MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

A Comprehensive Evaluation Framework and a Comparative Study for Human Detectors

Mohamed Hussein, Fatih Porikli, Larry Davis

TR2010-070 January 2010

Abstract

We introduce a framework for evaluating human detectors that considers the practical application of a detector on a full image using multi-size sliding window scanning. We produce DET (Detection Error Tradeoff) curves relating miss detection rate and false alarm rate computed by deploying the detector on cropped windows as well as whole images, using in the later either image resize or feature resize. Plots for cascade classifiers are generated based on confidence scores instead of varying the number of layers. To assess a method's overall performance on a given test, we use the ALMR (Average Log Miss Rate) as an aggregate performance score. To analyze the significance of the obtained results, we conduct 10-fold cross validation experiments. We applied our evaluation framework to two state of the art cascade-based detectors on the standard INRIA Person dataset, as well as a local dataset of near infrared images. We used our evaluation framework to study the differences between the two detectors on the two datasets with different evaluation methods. Our results show the utility of our framework. They also suggest that the descriptors used to represent features and the training window size are more important in predicting the detection performance than the nature of the imaging process, and that the choice between resizing images of features can have serious consequences.

IEEE Transactions on Intelligent Transportation Systems

A Comprehensive Evaluation Framework and a Comparative Study for Human Detectors

Mohamed Hussein, *Student Member, IEEE*, Fatih Porikli, *Senior Member, IEEE*, and Larry Davis, *Fellow, IEEE*

(Invited Paper)

***Abstract*—We introduce a framework for evaluating human detectors that considers the practical application of a detector on a full image using multisize sliding-window scanning. We produce detection error tradeoff (DET) curves relating the miss detection rate and the false-alarm rate computed by deploying the detector on cropped windows and whole images, using, in the latter, either image resize or feature resize. Plots for cascade classifiers are generated based on confidence scores instead of on variation of the number of layers. To assess a method's overall performance on a given test, we use the average log miss rate (ALMR) as an aggregate performance score. To analyze the significance of the obtained results, we conduct 10-fold cross-validation experiments. We applied our evaluation framework to two state-of-the-art cascade-based detectors on the standard INRIA Person dataset and a local dataset of near-infrared images. We used our evaluation framework to study the differences between the two detectors on the two datasets with different evaluation methods. Our results show the utility of our framework. They also suggest that the descriptors used to represent features and the training window size are more important in predicting the detection performance than the nature of the imaging process, and that the choice between resizing images or features can have serious consequences.**

***Index Terms*—Cascade, evaluation, histograms of oriented gradients (HOGs), human detection, near infrared, region covariance.**

I. Introduction

HUMAN detection is one of the most challenging tasks in computer vision with a long list of fundamental applications from intelligent vehicles and video surveillance to interactive environments. Unlike other detection problems, there exist significant appearance changes due to the pose variations and articulated body motion of humans, even for the same person. People, as a general class, dress in different colors and styles of clothing, carry bags, and hide behind umbrellas. They move together and occlude each other.

Manuscript received February 9, 2008; revised December 5, 2008. First published July 21, 2009; current version published September 1, 2009. This work was supported by Mitsubishi Electric Research Laboratories, Cambridge, MA. The Associate Editor for this paper was U. Nunes.

M. Hussein and L. Davis are with the Department of Computer Science, University of Maryland, College Park, MD 20742 USA (e-mail: mhussein@cs.umd.edu; lsd@cs.umd.edu).

F. Porikli is with Mitsubishi Electric Research Laboratories, Cambridge, MA 02139 USA (e-mail: fatih@merl.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TITS.2009.2026670

Despite these challenges, there has recently been a significant advancement in this area of research. Nevertheless, little attention has been given to the evaluation of detectors for practical applications. First, there is a notable mismatch between the way detectors are evaluated and the way they are applied in real-world applications, such as smart vehicle systems. At one end, detectors are evaluated on "ideal" windows that are cropped to have the human subjects centered on them and resized to match the window size used in training. However, at the other end, detectors are applied to whole images, typically using a multiple-size sliding-window approach, which results in probe windows that are far from being ideal. Second, most of the evaluations are performed on a single dataset, which leaves practitioners with uncertainty about the detection performance on other datasets, possibly with different modalities, or the significance of one detector's advantage over the other. Third, for detectors based on cascade classifiers, typically, performance plots are created by changing the number of cascade layers. This technique sometimes leads to difficulty in comparing different methods when the resulting plots do not cover the same range of false-alarm rates.

The main contribution of this paper is an evaluation framework that handles the shortcomings of the existing evaluations. The main features of our evaluation are given here:

1) comparing between evaluation on cropped windows and evaluation on whole images to get better prediction for a detector's performance in practice and how it differs from ideal settings;
2) using 10-fold cross validation to be able to study the significance of the obtained results;
3) plotting detection error tradeoff (DET) curves based on confidence scores for detectors based on cascade classifiers, instead of plotting them based on varying the number of layers;
4) introducing an aggregate performance score and using it as the main metric to statistically compare methods;
5) comparing between building a multisize image pyramid while fixing the scanning window size, and using a single image size and changing the scanning window size, when applying the detector on whole images. We refer to these two choices as *resizing images* and *resizing features*, respectively. This is an example of an implementation choice that can have a significant effect on

the detection performance, depending on the evaluated detector;

6) evaluation on both near-infrared images and visible images.

The goal of our study is not to provide a performance comparison of the state-of-the-art human detection techniques. Instead, our goal is to introduce a comprehensive evaluation framework and to highlight the mismatch between the typical evaluation techniques and the practical deployment of the detectors. We utilized the two detectors in [1] and [2] to demonstrate our evaluation framework. To the best of our knowledge, these are the best performing human detectors based on rejection cascades. We focus on rejection cascades, because they are appealing for practical applications, as explained in Section III. Despite that our presentation focuses on human detection, our framework and observations apply to other objects as well.

Our experimental results show the utility of our framework in understanding the performance of a human detector in practice. They suggest that the descriptors used to represent features, histograms of oriented gradients (HOGs), or region covariances in our study and the size of the training window are more important in predicting the detection performance than the nature of the imaging process, such as the imaged electromagnetic band. They also show that the choice between resizing images or features can have a significant impact on the performance, depending on the used descriptor.

This paper is organized as follows: Section II gives a brief overview of the human detection techniques. In Section III, we briefly describe the two pedestrian detectors used in our evaluation. In Section IV, we explain the elements of our evaluation framework. In Section V, we introduce the two datasets we use and how we prepared them for the experiments. In Section VI, we present the results and analysis of our evaluation. Finally, the conclusion is given in Section VII.

