Statistical and Spatial Consensus Collection for Detector Adaptation

Enver Sangineto

DISI, University of Trento, Italy
Enver.Sangineto@unitn.it

Abstract. The increasing interest in automatic adaptation of pedestrian detectors toward specific scenarios is motivated by the drop of performance of common detectors, especially in video-surveillance low resolution images. Different works have been recently proposed for unsupervised adaptation. However, most of these works do not completely solve the *drifting* problem: initial false positive target samples used for training can lead the model to drift. We propose to transform the outlier rejection problem in a weak classifier selection approach. A large set of weak classifiers are trained with random subsets of unsupervised *target* data and their performance is measured on a labeled *source* dataset. We can then select the most accurate classifiers in order to build an ensemble of weakly dependent detectors for the target domain. The experimental results we obtained on two benchmarks show that our system outperforms other pedestrian adaptation state-of-the-art methods.

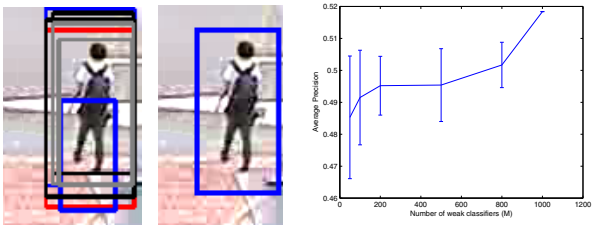
Keywords: Pedestrian Detection, Unsupervised Domain Adaptation, RANSAC.

1 Introduction

There is an increasing interest of the Computer Vision research community in transfer learning and domain adaptation techniques in the recent years as witnessed by the large number of papers on these subjects. The motivation behind this interest is due to the bottleneck of the current classifiers' training procedures which usually need hundreds or thousands of manually labeled samples. Indeed, manual annotation is an expensive and time consuming activity, and, in some domains, data acquisition can be a difficult task. On the other hand, the performance of current detectors usually drastically drops when used (tested) in scenarios different from the training data [18,10,6]. This is sometimes called *the dataset bias* problem [18]: a classifier trained with a specific image resolution, viewpoint, illumination conditions, etc., will have a poor generalization ability in a testing situation not fitting the training dataset characteristics. In order to alleviate this problem different works have been recently proposed which directly or indirectly use labeled data from a known domain (the *source dataset*) together with unsupervised or semisupervised data, acquired from the specific *target domain* (i.e., the scenario of interest).

D. Fleet et al. (Eds.): ECCV 2014, Part III, LNCS 8691, pp. 456–471, 2014.

In this paper we focus on the pedestrian detection case, in which the human body is the class of interest. However, no specific assumption on the positive class is done, except the existence of a sufficiently reliable object detector which is used in the very first stage of our algorithm for extracting candidate target samples. Collecting target samples using a generic pedestrian detector is an approach adopted in different other works, such as, for instance [32,29,34,28]. However, the question is: since the generic detector is supposed to poorly perform on the target domain, and many false positives will presumably belong to the initial candidate set, can we train an *adapted* detector using these candidates and improve the accuracy of the generic detector?



(a) (b) (c)

Fig. 1. (a)-(b) Spatial consensus collection. (a) Four out of five classifiers of our detector ensemble correctly hit the pedestrian in the figure. Bounding boxes of different colors correspond to positive answers of different classifiers. (b) The final answer of the ensemble. Note that the small blue rectangle on (a) has not been clustered together with all the other rectangles because of the scale difference and has not contributed to computing the average final rectangle on (b). (c) AP on the CUHK Square Test dataset as a function of M. Error bars show ± 1 standard deviation from the mean.

We propose to solve this issue transforming the target sample selection problem in a classifier selection problem. The target bounding boxes extracted using a baseline detector run on the target videos are randomly grouped in different overlapping subsets. In turn, each subset is used to train a different classifier. We train in this way a large vocabulary V of (weak) classifiers. Each element $C \in V$ will have an accuracy depending on the number of outliers (false positives of the baseline detector) included in its specific training set. If we could measure this accuracy, we could prune V. However, due to the lack of labeled target data (unsupervised training assumption), we cannot directly compute the accuracy of C. Nevertheless, we can use a different training set (the *source* dataset), for a rough estimate of this accuracy. There is an analogy between the approach we propose and RANSAC [13], where a statistical model is computed using a random subset of the available data and then it is verified using the rest of the data. In our case, the statistical model is C and the verification phase is performed using a different but similar dataset. V is pruned selecting the most promising classifiers which will be used at testing time as an ensemble of detectors.

The second novelty we propose concerns the way in which the ensemble detector reaches a decision on a test image. Rather than using a common voting approach [36], in which the final decision is taken *on each input subwindow*, our ensemble decisions are based on an agreement on *spatially related subwindows*. The underlying rationale is that different classifiers can disagree on a specific image window but they usually agree on windows which are close to a real instance of the class of interest (see Fig. 1(a)-1(b)).

The details of our approach are presented in Sec. 3-5, after a brief overview of the literature in Sec. 2. Experimental results using different protocols are illustrated in Sec. 6 and we finally conclude in Sec. 7.

2 Related Work

In [25] Pan and Yang present a survey on transfer learning and related areas, including domain adaptation [3], with a taxonomy of the existing approaches and the possible settings. One of the most important differences among settings is the availability at training time of (at least few) labeled target data. In this paper we assume the complete lack of labels for target data.

