Scene-Specific Pedestrian Detection for Static Video Surveillance

Xiaogang Wang, *Member*, *IEEE*, Meng Wang, *Student Member*, *IEEE*, and Wei Li, *Student Member*, *IEEE*

Abstract—The performance of a generic pedestrian detector may drop significantly when it is applied to a specific scene due to the mismatch between the source training set and samples from the target scene. We propose a new approach of automatically transferring a generic pedestrian detector to a scene-specific detector in static video surveillance without manually labeling samples from the target scene. The proposed transfer learning framework consists of four steps. 1) Through exploring the indegrees from target samples to source samples on a visual affinity graph, the source samples are weighted to match the distribution of target samples. 2) It explores a set of context cues to automatically select samples from the target scene, predicts their labels, and computes confidence scores to guide transfer learning. 3) The confidence scores propagate among target samples according to their underlying visual structures. 4) Target samples with higher confidence scores have larger influence on training scene-specific detectors. All these considerations are formulated under a single objective function called confidence-encoded SVM, which avoids hard thresholding on confidence scores. During test, only the appearance-based detector is used without context cues. The effectiveness is demonstrated through experiments on two video surveillance data sets. Compared with a generic detector, it improves the detection rates by 48 and 36 percent at one false positive per image (FPPI) on the two data sets, respectively. The training process converges after one or two iterations on the data sets in experiments.

Index Terms—Pedestrian detection, transfer learning, confidence-encoded SVM, domain adaptation, video surveillance

1 Introduction

Pedestrian detection is of great interest in video surveillance. Many existing works [1], [2], [3] are based on background subtraction, which is sensitive to lighting variations and scene clutters, and has difficulty in handling the grouping and fragmentation problems [2]. In recent years, appearance-based pedestrian detectors [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20] trained on large-scale data sets have become popular. But it is still a big challenge to train a generic pedestrian detector working reliably on all kinds of scenes. It not only requires a huge training set to cover a large variety of viewpoints, resolutions, lighting conditions, motion blur effects, and backgrounds under numerous conditions, but also a very complex classifier to handle so many variations. It is observed that the performance of state-of-the-art pedestrian detectors trained on a general data set drops significantly when they are applied to videos taken from specific scenes [20]. A generic pedestrian detector is often trained from the INRIA pedestrian data set [4] in the literature. When it is applied to the MIT Traffic data set [21] and our CUHK Square data set, the results have many false alarms and miss detections. This is due to the mismatch between the training samples in the INRIA data set and the samples from the target scenes on camera settings, viewpoints, resolutions, illuminations, and backgrounds. See examples in Fig. 1.

Most surveillance cameras are stationary. If a scene is fixed, the variations of positive and negative samples will be significantly reduced, since videos captured with a single camera only have limited variations on viewpoints, resolutions, lighting conditions, and backgrounds. It is easier to train a pedestrian detector with high accuracy using samples from the target scene. A straightforward way is to train the detector with manually labeled samples from the target scene. But repeating the manually labeling work for every camera view is costly and not scalable. A practical way is to automatically adapt a generic detector to a target scene given a batch of video frames collected from that scene for training, with little or a very small amount of labeling effort. Some efforts [22], [23], [24], [25], [26] have been made in this direction. Many of them are based on ad hoc rules. Their detectors have the risk of drifting during training and it takes the training process multiple rounds to converge.

We tackle this problem by proposing a transfer learning framework and exploring a set of context cues to automatically select scene-specific training samples. Our process of training a scene-specific pedestrian detector is shown in Fig. 2. It starts with a generic pedestrian detector, which is applied to unlabeled samples in videos collected from the target scene. Based on detection results and context cues, some positive and negative samples from the target scene are automatically selected. Since the labels of the selected samples are predicted by detection scores and context cues

• *The authors are with the Department of Electronic Engineering, The Chinese University of Hong Kong, Ho Sin Hang Engineering Building, Shatin, Hong Kong.*

Manuscript received 10 July 2012; revised 8 Apr. 2013; accepted 16 June 2013; published online 21 June 2013.
Recommended for acceptance by G. Mori.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-2012-07-0521.
Digital Object Identifier no. 10.1109/TPAMI.2013.124.

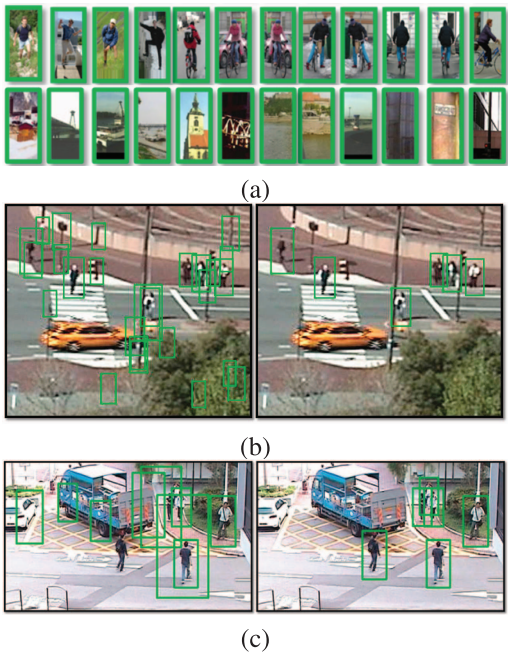


Fig. 1. (a) Positive (first row) and negative (second row) samples from the INRIA data set [4]. (b) Detection results on the MIT Traffic data set and (c) the CUHK Square data set. The left is the results of a generic detector (HOG-SVM [4]) trained on the INRIA data set. The right is the results of a scene-specific detector (also HOG-SVM) automatically trained by our approach.

and could be wrong, their confidence scores are estimated. The selected samples and their confidence scores are used to retrain the scene-specific detector by transfer learning. The updated scene-specific detector is applied to the samples from the target scene again to select more samples for the next round of training. It repeats until convergence. The preliminary result was published in [27].

Transfer learning has been used to solve many domain adaptation problems. However, there is only very limited work [28] studying it in pedestrian detection and some important issues regarding this problem are not well studied yet. For example, how to adapt the distribution of a source training set to a target scene? Many samples selected from the same target scene share visual similarities, and form clusters or manifold structures. How to incorporate such visual structures into transfer learning? Predicting sample labels in a target scene with motion alone is not reliable. Then what are the effective context cues for label prediction and how to combine them? How to integrate training scene-specific detectors and predicting labels of target samples in a principled way, such that transfer learning is robust to wrongly predicted labels and have good learning efficiency in the meanwhile? These issues will be studied in this paper.

1.1 Related Work

Compared with extensive research on generic pedestrian detectors, existing works on scene-specific pedestrian detectors are limited. They typically design a labeler which automatically selects positive and negative samples from the target scene to retrain the generic detector. To effectively improve the performance, the training samples selected by the automatic labeler must be reliable and informative to the original detector. Self-training has been used in [23]. Samples confidently classified by the detector are used to retrain the detector. Since the detector itself is the labeler and not reliable, the selected samples are not informative and are likely to have wrong labels, which make the detector drift. Nair and Clark [29] have used background subtraction results to label training samples for an appearance-based pedestrian detector. The accuracy of the background subtraction labeler is low and it introduces biased labeling which misleads the learning of the detector. For example, static pedestrians may be labeled as non-pedestrian samples. It is unlikely for pedestrians with clothes of similar color to the background to be labeled as pedestrian samples. To make the automatic labeler reliable, Wang and Wang [26] have integrated multiple cues of motions, path models, locations, sizes and appearance to select confident positive and negative samples from the target scene.

Some automatic labelers are designed under the co-training framework [22], [24], [30], [31]. Two detectors based on different types of features are trained iteratively. The prediction of one detector on unlabeled samples is used to enlarge the training set of the other. Levin et al. [22] have built two car detectors using gray images and background subtracted images. Javed et al. [30] have used Haar features and PCA global features to classify blobs into pedestrians, vehicles, and background. They all require manually labeling a small training set from the target scene for initialization. In order for co-training to be effective, the two detectors need to be independent, which is difficult. Dalal et al. [32] have shown that the appearance-based and motion-based pedestrian detectors are highly correlated.

Besides our early work [26], only motion has been used as the context cue and the visual structures of samples from the same target scene have not been considered. In these approaches discussed above, target samples are selected by hard-thresholding confidence scores obtained from the appearance-based detectors or context cues. Hard-thresholding is unreliable and discards useful information. An aggressive threshold makes the detector drift, while a conservative threshold makes the training inefficient and results in many rounds of retraining to converge. Transfer learning provides a principled way to solve domain adaptation problems. It has been successfully applied to object recognition [33], scene categorization [34], action recognition [35], retrieval [36], [37], and visual concept classification [38]. In cross-domain SVM [39] and TrAdaBoost [40], samples in the source data set and the target data set are reweighted differently. Wu and Srihari [41] have introduced a weighted soft margin SVM to incorporate prior knowledge in the training process. Pang et al. [28] have proposed a transfer learning approach to adapt weights of weak classifiers learned from a source data set to a target data set to handle the variation of viewpoints. It assumes that some samples in the target set are manually labeled and does not consider context cues.

When cameras are stationary, the distributions of negative samples are region-specific. Roth et al. [25], [31] have trained a separate detector for each local region. Stalder and Grabner [42] have used tracking and manually input scene geometry to assist labeling. Ali et al. [43] have proposed FlowBoost to learn a scene-specific detector from a sparsely labeled training video assisted by tracking,