II. Human Detection

Human detection methods can be categorized into two groups based on the camera setup. For static camera setups, object motion is considered as the distinctive feature. A motion detector, either a background subtraction or an image segmentation method, is applied to the input video to extract the moving regions and their motion statistics [3], [4]. A real-time moving human detection algorithm that uses Haar wavelet descriptors extracted from space-time image differences was described in [5]. Using AdaBoost, the most discriminative frame difference features were selected, and multiple features were combined to form a strong classifier. A rejection cascade that is constructed by strong classifiers to efficiently reject negative examples is adopted to improve the detection speed. A shortcoming of the motion-based algorithms is that they fail to detect stationary pedestrians. In addition, such methods are highly sensitive to viewpoint and illumination changes.

The second category of methods is based on detecting human appearance and silhouette by either applying a classifier at all possible subwindows in the given image or assembling local human parts [6]–[10] according to geometric constraints to form the final human model. A classic appearance-based approach is template matching, as in [11] and [12]. In this approach, a hierarchy of human body templates is built to efficiently be matched to the edge map of an input image via distance transform. Template matching is prone to producing false alarms in heavily cluttered areas. Another popular appearance-based method is the principal component analysis (PCA), which projects given images onto a compact subspace. While providing visually coherent representations, PCA tends to be easily affected by the variations in pose and illumination conditions. To make the representation more adaptive to changes, local receptive field features are extracted from silhouettes using multilayer perceptrons by means of their hidden layer [13] and are then provided to a support vector machine (SVM). In [14], a polynomial SVM was learned using Haar wavelets as human descriptors. Later, the work was extended to multiple classifiers trained to detect human parts, and the responses inside the detection window are combined to give the final decision [15]. In [16], human parts were represented by co-occurrences of local orientation features, and separate detectors were trained for each part using AdaBoost. Human location was determined by maximizing the joint likelihood of part occurrences combined according to the geometric relations.

In [17], local appearance features and their geometric relations were combined with global cues by top–down segmentation based on per-pixel likelihoods. In [18], an SVM classifier, which was shown to have false positive rates of at least one to two orders of magnitude lower for the same detection rates than the conventional approaches, was trained using densely sampled HOGs inside the detection window. This approach was extended to optionally account for motion by extending the histograms to include flow information in [19]. More recently, it was also applied to deformable part models as in [20] and [21]. A near real-time system was built based on it using a cascade model in [1]. Cascade models have also been successfully used with other types of features, such as edgelet features [22], region covariance [2], shapelet features [23], or heterogenous features [24].

III. Evaluated Detectors

The two human detectors that we used in our evaluation are based on a rejection cascade of boosted feature regions. They differ in how they describe the feature regions and how the weak classifiers are trained. One detector uses region covariance to describe feature regions and uses classification on Riemannian manifolds for the weak classifiers [2]. We refer to this detector as COV. The other detector uses HOGs to describe feature regions and uses conventional linear classification [1]. We refer to this detector as HOG. For the sake of completeness, we briefly describe here the notion of a rejection cascade of boosted feature regions and the descriptors used by the two classifiers. See the original papers for more details.

A. Rejection Cascade of Boosted Feature Regions

Rejection cascades of boosted feature regions were popularized by their success in the area of face detection [25]. They are based on two main concepts: 1) *boosted feature regions* and 2) *rejection cascades*.

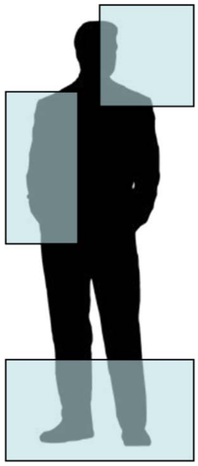


Fig. 1. Shaded rectangular subregions of the detection window are possible features to be combined to build stronger boosted features.

In boosting [26], a *strong classifier* is built by combining a number of *weak classifiers*. Boosting *feature regions* can be understood as combining simple feature regions to build a strong representation of the object that can be used to distinguish the object from other objects. The feature regions in our case are rectangular subregions from the *feature maps* of input images, as shown in Fig. 1. The concept of a feature map is explained in Section III-B.

A *rejection cascade* is composed of a number of classification layers. As shown in Fig. 2, a test pattern is examined by layers of the cascade one after another until it is rejected by one of them or until it is accepted by the final layer, in which case, it is classified as a positive example. During training of the cascade, the first layer is trained on all positive examples and a random sample of negative examples. Each subsequent layer is trained on all positive examples and the false positives of the preceding layers. This way, each layer handles harder negative examples than all the preceding layers. The benefit of this mechanism is twofold: One is the possibility of using a huge number of negative examples in training the classifier, which is not possible in training a traditional single-layer classifier. The other is that, during testing, most negative examples are quickly rejected by the initial layers of the cascade, and only hard negative examples are handled by the latter layers. Since, in our applications, it is likely that most of the examined patterns are negative, rejection cascades are computationally efficient since they quickly reject easy negative examples while spending more time on the hard negative or the positive examples. In our implementation, each cascade layer is trained using the LogitBoost algorithm [26].

B. Region Covariances

Region covariances were first introduced as descriptors in [27] and then used for human detection [2], which outperformed other state-of-the-art classifiers. Let I be a $W \times H$ 1-D intensity or a 3-D color image, and let F be a $W \times H \times d$ dimensional feature map extracted from I, i.e.,

$$F(x,y) = \Phi(I,x,y) \tag{1}$$

where function Φ can be any mapping, such as intensity, color, gradients, and filter responses. For a given rectangular region $R \subset F$, let $\{\mathbf{z}_i\}_{i=1,\ldots,S}$ be the d-dimensional feature points inside R. Region R is represented with the $d \times d$ covariance matrix of the feature points, i.e.,

$$\mathbf{C}_R = \frac{1}{S-1}\sum_{i=1}^{S}(\mathbf{z}_i - \mu)(\mathbf{z}_i - \mu)^T \tag{2}$$

where μ is the mean of the points.

For the human detection problem, the mapping $\Phi(I,x,y)$ is defined as

$$\left[x \quad y \quad |I_x| \quad |I_y| \quad \sqrt{I_x^2+I_y^2} \quad |I_{xx}| \quad |I_{yy}| \quad \arctan\frac{|I_x|}{|I_y|} \right]^T \tag{3}$$

where x and y represent the pixel location, $I_x, I_{xx}, \ldots$ are intensity derivatives, and the last term is the edge orientation. With this definition, the input image is mapped to a $d = 8$ dimensional feature map. The covariance descriptor of a region is an 8×8 matrix, and due to symmetry, only the upper triangular part is stored, which has only 36 different values. To make the descriptor invariant to local illumination changes, the rows and columns of a subregion's covariance matrix are divided by the corresponding diagonal elements in the entire detection window's covariance matrix.

Region covariances can efficiently be computed in $O(d^2)$ computations, regardless of the region size, using integral histograms [27], [28]. Covariance matrices and, hence, region covariance descriptors do not form a Euclidean vector space. However, since covariance matrices are positive definite matrices, they lie on a connected Riemannian manifold. Therefore, classification on Riemannian manifolds is more appropriate to be used with these descriptors [2].

