INTERACTIVE LEARNING A PERSON DETECTOR: FEWER CLICKS – LESS FRUSTRATION[1]

Peter M. Roth[2], Helmut Grabner[2], Christian Leistner[2], Martin Winter[2], and Horst Bischof[2]

Abstract

To train a general person detector a huge amount of training samples is required to cope with the variability in the persons' appearance and all possible backgrounds. Since this data is often not available we propose an interactive learning system, that enables an efficient training of a scene specific person detector. For that purpose we apply a two stage approach. First, a general detector is trained autonomously from labeled data. Later on this detector is improved and adapted to a specific scene by user interaction. Thus, only highly valuable samples are selected and only a small number of updates is necessary to adapt to a specific scene. In particular, for learning we apply off-line boosting in the first stage and on-line boosting in the second stage. Since we use the same underlying representation for both methods, we can efficiently re-train an existing classifier. In the experiments we applied the proposed approach for different scenarios and showed that the detection results (recall and accuracy) can be significantly improved by hand-labeling only a few novel samples.

1. Introduction

Due to the increasing number of cameras mounted for security reasons, automatic visual surveillance systems are required to analyze the upcoming data. One important task for such automatic systems is the detection of persons. Hence, there was a considerable interest on this topic and several approaches have been proposed to solve this problem.

Early approaches used change detection (motion detection) to find (moving) persons. Therefore, a background model was estimated and pixels, that could not be described by the background model were reported as foreground pixels. These pixels were grouped into blobs and the actual detection was performed based on blob analysis (e.g., [18]). However, these approaches have several limitations (e.g., varying backgrounds, crowds of persons, etc.) and can thus not be applied to more complex scenarios. To cope with these problems several approaches based on modeling the appearance of the object have been proposed. These methods can be sub-divided into three main groups according to the way they describe persons. For the first group of methods global image features such as edge templates, shape features or Implicit Shape Models (e.g., [8]) are used to describe a person's

[1]This work was supported by the FFG project AUTOVISTA (813395) under the FIT-IT program, the FFG project EVis (813399) under the FIT-IT program, and the Austrian Joint Research Project Cognitive Vision under projects S9103-N04 and S9104-N04. In addition, the work was partially funded by the Biometrics Center of Siemens IT Solutions and Services, Siemens Austria.

[2]Institute for Computer Graphics and Vision, Graz University of Technology, Austria
{pmroth, hgrabner, leistner, winter, bischof}@icg.tugraz.at

shape. The second group is focused on local image features such as Haar-wavelets [23], Histogram of Gradients [1], and local covariance matrices [21] to learn the appearance. In contrast, methods of the third group try to represent humans by their (articulated) parts (e.g., [24]).

The aim of all of these methods is to train a general person model, which should be applicable to different scenarios and tasks. Several approaches based on machine learning algorithms have been proposed. For that purpose, a classifier is build using a learning algorithm (e.g., AdaBoost [2]), which is subsequently applied using a sliding window technique on all possible sub-windows of a given image. For all of these methods a large training set is required, that captures all variability of persons and backgrounds. But even if general classifiers are trained from a very large number of training samples they often fail in practice. Moreover, from empirical studies (e.g., [10]) it can be seen that acceptable recall rates are only obtained, if the number of allowed false positives is very high. Thus, the main limitation of such approaches is that a representative dataset is needed for training. But not all variability, especially for the negative class (i.e., possible backgrounds) can be captured during training resulting in a low recall and an insufficient precision. To overcome these problems specific classifiers can be applied, which are designed to solve a specific task (e.g., object detection for a specific setup). Furthermore, adaptive classifiers using an on-line learning algorithm can be applied (e.g., [11, 15, 25]). Thus, the system can adapt to changing environments (e.g., changing illumination conditions) and the variations need not to be handled by the overall model. In fact, in this way the complexity of the problem is reduced and a more efficient classifier can be trained.