Adaptive-SVMs [35] are extended in [1] in an object detection scenario by introducing a sort of geometric parameter transfer (regularization is enforced among spatially close cells of a HOG-based SVM template). A similar idea is presented in [2], in which the target SVM parameters are regularized with parts borrowed from source templates. In [20] new object classes are learned borrowing *sample instances* from other similar classes, possibly after applying various geometric transformations. All these transfer learning approaches require at least a few target labeled data in order to refine the target classifier's parameters.

In [19] a shared intra-class representation, based on semantic attributes, is learned using only source samples, after that no training phase is necessary for the target class. In [27] an Information-theoretic Metric Learning is used to learn a linear transformation from the source to the target domain. A Transductive SVM together with virtual samples obtained using computer graphics techniques is used in [30] for adaptation in a pedestrian detection scenario.

In [26] a scene specific detection task is dealt with by a grid of classifiers, where each classifier needs only to learn the visual pattern of the corresponding cell on the image grid. In [22] a domain-specific classifier is selected from an initial set of pre-trained classifiers using model recommendation methods. In [23] background subtraction is used for collecting target samples. However, background subtraction can be unstable, especially in outdoor scenes.

In [32,34,29,28] a baseline pedestrian detector is run on the target videos in order to collect an initial set of positive samples. In [32] *source* samples (INRIA pedestrians) are weighted using the average distance (in HOG space) to the k nearest neighbours *target* samples and are used for training an adapted classifier. The new detector is run on the target video frames in order to acquire new target data and the process is iterated until convergence. However, if the initial target samples include a large number of false positives, the whole process can converge toward a wrong model (*drifting problem*).

In [34] a similarity score between the test image and the initial candidate pedestrian set is computed using a pre-built vocabulary tree and a threshold is used in order to reject non-pedestrian test images. However, the rejection threshold and different other parameters need to be manually set on each target scenario. In [29] the accuracy of a generic pedestrian detector is boosted using a second, adapted detector (a random fern classifier [24]) trained on target data only. Target data are acquired using the baseline detector and clustered in trajectories using position, size and appearance affinities. Different motion and spatial heuristics are used to partition the trajectories in positive and negative samples (similarly to [28]).

A problem common to all these methods is that the candidate set of pedestrians, acquired with the generic detector, is either pruned using a set of ad hoc heuristics and confidence thresholds (which need to be tuned on the target domain, an hard task if we suppose the lack of target labels) or used as it is, which likely leads to a drift of the train process. We show in Sec. 4 how a large amount of outliers can be tolerated in the initial training set using a RANSAC-like strategy and the source dataset for verifying the accuracy of the trained models.

3 Collecting Candidate Pedestrians

In the first phase of our algorithm we collect positive pedestrian samples running a generic pedestrian detector on the frames of the target videos. We use the Dalal and Triggs [7] pedestrian detector provided by the OpenCv implementation. Other more sophisticated baseline detectors can be used for this task, such as, for instance, [12], but, since the target videos used for our experiments have been acquired in far field traffic scenes (see Sec. 6), and the resulting resolution is very low, part based detectors have a very poor performance on this scenario [32]. Since false positives frequently happen in a same position along a video captured with a stationary camera, similarly to [32,33], we discard detections with a very large mutual overlapping on the same position. Other detections are discarded when they are close to the image borders because people entering and exiting from the scene are usually truncated (only partially visible).

Let $B = \{b_1, b_2, ...\}$ be the set of remaining bounding boxes obtained with the baseline detector. We estimate the mean μ and the standard deviation σ of the height values of the bounding boxes in B and we compute an upper (u) and a lower (l) bound on such values:

$$l = max(\mu - 3\sigma, min\{h(b)|b \in B\}), \tag{1}$$

where $h(b)$ is the height of the bounding box b and u is computed similarly. We prune B discarding those bounding boxes out of the range $[l, u]$, obtaining $B' = \{b|b \in B, l \leq h(b) \leq u\}$. Then we rank B' comparing its elements with our source dataset S (we use the INRIA dataset [7]). Specifically, let $S_P = \{p_1, p_2, ...\}$ be the set of the positive sample bounding boxes of S (recall that S is labeled). Moreover, let $f(b)$ be the feature vector of b. We use the common HOG features [7]. A dissimilarity score s for each $b \in B'$ can be computed by means of:

$$s(b) = \sum_{p \in S_P} ||f(b) - f(p)||_2^2. \quad (2)$$

Eq. (2) is conceptually similar to the function used in [17] for computing the similarity between source support vectors and target data. We use it to rank B' and we obtain $R = \{b_{i_1}, ..., b_{i_m}\}$ such that: $i_j < i_l \Rightarrow s(b_{i_j}) \leq s(b_{i_l})$ and $m = |B'|$. Before extracting HOG features, all the elements in B' and S_P are normalized to a standard bounding box whose height is l and the width is $l/2$. This operation is important because the elements in S_P (the INRIA pedestrian images) have a high resolution, while the elements in B', acquired from a low resolution video, have much less gradient information.