C. HOGs

HOGs were first applied to human detection in [18], which achieved a significant improvement over other features used for human detection at that time. HOGs were used in the framework of rejection cascades of boosted feature regions in [1] to deliver comparable performance to [18] at a much higher speed.

To compute the HOG descriptor of a region, the region is divided into four cells in a 2×2 layout. A nine-bin histogram is built for each cell. Histogram bins correspond to different gradient orientation directions. Instead of just counting the number of pixels with a specific gradient orientation in each bin, gradient magnitudes at the designated pixels are accumulated. Bilinear interpolation is used between the orientation bins of the histogram and spatially among the four cells. The four histograms are then concatenated to make a 36-D feature vector, which is then normalized. In our implementation, we use L_2 normalization for HOG descriptors.

Like region covariance descriptors, HOG descriptors can be computed fast using integral histograms. Bilinear interpolation among cells is computed fast using the kernel integral image approach [29].

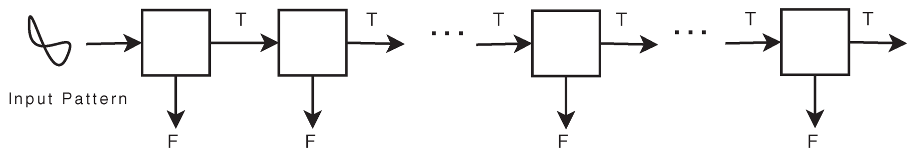


Fig. 2. Rejection cascade consists of layers. A test pattern is examined by layers in the cascade from left to right until being rejected. A pattern is accepted if all layers accept it.

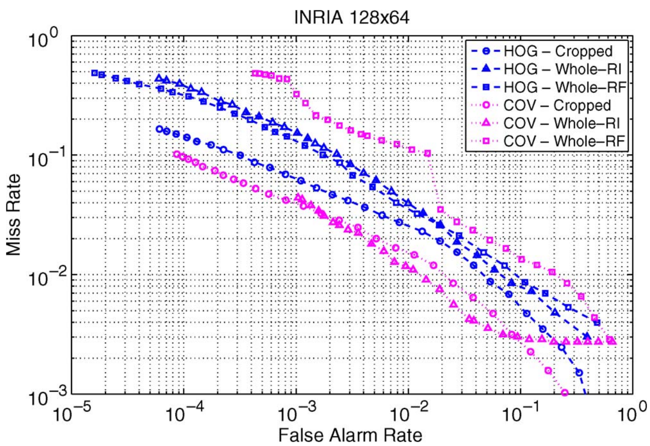


Fig. 3. DET layer plots for the INRIA dataset with window size 128 × 64.

IV. Evaluation Framework

In most recent studies on human detection, evaluation results are presented in DET curves, which relate the false alarm rate per window to the miss rate of the classifier in a log-log scale plot. Typically, positive examples used in the evaluation are adjusted to have the same subject alignment and size used in training the classifiers, and negative examples are human free. In this section, we identify several shortcomings of this evaluation approach. We explain how we address these shortcomings in our evaluation framework.

A. Score Plots for Cascade Classifiers

Typically, points on the DET curves of cascade classifiers are generated by changing the number of cascade layers. The problem with this approach is that the generated plots are not guaranteed to cover a particular range for either the horizontal or the vertical axis, which makes it hard to compare different methods. Fig. 3 shows examples of such plots. To overcome this problem, in our evaluation, we compute a confidence score for each sample and generate the plots based on these scores. We assume that each layer of the cascade can give a confidence score $\varphi(\mathbf{x}) \in (0, 1)$ to any given example $\mathbf{x}$. The overall confidence score over an n-layer cascade can be expressed as

$$\Phi(\mathbf{x}) = \mathcal{N}(\mathbf{x}) + \varphi_l(\mathbf{x}) \tag{4}$$

where $\mathcal{N}(\mathbf{x})$ is the number of layers that accepted $\mathbf{x}$, and $\varphi_l(\mathbf{x})$ is the confidence score of the last layer that examined it. The score in (4) reflects the way a cascade classifier works. It gives higher scores to examples that reach deeper in the cascade. If two examples leave the cascade at the same layer, their confidence scores will differ by the confidence scores assigned by the last layer. This way, we get a real-valued score. We can create DET curves from these scores by changing the threshold above which a test example is considered positive. At each point on the curve, we appropriately set the threshold to generate a specific level of the false alarm rate. Then, we measure the miss rate at this threshold value. This way, we have control over the range of false alarm rates to cover. Fig. 7 shows the same results of Fig. 3 using confidence scores.

In our implementation, each layer of the cascade is a boosted classifier. The real-valued outcome of such a classifier is proportional to the number of weak classifiers in it. Hence, we normalize this outcome by the number of weak classifiers to produce the layer's score in the range $(-6, 6)$. Then, this value is mapped to the range $(0, 1)$ using sigmoid function $\exp(x)/(\exp(x) + \exp(-x))$.

B. Evaluation on Whole Images

Evaluation on cropped windows is an optimistic estimate of the detector's performance in practice. Typically, detectors are applied to whole images using a multiple-size sliding-window scanning. The windows fed to the classifier in this case can rarely have humans centered on them or have the proper size, which would yield a lower performance than in the case of application to cropped windows. We evaluated the classifiers on both cropped windows and whole images to compare them. In the case of evaluation on cropped windows, the positive and negative examples are well defined. However, in the case of evaluation on whole images, the situation is different. In this case, scanned windows are not all perfect positive or negative examples since they may contain parts of humans or full humans who are not in the proper location or relative size. In many applications, if the detection window is slightly shifted, or slightly smaller or larger than the subject, it is still useful. Therefore, we should not consider such windows as negative examples and penalize the classifier for classifying them as positives. However, if we consider all scanned windows that are close to a human subject as positive examples, we will be penalizing the classifier for missing any of them, although detecting just one is good enough in practice.

Based on these considerations, in the case of evaluation on whole images, we consider any scanned window that is significantly far from all annotated human subjects in the image as a negative example. A missed detection is counted if an annotated human subject is significantly far from all scanned windows that are classified as positives by the classifier. In other words, a missed detection is counted if all scanned windows that are close enough to an annotated human subject are classified as negatives. The measure of closeness we use is the *overlap ratio*.

