Automatic Adaptation of a Generic Pedestrian Detector to a Specific Traffic Scene

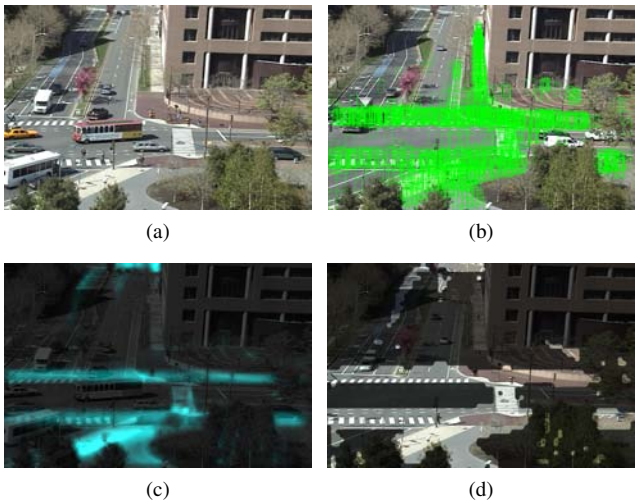
Meng Wang and Xiaogang Wang
{mwang,xgwang}@ee.cuhk.edu.hk
Department of Electronic Engineering, The Chinese University of Hong Kong

Abstract

In recent years significant progress has been made learning generic pedestrian detectors from manually labeled large scale training sets. However, when a generic pedestrian detector is applied to a specific scene where the testing data does not match with the training data because of variations of viewpoints, resolutions, illuminations and backgrounds, its accuracy may decrease greatly. In this paper, we propose a new framework of adapting a pre-trained generic pedestrian detector to a specific traffic scene by automatically selecting both confident positive and negative examples from the target scene to re-train the detector iteratively. An important feature of the proposed framework is to utilize unsupervisedly learned models of vehicle and pedestrian paths, together with multiple other cues such as locations, sizes, appearance and motions to select new training samples. The information of scene structures increases the reliability of selected samples and is complementary to the appearance-based detector. However, it was not well explored in previous studies. In order to further improve the reliability of selected samples, outliers are removed through multiple hierarchical clustering steps. The effectiveness of different cues and clustering steps is evaluated through experiments. The proposed approach significantly improves the accuracy of the generic pedestrian detector and also outperforms the scene specific detector retrained using background subtraction. Its results are comparable with the detector trained using a large number of manually labeled frames from the target scene.

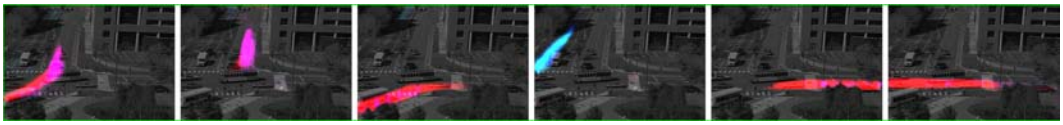
1. Introduction

Detecting pedestrians from video sequences is of great interest in video surveillance in traffic scenes (see an example in Figure 1 (a)). It is useful when analyzing typical and abnormal behaviors of pedestrians, detecting dangerous activities and counting pedestrians along different paths. Many existing works [14, 23] on pedestrian detection in video surveillance were based on background subtraction. However, it is well known that background subtraction is sensitive to lighting variations and scene clusters, and has difficulty in handling the grouping and fragmentation problems [1]. In recent years, appearance-based pedestrian detectors [4, 12, 22, 7, 2] based on large-scale training sets became more and more popular and have achieved great success. There is a huge literature [8] on this topic. However, it is difficult to train a generic appearance-based pedestrian detector which works robustly in different scenes because there is a large diversity of both positive and negative examples and there are also large variations of viewpoints, illuminations, resolutions and backgrounds across different scenes. For example, it was shown that the detection rate of the popular HOG pedestrian detector [4] trained on the

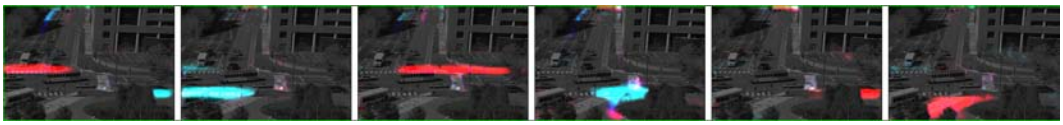


(a) (b) (c) (d)

Figure 1: (a) Typical traffic scene from the MIT traffic data set [20]. (b) Distribution of manually labeled pedestrian bounding boxes from this traffic scene. (c) Spatial distributions of pedestrian paths unsupervised learned using the approach in [20]. (d) Estimated regions of pedestrian paths by thresholding the distribution density in (c).