Instead of using the whole S_P in Eq. (2) we could restrict the sum to only the k nearest neighbours of $f(b)$ in S_P (e.g., similarly to the source sample weighting process in [32]). However, without using sophisticated data structures, this is computationally equivalent to our procedure and would require the estimation of the parameter k (sufficient number of neighbours). In our experiments, the results obtained with Eq. (2) and a k nearest neighbours approach with various values of k were basically equivalent, hence we decided in favour of the simplest solution.

We discard the second half of R based on the assumption that the baseline detector poorly performs on the target domain and, hence, most of the elements in B (R) are false positives. A finer solution is to discard a portion of R depending on a measure of the dissimilarity between the distributions generating B' and S_P, such as the Maximum Mean Discrepancy. However, in the current implementation we adopted this simple truncation because in our experiments we found it sufficient to achieve good results across different videos and fairly stable (e.g., $\pm 10\%$ in the truncation ratio gives basically the same overall system's accuracy). Let $T = \{b_1, ..., b_n\}$ correspond to the first half of R ($n = m/2$). In the following we use T as our target positive sample set.

The values l and u are used also at testing time to limit the number of analysed subwindows (Sec. 5) and at training time (Sec. 4) to specialize the classifiers to a specific scale range. This is reasonable since our goal is the construction of a detector for a specific scenario and viewpoint, in which the pedestrian scales are supposed to be constant over time. In [32,33] Wang et al. used our same target videos and assumed a similar mono-modal Gaussian distribution over the pedestrian sizes but they used mean shift [5] as to estimate the main mode of the distribution and the range $[l, u]$. However, the bandwidth of the mean shift kernel needs to be manually set thus we preferred a simpler but completely automatic procedure.

4 Transforming the Sample Selection Problem into a Classifier Selection Problem

The ranking and pruning operations described in Sec. 3 help in eliminating a lot of false positives. However, they are not sufficient to guarantee the lack of

outliers in T, whose number depends on the accuracy of the baseline detector and the difficulty of the target domain. Using visual inspection, in our experiments we empirically found an average of about $40 - 50\%$ of outliers in T (errors of the baseline detector). Since we want to use the elements in T for training our classifier, we would ideally need an oracle able to select a subset $T_G \subseteq T$ of "good" positive samples to use for training. This idea can be extended since we not only want that the elements in T_G are correct pedestrians images, but we also want that they are the most informative (or discriminative) for our learning task: for instance, we would like to avoid including images of the same pedestrian in the same pose. This problem can be formulated as follows:

$$T_G = \arg\min_{T_i \subseteq T} E(C_{T_i}, \mathcal{D}^t), \tag{3}$$

where C_{T_i} is a classifier trained using the set of positive samples T_i and a given random set of negatives N_i (see Sec. 4.1), $\mathcal{D}^t$ is the target domain and $E()$ is the generalization error. Since we do not have labeled samples extracted from $\mathcal{D}^t$, we use samples of the source domain $\mathcal{D}^s$, specifically, $S = S_P \cup S_N$, where S_N is the set of negatives in the INRIA dataset. Thus, Eq. (3) is approximated by:

$$T_G = \arg\min_{T_i \subseteq T} L(C_{T_i}, S), \tag{4}$$

where $L()$ is a suitable loss function for computing the empirical risk on S and

$$C_{T_i} = \arg\min_{C \in \mathcal{C}} \mathcal{R}(C) + \theta\lambda(T_i, N_i), \tag{5}$$

being $\mathcal{C}$ a model of classifiers (e.g., Support Vector Machines), $\lambda(T_i, N_i)$ a loss function computed over T_i and N_i (note that, generally speaking, $\lambda() \neq L()$), $\mathcal{R}()$ is a suitable regularization and finally θ is a weight.

The minimization involved in Eq. (4)-(5) is clearly non-convex. It can be easily shown that $L(C_{T_i}, S)$ is not submodular [14] and it is not adaptive-monotone [15] (see the Supplementary Material of this paper), thus Eq. (4)-(5) cannot be approximated using greedy submodular function optimization techniques [14,15,16]. Moreover, an exhaustive approach in which all the possible subsets of T are used for training a classifier is intractable. We propose to solve this problem using a RANSAC-like approach [13]. We fix the cardinality n_g ($n_g < n$) of T_i for all i and we build T_i randomly drawing n_g elements of T with replacement. T_i is then used to train a classifier C_{T_i}. We iterate this process a large number of times, obtaining a vocabulary V of weak classifiers. Then, we "verify" each statistical model (classifier) in V using S and $L()$ and we select a small subset of V forming an ensemble which is our final classifier. In the following we provide the details.

4.1 Training Details

The strategy above proposed is independent of the specific class of classifiers ($\mathcal{C}$) used for building C_{T_i}. In our implementation we used HOG features and holistic

(non part-based) classifiers based on linear SVMs, as described in [7]. We also followed the suggestions contained in [7] for setting the training parameters, such as, for instance, the SVM parameter 'C', set to 0.01 (which corresponds to θ in Eq. (5)). Due to the limited number of candidate target pedestrians in T, we set n_g to be a small number: $n_g = 400$, but we actually draw only 200 elements from T and we obtain T_i by horizontally flipping all the selected bounding boxes. Jittering could also be used but we do not use it in the current implementation. The number of positives used for training a classifier is lower than the positive samples used in [7]. However, our weak classifiers are used at testing time as an ensemble of weakly-dependent classifiers (weak statistical dependence is due to the partial overlapping of training data), which mutually compensate their errors. Moreover, we will show in Sec. 5 how we can further boost the ensemble performance merging spatially coherent answers.