Let $|R|$ be the area of region R. Consider two regions R_1 and R_2. The overlap ratio between them is defined as

$$\mathcal{O}(R_1, R_2) = \frac{|R_1 \cup R_2|}{|R_1 \cap R_2|}. \quad (5)$$

This ratio is minimum (1) when the two regions are perfectly aligned and is maximum (∞) when they have no overlap. In our evaluation, we consider a scan window negative if its overlap ratio to the closest annotated human subject is above 16. We count a miss detection if all scanned windows within an overlap ratio of 2 around an annotated human subject are all classified as negatives. The latter threshold is the same used in the Pascal challenge [30]. According to these thresholds, there are windows that are counted as neither positives nor negatives. The upper threshold is rather conservative so that we do not consider a window negative, unless it is too far from all annotated human subjects. For assigning scores to windows, negative windows' scores are computed as in (4), and each annotated human subject is assigned the maximum score over all positive windows associated with it.

Another option in presenting the performance on whole images would be to use precision recall (PR) curves. It was shown [31] that PR and receiver operating characteristic (ROC) curves are closely related in the sense that the dominant curve in one is the dominant curve in the other if they are generated using the same points. We preferred using DET curves, which are the log-log version of ROC curves, so that the performance on whole images can be compared with that on cropped windows in our results and other published results. In addition, to generate a PR plot, nearby detection windows have to be consolidated. First, we selected not to confound the detector's performance by a particular choice of this postprocessing step. Second, in our framework, consolidation will have to be applied at each point of the plot, which is prohibitively expensive.

1) Resizing Images Versus Resizing Features: An implementation choice for evaluation on whole images turns out to have a strong effect on the detection performance. We train each classifier on single-size images. In the case of applying them on whole images, which contain humans of different sizes, we have two options: One is to resize the images so that our scanning window size becomes the same as the training size. We refer to this option as *resizing images*. The other option is to resize the features selected by the classifier while maintaining their relative sizes to the scan window. We refer to this option as *resizing features*. Resizing features are faster since the preprocessing of the image, e.g., computing gradients and integral histograms, is performed only once. We made evaluations on whole images using the two options to compare them.

C. Statistical Analysis

Statistical analysis of detection performance is rarely conducted for human detection, possibly due to the long training time. To our knowledge, the only study that provided statistical analysis was [13], where a confidence interval for each point on the ROC curve was computed based on six observations (three training sets × two testing sets). We found it confusing to plot confidence intervals with the plots since, in our evaluation, plots intersect and come close to one another. Instead, we compute confidence intervals for the aggregate performance score, Average Log Miss Rate (ALMR), which is explained in Section IV-D. We conduct a 10-fold cross validation for all our experiments. Therefore, for each experiment, we obtain ten different curves. Each curve yields an ALMR score. To compare different experiments, we plot the average curve for each experiment. We also present a box plot for the mean, confidence interval, and range of the ALMR scores for all experiments in a separate plot. Confidence intervals are computed at the 0.95 confidence level.

TABLE I
COMPARISON BETWEEN THE TWO DATASETS USED IN OUR EVALUATION. TRACKS* ARE DEFINED ONLY IN THE CASE OF THE MERL-NIR DATASET

	INRIA	MERL-NIR
Electromagnetic Band	Visible	Near Infrared
Source of Images	Personal Photos	Interlaced Video Frames
Total Number of Images	2572	46000
Image Size	Variable	720×480
Number of Images Containing Humans	901	9823
Number of Human Samples	1825	11895
Number of Tracks	N/A	285
Min Person Height	48	20
Max Person Height	832	323
Mean of Person Height	290	92.66
Standard Deviation of Person Height	147.83	59.92
Median Person Height	260	72
Mode Person Height	208	50

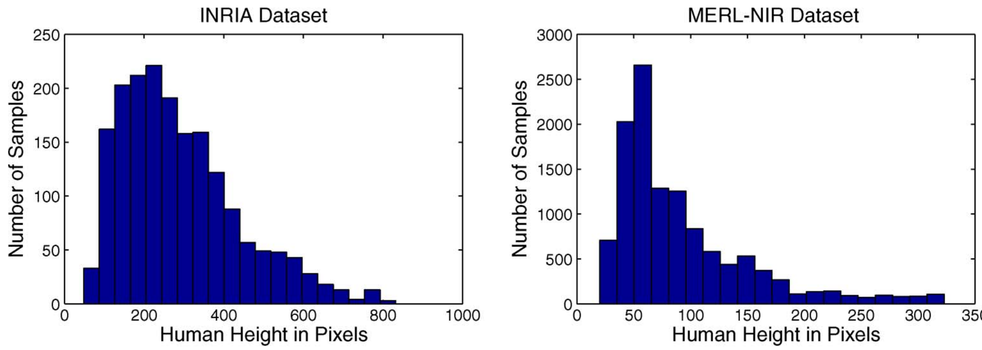
*A track is a sequence of windows containing the same person in consecutive frames. More than one track can be associated with one person if she becomes partially or totally occluded and then fully visible again.

D. Computing an Aggregated Performance Score

To analyze the significance of one method's advantage over another, we need an aggregated score that captures the difference between them over the entire curve. The log-log plots emphasize the relative difference, instead of the absolute difference between two curves. We need a score that emphasizes the same difference to be consistent with the difference perceived from the plots. For two curves a and b, such a score can be expressed as

$$R_{ab} = \frac{1}{n}\sum_{i=1}^{n} \log \frac{mr_i^a + \epsilon}{mr_i^b + \epsilon} \quad (6)$$

where mr is a miss rate value, ϵ is a small regularization constant, and the sum is over the points of the DET curve. We use 10 as the logarithmic base and $\epsilon = 10^{-4}$ in our experiments.



(a) (b)

Fig. 4. Distribution of human height in pixels in the two datasets used in our evaluation. (a) INRIA dataset. (b) MERL-NIR dataset.

We found the value of ϵ not significant in comparing curves. If this score is positive, it indicates that curve a misses more on average, and *vice versa*.

Instead of having a score for each pair of curves, it is better to have a score for each curve and compare the curves by comparing the scores. The score R in (6) can be expressed as

$$R_{ab} = \frac{1}{n}\sum_{i=1}^{n}\log\left(mr_i^a+\epsilon\right) - \frac{1}{n}\sum_{i=1}^{n}\log\left(mr_i^b+\epsilon\right). \quad (7)$$

This suggests that we can represent the performance of each curve as the average of the logarithm of the miss rate values over the curve. However, this score will always be negative. Therefore, we switch its sign to reach the following expression for the ALMR score:

$$\text{ALMR} = \frac{-1}{n}\sum_{i=1}^{n}\log(mr_i+\epsilon). \quad (8)$$

The higher the value of the ALMR score, the lower the miss rate over the curve on average, i.e., the better. The ALMR score is related to the R score in (6) and (7) by

$$R_{ab} = \text{ALMR}_b - \text{ALMR}_a. \quad (9)$$

The ALMR is related to the geometric mean of the miss rate values. It is also proportional to the area under the curve in the log-log domain when the curve is approximated using a staircase plot. Since our plots are on a log-log scale and the points are uniformly spaced, the ALMR score contains more samples from the low false alarm rate values. This is useful since, in many applications, we are more interested in the low-false-alarm-rate range.