(a) Some models of vehicle paths.



(b) Some models of pedestrian paths.

Figure 2: Examples of models of vehicle and pedestrian paths learned by [20] from the MIT traffic data set [20]. Each model is a distribution of over locations and moving directions of objects. Colors indicate moving directions: red ($\rightarrow$), cyan ($\leftarrow$), magenta ($\uparrow$) and green ($\downarrow$). The intensity of colors indicates the distribution over space.

INRIA data set dropped significantly when being tested on the Caltech benchmark video data set [6].

In video surveillance most cameras are stationary. If the scene is fixed, the diversity both of positive and negative examples will be significantly reduced. Therefore it is attractive to learn a scene specific detector with a higher accuracy in the target scene than a generic detector. Although higher accuracy can be achieved if the detector is trained using manually labeled examples from the target scene, repeating the manually labeling work for every different scene is costly and this approach is not scalable. A more 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, however without manually labeling. Our work is along this direction. The focus is how to automatically select training examples from the target scene.

1.1. Related Work

Compared with extensive research done on generic object detectors, existing works on scene specific pedestrian detectors are limited. They typically designed a labeler which automatically selected positive and negative examples from the target scene to re-train the generic detector. In order to effectively improve the performance, the training examples selected by the automatic labeler must be reliable and informative to the detector. Semi-supervised self-training was used in [15]. Examples confidently classified by the detector were used to re-train the detector. Since the detector itself was the labeler and not reliable, the selected examples were not informative and likely to have the wrong labels, which made the detector drift. Nair et al. [13] used background subtraction results to label training examples for an appearance-based pedestrian detector. The accuracy of the background subtraction labeler was low and it introduced biased labeling which misled the learning of the detector. For example, static pedestrians might be labeled as non-pedestrian examples. It was unlikely for pedestrians with clothes of a similar color to the background to be labeled as pedestrian examples.

Some automatic labelers were designed under the co-training framework [11, 10, 16, 21]. Two detectors based on different types of features were trained iteratively. The prediction of one detector on unlabeled examples was used to enlarge the training set of the other. For example, Levin et al. [11] built two car detectors using gray images and background subtracted images. They all required manually labeling a small training set from the target scene for initialization. In order for co-training to be effective, the detectors need to be independent, which is difficult to achieve. Dalal et al. [5] showed that the appearance-based and motion-based pedestrian detectors were highly correlated.

When cameras are stationary, the distribution of the negative class is region-specific. Roth et al. [16, 17] introduced classifier grids to train a separate detector for each local region. Stalder et al. [18] used tracking and manually-input scene geometry to assist labeling.

1.2. Our Approach

We focus on traffic scenes, which are more challenging and where moving objects consist mainly of pedestrians and vehicles. Eagle-eye perspective is assumed. The movements of pedestrians and vehicles are regularized by scene structures and they follow certain motion patterns. The models of pedestrian and vehicle paths can increase the reliability of the automatic labeler. It is more reliable to select positive examples on pedestrian paths (see Figure 1 (b)). Because it is rare for vehicles to move on pedestrian paths, knowing that examples on a pedestrian path are either pedestrians or negative examples from the background, the automatic labeling then becomes easier. Negative examples on the background and vehicle paths can also be better selected with the assistance of path models. Because the models of paths are distributions over locations, they are less correlated with appearance and can select more informative examples for re-training. If the locations help to select positive examples on pedestrian paths, after being re-trained,