A set of negatives N_i for the i-th classifier is collected as follows. From the target videos we randomly extract a few frames. Then we randomly select a few tens of windows from each of these frames. The total number of elements of N_i is five times n_g, using the same proportion adopted in [7]. The size of the randomly selected windows in N_i is bounded by $[l, u]$ (Sec. 3). Occasionally, some of the windows in N_i can overlap with pedestrians, so some of them can actually be false negatives. However we follow [26], in which a similar technique is used for unsupervised selection of negatives, exploiting the assumption that false negatives (random overlap of the selected windows with instances of the class of interest) happen with a quite low probability. As a consequence, both the current positive sample set T_i and the current negative set N_i can be noisy. From T_i and N_i we extract HOG features and we train a linear SVM (for details we refer to [7]).

Finally, we bootstrap the obtained classifier collecting hard target negatives (again following the strategy proposed in [7]). Specifically, we randomly select a second set of frames from the target videos and we run the just trained classifier on all the subwindows of these frames whose size is bounded by $[l, u]$. Hard negatives are all the positive answers of the classifier on the input windows and they are merged with N_i for a second turn of training, finally obtaining our weak classifier C_i. During the bootstrap phase we discard those hard negatives which overlap with elements in T, using the intersection over union criterion adopted in the PASCAL challenge. Nevertheless, since T does not represent *all* the pedestrians in the target videos (because the recall rate of the baseline classifier is very low), some of the hard negatives are possibly true positives, hence hard negatives can be noisy. However, since each frame of the training videos is composed of a huge number of subwindows, this situation happens in a minor number of cases. In our experiments the noisy hard negatives and the bootstrap retraining of the classifier largely helps in boosting the performance of the final classifiers.

We iterate the whole procedure collecting a vocabulary $V = \{C_i\}_{i=1}^{M}$ of weak classifiers. We set $M = 1000$. Each element C_i in V is then scored using a loss function $L()$ (Eq. (4)). We tested different loss functions, based on the overall

error, the recall or the precision of C_i on S. Surprisingly, the best results have been obtained with a precision-based criterion, which does not take into account false negatives on S. This is probably due to the way in which we compute the ensemble decision (Sec. 5), which exploits the precision of each single classifier (low number of false positive answers). More in details, our loss function is:

$$L(C_i, S) = 1 - \frac{TP}{TP + FP}, \tag{6}$$

where TP is the number of true positives on S (again using the PASCAL intersection over union criterion), while FP is the number of false positives, i.e., all the positively classified subwindows of all the images in S which do not sufficiently overlap with any true source positive sample. Size boundaries (l, u) here are not necessary and they are not used. We use Eq. (6) to associate each C_i with an error rate $e_i = L(C_i, S)$.

Training and computing the source dataset error of 1000 classifiers is a time consuming operation. With our non-optimized and non-parallelized C++ implementation, it takes about one day on a standard PC. However, it is a fully automatic process, in which there is no human intervention and can be faster and less expensive than manual annotation of hundreds or thousands of target samples. Moreover, since each classifier training and error computing procedure is completely independent from the other classifiers, the whole process can be easily parallelized.

Once the set of errors $\{e_i\}_{i=1}^{M}$ has been computed, we rank V in ascending order and we select the k top most elements. In our experiments we used $k = 5$. We show in Sec. 6 the influence of different values of k and M on the final performance of the ensemble detector. We indicate with $\mathcal{E} = \{C_i\}_{i=1}^{k}$ the final set of selected classifiers.

5 Spatial Consensus Collection

The common way to combine the outputs of a *classifier* ensemble is a (possibly weighted) voting procedure [36,11]. A test feature x is simultaneously input to all the classifiers of a given ensemble $\mathcal{E} = \{C_i\}_{i=1}^{k}$, obtaining k different outputs. Each output can be associated with a confidence weight which need to be *calibrated* [36]. Using a notation similar to [36], a simple, non-weighted *majority vote* rule can be expressed by means of:

$$\mathcal{E}(x) = \arg\max_{\omega \in \Omega} \sum_{1=1}^{k} v_{i,\omega}, \tag{7}$$

where Ω is the set of all the classes and, for each class $\omega \in \Omega$, $v_{i,\omega} = 1$ if classifier C_i chooses class ω, and 0 otherwise. This simple but effective rule is largely used in classifier ensembles and bagging approaches [36,11].

However, our final goal here is the construction of a *detector* ensemble. The difference is that, in a detection scenario, typically based on a sliding window

scan of the input image, input features x are not each other independent, being those features extracted from nearby and possibly overlapping windows highly correlated. In [8] Ding and Xiao observe that "local windows with positive classifier responses often cluster densely around human figures but distribute sparsely in the background". They exploit this observation building a context feature which takes into account the detector's local responses.

