improvement comes from the adapation framework, instead of augmented features.

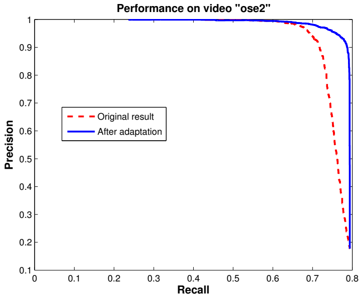


Figure 6: Adaptation using deformable model [6] on video "ose2".

Figure 5 and 6 present the adaptation performance based on [6] (the results on the other three videos showed similar plots). The original detection result already had a sharp drop in precision near the maximum recall as shown in the figure, which means it is very hard to further improve the performance. Even in this situation, our adaptation algorithm is still able to push the envelope, as shown in Figure 5 and 6. Due to the page limit. We show the complete adaptation results in Table 2.

To understand the limitation of the proposed approach, we show sample missing detections which remained to be difficult for our transfer learning framework to detect in Figure 7. Almost all these missing detections are small and blurry. They simply do not present sufficient visual details

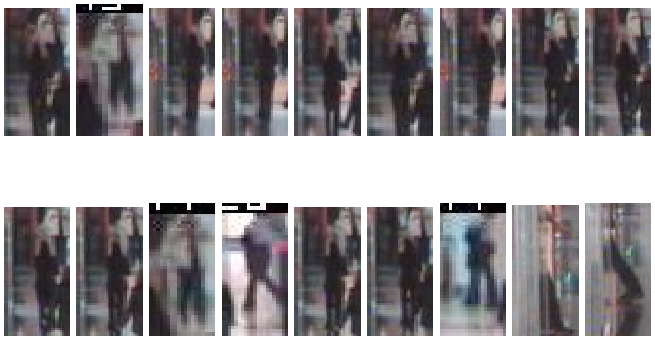


Figure 7: Difficult to be detected objects in video "ose2"

to be detected, which are quite understandable.

4.3. Identity grouping

In this section, we perform the experiment on identity grouping with the detection results from the HOG-LBP detector. We compared our identity grouping method with K-means grouping. For K-means grouping, we concatenate features from each patch to form a long vector. The grouping performance is evaluated according to the portion of the examples which has been correctly grouped. An example is considered to be correctly clustered if its identity is the majority of the identities in the cluster. We adopt the following evaluation criterion,

$$P = \frac{\sum_{i=1}^{k} \mathbf{F}(mode(g_i))}{N}, \tag{8}$$

where $\mathbf{F}(mode(g_i))$ counts the number of examples in group i with identity $mode(g_i)$, which is the most frequently appeared identities in group i; k is the number of identities specified; N is the total number of positive examples evaluated. We set $k = 20$ for K-means clustering.

The grouping performances on both the positive pool and all detections are presented in Table 3 and Table 4, we can see that our grouping method is significantly better than K-means clustering. Our grouping method improves the precision by more than 0.2 on both sets. Nevertheless, the grouping performance on all detections is not as impressive as we expected. This is because that there are many very small objects detected, which are very difficult to be identified. Some examples of such detections are shown in Figure 8. Figure 9 shows sample grouping results obtained by our grouping method and K-means grouping method.

In this experiment, we focused on the purity of the cluster results. This consideration is due to the following fact: for real applications where the users are interactively confirming the group annotation, it is much more convenient for the users to have a clean cluster. The reason that our proposed grouping algorithm is better than the K-means grouping over the raw features is that our binary codes largely suppressed the effects of noise.

Figure 8: Small objects which are difficult to be identified.

Table 3: Identity grouping comparison on positive pool.

	Ols1	Ols2	Osow1	Olsr2	Ose2	Avg
our method	**0.825**	**0.980**	**0.870**	**0.948**	**0.854**	**0.895**
k-means	0.584	0.897	0.515	0.658	0.474	0.625

Table 4: Identity grouping comparison on all detections.

	Ols1	Ols2	Osow1	Olsr2	Ose2	Avg
our method	**0.584**	**0.460**	**0.553**	**0.699**	**0.618**	**0.583**
k-means	0.393	0.367	0.289	0.389	0.405	0.368

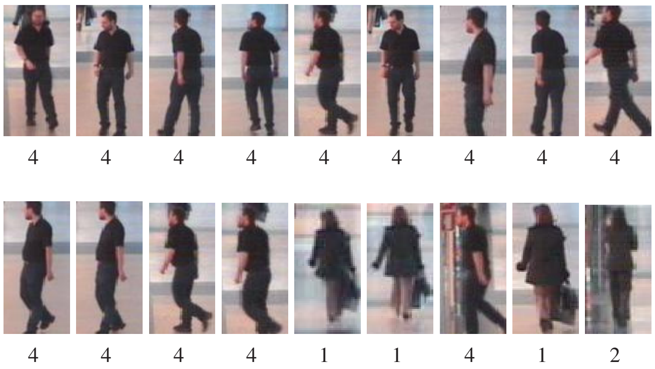


Figure 9: Sample grouping results: First row-examples grouped into the same group by our approach; Second row-examples grouped into the same group by k-menas clustering. The number below the image indicates the true ID.

5. Conclusion and future work

We proposed a simple and effective solution to improve the pure detection accuracy of off-the-shelf detectors trained from static images on target videos. The adapted detections can serve as a higher precision initialization for any other detection-by-tracking algorithms. Our nonparametric transfer learning scheme, namely *detection by detections*, needs neither the original training data nor the label information from the target video. Experiments on several challenging videos with two state-of-the-art object detectors show that our framework is insensitive to system parameters and always improves the detection accuracy. The approach is also able to group detections into identity groups. Future work will explore the transfer learning of other types of detectors and various applications which are enabled by the proposed video object detection system.