Finally, in our evaluation, we call the difference between the ALMR scores of two experiments *significant* when the confidence intervals of the two experiments do not overlap. Otherwise, we call the difference insignificant.

V. Evaluation Datasets

We evaluated our detectors on two different datasets. The first is the INRIA-Person dataset, which is maintained by the Institut National De Recherche En Informatique Et En Automatique (INRIA), and publicly available online. We refer to this dataset by INRIA. The second dataset is maintained by Mitsubishi Electric Research Labs (MERL). We refer to this dataset by MERL-NIR, for MERL-Near InfraRed. The INRIA dataset was introduced in [18] and subsequently used to evaluate many human detectors. The MERL-NIR dataset consists of 46 000 frames from a video sequence. The video was shot from a vehicle touring an Asian city, using a near-infrared interlaced camera. From the frames that contained annotated human subjects, we uniformly sampled 1600 to be used as positive images. From the remaining frames, we randomly sampled 1100 to be used as negative images. The description of the two datasets, along with the statistics and histograms of human sizes, is given in Table I and Fig. 4. Sample whole images and cropped human windows used in training and testing are shown in Figs. 5 and 6. To conduct cross-validation experiments, we divided the whole positive images in each dataset into five sets of a roughly equal number of annotated human subjects. We perform

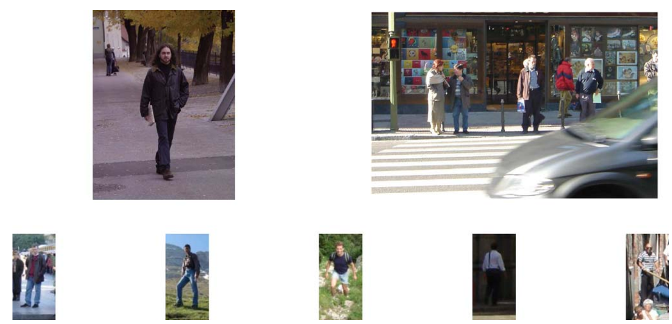


Fig. 5. Sample whole and cropped human images from the INRIA-Person dataset.

Fig. 6. Sample whole and cropped human images from the MERL-NIR dataset.

TABLE II
DIVISION OF EACH DATASET INTO FIVE POSITIVE SUBSETS AND TWO COMMON NEGATIVE SETS FOR 10-FOLD CROSS-VALIDATION EXPERIMENTS

		INRIA		MERL-NIR	
		Whole	Cropped	Whole	Cropped
Positive	Set # 1	179	730	320	766
	Set # 2	180	730	320	764
	Set # 3	180	730	320	764
	Set # 4	181	730	320	764
	Set # 5	181	730	320	764
Negative	Training	1218		800	
	Testing	453		300	

10-fold cross validation by using three sets for training and two sets for testing in each fold. Negative images used in training and testing are common in all experiments. Table II describes the contents of each set and the number of negative images in the two datasets. The number of cropped windows in the table includes the left–right reflection of each window.

VI. EVALUATION RESULT

We train the cascade classifiers to have 30 cascade layers. Each layer is trained using the LogitBoost algorithm [26] and adjusted to produce a 99.8% detection rate and a 65% false alarm rate, using the algorithm in [25]. The number of negative samples collected for each layer is set to 3.5 times the number of positive samples. Features are generated with the minimum side length set to 12.5% of the corresponding window side length, with a minimum of 8 pixels to have enough sample points to construct histograms and covariance matrices. The feature location stride and side length increment are set to half the minimum feature side length. For every five boosting iterations, 5% of the features are randomly sampled, with a maximum of 200. The limit on the number of sampled features is for all descriptors to fit in the memory, instead of being recomputed on every boosting iteration.

For evaluation on whole images, each image is scanned with nine window heights, starting from 75% of the training window height and using an increment of 30% of the last height used while preserving the aspect ratio. The scanning stride is set to 5% of the scanning window size in each dimension.

Our training and testing modules were run on a cluster of computers, with about 60 active nodes. Each node contained two Intel(R) Xeon(TM) CPU 3.06-GHz processors with 512 KB of cache memory and 4 GB of random access memory. The front end and compute operating system was CentOS release 4.5.

In the remainder of this section, we first present the evaluation results on the INRIA dataset with the default training and testing window size of 128 $\times$ 64. Then, we present the results on the MERL-NIR dataset, in which we use a window size of 48 $\times$ 24. Along with this set of results, we present the results for the INRIA dataset with window size 48 $\times$ 24 for the sake of comparison with the results on the MERL-NIR dataset. We present all the plots using the same limits in both axes for ease of comparison. In each plot, curves for the COV detector are drawn using dotted lines, and curves for the HOG detector

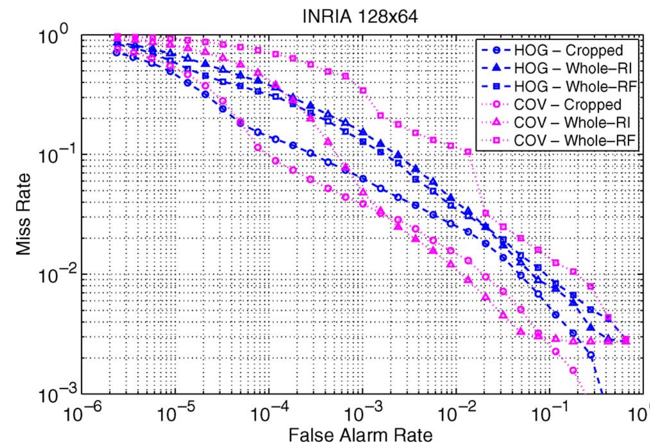


Fig. 7. DET score plots for the INRIA dataset with window size 128 $\times$ 64.

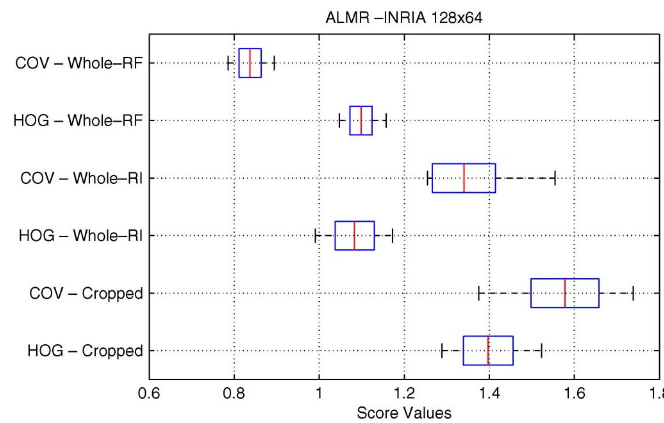


Fig. 8. Box plot for the mean, confidence interval, and range of the ALMR score for the plots in Fig. 7.

are drawn using dashed lines, with a different marker shape for each type of experiment. The legend of each experiment has two parts: The first is the descriptor, which is either HOG or COV. The second is the evaluation method, which is either Cropped, Whole-RI, or Whole-RF for cropped windows, whole images with resizing images, and whole images with resizing features, respectively.