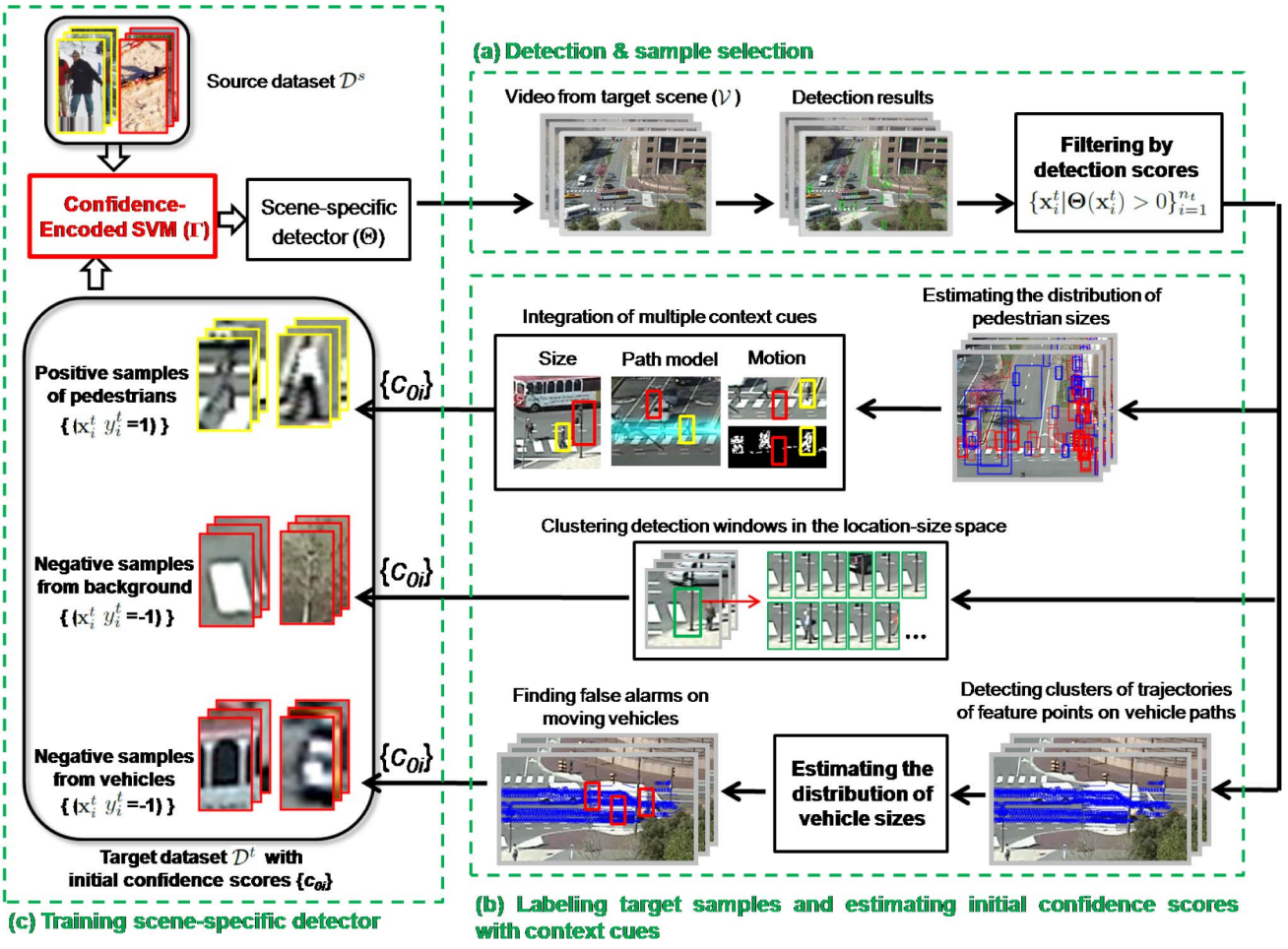


Fig. 2. Diagram of the proposed transfer learning framework. It has three modules: (a) detecting and selecting samples from the target scene, (b) labeling target samples and estimating their initial confidence scores with context cues, and (c) training the scene-specific detectors. Please see the detailed descriptions in Section 1.2.

when small labeling effort is allowed. Without retraining the detector, Jain and Learned-Miller [44] have adapted a generic detector to a new test domain using label propagation.

1.2 Motivations and Contributions

Our motivations can be explained from three aspects:

1. *Indegree-based weighting.* The distribution of a source data set used to train a generic detector usually does not match that of samples from a target scene. Some source samples are more similar to target samples, because they are taken with similar viewpoints, resolutions and lighting conditions, or negative samples come from the same background categories (trees, streets, and buildings). It is desirable to assign larger weights to such samples for training. We build a graph according to visual similarities between source and target samples, and weight source samples according to their indegrees from target samples. The indegrees detect the boundary between the distributions of source and target samples.
2. *Multiple context cues.* Besides the motion cue commonly used in existing works, we explore a rich set of context cues including scene structures, locations, and sizes to guide transfer learning. They are used to select confident positive/negative samples from the target scene to train the appearance-based scene-specific detector. The context cues are complementary to image appearance and are also used to compute confidence scores of target samples.
3. *Confidence-encoded SVM.* The confidence scores are well incorporated into our proposed confidence-encoded SVM, in which target samples with small confidence scores have little influence on the training of the scene-specific detector. Confidence-encoded SVM provides a more principled and reliable way to utilize the context information than existing approaches [22], [26], [29], which selected target samples by hard-thresholding confidence scores and caused the problems of drifting and inefficient training. Using context cues alone, only a small portion of target samples have high confidence scores and they may predict wrong labels. Confidence-encoded SVM propagates confidence scores among target samples along a graph and correct wrong labels according to underlying visual structures of samples. It improves the efficiency of transfer learning.

TABLE 1
Notations Used in This Paper

$(\mathbf{w}, b)$	parameters of SVM weights and bias
$\mathcal{D}^s$	source dataset
$\mathcal{D}^t$	target dataset
$\lvert\mathcal{D}^t\rvert$	number of samples in the target dataset at iteration t
$\mathcal{V}$	video sequence from the target scene
$\boldsymbol{\Theta}$	pedestrian detector
$\boldsymbol{\Xi}$	assigns labels to target samples based on context cues
$\mathbf{c}_0$	initial confidence score estimation on $\mathcal{D}^t$
$\boldsymbol{\Phi}$	assigns $\mathbf{c}_0$ to $\mathcal{D}^t$ according to scene context information
$\boldsymbol{\Gamma}$	training process of Confidence-Encoded SVM
$\mathbf{c}$	propagated confidence on $\mathcal{D}^t$
$\boldsymbol{\nu}$	confidence on $\mathcal{D}^s$
$\boldsymbol{\Psi}$	assigns $\boldsymbol{\nu}$ on $\mathcal{D}^s$
G	objective function of Confidence-Encoded SVM

All these considerations are integrated under a single objective function in confidence-encoded SVM. The effectiveness of the proposed framework is demonstrated through experiments on two video surveillance data sets. It significantly outperforms the generic pedestrian detector and other domain adaptation methods. A new CUHK Square data set with labeled ground truth is introduced to evaluate pedestrian detection in video surveillance. This framework can also be generalized to some other pedestrian detectors such as the deformable part-based model (DPM) [45]. The context cues in our approach assume static camera views. If other context cues can be extracted from moving camera views, confidence-encoded SVM can also be applied.

2 Approach Overview

The diagram of the proposed transfer learning framework is shown in Fig. 2 and its algorithm is summarized in Algorithm 1. Some mathematical notations are summarized in Table 1. The scene-specific pedestrian detector is trained with both a source data set and a target data set whose samples are automatically selected from the target scene.

Algorithm 1. The Proposed Transfer Learning Framework
Input:
The generic detector $(\mathbf{w}_0, b_0)$
The source data set $\mathcal{D}^s$
A video sequence $\mathcal{V}$ from the target scene
The target data set $\mathcal{D}^t \leftarrow \emptyset$
Output:
The scene-specific detector $(\mathbf{w}, b)$
for $r = 1, \ldots, R$ **do**
$(\mathbf{w}_r, b_r) \leftarrow (\mathbf{w}_{r-1}, b_{r-1})$
$\{\mathbf{x}_i^t\} \leftarrow \boldsymbol{\Theta}(\mathbf{w}_r, b_r, \mathcal{V})$
$\{y_i^t\} \leftarrow \boldsymbol{\Xi}(\{\mathbf{x}_i^t\})$
$\mathcal{D}_r^t = \{(\mathbf{x}_i^t, y_i^t)\}$
if $\lvert\mathcal{D}_r^t \cap \bar{\mathcal{D}}^t\rvert / \lvert\mathcal{D}^t\rvert < 0.005$ **then**
/* Convergence condition is reached */
break;
end if
$\mathcal{D}^t \leftarrow \mathcal{D}^t \cup \mathcal{D}_r^t$
$\mathbf{c}_0 \leftarrow \boldsymbol{\Phi}(\mathcal{D}^t)$
$(\mathbf{w}_r, b_r) \leftarrow \boldsymbol{\Gamma}(\mathcal{D}^s, \mathcal{D}^t, \mathbf{c}_0, (\mathbf{w}_r, b_r))$
end for
$(\mathbf{w}, b) \leftarrow (\mathbf{w}_R, b_R)$

The scene-specific detector is retrained iteratively. It starts with a generic detector $\boldsymbol{\Theta}$ using HOG-SVM [4] trained on the source data set $\mathcal{D}^s = \{(\mathbf{x}_i^s, y_i^s)\}_{i=1}^{n_s}$.[1] $\mathbf{x}_i^s$ is a source sample and $y_i^s \in \{-1, 1\}$ is its label. 1 and -1 indicates positive and negative samples. $\boldsymbol{\Theta} = (\mathbf{w}_0, b_0)$ is parameterized by the weights and the bias term of linear SVM. As shown in Fig. 2a, an unlabeled video $\mathcal{V}$ is captured from the target scene. Once $\boldsymbol{\Theta}$ is applied to $\mathcal{V}$, a target data set $\mathcal{D}^t = \{(\mathbf{x}_i^t, y_i^t) | \boldsymbol{\Theta}(\mathbf{x}_i^t) > 0\}_{i=1}^{n_t}$ is obtained by selecting target samples $\mathbf{x}_i^t$ with positive detection scores.

Since the generic detector is far from perfect, there are a significant portion of false positives in $\mathcal{D}^t$. In Fig. 2b, context cues help to assign a label y_i^t and an initial confidence score $c_{0i} \in [-1, 1]$ to each target sample $\mathbf{x}_i^t$. $c_{0i} = 1$ or $c_{0i} = -1$ indicates the highest confidence on the predicted label y_i^t and $c_{0i} = 0$ indicates no confidence. The target data set includes positive samples of pedestrians, negative samples from the background, and negative samples from moving vehicles. Their initial confidence scores are computed with different context cues in different ways. The details are given in Section 3. Positive samples are selected from detection windows inside pedestrian paths and with positive detection scores ($\boldsymbol{\Theta}(\mathbf{x}_i^t) > 0$). Their confidence scores are estimated by integrating the context cues of object sizes, path models, and motions (see Section 3.1). The negative samples from the background are selected through clustering detection windows in the location-size space (see Section 3.2). We only consider detection windows misclassified by the current detector and close to the decision boundary as candidates ($0 < \boldsymbol{\Theta}(\mathbf{x}_i^t) < 0.5$). To find false alarms on moving vehicles, vehicle motions are detected by clustering the trajectories of feature points on vehicle paths. Vehicles and pedestrians are separated with the size cue (see Section 3.3). Detection windows ($\boldsymbol{\Theta}(\mathbf{x}_i^t) > 0$) are selected as confident negative samples if they are on large trajectory clusters and on vehicle paths. If a detection window does not hit any of the above conditions, it is not used to train the detector at the current iteration but could be selected in later rounds when the detector is updated.