Unfortunately, unsupervised on-line learning methods tend to incorporate wrong updates, which reduces the performance of the detector. The detector might start to drift and finally ends up in an unreliable state [7]. In order to avoid drifting and to ensure a representative dataset Grabner et. al. [6] proposed an interactive training method for learning a scene specific (car) detector. They start with an empty classifier having zero knowledge and adapt it to a specific scene by taking into account scene-specific samples, which are labeled by a human operator. Thus, only very informative labels (near the current decision boundary) are taken into account and the label noise is minimized. However, in order to get a well performing classifier a huge number of reliable updates (i.e., hand-labeled samples) is needed.

The main goal in this work is to significantly reduce the human labeling effort when training such a classifier. Therefore, we propose to train a general seed classifier by off-line boosting first, which is later improved and adapted to a specific scene using on-line boosting. In contrast to existing scene adaption methods, that start from a general model, we apply an on-line learning method, which allows a direct user interaction. However, we enforce that the off-line and on-line boosting stages share the same statistical representation. Hence, the classifier trained off-line can directly be re-trained on-line. This tremendously reduces the amount of necessary human interaction.

The outline of the remaining paper is as follows: First, in Section 2. we summarize off-line and on-line boosting for feature selection. Next, in Section 3. we introduce the interactive training framework and discuss the knowledge transfer between different classifiers. Experimental evaluations are given in Section 4. Finally, we give a conclusion and an outlook in Section 5.

2. Off-line and On-line Boosting for Feature Selection

2.1. Off-line Boosting for Feature Selection

Boosting, in general, is a widely used technique in machine learning for improving the accuracy of any given learning algorithm (see [3] and [16] for good introduction and overview). In fact, boosting converts a set of weak learning algorithms into a strong one. In this work, we focus on the (discrete) AdaBoost algorithm, which has been introduced by Freund and Shapire [2]. The algorithm can be summarized as follows: given a training set $\mathcal{X} = \{\langle \mathbf{x}_1, y_1 \rangle, ..., \langle \mathbf{x}_L, y_L \rangle \mid \mathbf{x}_i \in \mathbf{R}^m,\ y_i \in \{-1, +1\}\}$ of L samples, where $\mathbf{x}_i$ is a sample and y_i is its corresponding positive or negative label, and a weight distribution $p(\mathbf{x})$, that is initialized uniformly distributed: $p(\mathbf{x}_i) = \frac{1}{L}$. Then, a weak classifier h is trained using $\mathcal{X}$ and $p(\mathbf{x})$, which has to perform only slightly better than random guessing (i.e., the error rate of a classifier for a binary decision task must be less than 50%). Depending on the error e of the weak classifier, a weight α is calculated and the samples' probability $p(\mathbf{x})$ is updated. For misclassified samples the corresponding weight is increased while for correctly classified samples the weight is decreased. Thus, the algorithm focuses on the hard examples. The whole process is iteratively repeated and a new weak classifier is added at each boosting iteration until a certain stopping criterion is met. Finally, a strong classifier $H_{off}(\mathbf{x})$ is estimated by a linear combination of all N trained weak classifiers:

$$H_{off}(\mathbf{x}) \quad = \quad \text{sign}\left(\sum_{n=1}^{N} \alpha_n h_n(\mathbf{x})\right) \ . \tag{1}$$

Furthermore, boosting can be applied for feature selection [19]. The basic idea is that each feature corresponds to a weak classifier and that boosting selects an informative subset from these features. Thus, given a set of k possible features $\mathcal{F} = \{f_1, ..., f_k\}$ in each iteration n a weak hypothesis is built from the weighted training samples. The best one forms the weak hypothesis h_n, that corresponds to the selected feature f_n. The weights of the training samples are updated with respect to the error of the chosen hypothesis. In fact, various different feature types may be applied, but similar to the seminal work of Viola and Jones [22] in this work we use Haar-like features, which can be calculated efficiently using integral data-structures.