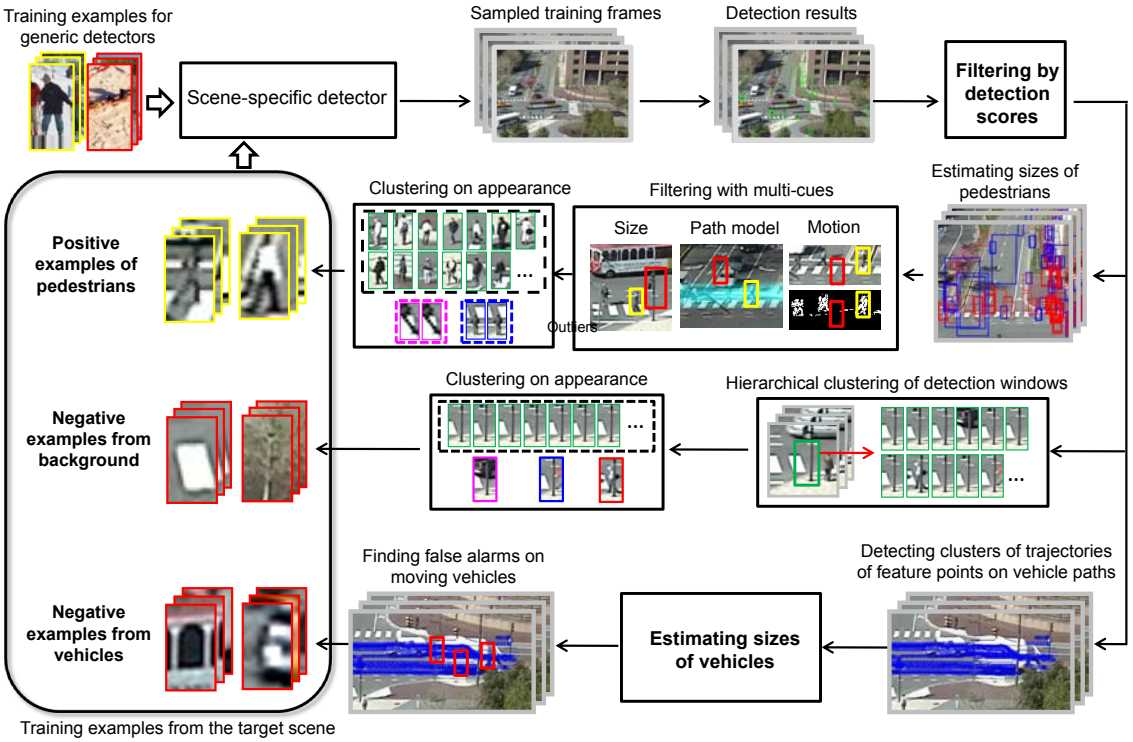


Figure 3: Diagram of our approach.

the detector can detect more pedestrians outside pedestrian paths based on appearance.

However, this information was not well explored in previous works, partially because obtaining the models of scene structures required manual input or reliable detectors and trackers as prerequisites. Manually inputting scene structures is not only costly but also inaccurate. As shown in Figure 2, it is difficult to manually draw the boundaries of paths to accurately match the moving patterns of objects. Some paths cannot be identified from the background image. In our previous work [20] proposed an approach of automatically learning the motion patterns of objects from simple location motions (see examples in Figure 2) was proposed. Benefiting from this outcome, our approach uses the models of pedestrians and vehicles paths learned by [20] to train scene specific pedestrian detector. Other cues such as locations, sizes, appearance and motions are also integrated to select training examples in the target scene. To improve reliability, we remove outliers through multiple clustering steps.

The effectiveness of different cues and clustering steps is evaluated through experiments. The proposed approach significantly improves the accuracy of the generic pedestrian detector and also outperforms the scene specific detector re-trained using the background subtraction. Its results are comparable with the detector trained using manually labeled examples from the target scene.

2. Data Set

We conduct the experimental evaluations on the MIT Traffic data set[1]. It consists of around $162,000$ frames from a 90 minutes long video sequence (30 fps), which was recorded by a stationary camera facing a street intersection. This video includes both pedestrian and vehicle movements with occlusions and varying illumination conditions. We uniformly sample 420 frames from the first half 45 minutes long video to train the scene-specific pedestrian detector and uniformly sample 100 frames from the second half 45 minutes long video to test the performance of the detector after re-training. The bounding boxes of pedestrians in the sampled 520 frames are manually labeled as ground truth and they are plotted in Figure 1 (a). However, they are ***NOT*** used during the training of the scene specific detector.

3. Method

The diagram of our approach is shown in Figure 3. It starts with a generic appearance-based pedestrian detector pre-trained from a common data set and the detector is it-

[1]The data set is available at `http://www.ee.cuhk.edu.hk/~xgwang/MITtraffic.html`. The manually labeled ground truth is also available from this webpage.