In Fig. 2c, a new scene-specific detector $(\mathbf{w}_r, b_r)$ is trained on $\mathcal{D}^s$, $\mathcal{D}^t$, and $\mathbf{c}_0 = \{c_{0i}\}$ with confidence-encoded SVM, whose details are given in Section 4. Once updated with $(\mathbf{w}_r, b_r)$, the detector $\boldsymbol{\Theta}$ is applied to $\mathcal{V}$ again to start the next round of training. The target data set is enlarged by adding new target samples whose detection scores change to positive with the updated detector. The retraining process stops when there are few target samples added or the maximum number of iterations is reached. Experiments on two different data sets show that our approach quickly converges after one or two rounds of training.

3 Context Cues

This section gives the details of selecting target samples and computing initial confidence scores. Positive samples selected according to Section 3.1 have labels $y_i^t = 1$, and negative samples selected according to Sections 3.2 and 3.3 have labels $y_i^t = -1$. The initial confidence scores $\mathbf{c}_0$ are estimated according to context cues. If $y_i^t = 1$, $\mathbf{c}_0$ is estimated with (1) and (5) in Section 3.1; otherwise, with

1. The INRIA data set [4] is used in this work. At initialization, the scene-specific detector is equivalent to the generic detector.

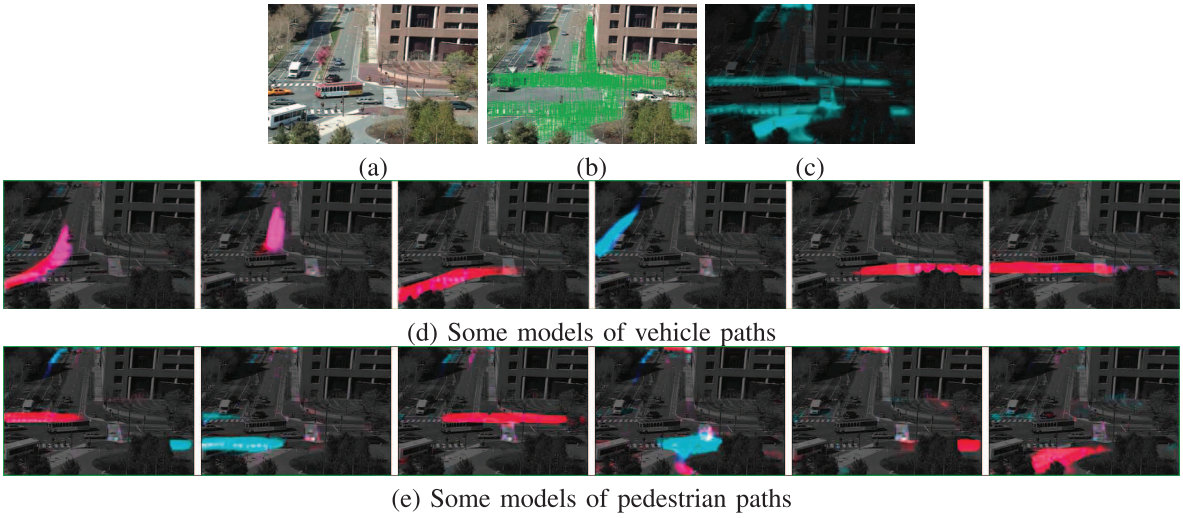


Fig. 3. (a) MIT Traffic scene [21]. (b) Distribution of manually labeled pedestrian windows. (c) Spatial distributions of pedestrian paths unsupervisedly learned in [21]. (d) and (e) Models of vehicle and pedestrian paths. They are distributions over locations and moving directions. Colors indicate moving directions: red ($\rightarrow$), cyan ($\leftarrow$), magenta ($\uparrow$), and green ($\downarrow$).

(6) and (7) in Section 3.4. A detection window is denoted with (x, y, s), where x and y are the coordinates of its center, s is the height. The aspect ratio of detection windows is 2:1. Normally, the back-end of a detector clusters detection windows based on sizes and locations, yielding merged windows at the final result. Instead, we select training samples from unmerged windows and this leads to a more robust scene specific detector. The sampled video frames are scanned with the detector at multiple scales.

3.1 Positive Samples of Pedestrians

Scene structures: path models of pedestrians. The motions of pedestrians and vehicles are regularized by scene structures and follow certain patterns. The models of pedestrian and vehicle paths can increase the reliability of the automatic labeler. It is more reliable to select positive samples on pedestrian paths (see Fig. 3b). It is rare for vehicles to move on pedestrian paths. Since samples on a pedestrian path are either pedestrians or negative samples from the background, the automatic labeling task becomes easier. Because the models of paths are distributions over locations, they are less correlated with appearance and can select more informative samples for retraining. After being retrained, the detector can detect more pedestrians outside pedestrian paths based on appearance.

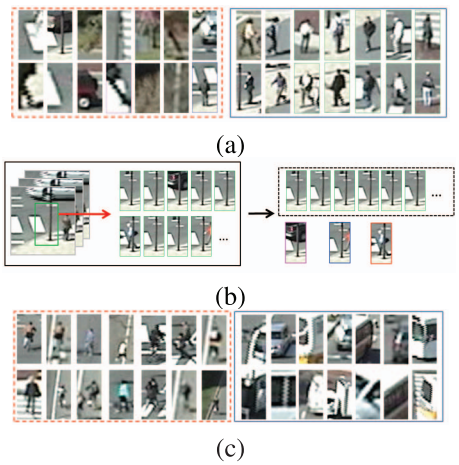


Fig. 4. (a) Examples detected by the generic detector, with positive scores and within the regions of pedestrian paths. They include false alarms (left) to be purified (see Section 3.1). (b) A background false alarm cluster (left) obtained by clustering on locations and sizes includes a few pedestrians accidentally passing by. The false alarms are clustered in appearance and the true positives are removed as outliers (right) (see Section 3.2). (c) Examples detected on vehicle paths. Some true positives are included (left).

This information has not been widely used partially because obtaining scene structures requires manual segmentation or reliable detectors as prerequisites. Manual segmentation is costly and inaccurate. In Figs. 3d and 3e, it is difficult to manually draw the boundaries of paths to accurately match motion patterns, because the view is not top-down. Some paths cannot be recognized from background images. Our previous work [21] has proposed an approach of automatically learning motion patterns from local motions (see examples in Figs. 3d and 3e). Without object detection or tracking, it detects moving pixels by computing the intensity differences between successive frames. Path models are unsupervised learned by exploring the cooccurrence of moving pixels with a Bayesian model. Instead of outputting binary segmentation maps, the path models learned from videos have probabilistic distributions over space and can be used to compute confidence scores in (1). Therefore, we use the models of pedestrians and vehicles paths learned by [21] as a context cue.[2]

To obtain confident positive samples, we only select detection windows, which fall in the regions of pedestrian paths (see Fig. 3d) and whose detection scores are positive. Using path models alone is not reliable. As shown in Fig. 4a, these candidates include a lot of negative samples to be purified with other context cues including sizes, locations, and motions. The initial confidence scores $\mathbf{c}_0$ of the candidates are assigned according to context cues through $\mathbf{c}_0 \leftarrow \mathbf{\Phi}(\mathcal{D}^t)$ shown in Algorithm 1.

2. Given the output of [21], the user needs to label a path model to be a pedestrian path or a vehicle path. But this workload is light.

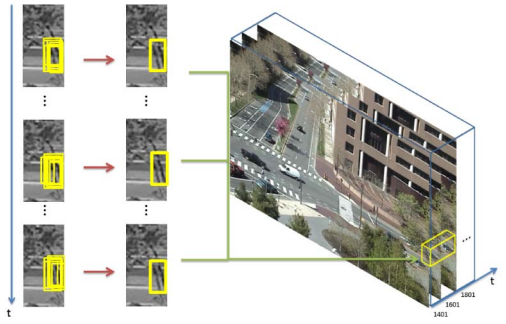


Fig. 5. Hierarchical clustering detection windows in the location-size space. Detection windows in the same frame are first clustered and merged. Merged windows are further clustered across frames. Clusters of detection windows at the same location over a long period are selected as negative samples from the background.

Sizes: estimating distributions of pedestrian sizes. To estimate the size range of pedestrians in the target scene, we construct the histograms of the sizes of detected windows. The mode $\overline{s}$ of the histogram is found by mean shift [46] and the variance (σ) of the mode is also estimated. Pedestrians appear in different sizes because of perspective distortions. Their size variation is modeled as a Gaussian distribution $G(\overline{s}, \sigma)$ to be integrated with other cues later. Size variation could be better modeled through estimating the perspective transformation [47] or estimating different Gaussian distributions in different local regions.

Locations: hierarchical clustering detection windows. It is uncommon for pedestrians to stay at the same location for a long time. If a background patch is misclassified as a pedestrian, similar false alarm patterns tend to repeatedly appear at the same location over a long period. By hierarchical clustering in Fig. 5, we find such samples and exclude them from confident positive samples. Hierarchical clustering on locations and sizes of detection windows has two stages, clustering within a frame and across frames. Clustering within a frame is similar to window merging used in sliding-window-based detection [4]. A sliding-window-based detector gives multiple detections around the location of one pedestrian. Mean shift based on locations and sizes of windows (x, y, s) clusters and merges these windows into one window (x_m, y_m, s_m). The bandwidth is chosen as $\bar{s}/3$, tuned on the INRIA data set. The merged windows are further clustered across frames using mean shift based on (x_m, y_m, s_m). Large clusters across more than 3 minutes are removed from confident positive samples and selected as candidates of confident negative samples from the background. They are not necessarily true negative samples and will be further processed in Section 3.2.

Motions. A detection window on a pedestrian often contains more moving pixels than that on the background. Denote the current frame as I_t. Two reference frames I_{t-50} and I_{t+50} 50 frames are selected. By calculating the frame difference as $0.5(|I_t - I_{t-50}| + |I_t - I_{t+50}|)$, moving pixels inside a detection window are thresholded and counted.