A. Evaluation on INRIA 128 $\times$ 64

In this set of experiments, we evaluate our two detectors on the INRIA dataset using the original window size of 128 $\times$ 64, where each positive window is adjusted so that the height of the human body in it is 96 pixels.

Fig. 7 shows the DET score plots for this set of experiments. Each curve is the average of the ten curves produced by cross validation. However, the curves often intersect one another, and there is no clear winner. Therefore, we will rely on the ALMR score statistics to compare experiments when it is hard to reach a conclusion by inspecting the curves.

Fig. 8 shows the statistics of the ALMR score for each curve in Fig. 7. Note how comparing the mean values of the ALMR scores of two curves matches well with how the curves themselves compare with one another on average. The difference between the mean scores of two curves reflects the average relative advantage of one curve over the other in terms of miss rate. For example, the mean ALMR scores for the HOG-Cropped and COV-Cropped experiments are approximately 1.6 and 1.4, respectively. This means that, on average, the miss rate of the HOG detector is $10^{0.2} \simeq 1.6$ times the miss rate of the COV detector, which is consistent with how the curves compare with one another.

For evaluation on cropped windows, the ALMR score shows the significant advantage of the COV detector on average. The confidence intervals of the two scores do not overlap. On average, COV leads by about 0.2 points. Note how the ranges of the ALMR scores are large to the extent that they overlap. This signifies the importance of using statistical analysis to have a reliable estimate for a detector's performance.

For evaluation on whole images, the COV detector maintains its lead over the HOG detector. The lead this time is even more evident since the ranges of the ALMR scores do not overlap. On average, COV leads by about 0.2 points. However, the performance of the two detectors significantly deteriorates in this case by losing about 0.3 points on the ALMR scale on average. This deterioration signifies the importance of evaluation on whole images to predict the detector's performance in a typical practical setting.

Finally, for evaluation on whole images with resizing features, the picture is totally different. Without even inspecting the ALMR score statistics, we can notice that the HOG detector consistently outperforms the COV detector. By inspecting the ALMR scores, we notice that this difference is significant. On average, HOG outperforms COV by about 2.5 points. The difference between the two detectors' behaviors in this case may be due to the difference between the two descriptors or due to the usage of learning on Riemannian manifolds in the case of COV. Further investigation is needed to understand this phenomenon. On the other hand, comparing the evaluation on whole images for the HOG detector with resizing images and resizing features, we find the difference between them to be insignificant. The mean score of each experiment lies in the confidence interval of the other. This gives the HOG detector higher advantage over COV in terms of processing time. The COV detector is at least ten times slower than the HOG detector. Resizing features saves about 40% of the processing time of the HOG detector without significant loss in detection performance. This makes the COV detector at least about 17 times slower than the HOG detector when resizing features is used for the latter.

Despite the advantage of the COV detector in most of the experiments, on average, it is worth noting that the HOG detector often slightly outperforms the COV detector in the very low false alarm rate range, which is below 10^{-4}. However, the points in this range of false alarm rates are often found only in the score-based plots and are missing from the layer-based plots (compare Fig. 7 with Fig. 3). This may indicate the possibility of obtaining a more consistent advantage for the COV detector if we continue training more cascade layers to cover the entire range of false alarm rate. However, this is difficult in practice. It takes about four days to train a COV classifier for 30 layers. The bottleneck of the training process is finding enough misclassified negative samples for each new layer to be trained, and this time increases with the number of layers.

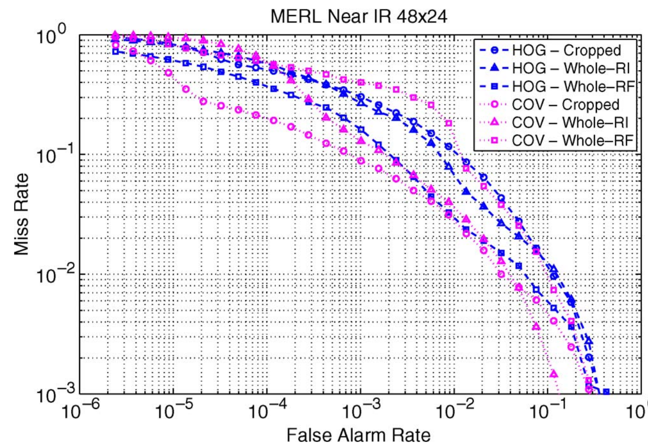


Fig. 9. DET score plots for the MERL-NIR dataset.

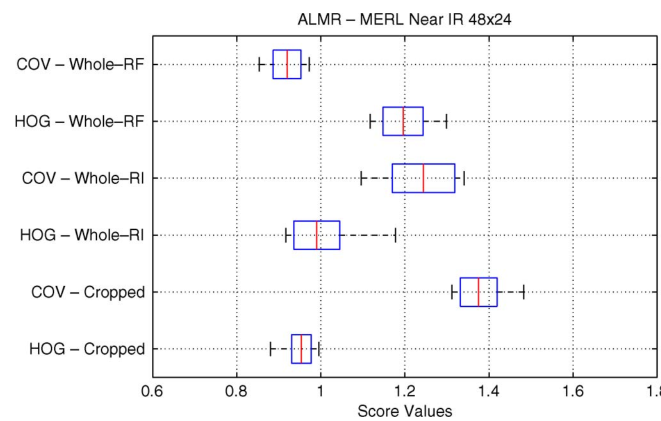


Fig. 10. Box plot for the mean, confidence interval, min, and max of the ALMR score for the plots in Fig. 9.

B. Evaluation on MERL-NIR

In this set of experiments, we evaluate our two detectors on the MERL-NIR dataset. Due to the smaller person heights in this dataset, compared with the INRIA dataset, as shown in Fig. 4, we have to use the reduced window size of 48×24 in this set of experiments. All positive windows are adjusted, so that the height of the human body is 36 pixels. Because of this reduction in window size, we expect reduced detection performance.