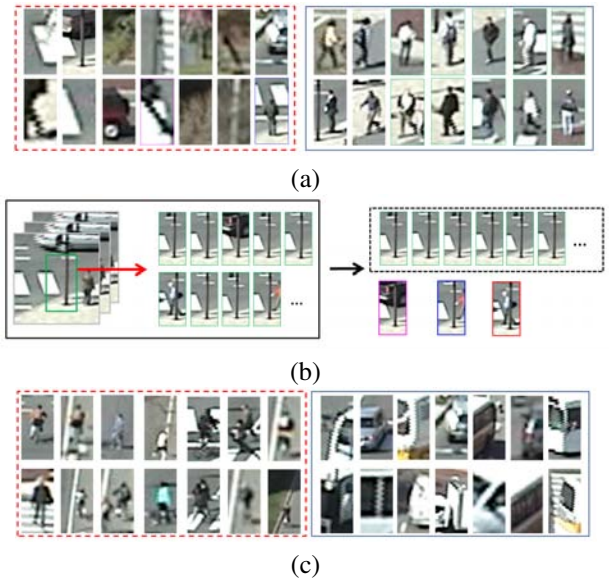


Figure 4: (a) Examples detected by the generic detector, with positive scores and within the regions of pedestrian paths. They include a lot of false alarms (left side) to be purified. See text in Section 3.1. (b) A background false alarm cluster (left side) obtained by clustering on locations and sizes includes a few pedestrians accidentally passing by the same location. They are removed by further clustering on appearance (right side), since background false alarms are also clustered in appearance. See text in Section 3.2. (c) Examples detected on vehicle paths. Some true positives are included (left side).

eratively re-trained. In each round, the detector is applied to the training video frames and three types of examples (confident positive examples of pedestrians, confident negative examples from the background and confident negative examples from vehicles) from the target scene are automatically selected to re-train the detector. We assume that the models of pedestrian and vehicle paths are learned and their regions are segmented using the approach in [20] [2].

To obtain the generic detector, we choose the HOG+SVM pedestrian detector [4], and train it on the INRIA data set. A detection window is denoted by $(x, y, 0.5s, s)$, where x and y are the coordinates of the center of the detection window, $0.5s$ and s are the width and the height. The HOG feature associated with a detection window is denoted as $\mathbf{f}_{x,y,s}$. The output of the linear SVM classifier takes the form,

$$\text{score} = \mathbf{a} \cdot \mathbf{f}_{x,y,s} + a_0, \tag{1}$$

where $\mathbf{a}$ and a_0 are the weights and bias learned by SVM. Since ours is a general framework, other generic pedestrian detectors and training sets can also be used. Normally, the back-end of a detector clusters detection windows based on their sizes and locations, yielding merged windows at the final result. Instead, we select training examples from unmerged windows and this leads to a more robust scene specific detector. The details of automatically selecting training examples are given in the following sub-sections.

[2] Given the output of [20], the user needs to label a path model to be a pedestrian path or a vehicle path. However, this workload is light.

3.1. Confident Positive Examples of Pedestrians

The sampled video frames are scanned with the pedestrian detector at multiple scales. Since it is more likely for pedestrians to appear on pedestrian paths, in order to obtain confident positive examples, we only consider detection windows, which fall in the regions pedestrian paths (as shown in Figure 1d) and whose scores given by Eq. (1) are positive, as candidates. As shown in Figure 4 (a), these candidates include a lot of negative examples to be purified in the further steps.

Estimating sizes of pedestrians. In order to estimate the size range of pedestrians in the target scene, we construct the histograms of the sizes of the detected windows. The mode $\overline{s}$ of the histogram is selected by mean shift [3] as the mean of the pedestrian sizes and the variance (σ) of the mode is also estimated. Pedestrians appear in different sizes in the scene because of perspective distortion. Their size variation is modeled as a single global Gaussian distribution $G(\overline{s}, \sigma)$ in our approach and this model will be integrated with other cues in a probabilistic way as described later. The size variation could be better modeled through estimating the perspective transformation of the scene [9] or estimating different Gaussian distributions in different local regions.

Hierarchical clustering of detection windows. In a traffic scene, it is uncommon for pedestrians to stay at the same location for a long time. On the other hand, if a background patch is misclassified as a pedestrian, similar patterns tend to repeatedly appear at the same location and be misclassified over a long period. Through hierarchical clustering illustrated in Figure 5, we find such examples and exclude them from selected confident positive examples, since they are more likely to be false alarms on the background. As shown in Figure 5, the hierarchical clustering on the locations and sizes of detection windows has two stages, clustering within a single frame and clustering across frames. Clustering within a single frame is similar to window merging commonly used in sliding-window based detection [4]. A sliding-window based detector usually gives multiple detections around the location of one pedestrian. Mean shift based on locations and sizes of windows (x, y, s) is used to cluster these windows and merge them into one window $(x_{\text{m}}, y_{\text{m}}, s_{\text{m}})$. The bandwidth is chosen as $\bar{s}/3$, which is 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 many frames (e.g.