2.2. On-line Boosting for Feature Selection

Contrary to off-line methods, during on-line learning each training sample is provided only once to the learner. Thus, all steps have to be on-line and the weak classifiers have to be updated whenever a new training sample is available. On-line updating the weak classifiers is not a problem since various on-line learning methods exist, that may be used for generating hypotheses. The same applies for the voting weights α_n, which can easily be computed if the errors of the weak classifiers are known. The crucial step is the computation of the weight distribution since the difficulty of a sample is not known a priori. To overcome this problem Oza et al. [12, 13] proposed to compute the importance λ of a sample by propagating it through the set of weak classifiers. In fact, λ is increased proportional to the error e of the weak classifier if the sample is misclassified and decreased otherwise.

Since the approach of Oza can not directly be used for feature selection Grabner and Bischof [4] introduced *selectors* and performed on-line boosting on these selectors and not directly on the weak classifiers. A selector $h_n^{sel}(\mathbf{x})$ can be considered a set of M weak classifiers $\{h_1(\mathbf{x}), \ldots, h_M(\mathbf{x})\}$, that

are related to a subset of features $\mathcal{F}_n = \{f_1, \ldots, f_M\} \in \mathcal{F}$, where $\mathcal{F}$ is the full feature pool. At each time the selector $h_n^{sel}(\mathbf{x})$ selects the best weak hypothesis

$$h^{sel}(\mathbf{x}) = \arg\min_m e\left(h_m^{weak}(\mathbf{x})\right) \tag{2}$$

according to the estimated training error

$$\hat{e} = \frac{\lambda_{wrong}}{\lambda_{wrong} + \lambda_{corr}}, \tag{3}$$

where λ_{corr} and λ_{wrong} are the importance weights of the samples seen so far, that were classified correctly and incorrectly, respectively. The work-flow of the on-line boosting for feature selection can be described as follows: A fixed number of N selectors $h_1^{sel}, .., h_N^{sel}$ is initialized with random features. The selectors are updated whenever as a new training sample $\langle \mathbf{x}, y \rangle$ is available and the weak classifier with the smallest estimated error is selected. Finally, the weight α_n of the n-th selector h_n^{sel} is updated and the importance λ_n is passed to the next selector h_{n+1}^{sel} and a strong classifier is computed by a linear combination of N selectors:

$$H_{on}(\mathbf{x}) = \text{sign}\left(\sum_{n=1}^{N} \alpha_n h_n^{sel}(\mathbf{x})\right). \tag{4}$$

Contrary to the off-line version, an on-line classifier is available at any time of the training process.

3. Learning Framework

When learning a classifier the samples are usually drawn randomly from a fixed set (*i.i.d.*). The set represents the underlying distributions of positive and negative samples. Hence, a great number of samples is needed. Such a sampling strategy, which is often referred to as *passive learning* [9], would result in a slow convergency.

3.1. InterActive Learning

To overcome these problems, an adaptive learning algorithm, taking advantage of the ideas of *active learning* (e.g., [9,20,26]), can be applied. In general, an active learner can be considered as a quintuple $(C, Q, S, \mathbf{L}, \mathbf{U})$ [9], where C is a classifier, Q is a query function, S is a supervisor (teacher), and $\mathbf{L}$ and $\mathbf{U}$ are a set of labeled and unlabeled data, respectively. First, an initial classifier C_0 is trained from the labeled set $\mathbf{L}$. Given a classifier C_{t-1}, the query function Q selects the most informative unlabeled samples from $\mathbf{U}$ and the supervisor S is requested to label them. Using the thus labeled samples the current classifier is re-trained obtaining a new classifier C_t. This procedure is summarized in Algorithm 1.

When considering an adaptive system we can start from a small set of labeled data $\mathbf{L}$. But usually the unlabeled data $\mathbf{U}$ is not available in advance. In our case, when learning a person detector, we can define all patches in the processed input images as unlabeled data $\mathbf{U}$. Thus, the first crucial point is to define a suitable query function Q, which selects the most valuable samples. It has been shown [14] that it is more effective to sample the current estimate of the decision boundary than the unknown true boundary. Therefore, the most valuable samples are exactly those, that were misclassified by the current classifier. Hence, the algorithm is focused on the hard samples and the number of required training samples can be considerably reduced. Considering the person detection task the misclassified samples are the detected false positives and the missed true positives.