Figs. 9 and 10 show the DET plots and ALMR score statistics for this set of experiments. Similar to the results on the INRIA 128×64 dataset, the COV detector's lead over the HOG detector in the case of cropped windows and whole images with resizing images, and the HOG detector's lead in the case of whole images with resizing features are significant. However, there are several differences between the two sets of results. The first notable difference is the improved performance for both detectors in the case of resizing features with respect to the other types of evaluation. In the case of HOG, using resizing

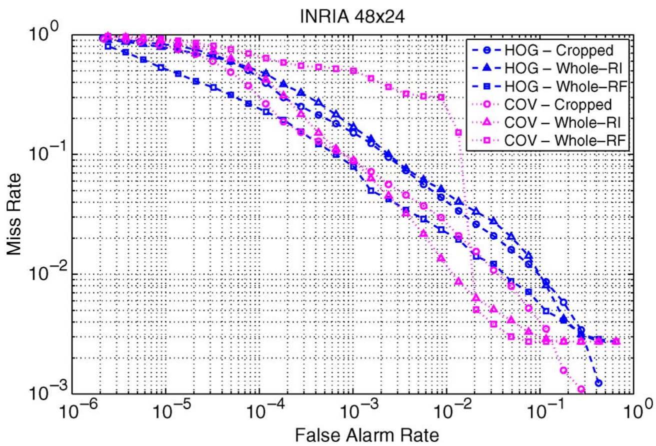


Fig. 11. DET score plots for the INRIA dataset with window size 48 × 24.

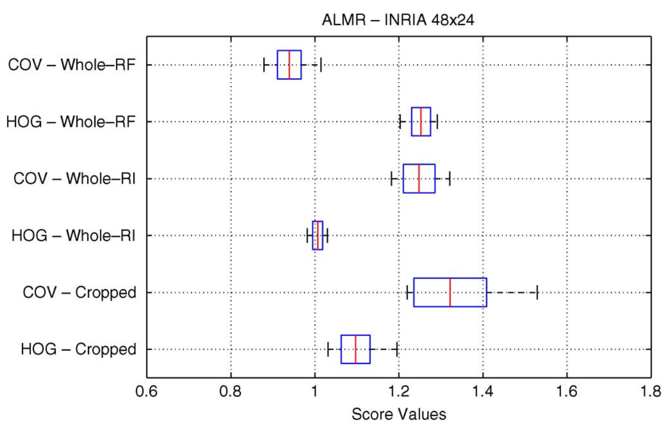


Fig. 12. Box plot for the mean, confidence interval, min, and max of the ALMR score for the plots in Fig. 11.

features became even better than using resizing images. The second notable difference is that the advantage of evaluation on cropped windows over evaluation on whole images with resizing images is no longer significant, with overlapping confidence intervals of the ALMR scores, and is reversed in the case of the HOG detector.

Before attempting to explain these differences, we present another set of results on the INRIA dataset, but with the window size reduced to match that used with MERL-NIR. In this set of experiments, all the INRIA dataset images used in training and testing are reduced in size with the same factor that reduces the window size of 128 × 64 to 48 × 24. Figs. 11 and 12 show the results of this set of experiments. Comparing this set of results with those obtained on the MERL-NIR dataset, by comparing Fig. 12 with Fig. 10, we find that they are very similar. Most of the differences between them are either small or insignificant. This observation gives us a clue about the differences between the results on the INRIA 128 × 64 dataset and those on the MERL-NIR dataset. It tells us that the difference is mostly due to the window size.

The reduced window size leads to a reduced stride when scanning whole images for evaluation since we set the stride to be 5% of the window side length. That makes the stride just 1 or 2 pixels in each dimension for a 48 × 24 window. In addition, using a reduced minimum scanning size results in a reduced scanning size range and, hence, a denser coverage of that range. These two factors could explain the reduction in the performance gap between the evaluation on cropped windows and the evaluation on whole images. With reduced window sizes and window size range, there is a higher chance that the scanning window will become close to annotated human subjects while having them centered. In addition, with a smaller range of scanning window sizes, the effect of resizing features, compared with resizing images, should be less significant. Nevertheless, the enhanced performance of resizing features, compared with resizing images, in the case of HOG needs further investigation.

Finally, by comparing the ALMR scores in the case of evaluation on cropped images when using a large scan window size (see Fig. 8) with the use of a small scan window size (see Figs. 10 and 12), we observe that the performance on small window sizes is significantly worse. Note that evaluation on cropped windows actually evaluates the classifier and not how it is used in the detection task. A classifier trained on a large window size has a richer set of features to select from. Therefore, it is expected to perform better, as the results show.

VII. Conclusion

We have presented a comprehensive evaluation framework for object detectors that is geared toward a typical practical deployment paradigm. We have demonstrated its utility on two state-of-the-art human detection algorithms that are based on cascade classifiers on two different datasets, covering two bands of the electromagnetic spectrum, which are visible and near infrared. In our evaluation, we have compared between the typically used evaluation on cropped windows and the more practical evaluation on whole images. We have introduced enhanced DET plot generation based on confidence scores, instead of variation of the number of layers in cascade classifiers. We have introduced an aggregate performance score to summarize such plots for ease of comparison. We have used 10-fold cross validation to statistically analyze our results.

Our experiments have shown the effectiveness of our framework and have led to the findings given here.

1) The COV detector maintains a significant lead over the HOG detector on average. However, sometimes, it is very close or slightly inferior in the very low false alarm rate range, and it is at least 17 times slower.
2) Application of detectors on whole images can yield a significant reduction in detection performance than what can be observed upon evaluation on cropped windows. However, when the application deploys a dense scanning in terms of strides and window sizes, the difference between them may not be significant.
3) Detection performance may not significantly be affected by applying the same algorithm to images in the near-infrared band, instead of the visible band. However, it is significantly affected by the window size used in training the classifiers.

4) Whether to use resizing images or resizing features, when applying a detector to whole images, can have a significant effect on the detection performance, depending on the detector used. While the HOG detector can deliver the same or better performance when resizing features, the COV detector delivers a significantly deteriorated performance.

Many directions can be taken for future extensions and enhancements of our framework. It is not clear how the extended plots we obtain for cascade classifiers using confidence scores are comparable with plots obtained by increasing the number of layers in the cascades. The ALMR aggregate confidence score gives an overall performance measure, assuming that the performance over the entire range of the false alarm rate is important. An investigation using a weighted or limited-range version of the score for some applications can be useful. Comparison with PR curves and what we learn from both DET and PR curves on the evaluation on whole images needs to be studied further. Finally, the framework, in general, needs to be applied to other state-of-the-art detectors, particularly those that do not rely on cascade classifiers.

Acknowledgment

The authors would like to thank J. McAndless for taking over the tedious job of creating ground truth annotations for the near IR dataset.

References

[1] Q. Zhu, S. Avidan, M.-C. Yeh, and K.-T. Cheng, "Fast human detection using a cascade of histograms of oriented gradients," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recog.*, New York, Jun. 2006, pp. 1491–1498.

[2] O. Tuzel, F. Porikli, and P. Meer, "Human detection via classification on Riemannian manifolds," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recog.*, 2007, pp. 1–8.

[3] I. Haritaoglu, D. Harwood, and L. Davis, "w^4: Real-time surveillance of people and their activities," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 809–830, Aug. 2000.