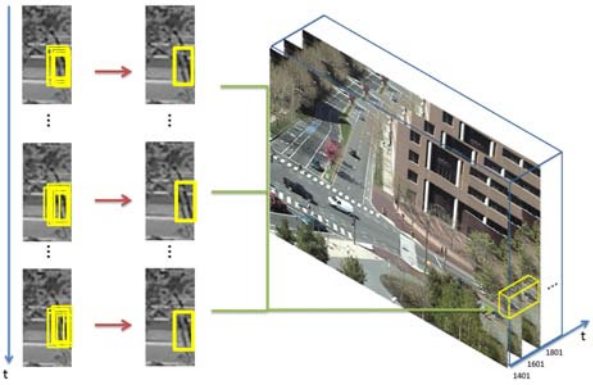


Figure 5: Hierarchical clustering of detection windows.

longer than 3 minutes in our implementation) are removed from confident positive examples and selected as candidates of confident negative examples from the background. Note that they are not necessarily negative examples and will be further processed in Section 3.2.

Filtering with Multi-cues. Confident positive examples of pedestrians 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. n is the number of moving pixels in the window and N is the total number of pixels in the window. Then the log likelihood of this detected window being a pedestrian is given by the joint probability,

$$L_{\mathrm{p}}(z) = \log p_{\mathrm{s}}(s|\bar{s}, \sigma) + \log p_{\ell}((x, y, s)|\phi_k) + \log p_{\mathrm{m}}(n, N). \quad (2)$$

p_{s} models the pedestrian sizes as a Gaussian distribution and therefore,

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

$\log p_{\ell}((x, y, s)|\phi_k)$ is the log likelihood based on the models of pedestrian paths. Suppose the detection window contains N pixels whose locations 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 where the window is detected. Then,

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

A detection window on a pedestrian often contains more moving pixels than that on the background. $\log p_m(n, N)$ is the log likelihood based on the motion cue,

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

Moving pixels are detected in a simple way. Suppose the current frame is I_t. Two reference frames I_{t-50} and I_{t+50} 50 frames before and after the current frame 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.

Similar to other self-training [15, 13] or co-training [11, 10, 16, 21] frameworks, the confident positive examples are found by thresholding $L_{\mathrm{p}}(z) > L_0$. The larger the threshold is, the more conservative the strategy of selecting examples is. In our approach, L_0 can be decided by interpreting the probabilistic meanings of the three terms in Eq. (2). For example, in our experiments, L_0 is chosen as

$$L_0 = \log p_{\mathrm{s}}(\bar{s}+\sigma/2|\bar{s}, \sigma)+\log 0.75 \max(\{\phi_{kw}\})+\log 0.2.$$

Clustering on Appearance. The remaining examples after thresholding include a small portion of outliers from the background and vehicles. These outliers are removed by clustering the HOG features $\mathbf{f}_{x,y,s}$ by mean shift. Examples on pedestrians, vehicles and the background form different clusters on appearance and pedestrians take majority in the remaining examples. The bandwidth for mean shift is automatically decided by the criterion that 90% (in our experiments, this threshold in the range of $70\% - 95\%$ leads to satisfactory results) of examples fall into one cluster. Examples in this cluster are selected as confident positive examples of pedestrians.

3.2. Confident Negative Examples from the Background

In order to automatically select confident negative examples, we only consider detection windows whose scores satisfy $0 <$ score < 0.5 as candidates. These examples are misclassified by the detector and close to the decision boundary. They are informative to the detector and are also known as hard examples in literature [4, 7]. As explained in Section 3.1, false alarms on the background tends to repeat over time at the same location with similar appearance patterns. Therefore, their examples 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 a long period are selected as negative examples. However, as shown in Figure 4 (b), they may include a small number of pedestrians who accidentally pass by the same locations. To remove these positive examples, examples within each cluster are further clustered using mean shift on HOG features. Again, 90% examples are kept by automatically adjusting the bandwidth.

3.3. Confident Negative Examples from Vehicles

It is unreliable to directly count windows detected on vehicle paths as negative examples, since some pedestrians

and bicycles also move on the vehicle paths (some examples are shown in Figure 4 (c)). In order to select confident negative examples from moving vehicles, the existence of moving vehicles need to be first detected. This is achieved by feature point tracking and clustering. Corner feature points in the scene are detected and tracked using the KLT tracker [19]. Stationary points and short trajectories are removed. Then trajectories are clustered based on their temporal and spatial proximity by mean shift. Each trajectory cluster is assigned to one of the vehicle paths 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 estimating pedestrian size in Section 3.1. The trajectory clusters of pedestrians on vehicle paths are removed using the size evidence. If a 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 example on a moving vehicle.