Algorithm 1 Active Learning

Input: unlabeled samples **U**, classifier C_{t-1}
Output: classifier C_t
1: **while** teacher S can label samples u_j **do**
2: Apply C_{t-1} to all samples u_j
3: Let Q find the m most informative samples u_q
4: Let teacher S assign labels y_q to samples u_q
5: Re-train classifier: C_t
6: **end while**

The key idea of this paper is that a human operator, who is supported by the system, undertakes the task of the query function Q and the teacher S. Thus, it can be assumed that only valuable samples are selected and that (almost) all labels are correct. This assures a fast convergency, i.e., only a small number of updates is necessary. Consequently this reduces the human effort. In practice, the current classifier C_{t-1} is applied on the current image and the detection results are displayed by the system. Based on this output the human operator labels the informative samples. In fact, these are the reported false positives and the missed true positives. This fully supervised and interactive process is iterated until the desired performance (i.e., recall and accuracy) is reached. The interactive training process is summarized more formally in Algorithm 2.

Algorithm 2 InterActive Learning

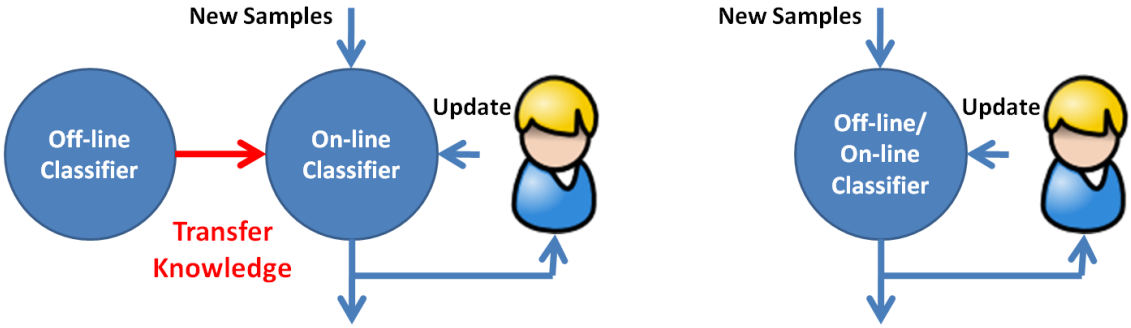
1: Initialize classifier C_0
2: **while** non-stop-criteria **do**
3: Evaluate current classifier C_{t-1} and display results
4: Manually label "good" samples u_q (positive and negative)
5: Re-train classifier: C_t
6: **end while**

3.2. Including Prior Knowledge

To reduce the human effort, i.e., the number of manually labeled samples, we investigate how we can benefit from the incorporation of prior knowledge when learning an initial classifier. Assuming that the prior knowledge is given as an off-line classifier, the goal is to transfer this knowledge to an on-line classifier, which can then be improved. This is depicted in Figure 3.2.

Classifier Transfer

The simplest way to make use of prior knowledge is to transfer pre-learned knowledge (prior) via labels. Precisely, the on-line classifier is updated with samples $\mathbf{x}$ of the novel scene, where the labels are provided by the off-line classifier: $\langle \mathbf{x}, H_{off}(\mathbf{x}) \rangle$. More sophisticated ways of knowledge transfer may be applied. For instance similar to [17] we can transfer the information from an off-line classifier H_{off} to an on-line classifier H_{on} by applying H_{off} as first weak classifier. In fact, the succeeding weak classifiers in the on-line ensemble compensate the errors of the prior off-line classifier.



(a) knowledge transfer (b) our approach: direct retraining

Figure 1. Knowledge transfer: (a) a new on-line classifier is build using the information, that is already captured by an off-line classifier; (b) proposed approach – an off-line trained classifier is directly re-trained.

As a drawback, if the off-line classifier has a high error (i.e., the two distributions are very "different") the complexity of the on-line classifier has to be large . This, however, yields to the common problem that many training samples have to be provided again.