Filtering with multicues. Confident positive samples are selected by integrating multiple cues of motions, models of pedestrian paths, and sizes of detection windows in a probabilistic way. Let $z = (x, y, s, n, N)$ be a detected window, where n is the number of moving pixels in the window and N is the total number of pixels. The likelihood of this detected window being a pedestrian is given by

$$L_{\mathrm{p}}(z) = p_{\mathrm{s}}(s|\bar{s}, \sigma) \cdot p_{\ell}((x, y, s)|\phi_k) \cdot p_{\mathrm{m}}(n, N). \tag{1}$$

p_{s} models pedestrian sizes as a Gaussian distribution

$$p_{\mathrm{s}}(s|\bar{s}, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(s-\bar{s})^2}{2\sigma^2}\right). \tag{2}$$

$p_{\ell}((x, y, s)|\phi_k)$ is the likelihood based on the models of pedestrian paths. Suppose the locations of pixels inside the detection window are $\{(x_j, y_j)\}_{j=1}^{N}$. $\phi_k = (\phi_{k1}, \ldots, \phi_{kW})$ (W is the number of discretized cells in the target scene) is the discrete spatial distribution of the pedestrian path k where the window is detected. Then,

$$\log p_{\ell}((x, y, s)|\phi_k) = \frac{1}{N} \sum_{j=1}^{N} \log p((x_j, y_j)|\phi_k). \tag{3}$$

$p_m(n, N)$ is the likelihood based on the motion cue

$$p_m(n, N) = \frac{n}{N}. \tag{4}$$

The initial confidence score $c_{0p}(z)$ for a detection window z is computed from $L_p(z)$ and normalized to $[0, 1]$,

$$c_{0p}(z) = \frac{L_p(z) - \min_{z'} L_p(z')}{\max_{z'} L_p(z') - \min_{z'} L_p(z')}. \tag{5}$$

3.2 Negative Samples from the Background

To select confident negative samples, we only consider detection windows whose scores satisfy $0 < \boldsymbol{\Theta}(\mathbf{x}_i^t) < 0.5$ as candidates. They are misclassified by the detector and close to the decision boundary. They are informative to the detector and known as hard samples [4], [8]. As explained in Section 3.1, false alarms on the background tend to repeat over time at the same location with similar appearance patterns. Their samples tend to be highly clustered in both the location-size space and the appearance space. After hierarchical clustering on sizes and locations as described in Section 3.1, clusters of detection windows observed at the same locations over long periods are selected as negative samples. As shown in Fig. 4b, they may include a small number of pedestrians who accidentally pass by the same locations. The effect of wrongly labeled samples is reduced by transfer learning introduced in Section 4.

3.3 Negative Samples from Vehicles

It is unreliable to directly count windows detected on vehicle paths as negative samples, since some pedestrians and bicycles also move on vehicle paths (see Fig. 4c). To select confident negative samples, the existence of moving vehicles need to be first detected. It is achieved by feature point tracking and clustering. Feature points are detected and tracked using the KLT tracker [48]. Stationary points and short trajectories are removed. Then trajectories are clustered based on temporal and spatial proximity by mean shift. Each trajectory cluster is assigned to a vehicle path or removed[3] based on the spatial overlap between the cluster and the path. The remaining trajectory clusters mainly correspond to vehicles. The size range of vehicles along each vehicle path is estimated using mean shift in a similar way as in Section 3.1. The trajectory clusters of pedestrians on vehicle paths are removed using the size evidence. If a

3. The removed clusters are from pedestrians or background clutters.

detection window is on a trajectory cluster which is on a vehicle path and whose size is large enough, the detection window is selected as a confident negative sample.

3.4 Initial Confidence for Negative Samples

For a selected negative sample z from the background or vehicles, its likelihood of being a negative sample is

$$L_n(z) = (1 - p_s)(1 - p_m)(1 - p_l), \tag{6}$$

where p_s, p_m, and p_l are computed in the same way as (2), (3), and (4). The initial confidence score of z is computed from $L_n(z)$ and normalized to the range of $[-1, 0]$,

$$c_{0n}(z) = -\frac{L_n(z) - \min_{z'} L_n(z')}{\max_{z'} L_n(z') - \min_{z'} L_n(z')}. \tag{7}$$

4 Training Scene-Specific Detectors

Given the source data set $\mathcal{D}^s$, the selected target data set $\mathcal{D}^t$ and its initial confidence scores $\mathbf{c}_0 = \{c_{0i}\}$ computed in Section 3,[4] the remaining challenge is how to retrain the scene-specific detector. Since $\mathcal{D}^t$ has included wrongly labeled target samples, the retraining process needs to be carefully designed to avoid drifting. Some ad hoc rules have been adopted in [26]. It only selects target samples with high confidence scores using hard thresholding and removes outlier target samples using mean shift clustering. This approach has certain drawbacks. Both the threshold and the bandwidth of mean shift need to be carefully chosen. An aggressive threshold or bandwidth makes the detector drift, while a conservative threshold or bandwidth makes the training inefficient and results in more rounds of retraining to converge. It discards some useful samples and ignores the confidence scores after thresholding.

Algorithm 2. Confidence-Encoded SVM
Input:
The current detector $(\mathbf{w}_0, b_0)$
The source data set $\mathcal{D}^s$
The target data set $\mathcal{D}^t$
The initial confidence scores of target samples $\mathbf{c}_0$
Output:
The scene-specific detector $(\mathbf{w}_u, b_u)$
$\boldsymbol{\nu} \leftarrow \boldsymbol{\Psi}(\mathcal{D}^s, \mathcal{D}^t, \mathbf{c}_0)$
$k = 0$
repeat
$k \leftarrow k + 1$
$\mathbf{c}_k \leftarrow \underset{\mathbf{c}}{\operatorname{argmin}}\ G(\mathbf{c}, \mathbf{w}_{k-1}, b_{k-1}; \mathbf{c}_0, \boldsymbol{\nu}, \mathcal{D}^s, \mathcal{D}^t)$
$(\mathbf{w}_k, b_k) \leftarrow \underset{\mathbf{w},b}{\operatorname{argmin}} G(\mathbf{c}_k, \mathbf{w}, b; \mathbf{c}_0, \boldsymbol{\nu}_r, \mathcal{D}^s, \mathcal{D}^t)$
until Converge
$(\mathbf{w}_u, b_u) \leftarrow (\mathbf{w}_k, b_k)$

We propose a transfer learning approach to update the detector as summarized in Algorithm 2. A source sample $\mathbf{x}_i^s$ is reweighted by ν_i according to its visual distance to target samples. A new scene-specific detector $(\mathbf{w}_u, b_u)$ is trained on both $\mathcal{D}^s$ and $\mathcal{D}^t$ given the current detector $(\mathbf{w}_0, b_0)$, $\boldsymbol{\nu} = \{\nu_i\}$, and $\mathbf{c}_0 = \{c_{0i}\}$ under the proposed confidence-encode SVM in (11). In confidence-encoded SVM, initial confidence estimation $\mathbf{c}_0$ propagates to confidence scores $\mathbf{c}$ which are jointly estimated with $(\mathbf{w}_u, b_u)$. It does not require any thresholding or clustering step to remove unreliable target samples or outliers, since confidence-encoded SVM is robust to wrongly labeled target samples. It does not require tuning parameters and is convenient to use. The details are given in the following sections.

4. c_{0i} is set as $c_{0p}(z)$ in (5) if a target sample is labeled as positive according to context cues and as $c_{0n}(z)$ in (7) if labeled as negative.

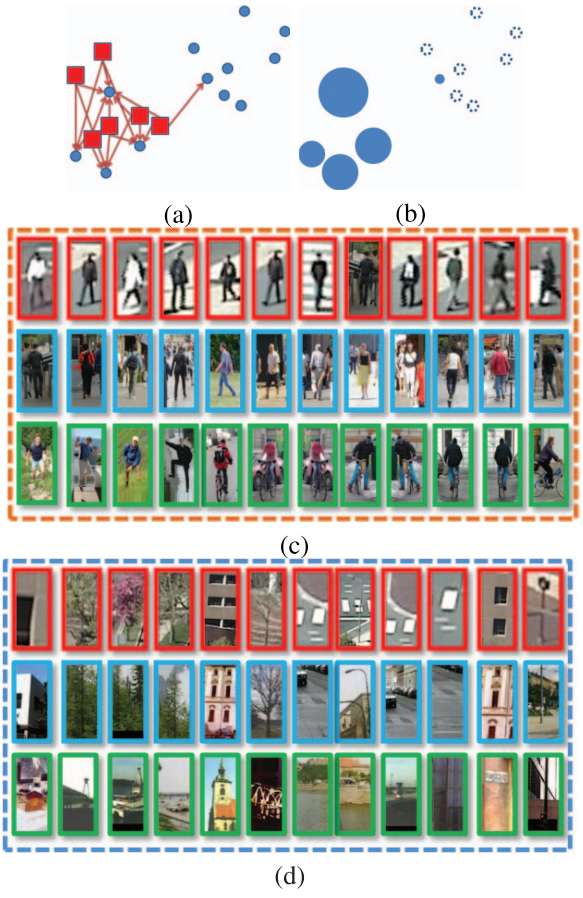


Fig. 6. (a) Red squares indicate target samples and blue points indicate source samples. Each target sample has K ($K = 3$) directed edges pointing toward it K nearest neighbors in the source set. If a source sample is an outlier of the target set, it has a small indegree. (b) The sizes of points indicate the indegrees of source samples. Some source samples have zero indegree and they are denoted as dashed circles. (c) Positive target samples (first row), positive source samples with large indegrees (second row), and positive source samples with zero indegree (third row). (d) Negative target samples (first row), negative source samples with large indegrees (second row), and negative source samples with zero indegree (third row). The source set is the INRIA data set and the target set is the MIT Traffic data set.

4.1 Reweighting Source Samples

As shown in Fig. 6, some source samples better match the target data set and should gain larger weights in training. To weight source samples, a graph between $\mathcal{D}^t$ and $\mathcal{D}^s$ is built. Nodes are samples and edges are created based on K-nearest-neighbors (KNNs). Under the L2 distance,

$$d_{j,i} = \left\|\mathbf{x}_j^t - \mathbf{x}_i^s\right\|^2, \tag{8}$$

there is an edge pointing from a target sample $\mathbf{x}_j^t$ to each of its KNNs in $\mathcal{D}^s$ as shown in Fig. 6a with weight

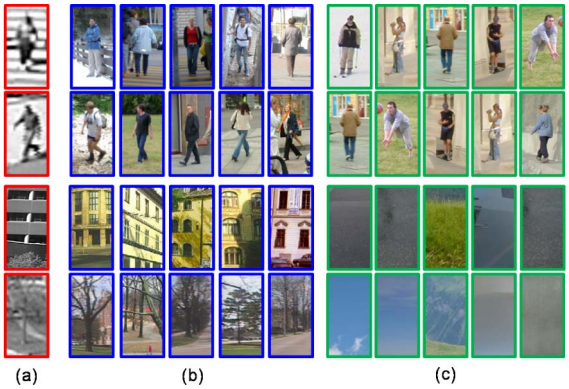


Fig. 7. (a) Positive and negative query examples from the target set (MIT Traffic data set). (b) Top five matched samples from the source set (INTIA) for each query with HOG features and L_2 distance. The top matched positive samples have similar poses as queries. When buildings and trees are selected as negative queries from the target set, similar buildings and trees are selected from the source set by HOG. (c) Top five matched samples from the source set for each query with raw pixels and L_2 distance. They have different poses and semantic categories than the queries.

$$w_{j,i} = \exp\left(-\frac{d_{j,i}^2}{\sigma^2}\right), j = 1, \ldots, n_t, i = 1, \ldots, n_s. \quad (9)$$

The indegree of a source sample is defined as

$$\text{indegree}\left(\mathbf{x}_i^s\right) = \sum_{\mathbf{x}_i^s \in \text{KNN}\left(\mathbf{x}_j^t\right)} w_{j,i}. \quad (10)$$

As shown in Fig. 6, if a source sample is an inlier of the target set, there are a large number of edges pointing toward it and it has a large indegree. Otherwise, its indegree is small. Indegree is widely studied in complex network [49]. It is very effective on detecting the boundary between distributions. Transfer learning is to match the distribution of source samples to that of target samples by reweighting. It is important to detect the boundary between the distributions of target and source data sets, and assign large weights to samples in their overlapping regions. Indegree has not been studied in transfer learning yet. Most transfer learning algorithms [39] directly use KNNs to estimate the distance between a source sample and the target data set.