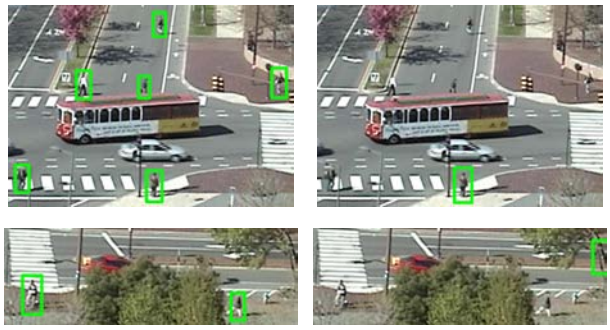
3.4. Final Scene Specific Pedestrian Detector

Once the scene specific pedestrian detector has been well trained on the sampled video frames, it will be used to detect pedestrians in new frames purely based on appearance without the assistance of other cues. Although the multiple cues discussed above are effective on selecting training examples, they cannot guarantee high detection accuracy[4]. For example, if the detector relies on path models, pedestrians walking on the vehicle paths may be missed (these pedestrians are of great interest in video surveillance). Replying on motions and sizes, some stationary pedestrians and small pedestrians may be missed. The final detector gives multiple detection windows around the location of a pedestrian. The windows are merged to give the final result.

4. Experiment Results

Experiments are conducted on the MIT traffic data set described in Section 2. We adopt the PASCAL criterion [6] that a detection is correct if the ratio between the intersection and the union is larger than 0.5 comparing the detection window and the ground truth. The ROC curve is used as the evaluation metric. We have particular interest in the detection rates when the false alarm rate (FAR) is 10^{-6}, as it corresponds approximately to 1 false alarm per frame on this data set. This comparison result is shown in Table 1.

1) Overall performance. Figure 7 (a) and (b) plot the ROC curves of the initial generic detector and our scene-specific detector after different rounds of re-training. They are evaluated on both the training set and the testing set[5]. The performance of the generic detector is low on our data set and achieves a detection rate of only 21% at FAR $= 10^{-6}$. This result is consistent with the observation in [6].Our scene specific detector converges[6] after 10 rounds of automatic re-training. It greatly enhances the detection rate to 61% (on the training set) and 62% (on the testing set) at FAR $= 10^{-6}$. An example is shown in Figure 6.



(a) (b)

Figure 6: Detection examples. Our scene-specific detector (a) significantly enhances the detection rate of the generic detector (b).

2) Comparison with the automatic labeler based on background subtraction. We compare with a scene-specific detector re-trained using the automatic labeler based on background subtraction. Positive and negative examples for re-training are automatically selected according to the number of foreground pixels within the detection windows. A similar strategy was used in [13]. [13] used Haar features + Boosting. To make the comparison consistent, we still use SVM + HOG. Its best ROC curve (after the 5-th round of re-training) on the testing set is plotted in 7 (c). This scene-specific detector does not perform as well as ours because many false alarms on moving vehicles are selected as positive examples and stationary pedestrians are selected as negative examples. Starting from the 7-th round of re-training, the detector drifts and its performance dramatically deteriorates.

3) Comparison with the scene-specific detector trained using manually labeled examples from the target scene. We also train detectors using different numbers of manually labeled frames in the training set. These detectors are bootstrapped according to the strategy in [4][7]. Their ROC curves on the testing set are plotted in Figure 7 (d). Intuitively, the detector trained using all the manually labeled

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

[4]The purpose of using these cues is to find some confident examples without introducing bias on appearance but not all the examples.

[5]The ROC curves on the testing set are slightly higher than on the training set, because the sampled training frames are more difficult.

[6]Convergence means that the performance of the detector does not improve significantly anymore.

[7]After the first round of training, the detector is re-trained by adding more hard negative examples found in the frames.

GE	**Ours**	SS_B	SS_M(420)
0.21	**0.62**	0.43	0.66
SS_M(300)	SS_M(150)	SS_M(100)	SS_M(50)
0.62	0.52	0.45	0.42

Table 1: Detection rates of different detectors when FAR $= 10^{-6}$. **GE**: generic detector; **Ours**: our scene-specific detector; **SS_B**: scene-specific detector using background subtraction as the automatic labeler; **SS_M**(n): scene-specific detector trained using n manually labeled frames.

frames from the target scene is the best one we can get. The ROC curve of our detector is slightly lower than that of this best one. But it performs better than the detectors trained using $50 \sim 300$ manually labeled frames. As shown in Table 1, the detection rate of our detector is the same as the one trained using 300 manually labeled frames when FAR $= 10^{-6}$.