We propose to take advantage of the same observation differently, by combining the outputs of our detector ensemble over different windows which are close in the scale and translation space. In our case we have only two classes: $\Omega = \{0, 1\}$, the positive (1) and the negative (0) class. As is well known, the negative class is much more frequent in a sliding window approach [31], thus the classifiers' responses corresponding to the two classes need to be managed asymmetrically. A spatially-dependent majority vote rule can be formulated as follows:

$$\mathcal{E}(G) = \mathbf{1}_{||\mathbf{v}_G||_0 > k/2}(G). \tag{8}$$

In Eq. (8) $\mathbf{1}()$ is the indicator function and k is the cardinality of the ensemble. $G = \{\mathbf{d}_1, \mathbf{d}_2, ...\}$ is a spatial cluster of *positive* detections. Each $\mathbf{d}_j = (b_j, i)$ in G is a pair composed of an image window b_j and the index i of the corresponding classifier C_i whose outcome on b_j was positive:

$$C_i(f(b_j)) = 1. \tag{9}$$

Note that a same image window b_j in G can be associated with more than one positive detection (e.g., $(b_j, i_1), (b_j, i_2) \in G$). $\mathbf{v}_G = (v_1, ..., v_k)^T$ is a k-dimensional *vote* vector, such that, for each i $(1 \leq i \leq k)$:

$$v_i = min(1, |\{(b, i) \in G\}|), \tag{10}$$

and $|A|$ is the cardinality of set A. Note that $\mathbf{v}_G$ is a vote vector collecting votes *over classifiers* and not over classes. Finally, $||\mathbf{x}||_0$ is the 0-norm which counts the number of non-zero elements in vector $\mathbf{x}$.

The intuitive idea behind Eq. (8) is quite straightforward. Given a cluster of nearby positive detections G, we simply count the number of *different* classifiers which contributed to G. If this number is higher than half of the ensemble cardinality (simple, non-weighted majority), then the decision of the ensemble on G is 1, and 0 otherwise (see Fig. 1(a)-1(b)).

Also the implementation is quite straightforward. We independently run every C_i on the whole image, scanning all those windows whose size is included in the range $[l, u]$ (see Sec. 3). Then we collect all the positive detections of *all* the classifiers in a set $D = \{\mathbf{d}_1, \mathbf{d}_2, ...\}$. We do not use classifiers' confidences (and, thus, we do not need any calibration among classifiers). After that, we perform standard clustering of the rectangles in D. We adopted the common procedure described in [31] and briefly summarized in Sec. 5.1. The clustering outcome is a set of detection groups $G_1, G_2, ...$ Each G_h is composed of windows spatially close and with a similar scale. We can now apply Eq. (8) on each G_h. If $\mathcal{E}(G_h) = 1$, then we compute the average rectangle $\bar{b}_h$ using $\{b_j\}_{(b_j,i) \in G_h}$.

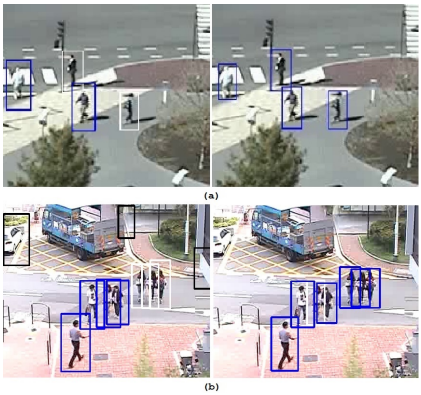


Fig. 2. Some detection results of our system (right) and the Dalal and Triggs method (left) on the MIT Traffic dataset (a) and the CUHK Square dataset (b). Black rectangles are false positives, white rectangles false negatives and blue rectangles true positives.

Finally, $\bar{b}_h$ is a positive window of our detector ensemble. Fig. 2 shows some examples of results. The *whole* testing phase on a large 1152×1440 image takes about 3 seconds on a standard PC (non-parallelized and non-optimized code).

The proposed spatially-dependent majority vote rule is similar to the context score used in [4,12] and in [21], which is based on a max-pool aggregation of the detection scores produced by different classifiers on overlapping image windows. However, in [4,12] the context score is then input to a context-score-based SVM which needs to be trained with supervised data that we do not have (since $\mathcal{D}^t$ is unlabeled). Similarly, in [21] the context score is linearly combined using the pre-computed pairwise co-occurrence frequencies of different classifiers, which need to be trained in a supervised manner.

5.1 Clustering Multiple Detections

Even if clustering of positive detection windows is a standard procedure for object detection approaches, for completeness we briefly summarize here the algorithm we adopted. We followed the well known approach described, for instance, in [31]. Given a set of bounding boxes $B = \{b_1, b_2, ...\}$, B is partitioned in disjoint subsets according to this simple relation: b_i and b_j are in the same subset if the ratio of the intersection area over their union area is greater than 0.6. The parameter 0.6 is commonly adopted by many authors (e.g., [4,9]). The algorithm's output is the resulting partition. When used for Non Maxima Suppression (NMS), for each cluster an average bounding box is also computed.

6 Experiments

We used the datasets and the experimental protocols adopted in [32] in order to compare our approach with the methods and the results reported in the same

article. A detection window is considered a true positive when the intersection area with a ground truth bounding box over the union of the two rectangles is at least 50% (PASCAL rule). The x-axis of the ROC curves is the number of False Positive Per Image (FPPI).