In Figs. 6c and 6d, it is observed that positive source samples with large indegrees have similar viewpoints as the target samples, and negative source samples with large indegrees are from the same background categories (trees, buildings, roads, and poles) as the target samples.

The confidence score ν_i of a source sample is computed as a sum of indegrees weighted by the initial confidence scores of the target samples, $\nu_i = \sum_{\mathbf{x}_i^s \in \text{KNN}(\mathbf{x}_j^t)} w_{j,i} c_{0j}$. ν_i is further normalized to the range of $[-1, 1]$.

In this work, HOG is used to compute the distance between source and target samples in (8), because the detector is based on HOG and we need to make the two sets of features consistent. The point of transfer learning for domain adaptation is to weight source samples such that their distribution in the feature space is similar to that of target samples at the classification stage. If features to weight source samples are different than those used for classification, we cannot justify the distributions of source and target samples become more similar in the feature space for classification after weighting. Fig. 7 compares matching source and target samples with HOG and raw pixels. The matching result with raw pixels is much worse, since it is easily affected by lightings, misalignment, and intensity distributions. We choose L2 distance to match HOG, since it is commonly used and works well in our experiments. Other metrics could also be effective.

4.2 Confidence-Encoded SVM

Confidence-encoded SVM is an extended version of L2-regularized L2-loss SVM, with objective function G:

$$\begin{aligned} \min_{\mathbf{c},\mathbf{w},b} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n_s}\left(\nu_i \xi_i^s\right)^2 + C\sum_{j=1}^{n_t}\left(c_j \xi_j^t\right)^2 \\ & + \frac{\mu}{2}\mathbf{c}^T\mathbf{L}\mathbf{c} + \frac{\lambda}{2}(\mathbf{c}-\mathbf{c}_0)^T\mathbf{A}(\mathbf{c}-\mathbf{c}_0) \\ \text{s.t.} \quad & y_i^s\left(\mathbf{w}^T\mathbf{x}_i^s + b\right) \geq 1 - \xi_i^s, i = 1, \ldots, n_s, \\ & y_j^t\left(\mathbf{w}^T\mathbf{x}_j^t + b\right) \geq 1 - \xi_j^t, j = 1, \ldots, n_t, \\ & \xi_i^s \geq 0, i = 1, \ldots, n_s, \\ & \xi_j^t \geq 0, j = 1, \ldots, n_t, \end{aligned} \quad (11)$$

where C, μ, and λ are preset parameters. $\mathbf{c} = (c_1, \ldots, c_{n_t})$ are the propagated confidence scores on the target data set. They are jointly estimated with SVM parameters. The slack penalty of misclassifying a source (target) sample $\mathbf{x}_i^s(\mathbf{x}_j^t)$ is proportional to its confidence score ν_i (c_j). The lower confidence a sample has, the smaller influence it has on training SVM. Some approaches [22], [23], [24], [26], [30] selected positive/negative target samples by hard-thresholding confidence scores and treated them equally when training SVM. It is special case of ours, considering c_j can only be $1, -1$, or 0. Our approach does not require thresholding which causes errors. If the threshold is aggressive, some wrongly labeled samples are used to train SVM and cause the drifting problem. If the threshold is conservative, not enough samples are selected and the performance of the detector improves slowly after many rounds of training. It also does not make sense to treat all the training samples with different confidence equally after thresholding.

4.2.1 Confidence Propagation

Using context cues alone, only a small portion of target samples have high confidence scores and some predicted labels are wrong (see examples in Fig. 8). Image patches from the same scene form clusters and manifolds based on their visual similarities. If two image patches are visually similar, they should have the same label because they are captured under the same condition.[5] We propagate confidence scores to obtain more samples with high confidence and reduce the confidence of samples with wrong labels.

Estimation of confidence scores $\mathbf{c}$ depends on three terms in (11). $\mathbf{c}^T\mathbf{L}\mathbf{c}$ comes from graph Laplacian and requires that visually similar samples have similar confidence scores. A pairwise weight matrix $\mathbf{W}$ is calculated from $\mathcal{D}^t$ by

$$w_{i,j} = \exp\left(-\frac{\left\|\mathbf{x}_i^t - \mathbf{x}_j^t\right\|^2}{\sigma^2}\right). \quad (12)$$

5. This assumption may not hold if image patches are from different scenes. Scene variations may make two image patches of different categories similar in appearance.

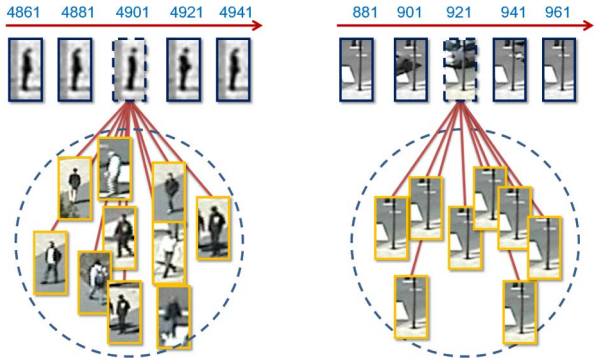


Fig. 8. Left: A pedestrian is stationary for a long period and therefore is labeled as a negative sample with an initial high confidence score according to the motion cue. Its confidence score gets close to zero after confidence propagation because a large number of other samples with similar visual appearance to it are labeled as positive samples with high confidence scores. Therefore it will not have a bad influence on training. Right: A background patch is labeled as a negative sample with a low initial confidence score because a vehicle happens to pass by and causes motions. Its confidence score becomes high after confidence propagation because some similar background patches are labeled as negative samples with high confidence scores.

It is sparsified by setting $w_{ij} = 0$, if $\mathbf{x}_i$ and $\mathbf{x}_j$ are not the KNNs of each other. A diagonal matrix $\mathbf{D}$ is defined by $\mathbf{D}_{ii} = \sum_{j=1}^{n_t} w_{ij}$. Then the graph Laplacian is $\mathbf{L} = \mathbf{D} - \mathbf{W}$. Although our work only considers visual distances, other cues which characterize the structures of samples can also be used to compute $\mathbf{L}$. For example, temporal consistency of samples can be exploited if tracking is available.

$\mathbf{A}$ is a diagonal matrix, where $\mathbf{A}_{jj} = |c_{0j}|$. Therefore,

$$(\mathbf{c} - \mathbf{c}_0)^T \mathbf{A} (\mathbf{c} - \mathbf{c}_0) = \sum_{j=1}^{n_t} (c_j - c_{0j})^2 |c_{0j}| \tag{13}$$

is used to regularize $\mathbf{c}$ from its initial estimation $\mathbf{c}_0$. If c_{j0} is low, which means that the context information does not has a strong opinion on the label of $\mathbf{x}_j^t$, then its confidence score can be easily influenced by other samples with less penalty. Otherwise, its confidence score can be changed only when there is strong evidence from other samples.

The third term $\sum_{j=1}^{n_t} (c_j \xi_j^t)^2$ tends to assign small confidence scores to samples misclassified by SVM (with large ξ_j^t), since the context information and appearance-based classifier have disagreement on them.

4.2.2 Optimization

We optimize (11) iteratively. Denote the objective function by $G(\mathbf{c}, \mathbf{w}, b)$. Optimization starts with an initial model $(\mathbf{w}_0, b_0)$. At each iteration k, let $\mathbf{c}_k$ minimize $G(\mathbf{c}, \mathbf{w}_{k-1}, b_{k-1})$. Since it is a convex quadratic function, the optimal $\mathbf{c}_k$ can be found by setting its derivative to be 0. We obtain the parameters $(\mathbf{w}_k, b_k)$ of a new model by minimizing $G(\mathbf{c}_k, \mathbf{w}, b)$ using a modified version of LIBLINEAR [50], which is based on the trust region Newton method. This algorithm converges since the objective function monotonically decreases after each step. According to our experimental results, it usually converges within five iterations. Fig. 9 shows an example of how the confidence scores and detection scores by SVM change after three iterations. After convergence, the detection scores and confidence scores tend to agree with each other.

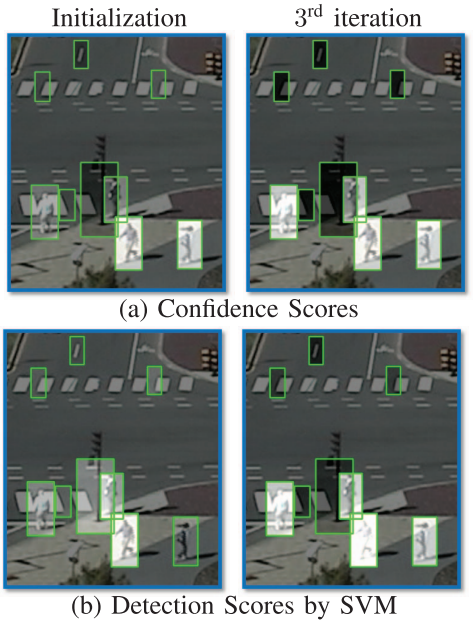


Fig. 9. (a) Confidence scores and (b) detection scores by SVM change after three iterations when optimizing confidence-encoded SVM. Windows are image patches in $\mathcal{D}^t$. A bright window indicates a score close to 1 and a dark one indicates a score close to -1. At initialization, there are large differences between confidence and detection scores. After three iterations, they look more consistent and correct. Experiment is on the MIT Traffic data set.

Confidence-encoded SVM is a latent variable model and its objective function is nonconvex. Other optimization methods [51], [52] could be adopted to obtain a better local minimum. For example, Kumar et al. [52] proposed *self-paced learning*. Instead of considering all the training samples simultaneously, its learning first only uses easy samples and then includes harder ones. More advanced optimization strategies could be studied in the future work.