4) Effectiveness of different cues for selecting confident positive examples. As shown in Eq (2), the cues of detection window sizes, models of pedestrian paths and motions are integrated to select confident positive examples. Figure 7 (e) plots the ROC curves of removing each of the three cues separately. It shows that the models of pedestrian paths are most effective. Removing this cue during re-training, the detection rate of the final scene specific detector will significantly decrease by 17% at FAR $= 10^{-6}$.

5) Effectiveness of removing outliers through clustering on appearance. In both Section 3.1 and Section 3.2, some outliers are removed through clustering on appearance. Our experiments show that this is crucial for the convergence of the scene specific detector. Without these clustering steps, the detector drifts after several rounds of re-training due to the massive inclusion of training examples with wrong labels. In our approach, mean shift is used for clustering and its bandwidth is automatically selected to reject 10% examples as outliers. In practice, we find that this threshold (the mean shift rejection rate) is highly configurable and can be set as $5\% \sim 30\%$. The ROC curves of choosing different thresholds are shown in Figure 7 (f). We recommend a relatively high threshold, since it reduces the risk of detector drifting, although it may result in more rounds of re-training to converge.

5. Conclusions and Discussions

We propose a novel framework of adapting a generic pedestrian detector to a specific traffic scene by automatically selecting confident positive and negative examples from the target scene. It integrates the models of pedestrian and vehicle paths with other cues to make the selected examples informative and reliable. Experiment on the MIT Traffic data set shows that our approach significantly improves the detection rate from 21% to 62% given FAR $= 10^{-6}$ compared with the generic detector. It even outperforms the scene-specific detector trained directly using fewer than 300 manually labeled frames.

Our approach only has two parameters to be set empirically: L_0 described in Section 3.1 and the mean shift rejection rate in Figure 7 (f). They controls how aggressive the automatic training process is. Similar parameters also exist in other approaches of automatically training scene specific detectors [10, 11, 12, 13, 14, 15, 16]. Our approach has robustness to these parameters within certain range. It is also possible to tune the two parameters using a few manually labeled frames. As shown in Figure 7 (d), given a small number of labeled frames, our approach greatly outperforms the approach of directly using these labels to train the detector.

6. Acknowledgement

This work is supported by the General Research Fund sponsored by the Research Grants Council of Hong Kong (porject No. CUHK417110).

References

[1] B. B., X. Wang, and W. Grimson. Multi-class Object Tracking Algorithm that Handles Fragmentation and Grouping. 2007.

[2] L. Bourdev, S. Maji, T. Brox, and J. Malik. Detecting people using mutually consistent poselet activations. In *Proc. ECCV*, 2010.

[3] D. Comaniciu and P. Meer. Mean shift: a robust approach toward feature space analysis. *IEEE Trans. on PAMI*, 24:603–619, 2002.

[4] N. Dalal and B. Triggs. Histogram of oriented gradients for human detection. In *Proc. CVPR*, 2005.

[5] N. Dalal, B. Triggs, and C. Schmid. Human detection using oriented histogram of flow and appearance. In *Proc. ECCV*, 2006.

[6] P. Dollar, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: A benchmark. In *Proc. CVPR*, 2009.

[7] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *Proc. CVPR*, 2008.

[8] D. Geronimo, A. M. Lopez, A. D. Sappa, and T. Graf. Survey of pedestrian detection for advanced driver assistance systems. *IEEE Trans. on PAMI*, 32:1239–1258, 2010.

[9] D. Hoiem, A. a. Efros, and M. Hebert. Putting Objects in Perspective. *International Journal of Computer Vision*, 80(1):3–15, Apr. 2008.