Datasets. We used as target datasets the videos adopted in [32]: the MIT Traffic dataset and the CUHK Square dataset (see Fig. 2 for some examples). They are two videos of two different traffic scenes (respectively, 90 and 60 minutes long), captured with a stationary camera in far field. In both videos there are low resolution pedestrians, vehicles and frequent occlusions. In [32] 420 frames were uniformly sampled from the first 45 minutes of the MIT Traffic video and used for training, and other 100 frames were uniformly sampled from the last 45 minutes and used for testing. Similarly, 350 frames were uniformly sampled from the first 30 minutes of the CUHK Square video and used for training and other 100 frames were sampled from the last 30 minutes for testing. Note that the authors in [32] provide annotations also for the train frames, but these annotations *were not used for training* (neither by us nor by the other methods we compare with). Indeed they are used only for testing purposes in a *transductive* learning paradigm (testing is done on the same video used for training because labeled data are not used). We adopted the same protocol, using the same frames for training our system (i.e., collecting the candidate pedestrian set T, see Sec. 3: one T used for *both* CUHK Square Train and Test, and one T used for *both* MIT Traffic Train and Test) and then we tested our approach on both the train and the test frames (Fig. 3). Two different detector ensembles have been trained, one on the MIT Traffic and the second on the CUHK Square dataset. In both cases we used the INRIA dataset [7] as source dataset.

Parameter Setting. Few parameters need to be set in our approach because we followed the standard setting of the common HOG+SVM approach proposed in [7] for the training phase (Sec. 4.1) and we adopted the NMS setting of [4,9] for the spatial cluster of the positive windows (Sec. 5.1). The number of positive samples n_g for each weak classifier (Sec. 4.1) was set to 400 (200 not considering flipping) because of the low number of candidate pedestrians extracted by the baseline detector in the target videos (a few hundreds per video). We believe that our method can largely benefit of a possible higher number of initial candidates. We used the first half of the CUHK Square Train frames as a validation set in order to select the values of all the remaining parameters, such as k and M (Sec. 4.1) and *we kept constant these parameters in all the experiments and across all the target domains*. We show below the effects of different choices for k and M using the Test frames of the same target video.

Comparison with Other Methods. We compare our approach with other state-of-the-art systems tested on the same datasets: Wang CVPR12 [32], Wang CVPR11 [33], Nair CVPR 04, a modified version of [23] (see [32] for details) and Dalal CVPR 05, the baseline HOG+SVM detector trained on INRIA which we used for the initial candidate pedestrian extraction (Sec. 3). All the results, except ours (called Statistical and Spatial Consensus Collection, SSCC) and Dalal CVPR 05, have been taken from [32]. Note that in [32] the authors also

use as a baseline a HOG+SVM detector but obtained different results. This is probably due to the fact that we used the OpenCv implementation of the Dalal and Triggs method. Moreover, for a fair comparison, we adopted for (Dalal CVPR 05) the same NMS algorithm used in our system and described in Sec. 5.1. In fact we observed that our self-implemented NMS procedure gives better results than the OpenCV NMS.

The results are shown in Fig. 3. The ROC curves are computed as follows. We use a unique threshold τ for all the k classifiers' confidence values discarding all those detections $\mathbf{d}_j$ (Sec. 5) whose SVM confidence is less than τ. Varying τ we obtain different points on the ROC. Note that the classifiers' confidences are not used in Eq. (8). In fact Eq. (8) is simple and effective because it collects spatially-close votes and it does not need supervised weight learning for combining the classifiers' confidences as it is necessary in [4,12,21]. ROC curves can also be computed thresholding $|G|$ without using confidences at all. However, since G is usually small, we obtained only few ROC points using this method.

As it is clear from Fig. 3, we outperform all the other methods and the improvement is particularly sharp in the CUHK Square dataset. Specifically, the (large) improvement with respect to our baseline detector (Dalal CVPR 05) shows that the completely unsupervised method we proposed here for detector adaptation in a specific scene can effectively obtain much better results than a generic detector without any need of manual sample annotation. In the Supplementary Material we show further experiments using these datasets.

Experiments on Different Parts of Our Method. In all the remaining experiments we used the CUHK Square Test dataset. Tab. 1 shows the Average Precision (AP) difference between the full approach with an ensemble of 5 classifiers (SSCC-5) and the case in which a different number k of final classifiers is selected (SSCC-k). Classifier selection is always performed using the loss function of Eq. (6). (SSCC-1) is a single classifier: all the others are ensembles in which an agreement is reached using Eq. (8). The results reported in Tab. 1 show that, with $k > 1$, the cardinality of the ensemble only marginally influences the accuracy of the system, which is a good news, because it means that this parameter does not need to be set using target data.

In the same table we report the results of a "standard", non-spatially dependent majority vote for the ensemble decision (see Sec. 5), which we call Standard Ensemble Decision Rule (SEDR-5). The results for (SEDR-5) were obtained as follows. We used exactly *the same* classifier ensemble $\mathcal{E}$ of (SSCC-5). The only difference is at testing time, because in (SEDR-5) we collect the ensemble consensus using the rule described in Eq. (7) instead of Eq. (8). More in details, *for each window* b of the sliding window process, we extract its HOG representation $f(b)$ and we input $f(b)$ to all the classifiers in $\mathcal{E}$. Then we use the majority vote (Eq. (7)) for computing the class of b. Positive windows are collected in a set B (note that we do not need classifier indexes here) and NMS is applied as described in Sec. 5.1. The improvement of (SSCC-5) over (SEDR-5) shows the advantage of a spatially-dependent consensus collection for a detector ensemble based on a sliding window image scan. Quite surprisingly, (SEDR-5) is even worse than