4.3 Final Scene-Specific Pedestrian Detector

Once the scene-specific detector is trained on sampled video frames, it can be used to detect pedestrians in new frames purely based on appearance without other cues. Although the context cues are effective on selecting training samples, they cannot guarantee high detection accuracy when being used alone in the final detector.[6] If the detector relies on path models, pedestrians walking on vehicle paths may be missed. Replying on motions or sizes, stationary or small pedestrians may be missed. The final detector could be improved by integrating the outputs of the appearance-based detector and context cues. But it is hard to decide combination weights, since manually labeled training samples from target scenes are unavailable. If automatically selected and labeled target samples are used to train combination weights, bias would be introduced since those samples have high confidence scores according to context cues. In this paper, our final detector only considers appearance in the following experimental evaluation.

6. The purpose of using these cues is to find some confident samples without introducing bias on appearance but not to detect all the samples.

(a) (b)

Fig. 10. (a) MIT Traffic data set and (b) CUHK Square data set.

5 Generalization to Other Pedestrian Detectors

Our approach can also be applied to some other detectors such as the deformable part-based model (DPM) [8]. Similar to (11), we encode the confidence scores and the smoothing term into the objective function:

$$\begin{aligned}\min_{\mathbf{c},\boldsymbol{\beta}} \quad & \frac{1}{2}\|\boldsymbol{\beta}\|^2 + C\sum_{i=1}^{n_s}\left(\nu_i\xi_i^s\right)^2 + C\sum_{j=1}^{n_t}\left(c_j\xi_j^t\right)^2 \\ & + \frac{\mu}{2}\mathbf{c}^T\mathbf{L}\mathbf{c} + \frac{\lambda}{2}(\mathbf{c}-\mathbf{c}_0)^T\mathbf{A}(\mathbf{c}-\mathbf{c}_0) \\ \text{s.t.} \quad & y_i^s f_{\boldsymbol{\beta}}(\mathbf{x}_i^s) \geq 1-\xi_i^s, i=1,\ldots,n_s, \\ & y_j^t f_{\boldsymbol{\beta}}(\mathbf{x}_j^t) \geq 1-\xi_j^t, j=1,\ldots,n_t, \\ & \xi_i^s \geq 0, i=1,\ldots,n_s, \\ & \xi_j^t \geq 0, j=1,\ldots,n_t.\end{aligned} \tag{14}$$

where $\boldsymbol{\beta}$ is the composite model parameter including both the root filter and part filters, as well as the placement deviation penalties. The classification score $f_{\boldsymbol{\beta}}(\mathbf{x})$ is obtained by an optimal placement $\mathbf{z}$, which is a latent variable:

$$f_{\boldsymbol{\beta}}(\mathbf{x}) = \max_{\mathbf{z}\in\mathbf{Z}(\mathbf{x})} \boldsymbol{\beta}\cdot\boldsymbol{\Phi}(\mathbf{x},\mathbf{z}).$$

Optimization can be done in the fashion as in Section 4.2.

The computation cost of DPM is about three times higher than HOG-SVM. This is a major concern in real-time surveillance applications. DPM also requires higher resolutions while many pedestrians are small in surveillance videos. The typical pedestrian size is 40×20 in the MIT Traffic data set and 24×12 in the CUHK Square data set.

6 Experimental Results

Experiments are conducted on the MIT Traffic data set [26] and the CUHK Square data set which is constructed by us.[7] The two scenes are shown in Fig. 10. We adopt the PASCAL "50 percent rule," i.e., the overlapping region between the detection window and the ground truth must be at least 50 percent of the union area. Recall rate versus false positive per image (FPPI) is used as the evaluation metric.

6.1 Data Sets

MIT Traffic data set is a 90-minute long video at 30 fps. It captured a street intersection with an eagle-eye perspective and was recorded with a stationary camera. Occlusions and varying illumination conditions apply. Four hundred twenty frames are uniformly sampled from the first 45 minutes video to train the scene-specific detector. Hundred frames are uniformly sampled from the last 45 minutes video for testing.

CUHK Square data set is also captured by a stationary camera from a bird-view. It is a 60-minutes long video at 25 fps. Since the camera was much lower than that in the MIT Traffic data set, perspective deformation is more challenging. Three hundred fifty frames are uniformly sampled from the first 30 minutes video for training. Hundred frames uniformly sampled from the last 30 minutes video for testing.

In both data sets, the bounding boxes of pedestrians are manually labeled as ground truth. Note that when our approach trains the scene-specific detector, it does not use any labeled samples from the videos. An the test stage, only the appearance-based detector without context cues is used.

6.2 Parameter Setting

In (11), $C = 1/(\frac{1}{n_s+n_t}(\sum_{i=1}^{n_s}\|\mathbf{x}_i^s\| + \sum_{i=1}^{n_t}\|\mathbf{x}_i^t\|))^2$, and $\mu = \lambda = 1$. The performance of our approach is stable when μ and λ change in a relatively large range. σ in (9) is defined by $\sigma^2 = \frac{1}{n_t\cdot n_s}\sum_{i=1}^{n_t}\sum_{j=1}^{n_t} d_{ji}^2$, where d_{ji} is given by the L2 distance between a source sample $\mathbf{x}_i^s$ and a target sample $\mathbf{x}_j^t$. In (12), σ is given by $\sigma^2 = \frac{1}{(n_t-1)^2}\sum_{j=1}^{n_t}\sum_{i=1}^{n_t} d_{ij}^2$. The experiments on the two data sets use the same fixed-value parameters for μ and λ, and compute parameters in the same way.

6.3 Results

6.3.1 Comparison of Scene-Specific Detectors

We compare with the following approaches. When we talk about detection rates, it is assumed that FPPI $= 1$.

- A generic HOG+SVM detector trained on the INRIA data set (Generic).
- Adapting a generic detector to the target scene by integrating multiple context cues with hard thresholding and clustering as proposed in [26]. It used exactly the same context cues as ours.
- Adapting a generic detector to the target scene using background subtraction to select samples (similar to [29], but its detector is HOG-SVM not boosting).
- A HOG-SVM detector is trained on the INRIA data set and all the manually labeled frames from the target training set, and then bootstrapped on target samples (INRIA+manual). Bootstrap is the same as [4].
- Several transfer learning approaches including transfer boosting [28], easy semi-supervised domain adaptation [53] (EasyAdapt), adaptive SVM [36] (AdaptSVM), and cross domain SVM [39] (CDSVM). These transfer learning approaches all require manually labeled target samples. We assume the INRIA data set and 50 manually labeled frames from the target scene are available for training.

Figs. 11a and 11b compare with two automatic scene adaptation approaches [26] and [29]). They do not require manually labeled target samples. Their training time was reported in Table 2. Although the training time of [29] is comparable with ours, its accuracy is significantly lower.

7. http://www.ee.cuhk.edu.hk/~xgwang/CUHK_square.html.

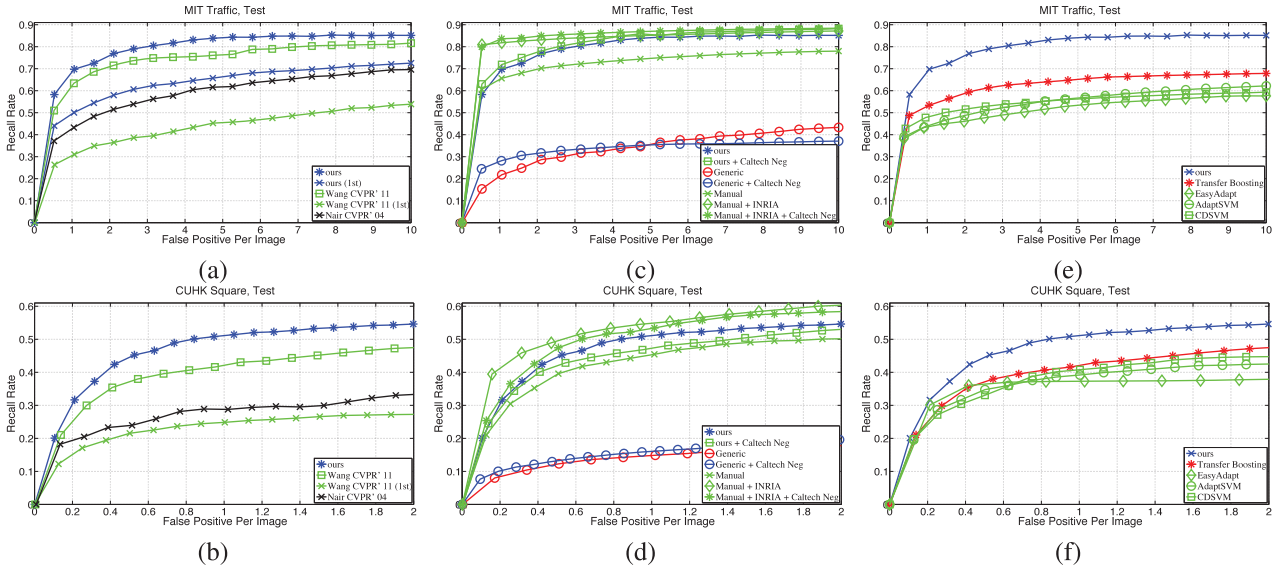


Fig. 11. (a) and (b) compare with [26] and [29], which do not require manually labeled target samples for training. For our approach and [26], the results after the first round of training are also reported. (c) and (d) compare with the generic detector and the scene-specific detector trained with manually labeled frames from the target scene. MIT Traffic and CUHK Square data set have 420 and 350 frames in the target training set. They are all labeled for training the scene-specific detector. (e) and (f) compare with transfer learning approaches which require both source samples and manually labeled target samples for training. The INRIA data set and 50 manually labeled frames from the target scene are used by them.

TABLE 2
Efficiency and Accuracy of Different Methods

	MIT Traffic			CUHK Square		
Training Method	No. Rounds	Training Time	Final Detection Rate	No. Rounds	Training Time	Final Detection Rate
Ours	2	4 hours	69%	1	1.2 hours	51%
Wang CVPR'11	10	18 hours	62%	7	8 hours	44%
Nair CVPR'04	5	2.5 hours	42%	4	2 hours	28%
Generic	N/A	N/A	21%	N/A	N/A	15%
Manual + INRIA	2	1.7 hours	81.5%	2	0.9 hours	55%

All tested on a machine with Intel Xeon W5580 3.2G CPU.

Compared with [26], our approach converges with fewer rounds, and leads to a higher performance (7 percent improvement on detection rate). Although our approach takes slightly longer time at each round of training than [26], its total training time is much lower, because [26] is based on ad hoc rules and hard-thresholding, which reduce its efficiency. Another point of achieving good efficiency is to extract useful information as much as possible from fewer target training samples. At the first round of training, both [26] and ours have the same target set and initial confidence scores $\mathbf{c}_0$, since they utilize the same context cues. Confident-encoded SVM achieves a 48 percent detection rate after the first round training, while [26] only achieves 30 percent. It shows that our approach makes better use of the same target training data.