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

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

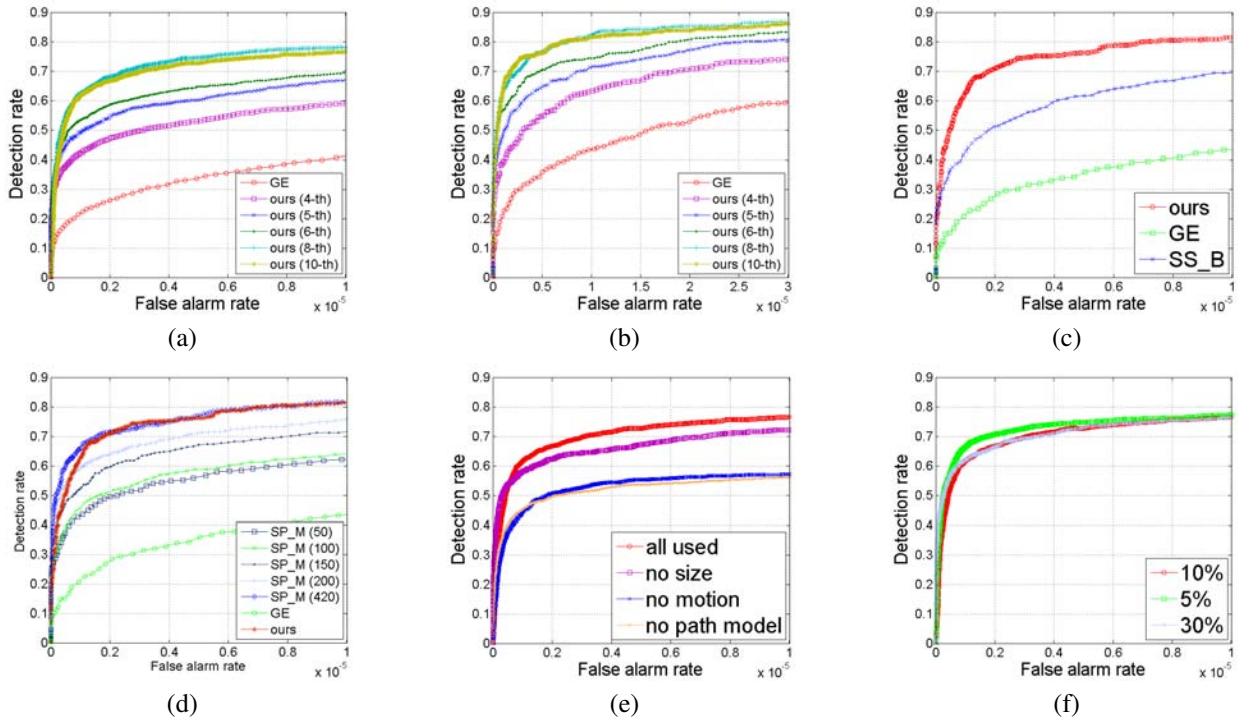


Figure 7: ROC curves of different detectors. (a)-(b): the initial generic detector (**GE**) and our scene-specific detector (**Ours**) after different numbers of rounds of retraining. (a) is on the training set and (b) is on the testing set. (c) Our final detector after the 10-th round of re-training and the scene-specific detector using the background subtraction as the automatic labler (**SS_B**) after the 5-th round of re-training. SS_B achieves its best performance after the 5-th round of re-training. It starts to drift after the 7-th round of re-training. Evaluation is on the testing set. (d) Scene-specific detectors (**SS_M**(n)) trained using different numbers (n) of manually labeled frames. Evaluation is on the testing set. (e): our detectors obtained when removing one of the cues (size, motion and path model) during the selection of confident positive examples. (f): our detectors obtained when rejecting different percentages of examples as outliers for mean shift clustering on appearance. Both (e) and (f) are evaluated on the training set, since they directly reflects the effectiveness of the automatic labeler, whose task is to make the detector well fit the training data unsupervisedly.

[12] Z. Lin, L. Davis, D. Doermann, and D. Dementhon. Hierarchical part-template matching for human detection and segmentation. In *Proc. ICCV*, 2007.

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

[14] C. Nakajima, M. Pontil, B. Heisele, and T. Poggio. Full-body recognition system. *Pattern Recognition*, 36:1977–2006, 2003.

[15] C. Rosenberg, M. Hebert, and H. Schneiderman. Semi-supervised self-training of object detection models. In *Proc. of IEEE Workshop on Application of Computer Vision*, 2005.

[16] P. Roth, H. Grabner, D. Skocaj, H. Bishof, and A. Leonardis. On-line conservative learning for person detection. In *Proc. IEEE Int'l Workshop on PETS*, 2005.

[17] P. Roth, S. Sterning, H. Grabner, and H. Bishof. Classifier grids for robust adaptive object detection. In *Proc. CVPR*, 2009.

[18] S. Stalder, H. Grabner, and L. van Gool. Exploring context to learn scene specific object detectors. In *Proc. IEEE Int'l Workshop on PETS*, 2009.

[19] C. Tomasi and T. Kanade. Detection and tracking of point features. *Int'l Journal of Computer Vision*, April 1991.

[20] X. Wang, X. Ma, and W. Grimson. Unsupervised activity perception in crowded and complicated scenes using hierarchical bayesian models. *IEEE Trans. on PAMI*, 31:539–555, 2009.

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

[22] L. Zhang, B. Wu, and R. Nevatia. Detection and tracking of multiple humans with extensive pose articulation. In *Proc. ICCV*, 2007.

[23] T. Zhao, R. Nevatia, and B. Wu. Segmentation and tracking of multiple humans in crowded environments. *IEEE Trans. on PAMI*, 30:1198–1211, 2008.