(SSCC-1) (no ensemble): 0.4845 AP versus 0.4959 AP, respectively. We believe that this is a consequence of the interaction with the NMS stage. In fact classifiers usually agree on neighbouring windows but rarely exactly on the same window (Fig. 1(a)-1(b)). Consequently, the set B (Sec. 5.1) is more fragmented, producing slightly more false positives. As an explanatory example, suppose the best classifier (SSCC-1) outputs a cluster of positive windows around a true pedestrian. If the other classifiers do not agree on the same windows, many of them are discarded using Eq. (7). The subsequent NMS can produce different clusters, some of which non-sufficiently overlapped with the true pedestrian.

Table 1. AP over the CUHK Test dataset with different ensembles and decision rules

SEDR5	SSCC1	SSCC3	SSCC5	SSCC7	SSCC9	SSCC11
0.4845	0.4959	0.5118	0.5184	0.5183	0.5172	0.5175

Table 2. AP over the CUHK Test dataset using different loss functions for classifier selection

Precision	Recall	Error	Random	Random-1
0.5184	0.4750	0.4789	0.4494 (0.0232)	0.4382 (0.0408)

In Fig. 1(c) we show how the AP of our method varies as a function of the cardinality M of the classifier vocabulary V (Sec. 4.1). In this experiment, the cardinality of the final ensemble is fixed ($k = 5$) and we always used the loss function of Eq. (6) and the decision rule of Eq. (8). For every discrete value of M we randomly pre-selected M classifiers from our vocabulary V (before computing their accuracy on S) and we averaged the results over 10 tests.

Finally, Tab. 2 shows the impact of using different loss functions when computing the classifier's error on S (Sec. 4.1). All the tested ensembles are composed of 5 classifiers. The only difference is the loss function adopted for scoring their accuracy. *Precision* indicates the approach presented in the other sections of this paper, based on Eq. (6). Conversely, *Recall* is defined by:

$$L(C_i, S) = 1 - \frac{TP}{TP + FN}, \tag{11}$$

where FN is the number of false negatives (missed detections of C_i on S) and TP is the same as in Sec. 4.1. *Error* is based on:

$$L(C_i, S) = FP + FN. \tag{12}$$

In *Random* we simply randomly chose 5 classifiers from the vocabulary V. In case of *Random*, we repeated the experiment 10 times and the results were averaged in order to decrease the variance of the outcome (in brackets the standard deviation). Tab. 2 motivates our choice in favour of Eq. (6). *Random-1* was computed as *Random* but using only one classifier (drawn at random from V). Comparing *Random-1* with SSCC1 it is clear that using the source dataset for selecting the best classifier(s) is of crucial importance.

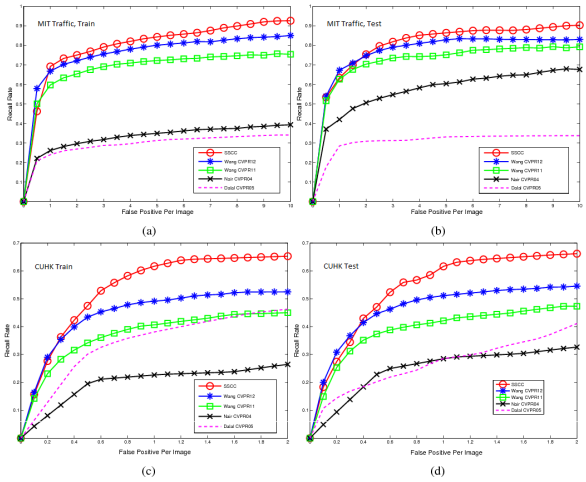


Fig. 3. ROC curves comparing our system (SSCC) with Wang CVPR12 [32], Wang CVPR11 [33], Nair CVPR 04 [23] and Dalal CVPR 05 [7]. The datasets used are: the MIT Traffic Train set (a), the MIT Traffic Test set (b), the CUHK Square Train set (c) and the CUHK Square Test set (d) (better seen at a high magnification).

7 Conclusions

In this paper we proposed two novelties: (1) Transforming the problem of rejecting outliers from an unsupervised target training dataset in a classifier selection problem using random subsets of the target data and a labeled source dataset for verification. (2) A spatially-dependent decision rule for detector ensembles. In contrast with most of the state-of-the-art people detector adaptation works, our method does not rely on sophisticated heuristics or target-dependent parameters for the rejection of outliers. Conversely, our proposed approach allows a simple yet effective and completely automatic construction of a small ensemble of detectors from a very noisy initial bunch of images.

We tested our approach on difficult, low resolution videos obtaining a large accuracy increment with respect to generic pedestrian detectors and state-of-the-art detector adaptation methods.