Direct Re-training

To avoid the drawbacks of the previously discussed methods for knowledge transfer and to further reduce the human effort for re-training, in this work, we propose to directly update the off-line trained classifier in an on-line manner. For that purpose, we have to ensure that all statistics, that are necessary for on-line updating, are stored during the off-line training phase. This can be done straightforwardly for all components:

Weak classifier: In order to build a weak hypothesis $h_n : \mathcal{X} \rightarrow \{-1, +1\}$ corresponding to an image feature f_n we apply a learning algorithm. Precisely, we estimate the distributions $P(y = 1|f_n(\mathbf{x}))$ and $P(y = -1|f_n(\mathbf{x}))$ for positive and negative samples, respectively, and apply a Bayesian decision rule to estimate h_n. Assuming that positive and negative feature values follow Gaussian distributions, we can calculate the mean and the variance from all off-line samples. These parameters can then easily be adjusted during the on-line learning stage [5].

Errors: The error of the weak classifier is used to select the best weak classifier within a selector to calculate the voting weight α and to update the importance λ. In the off-line case the error depends on the weights p_i of the training samples, that were classified correctly and incorrectly. These values have to be saved as λ_{corr} and λ_{wrong}, which can then be updated by the importance λ in the on-line case. Thus, by using Eq. (3) the estimated error can be re-calculated.

These modifications of the off-line learning process allow us to on-line re-train an off-line trained classifier later on. Thus, we can retain the information captured during the off-line training and we can still adapt an existing classifier to a new specific scene.

4. Experimental Evaluation

The purpose of the experiments is twofold. First, we want to show that using the proposed interactive learning strategy efficiently a person detector can be trained and, second, that by including prior knowledge, i.e., by using a pre-trained classifier, the learning process can be speeded up. In fact, we show that an existing general classifier can be adapted to a specific scene. To illustrate this we first trained a general off-line classifier using publicly available hand-labeled samples and then re-trained this classifier for specific scenes later on. For evaluation purposes we compared the detection results using the overlap-criterion with the defined ground-truths (required overlap 50%), where we used *precision*, *recall*, and *F-measure* (see e.g., [7]) as evaluation criteria. To allow a fair comparison we used the same complexity for all classifiers (including the one trained off-line). In particular, we applied 50 selectors, each holding a set of 250 weak classifiers corresponding to a feature.

4.1. Benchmark Data

The *CoffeeCam* dataset shows a corridor in a public building near to a coffee dispenser. Hence, we could capture various representative real-world scenarios: walking people, people standing around while waiting for their coffee, or people building small crowds while drinking coffee. For evaluation purposes we generated a training sequence containing 1200 frames and a challenging independent test set (containing groups of persons, persons partially occluding each other, and persons walking in different directions) and a corresponding ground-truth. In total the test sequence consists of 300 frames and contains 224 persons.

The *Caviar*[2] dataset contains sequences showing a corridor in a shopping mall from two different views. For our experiments we have selected and adapted one typical (more complex) sequence showing the frontal view (*ShopAssistant2cor*). The frame-rate of the original sequence was reduced and the images were converted to gray-scale. In total the test sequence consists of 144 frames and contains 364 persons. The provided ground-truth, which also contains partial persons (hands, head, etc.), was adapted such that only detectable persons are included. For training an independent sequence of 1200 frames, compiled from different sequences, was used.

The *PETS 2006*[3] dataset shows the concourse of a train station from four different views. For our experiments we have selected sequences from two of the four views (frontal view/Camera 3 and side view/Camera 4), which are different in view angle, size, and geometry. For evaluation Dataset S7 (Take 6-B), Camera 3 (*PETS3*) and Dataset S5 (Take 1-G), Camera 4 (*PETS4*) were adapted, i.e., the frame-rate was reduced and a ground-truth was estimated. Thus, the test sequence *PETS3* consists of 214 frames and contains 158 persons, whereas the test sequence *PETS4* consists of 308 frames and contains 1714 persons in total. In addition, independent test sequences were created, containing approximative 1000 frames, respectively.