Figs. 11c and 11d show that the adapted detector obtained by our approach significantly outperforms the generic detector. On the MIT Traffic and CUHK Square test sets, it improves the detection rates from 21 to 69 percent, and from 15 to 51 percent. In the literature [15], people have observed improvement when a detector is bootstrapped with additional negative samples even from a different data set. We bootstrap our adapted detector and the generic detector with negative samples from the Caltech training set [54], denoted as "Ours+Caltech Neg" and "Generic+Caltech Neg." Both detectors are slightly improved only on the MIT Traffic data set, because the scenes of the Caltech data set are similar to the MIT Traffic data set, but quite different than the CUHK Square data set. Additional source samples are helpful only when they are similar to target samples. "Generic+Caltech Neg" is much worse than our adapted detector. It shows that collecting training samples from the target scene is important. We also compare with the HOG-SVM detectors trained on manually annotated frames from the target scenes. All the 420 and 350 training frames of the two data sets are used.[8] If the detectors are only bootstrapped on the target set (denoted as "Manual"), their detection rates (66 and 45 percent, respectively) are lower than our adapted detector. If the INRIA data set is included as the source data (denoted as "INRIA+Manual"), it outperforms our adapted detector by 10 and 3 percent on the two data sets. It shows that source data is important especially when the target set is not large enough. The performance of "INRIA+Manual" can be viewed as an upper bound that our approach targets on, because it has all the data that our approach has and additional manual labels of all the target samples. Bootstrapped on additional negative samples (denoted as "INRIA+Manual+Caltech Neg"), the performance gets improved only on the MIT Traffic data set.

Figs. 11e and 11f compare with several transfer learning algorithms. All these algorithms assume that a small portion of the target set is manually labeled, and

8. The numbers of positive samples are 1,573 and 956, respectively.

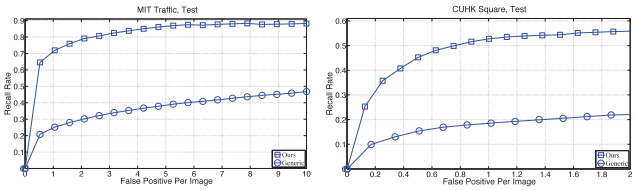


Fig. 12. Performance of our approach when Caltech instead of INRIA data set is used as the source set.

weight source training samples during the learning process. However, they do not utilize contextual information from the target scene. EasyAdaptSVM [53], AdaptSVM [36], and CDSVM [39] are general transfer learning algorithms and Transfer Boosting [28] is specially designed for pedestrian detection. But all have big gaps with our approach, even though they have additional manual labels of target samples.

6.3.2 Different Factors in Transfer Learning

Fig. 12 reports the result of our approach when using the Caltech data set [54] instead of INRIA [4] as the source set. Caltech has a much larger number of positive training samples than INRIA (4,640 versus 1,517). It leads to a better generic detector and a better adapted detector. The big improvement caused by our approach still can be observed.

Figs. 13a and 13b investigate the effectiveness of 1) including target samples for training, 2) confidence propagation, and 3) weighting source samples using indegrees, on the two data sets. Only weighting source samples without including target samples ("Source Only"), the detection rates drops by 43 and 15 percent on the MIT Traffic and CUHK Square data sets, respectively. Without confidence propagation ("No Propagation"), it takes two more rounds to converge on the MIT Traffic data set and the detection rate drops by 11 percent. On the CUHK Square data set, it takes one more round to converge and the detection rate drops by 13 percent. If source samples are not weighted ("No Source Weighting"), the detection rates drop by 6 and 7 percent on two data sets. If source samples are weighted using KNNs as [39] ("KNN"), which is commonly used in transfer learning, the detection rates drop by 5 percent on both data sets.

6.3.3 Effectiveness of Context Cues

Figs. 13c and 13d investigate the effectiveness of different context cues by removing each of them separately. The models of pedestrian paths are the most effective in the traffic scene, where pedestrians and vehicles are the two major classes of moving objects to be distinguished and their motions are strongly regularized by path models. The cue of sizes is least effective on the MIT Traffic data set, because there is large projective distortion in that scene.

6.3.4 Design of the Final Detector

Our final detector only considers the appearance cue. Table 3 evaluates the effectiveness of incorporating context cues in the final scene-specific detector as discussed in Section 4.3. The output of the adapted appearance-based detector is combined with the output of a context cue with

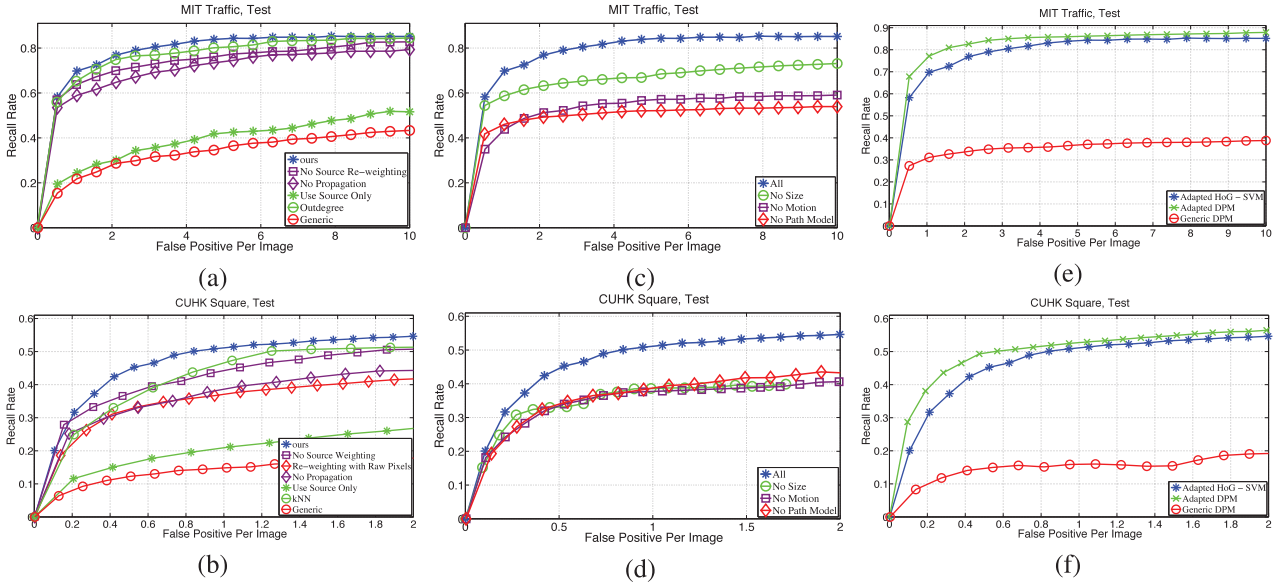


Fig. 13. (a) and (b) investigate the effectiveness of different factors in the transfer learning framework. (c) and (d) investigate the effectiveness of different context cues. (e) and (f) improvement of adapting the deformable part model [45] to target scenes with our framework.

TABLE 3
Improvements on Detection Rates by Combining Context Cues in the Final Scene-Specific Detector Compared with Using the Appearance-Based Detector Alone (69 Percent) on the MIT Traffic Data Set

w_p/w_m	1/10	1/3	1/2	1	2	3	10
$s_a + w_p s_p$	$+0\%$	$+0.4\%$	$+0.8\%$	$+0.5\%$	$+0.2\%$	$+0\%$	-8%
$s_a + w_m s_m$	$+0\%$	$+0\%$	$+0\%$	-0.4%	-0.8%	-1.3%	-12%

s_a, s_p, and s_m are the outputs of the appearance-based detector, path models, and motion cue. w_p and w_m are the weights of path models and motion cue.

a weight. Since the ranges of the two outputs could be very different, we change the weight from 0.1 to 10. No significant improvement is observed compared with using the appearance-based detector alone. When the context cue has a large weight, the detection rate drops significantly.

6.3.5 Extension to DPM

Our approach can be generalized to other detectors. Figs. 13e and 13f show the result of applying our framework to the deformable part model [45] described in Section 5. The adapted PDM has a huge improvement over the generic DPM, and outperforms the adapted HOG-SVM.

7 Discussions

Our training process converges after one or two rounds on the two data sets. Convergence means that few target samples are added and the performance changes little. It can be proved that our approach converges, but without guarantee on the number of iterations. If unlimited memory and computing power are assumed, it may work by taking all the target samples at one round and (11) is only optimized once. In practice, we keep adding new samples crossing the margin to the target set at each round. Therefore, it must converge; otherwise, all the samples would be eventually selected and it stops. Our approach is related to the bootstrapping strategy [4], [15], [45], [55] of mining hard samples iteratively. Both approaches keep adding hard negative samples close to the margin. Bootstrapping includes all the positive samples at the initial step, while our approach incrementally adds positive samples. Bootstrapping knows the labels of all the samples as ground truth, while ours relies on context cues to predict labels and estimate confidence scores. In order for our approach to converge quickly, context cues and confidence scores must be effectively used such that classification plane moves fast in a right direction at each round. The fast convergence on the two data sets shows the effectiveness of our approach.

We will work on the online version in the future work. A dynamically updated detector can better handle the variation of illumination and background. Our approach can also be improved by training different detectors for different regions. In each region, pedestrians appear in similar sizes and the background is homogeneous in texture. Region-based context cues are more reliable, and a region-based detector can better classify positive and negative samples in the same region, whose distributions are simpler.

8 Conclusions

In this paper, we propose a new transfer learning framework to automatically adapt a generic pedestrian detector to a specific scene with the minimum annotation requirement. The source data set, multiple context cues, and the visual structures of target samples are well integrated under the proposed confidence-encoded SVM. It quickly converges after one or two rounds of training on the two data sets. The whole system has only two free parameters (μ and λ). When they are fixed as 1 on both data sets, our approach significantly improves the detection rates by 48 and 36 percent at 1 FPPI on compared with the generic detector.

Acknowledgments

This work was supported by General Research Fund sponsored by Research Grants Council of Hong Kong (Project nos. CUHK 417110, CUHK 417011, and CUHK 419412).

References

[1] C. Nakajima, M. Pontil, B. Heisele, and T. Poggio, "Full-Body Recognition System," *Pattern Recognition,* vol. 36, pp. 1977-2006, 2003.

[2] B. Bose, X. Wang, and W.E.L. Grimson, "Multi-Class Object Tracking Algorithm that Handles Fragmentation and Grouping," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR),* 2007.

[3] T. Zhao, R. Nevatia, and B. Wu, "Segmentation and Tracking of Multiple Humans in Crowded Environments," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 30, no. 7, pp. 1198-1211, July 2008.

[4] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR),* 2005.