Acknowledgments

Gang Hua is partly supported by the start-up funds from Stevens Institute of Technology. Xiaoyu Wang and Tony X. Han are partially supported by the NGA NURI grant (HM1582-10-1-0013) and by the Leonard Wood Institute in cooperation with the U.S. Army Research Laboratory. The work was accomplished under Cooperative Agreement Number W911NF-07-2-0062. 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 Leonard Wood Institute, 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 heron.

References

[1] M. Andriluka, S. Roth, and B. Schiele. People-tracking-by-detection and people-detection-by-tracking. In *CVPR*, 2008. 1

[2] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. V. Gool. Robust tracking-by-detection using a detector confidence particle filter. In *ICCV*, 2009. 1

[3] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005. 1, 2, 4

[4] P. Dollár, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: A benchmark. In *CVPR*, 2009. 1

[5] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2008 (VOC2008) Results. 2

[6] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *CVPR*, 2008. 1, 2, 5, 6

[7] D. M. Gavrila and S. Munder. Multi-cue pedestrian detection and tracking from a moving vehicle. *IJCV*, 2006. 1

[8] H. Grabner, C. Leistner, and H. Bischof. Semi-supervised on-line boosting for robust tracking. In *ECCV*, 2008. 1

[9] M. Han and Y. G. Amit Sethi, Wei Hua. A detection-based multiple object tracking method. In *ICIP*, 2004. 1

[10] O. Javed, S. Ali, and M. Shah. Online detection and classification of moving objects using progressively improving detectors. In *CVPR*, 2005. 1

[11] C. H. Lampert, M. B. Blaschko, and T. Hofmann. Beyond sliding windows: Object localization by efficient subwindow search. In *CVPR*, 2008. 1, 2

[12] A. Levin, P. Viola, and Y. Freund. Unsupervised improvement of visual detectors using co-training. In *ICCV*, 2003. 1

[13] Z. Lin, G. Hua, and L. S. Davis. Multiple instance feature for robust part-based object detection. In *CVPR*, 2009. 1, 2

[14] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004. 2

[15] S. Maji, A. Berg, and J. Malik. Classification using intersection kernel support vector machines is efficient. In *CVPR*, 2008. 1, 2

[16] F. Moosmann, B. Triggs, and F. Jurie. Fast discriminative visual codebooks using randomized cluster forests. In *NIPS*, 2006. 2, 3

[17] S. Munder and D. Gavrila. An experimental study on pedestrian classification. *IEEE T-PAMI*, 2006. 1, 2

[18] V. Nair and J. J. Clark. An unsupervised online learning framework for moving object detection. *CVPR*, 2004. 1

[19] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *CVPR*, 2006. 2, 3

[20] M. Oren, C. Papageorgiou, P. Sinha, E. Osuna, and T. Poggio. Pedestrian detection using wavelet templates. In *CVPR*, 1997. 2

[21] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 2010. 2

[22] D. Park, D. Ramanan, and C. Fowlkes. Multiresolution models for object detection. In *ECCV*, 2010. 1

[23] C. Rosenberg, M. Hebert, and H. Schneiderman. Semi-supervised self-training of object detection models. In *The Seventh IEEE Workshop on ACV*, 2005. 1

[24] P. M. Roth, H. Grabner, D. Skocaj, and H. Bischof. On-line conservative learning for person detection. In *In Proc. VS-PETS*, 2005. 1

[25] P. M. Roth, S. Sternig, H. Grabner, and H. Bischof. Classifier grids for robust adaptive object detection. In *CVPR*, 2009. 1

[26] S. Stalder, H. Grabner, and L. V. Gool. Exploring context to learn scene specific object detectors. In *Proc. IEEE International Workshop on PETS*, 2009. 1

[27] O. Tuzel, F. Porikli, and P. Meer. Human detection via classification on riemannian manifolds. In *CVPR*, 2007. 1, 2

[28] P. Viola and M. Jones. Robust real-time object detection. *IJCV*, 2001. 2

[29] P. Viola, M. J. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. In *ICCV*, 2003. 2

[30] S. Walk, N. Majer, K. Schindler, and B. Schiele. New features and insights for pedestrian detection. In *CVPR*, 2010. 1

[31] X. Wang, T. X. Han, and S. Yan. An hog-lbp human detector with partial occlusion handling. In *ICCV*, 2009. 1, 2, 5

[32] J. Wright and G. Hua. Implicit elastic matching with random projections for pose-variant face recognition. In *CVPR*, 2009. 2, 3

[33] B. Wu and R. Nevatia. Detection of Multiple, Partially Occluded Humans in a Single Image by Bayesian Combination of Edgelet Part Detectors. *ICCV*, 2005. 1

[34] B. Wu and R. Nevatia. Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors. In *ICCV*, 2005. 2

[35] B. Wu and R. Nevatia. Improving part based object detection by unsupervised, online boosting. In *CVPR*, 2007. 1

[36] B. Wu, L. Zhang, V. K. Singh, and R. Nevatia. Robust object tracking based on detection with soft decision. In *IEEE Workshop on MVC*, 2008. 1

[37] M. Yang, F. Lv, W. Xu, and Y. Gong. Detection driven adaptive multi-cue integration for multiple human tracking. In *ICCV*, 2009. 1

[38] Q. Yu and G. Medioni. Integrated detection and tracking for multiple moving objects using data-driven mcmc data association. In *IEEE Workshop on MVC*, 2008. 1

[39] C. Zhang, R. Hamid, and Z. Zhang. Taylor expansion based classifier adaptation: Application to person detection. In *CVPR*, 2008. 1

[40] L. Zhang, Y. Li, and R. Nevatia. Global data association for multi-object tracking using network flows. In *CVPR*, 2008. 1