4.2. Off-line Classifier

We trained an off-line classifier using boosting [22] with the data provided by Dalal and Triggs [1][4]. In fact, we used 1000 positive samples. In addition, the negative samples were bootstrapped from a set of many random images, that do not contain persons. This classifier was then applied to different

[2] `http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1`, (February 13, 2008)
[3] `http://www.pets2006.net`, (February 13, 2008)
[4] `http://pascal.inrialpes.fr/soft/olt/`, (February 13, 2008)

test sequences. The results are summarized in Table 1. Since the trained classifier includes no scene specific information it has to cope with a "general" learning task and hence the precision is quite low even for simple sequences such as *CoffeeCam*. Thus, such a classifier can not be applied for practical applications. Moreover, it can be seen that the classifier fails completely for the *PETS3* dataset. This is not surprising since the classifier was trained from frontal and back views only whereas the test sequence contains mainly side views of persons.

	recall	precision	F-measure
CoffeeCam	0.78	0.90	0.84
Caviar	0.62	0.25	0.36
PETS3	0.27	0.06	0.10
PETS4	0.57	0.18	0.27

Table 1. Performance of off-line classifier.

4.3. On-line Classifier without prior Knowledge

For this experiment we trained a scene-specific on-line classifier from scratch using the interactive method described in Section 3.1. We randomly initialized an on-line classifier and manually performed updates using the graphical user interface. The results are summarized in Table 2. It can be seen that by scene-specific on-line learning the performance was dramatically increased. But since no prior information is used a huge amount of human interaction (labeling effort) is necessary to train a proper classifier. Especially, for complex scenarios such as *PETS4*. Nevertheless, compared to the off-line case the number of required labeled samples, which corresponds to the number of positive and negative updates (no. updates), is quite small!

	recall	precision	F-measure	no. updates
CoffeeCam	0.93	0.80	0.86	139
Caviar	0.75	0.65	0.70	366
PETS3	0.82	0.82	0.82	139
PETS4	0.75	0.79	0.77	724

Table 2. Performance of interactive on-line classifier – without using prior knowledge.

4.4. Re-trained On-line Classifier with Prior Knowledge

To further reduce the hand labeling effort we took into account the knowledge already available by the off-line classifier. For that purpose we directly re-trained this classifier as proposed in Section 3.2.

The results are illustrated in Table 3. As can be seen the already acquired information can directly be used and the manual effort is dramatically reduced. In fact, compared to the classifier trained from scratch the number of required updates was reduced to approximative a third, whereas the performance is comparable or even better. The only exception is *PETS3*. Since this scene has a slightly different viewpoint, especially a lot of new positive samples are needed, that can not be provided by the pre-trained classifier. Thus, for both cases, with or without using prior knowledge, the same human effort is necessary.

	recall	precision	F-measure	no. updates
CoffeeCam	0.91	0.76	0.83	44
Caviar	0.79	0.65	0.72	93
PETS3	0.92	0.88	0.90	142
PETS4	0.81	0.88	0.85	221

Table 3. Performance of interactive on-line classifier – using prior knowledge.

Finally, we show some qualitative results of the proposed method. Figure 2(a) shows original detections, that were obtained by the off-line classifier whereas Figure 2(b) shows the final results, that were obtained by improving the classifier using the interactive training module. It clearly can be seen that the results obtained from re-trained classifier are much better!

(a)

(b)

Figure 2. Person detection results: (a) off-line and (b) on-line improved by the proposed method for Caviar, PETS3, and PETS4, respectively.

An outcome of our experiments is that although classifiers can be trained from scratch for a specific scene using on-line boosting, the amount of human hand labeling effort can significantly be reduced by incorporating prior knowledge. In fact, a better (off-line) seed classifier reduces the amount of required interaction.

5. Conclusion and Outlook

In this paper we proposed an interactive method for learning a scene specific person detector. The main idea is that the current detector provides labels for patches extracted from the specific scene. These patches are verified by a human operator and can be used for an on-line update of the current classifier. In fact, the learning process can be started from scratch. To reduce the training time, i.e., the number of necessary updates, a general seed classifier is trained off-line first, that is then

improved via on-line learning. In particular, we used off-line and on-line boosting for that purpose. Both learning algorithms are based on the same representation. Thus, an off-line trained classifier can directly be re-trained in an on-line manner. In the experiments we could show that the performance of a pre-trained off-line classifier can be significantly improved. This was demonstrated for two publicly available benchmark datasets, i.e., *Caviar* and *PETS 2006*. Moreover, we could show that due to the pre-trained off-line prior the human effort can be reduced to a minimum. Future work will include the extension of the proposed framework for semi-autonomous annotation of new data, where the off-line prior provides suggestions for possible labels, which in fact, might also be used for updating the existing classifier.