References

1. Aytar, Y., Zisserman, A.: Tabula rasa: Model transfer for object category detection. In: ICCV (2011)
2. Aytar, Y., Zisserman, A.: Enhancing exemplar svms using part level transfer regularization. In: British Machine Vision Conference (2012)
3. Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., Vaughan, J.W.: A theory of learning from different domains. Mach. Learn. 79(1-2), 151–175 (2010)

4. Bourdev, L., Maji, S., Brox, T., Malik, J.: Detecting people using mutually consistent poselet activations. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part VI. LNCS, vol. 6316, pp. 168–181. Springer, Heidelberg (2010)
5. Cheng, Y.: Mean shift, mode seeking and clustering. IEEE Trans. on PAMI 17(8), 790–799 (1995)
6. Cortes, C., Mohri, M., Riley, M., Rostamizadeh, A.: Sample selection bias correction theory. In: Proceedings of the 19th International Conference on Algorithmic Learning Theory, pp. 38–53 (2008)
7. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR. pp. 886–893 (2005)
8. Ding, Y., Jing, X.: Contextual boost for pedestrian detection. In: CVPR. pp. 2895–2902 (2012)
9. Dollár, P., Tu, Z., Perona, P., Belongie, S.: Integral channel features. In: British Machine Vision Conference (2009)
10. Dollár, P., Wojek, C., Schiele, B., Perona, P.: Pedestrian detection: An evaluation of the state of the art. IEEE Trans. on Pattern Anal. Mach. Intell. 34(4), 743–761 (2012)
11. Duda, R.O., Hart, P.E., Strorck, D.G.: Pattern classification (2nd ed.). Wiley Interscience (2000)
12. Felzenszwalb, P.F., Girshick, R.B., McAllester, D.A., Ramanan, D.: Object detection with discriminatively trained part-based models. IEEE Trans. on Pattern Anal. Mach. Intell. 32(9), 1627–1645 (2010)
13. Fischler, M., Bolles, R.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM 24(6), 381–395 (1981)
14. Golovin, D., Krause, A.: Submodular Function Maximization. In: Tractability: Practical Approaches to Hard Problems (to appear)
15. Golovin, D., Krause, A.: Adaptive submodularity: Theory and applications in active learning and stochastic optimization. Journal of Artificial Intelligence Research (JAIR) 42, 427–486 (2011)
16. Guillory, A., Bilmes, J.: Simultaneous learning and covering with adversarial noise. In: ICML, pp. 369–376 (2011)
17. Jiang, W., Zavesky, E., Chang, S.F., Loui, A.C.: Cross-domain learning methods for high-level visual concept classification. In: ICIP, pp. 161–164 (2008)
18. Khosla, A., Zhou, T., Malisiewicz, T., Efros, A.A., Torralba, A.: Undoing the damage of dataset bias. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part I. LNCS, vol. 7572, pp. 158–171. Springer, Heidelberg (2012)
19. Lampert, C.H., Nickisch, H., Harmeling, S.: Learning to detect unseen object classes by betweenclass attribute transfer. In: CVPR (2009)
20. Lim, J.J., Salakhutdinov, R., Torralba, A.: Transfer learning by borrowing examples for multiclass object detection. In: Neural Information Processing Systems, NIPS (2011)
21. Malisiewicz, T., Gupta, A., Efros, A.A.: Ensemble of exemplar-svms for object detection and beyond. In: ICCV (2011)
22. Matikainen, P., Sukthankar, R., Hebert, M.: Model recommendation for action recognition. In: CVPR, pp. 2256–2263 (2012)
23. Nair, V., Clark, J.J.: An unsupervised, online learning framework for moving object detection. In: CVPR, pp. 317–325 (2004)
24. Ozuysal, M., Calonder, M., Lepetit, V., Fua, P.: Fast keypoint recognition using random ferns. IEEE Trans. Pattern Anal. Mach. Intell. 32(3), 448–461 (2010)

25. Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE Transaction on Knowledge and Data Engineering (2010)
26. Roth, P.M., Sternig, S., Grabner, H., Bischof, H.: Classifier grids for robust adaptive object detection. In: CVPR, pp. 2727–2734 (2009)
27. Saenko, K., Kulis, B., Fritz, M., Darrell, T.: Adapting visual category models to new domains. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part IV. LNCS, vol. 6314, pp. 213–226. Springer, Heidelberg (2010)
28. Sharma, P., Huang, C., Nevatia, R.: Unsupervised incremental learning for improved object detection in a video. In: CVPR, pp. 3298–3305 (2012)
29. Sharma, P., Nevatia, R.: Efficient detector adaptation for object detection in a video. In: CVPR, pp. 3254–3261 (2013)
30. Vázquez, D., López, A.M., Ponsa, D.: Unsupervised domain adaptation of virtual and real worlds for pedestrian detection. In: ICPR, pp. 3492–3495 (2012)
31. Viola, P., Jones, M.: Robust real-time face detection. Int. J. Computer Vision 57(2), 137–154 (2004)
32. Wang, M., Li, W., Wang, X.: Transferring a generic pedestrian detector towards specific scenes. In: CVPR, pp. 3274–3281 (2012)
33. Wang, M., Wang, X.: Automatic adaptation of a generic pedestrian detector to a specific traffic scene. In: CVPR, pp. 3401–3408 (2011)
34. Wang, X., Hua, G., Han, T.X.: Detection by detections: Non-parametric detector adaptation for a video. In: CVPR, pp. 350–357 (2012)
35. Yang, J., Yan, R., Hauptmann, A.G.: Adapting svm classifiers to data with shifted distributions, pp. 69–76. IEEE Computer Society (2007)
36. Zhang, C., Ma, Y.: Ensemble Machine Learning. Springer (2012)