[5] B. Wu and R. Nevatia, "Detection of Multiple Partially Occluded Humans in a Single Image by Bayesian Combination of Edgelet Part Detectors," *Proc. 10th IEEE Int'l Conf. Computer Vision (ICCV),* 2005.

[6] Z. Lin, L. Davis, D. Doermann, and D. Dementhon, "Hierarchical Part-Template Matching for Human Detection and Segmentation," *Proc. 11th IEEE Int'l Conf. Computer Vision (ICCV),* 2007.

[7] P. Sabzmeydani and G. Mori, "Detecting Pedestrians by Learning Shapelet Features," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR),* 2007.

[8] P. Felzenszwalb and D. McAllester, "A Discriminatively Trained Multiscale, Deformable Part Model," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR),* 2008.

[9] O. Tuzel, F. Porikli, and P. Meer, "Pedestrian Detection via Classification on Riemannian Manifolds," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 30, no. 10, pp. 1713-1727, Oct. 2008.

[10] X. Wang, X. Han, and S. Yan, "An Hog-LBP Human Detector with Partial Occlusion Handling," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR),* 2009.

[11] C. Wojek, S. Walk, and B. Schiele, "Multi-Cue Onboard Pedestrian Detection," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR),* 2009.

[12] W.R. Schwartz, A. Kembhavi, D. Harwood, and L.S. Davis, "Human Detection Using Partial Least Squares Analysis," *Proc. 12th IEEE Int'l Conf. Computer Vision (ICCV),* 2009.

[13] W. Ouyang and X. Wang, "A Discriminative Deep Model for Pedestrian Detection with Occlusion Handling," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR),* 2012.

[14] W. Ouyang and X. Wang, "Single-Pedestrian Detection Aided by Multi-Pedestrian Detection," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR),* 2013.

[15] M. Enzweiler and D.M. Gavrila, "Monocular Pedestrian Detection: Survey and Experiments," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 31, no. 12, pp. 2179-2195, Dec. 2009.

[16] L. Bourdev and J. Malik, "Poselets: Body Part Detectors Trained Using 3D Human Pose Annotations," *Proc. 12th IEEE Int'l Conf. Computer Vision (ICCV),* 2009.

[17] M. Enzweiler, A. Eigenstetter, B. Schiele, and D.M. Gavrila, "Multi-Cue Pedestrian Classification with Partial Occlusion Handling," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR),* 2010.

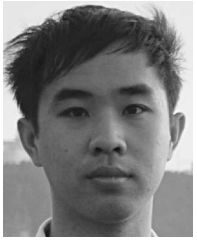
[18] D. Geronimo, A.M. Lopez, A.D. Sappa, and T. Graf, "Survey of Pedestrian Detection for Advanced Driver Assistance Systems," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 32, no. 7, pp. 1239-1258, July 2010.

[19] S. Walk, N. Majer, K. Schindler, and B. Schiele, "New Features and Insights for Pedestrian Detection," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR),* 2010.

[20] P. Dollar, B.C. Wojek, B. Schiele, and P. Perona, "Pedestrian Detection: An Evaluation of the State of the Art," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 34, no. 4, pp. 734-761, Apr. 2012.

[21] X. Wang, X. Ma, and E. Grimson, "Unsupervised Activity Perception in Crowded and Complicated Scenes Using Hierarchical Bayesian Models," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 31, no. 3, pp. 539-555, Mar. 2008.
[22] A. Levin, P. Viola, and Y. Freund, "Unsupervised Improvement of Visual Detectors Using Co-Training," *Proc. Ninth IEEE Int'l Conf. Computer Vision (ICCV),* 2003.
[23] C. Rosenberg, M. Hebert, and H. Schneiderman, "Semi-Supervised Self-Training of Object Detection Models," *Proc. IEEE Workshop Application of Computer Vision,* 2005.
[24] B. Wu and R. Nevatia, "Improving Part Based Object Detection by Unsupervised Online Boosting," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR),* 2007.
[25] P.M. Roth, S. Sternig, H. Grabner, and H. Bischof, "Classifier Grids for Robust Adaptive Object Detection," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR),* 2009.
[26] M. Wang and X. Wang, "Automatic Adaptation of a Generic Pedestrian Detector to a Specific Traffic Scene," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR),* 2011.
[27] M. Wang, W. Li, and X. Wang, "Transferring a Generic Pedestrian Detector Towards Specific Scenes," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR),* 2012.
[28] J. Pang, Q. Huang, S. Yan, S. Jiang, and L. Qin, "Transferring Boosted Detectors towards Viewpoint and Scene Adaptiveness," *IEEE Trans. Image Processing,* vol. 20, no. 5, pp. 1388-1400, May 2011.
[29] V. Nair and J.J. Clark, "An Unsupervised Online Learning Framework for Moving Object Detection," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR),* 2004.
[30] O. Javed, S. Ali, and M. Shah, "Online Detection and Classification of Moving Objects Using Progressively Improving Detectors," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR),* 2005.
[31] P.M. Roth, H. Grabner, D. Skocaj, H. Bishof, and A. Leonardis, "On-Line Conservative Learning for Person Detection," *Proc. IEEE Int'l Workshop Visual Surveillance and Performance Evaluation of Tracking and Surveillance (PETS),* 2005.
[32] N. Dalal, B. Triggs, and C. Schmid, "Human Detection Using Oriented Histogram of Flow and Appearance," *Proc. Ninth European Conf. Computer Vision (ECCV),* 2006.
[33] B. Kulis, K. Saenko, and T. Darrell, "What You Saw Is Not What You Get: Domain Adaptation Using Asymmetric Kernel Transforms," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR),* 2011.
[34] G. Qi, C. Aggarwal, Y. Rui, Q. Tian, S. Chang, and T. Huang, "Towards Cross-Category Knowledge Propagation for lEarning Visual Concepts," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR),* 2011.
[35] J. Liu, M. Shah, B. Kuipers, and S. Savarese, "Cross-View Action Recognition via View Knowledge Transfer," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR),* 2011.
[36] J. Yang, R. Yan, and A.G. Hauptmann, "Cross-Domain Video Concept Detection Using Adaptive SVMs," *Proc. 15th ACM Int'l Conf. Multimedia (Multimedia),* 2007.
[37] G. Qi, C. Aggarwal, and T. Huang, "Towards Semantic Knowledge Propagation from Text Corpus to Web Images," *Proc. 20th Int'l Conf. World Wide Web,* 2011.
[38] L. Duan, I.W. Tsang, D. Xu, and S.J. Maybank, "Domain Transfer SVM for Video Concept Detection," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR),* 2009.
[39] W. Jiang, E. Zavesky, S. Chang, and A. Loui, "Cross-Domain Learning Methods for High-Level Visual Concept Classification," *Proc. 15th IEEE Int'l Conf. Image Processing (ICIP),* 2008.
[40] W. Dai, Q. Yang, and G.R. Xue, "Boosting for Transfer Learning," *Proc. 24th Int'l Conf. Machine Learning (ICML),* 2007.
[41] X. Wu and R. Srihari, "Incorporating Prior Knowledge with Weighted Margin Support Vector Machines," *Proc. 10th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining,* 2004.
[42] S. Stalder and H. Grabner, "Cascaded Confidence Filtering for Improved Tracking-by-Detection," *Proc. 11th European Conf. Computer Vision (ECCV),* 2010.
[43] K. Ali, D. Hasler, and F. Fleuret, "FlowBoost—Appearance Learning from Sparsely Annotated Video," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR),* 2011.
[44] V. Jain and E. Learned-Miller, "Online Domain Adaptation of a Pre-Trained Cascade of Classifiers," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR),* 2011.
[45] P.F. Felzenszwalb, R.B. Girshick, D. McAllester, and D. Ramanan, "Object Detection with Discriminatively Trained Part-Based Models," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 32, no. 9, pp. 1627-1645, Sept. 2010.
[46] D. Comaniciu and P. Meer, "Mean Shift: A Robust Approach Toward Feature Space Analysis," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 24, no. 5, pp. 603-619, May 2002.
[47] D. Hoiem, A. Efros, and M. Hebert, "Putting Objects in Perspective," *Int'l J. Computer Vision,* vol. 80, no. 1, pp. 3-15, Apr. 2008.
[48] C. Tomasi and T. Kanade, "Detection and Tracking of Point Features," Technical Report, School of Computer Science, Carnegie Mellon Univ., Apr. 1991.
[49] A. Mislove, M. Marcon, K. Gummadi, P. Druschel, and B. Bhattacharjee, "Measurement and Analysis of Online Social Networks," *Proc. Seventh ACM SIGCOMM Conf. Internet Measurement,* 2007.
[50] R.E. Fan, K.W. Chang, and C.J. Hsieh, "LIBLINEAR: A Library for Large Linear Classification," *J. Machine Learning Research,* vol. 9, pp. 1871-1874, 2008.
[51] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum Learning," *Proc. Int'l Conf. Machine Learning (ICML),* 2009.
[52] M.P. Kumar, B. Packer, and D. Koller, "Self-Paced Learning for Latent Variable Models," *Proc. Conf. Neural Information Processing Systems (NIPS),* 2010.
[53] H. Daumé III, A. Kumar, and A. Saha, "Frustratingly Easy Semi-Supervised Domain Adaptation," *Proc. Workshop Domain Adaptation for Natural Language Processing,* 2010.
[54] P. Dollár, C. Wojek, and B. Schiele, "Pedestrian Detection: A Benchmark," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR),* 2009.
[55] R. Benenson, M. Mathias, R. Timofte, and L. Van Gool, "Pedestrian Detection at 100 Frames Per Second," *Proc. Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR),* 2012.

Xiaogang Wang received the BS degree in electrical engineering and information science from the Special Class for Gifted Young, University of Science and Technology of China in 2001, the MPhil degree in information engineering from the Chinese University of Hong Kong in 2004, and the PhD degree in computer science from the Massachusetts Institute of Technology. He is currently an assistant professor in the Department of Electronic Engineering at the Chinese University of Hong Kong. He was the area chair of the IEEE International Conference on Computer Vision (ICCV) 2011. He is an associate editor of the *Image and Visual Computing Journal*. His research interests include computer vision and machine learning. He is a member of the IEEE.



Meng Wang received the BEng degree in electronic engineering from the Chinese University of Hong Kong in 2010, where he is currently working toward the MPhil degree in the Department of Electronic Engineering. His research interests include object detection and machine learning. He is a student member of the IEEE.

Wei Li received the BS degree in computer science from Tsinghua University in 2011. He is currently working toward the MPhil degree in the Department of Electronic Engineering at the Chinese University of Hong Kong. His research interests include computer vision and machine learning. He is a student member of the IEEE.