References

[1] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume I, pages 886–893, 2005.

[2] Y. Freund and R. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

[3] Y. Freund and R. Schapire. A short introduction to boosting. *Journal of Japanese Society for Artificial Intelligence*, 14(5):771–780, 1999.

[4] H. Grabner and H. Bischof. On-line boosting and vision. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume I, pages 260–267, 2006.

[5] H. Grabner, C. Leistner, and H. Bischof. Time dependent on-line boosting for robust background modeling. In *Proc. International Conference on Computer Vision Theory and Applications*, 2007.

[6] H. Grabner, T. Nguyen, B. Gruber, and H. Bischof. On-line boosting-based car detection from arial images. *ISPRS Journal of Photogrammetry & Remote Sencing*, 2007.

[7] H. Grabner, P. Roth, and H. Bischof. Is pedestrian detection realy a hard task? In *Proc. IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, 2007.

[8] B. Leibe, A. Leonardis, and B. Schiele. Robust object detection with interleaved categorization and segmentation. *International Journal of Computer Vision*, 2007 (in press).

[9] M. Li and I. K. Sethi. Confidence-based active learning. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 28(8):1251–1261, 2006.

[10] S. Munder and D. M. Gavrila. An experimental study on pedestrian classification. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 28(11):1–6, 2006.

[11] V. Nair and J. J. Clark. An unsupervised, online learning framework for moving object detection. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume II, pages 317–324, 2004.

[12] N. C. Oza and S. Russell. Experimental comparisons of online and batch versions of bagging and boosting. In *Proc. ACM SIGKDD Intern. Conf. on Knowledge Discovery and Data Mining*, 2001.

[13] N. C. Oza and S. Russell. Online bagging and boosting. In *Proc. Artificial Intelligence and Statistics*, pages 105–112, 2001.

[14] J.-H. Park and Y.-K. Choi. On-line learning for active pattern recognition. *IEEE Signal Processing Letters*, 3(11):301–303, 1996.

[15] P. Roth, H. Grabner, D. Skočaj, H. Bischof, and A. Leonardis. On-line conservative learning for person detection. In *Proceeding IEEE Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, 2005.

[16] R. Schapire. The boosting approach to machine learning: An overview. In *Proc. MSRI Workshop on Nonlinear Estimation and Classification*, 2001.

[17] R. E. Schapire, M.Rochery, M. Rahim, and N. Gupta. Incorporating prior knowledge into boosting. In *Proc. International Conference on Machine Learning*, 2002.

[18] C. Stauffer and W. E. L. Grimson. Adaptive background mixture models for real-time tracking. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume II, pages 246–252, 1999.

[19] K. Tieu and P. Viola. Boosting image retrieval. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 228–235, 2000.

[20] M. Turtinen and M. Pietikänien. Labeling of textured data with co-training and active learning. In *Proc. Workshop on Texture Analysis and Synthesis*, pages 137–142, 2005.

[21] O. Tuzel, F. Porikli, and P. Meer. Human detection via classification on riemannian manifolds. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[22] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume I, pages 511–518, 2001.

[23] P. Viola, M. J. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. In *Proc. IEEE Intern. Conf. on Computer Vision*, volume II, pages 734–741, 2003.

[24] B. Wu and R. Nevatia. Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors. In *Proc. IEEE Intern. Conf. on Computer Vision*, volume I, pages 90–97, 2005.

[25] B. Wu and R. Nevatia. Improving part based object detection by unsupervised, online boosting. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[26] R. Yan, J. Yang, and A. Hauptmann. Automatically labeling video data using multi-class active learning. In *Proc. IEEE Intern. Conf. on Computer Vision*, volume I, pages 516–523, 2003.