[4] Y. Ran, I. Weiss, Q. Zheng, and L. Davis, "Pedestrian detection via periodic motion analysis," *Int. J. Comput. Vis.*, vol. 71, no. 2, pp. 143–160, Feb. 2007.

[5] P. Viola, M. Jones, and D. Snow, "Detecting pedestrians using patterns of motion and appearance," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, New York, 2003, vol. 1, pp. 734–741.

[6] P. Felzenszwalb and D. Huttenlocher, "Pictorial structures for object recognition," *Int. J. Comput. Vis.*, vol. 61, no. 1, pp. 55–79, Jan. 2005.

[7] S. Ioffe and D. A. Forsyth, "Probabilistic methods for finding people," *Int. J. Comput. Vis.*, vol. 43, no. 1, pp. 45–68, Jun. 2001.

[8] R. Ronfard, C. Schmid, and B. Triggs, "Learning to parse pictures of people," in *Proc. Eur. Conf. Comput. Vis.*, Copehagen, Denmark, 2002, vol. 4, pp. 700–714.

[9] K. Mikolajczyk, B. Leibe, and B. Schiele, "Multiple object class detection with a generative model," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, New York, 2006, vol. 1, pp. 26–36.

[10] A. Opelt, A. Pinz, and A. Zisserman, "Incremental learning of object detectors using a visual shape alphabet," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, New York, 2006, vol. 1, pp. 3–10.

[11] D. Gavrila and V. Philomin, "Real-time object detection for smart vehicles," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Fort Collins, CO, 1999, pp. 87–93.

[12] L. Zhao and L. S. Davis, "Closely coupled object detection and segmentation," in *Proc. ICCV*, 2005, pp. 454–461.

[13] S. Munder and D. M. Gavrila, "An experimental study on pedestrian classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 11, pp. 1863–1868, Nov. 2006.

[14] P. Papageorgiou and T. Poggio, "A trainable system for object detection," *Int. J. Comput. Vis.*, vol. 38, no. 1, pp. 15–33, Jun. 2000.

[15] A. Mohan, C. Papageorgiou, and T. Poggio, "Example-based object detection in images by components," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 4, pp. 349–360, Apr. 2001.

[16] K. Mikolajczyk, C. Schmid, and A. Zisserman, "Human detection based on a probabilistic assembly of robust part detectors," in *Proc. Eur. Conf. Comput. Vis.*, Prague, Czech Republic, 2004, vol. 1, pp. 69–81.

[17] B. Leibe, E. Seemann, and B. Schiele, "Pedestrian detection in crowded scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, San Diego, CA, 2005, vol. 1, pp. 878–885.

[18] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, San Diego, CA, 2005, pp. 886–893.

[19] N. Dalal, B. Triggs, and C. Schmid, "Human detection using oriented histograms of flow and appearance," in *Proc. Eur. Conf. Comput. Vis.*, Graz, Austria, 2006, pp. 428–441.

[20] P. Felzenszwalb, D. McAllester, and D. Ramanan, "A discriminatively trained, multiscale, deformable part model," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Anchorage, AK, 2008, pp. 1–8.

[21] D. Tran and D. Forsyth, "Configuration estimates improve pedestrian finding," in *Proc. NIPS*, 2007, pp. 1529–1536.

[22] B. Wu and R. Nevatia, "Detection of multiple, partially occluded humans in a single image by Bayesian combination of edgelet part detectors," in *Proc. 10th Int. Conf. Comput. Vis.*, Beijing, China, 2005, pp. 90–97.

[23] P. Sabzmeydani and G. Mori, "Detecting pedestrians by learning shapelet features," in *Proc. CVPR*, 2007, pp. 1–8.

[24] B. Wu and R. Nevaita, "Optimizing discrimination-efficiency tradeoff in integrating heterogeneous local features for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Anchorage, AK, 2008, pp. 1–8.

[25] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recog.*, 2001, pp. 511–518.

[26] J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: A statistical view of boosting," *Ann. Stat.*, vol. 28, no. 2, pp. 337–407, 2000.

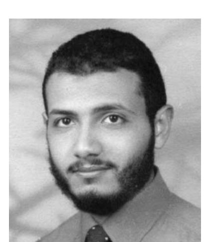
[27] O. Tuzel, F. Porikli, and P. Meer, "Region covariance: A fast descriptor for detection and classification," in *Proc. ECCV*, 2006, pp. 589–600.

[28] F. Porikli, "Integral histogram: A fast way to extract histogram features," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recog.*, 2005, pp. 829–836.

[29] M. Hussein, F. Porikli, and L. Davis, "Kernel integral images: A framework for fast non-uniform filtering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Anchorage, AK, 2008, pp. 1–8.

[30] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, *The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results*. [Online]. Available: http://www.pascalnetwork.org/challenges/VOC/voc2007/workshop/index.html

[31] J. Davis and M. Goadrich, "The relationship between precision-recall and ROC curves," in *Proc. ICML*, 2006, pp. 233–240.



Mohamed Hussein (S'08) received the B.Sc. and M.Sc. degrees in computer science from Alexandria University, Alexandria, Egypt, in 1998 and 2002, respectively, and the M.Sc. degree in computer science from the University of Maryland, College Park, in 2005. He is currently working toward the Ph.D. degree in computer science with the Department of Computer Science, University of Maryland. His Ph.D. research involves object detection and vision computing on graphics processing units.

Prior to starting research in computer vision in 2004, his background was in systems and networking. He spent nine months, split between 2007 and 2008, as an Intern with the Mitsubishi Electric Research Laboratories, Cambridge, MA. His current research interests include large-scale learning on modern parallel architectures with applications in computer vision.

Fatih Porikli (SM'04) received the Ph.D. degree, with specialization in video object segmentation, from the Polytechnic Institute of New York University, Brooklyn, NY.

He is currently a Senior Principal Research Scientist and Project Manager with Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA. Before joining MERL in 2000, he developed satellite image applications at Hughes Research Laboratories, Malibu, CA, in 1999 and 3-D systems at AT&T Research Laboratories, Middletown, NJ, in 1997. He is an Associate Editor for two journals. He has chaired more than a dozen workshops and is a member of the organizing committees of several flagship conferences, including ICCV, ECCV, CVPR, ISVC, ICIP, AVSS, ICME, and ICASSP. He also served as Area Chair of CVPR 2009, IV 2008, and ICME 2006. He has authored more than 80 technical publications. He is the holder of more than 50 pending patents. His current research interests include pattern recognition, biomedical data analysis, online learning and classification, computer vision, robust optimization, multimedia processing, and data mining, with many commercial applications ranging from surveillance to medical to intelligent transportation systems.

Dr. Porikli is a Senior Member of the Association for Computing Machinery and The International Society for Optical Engineers. He is the organizer of the 2010 IEEE AVSS Conference as the general chair. He was the recipient of the R&D 100 Scientist of the Year Award in 2006, the Best Paper Runner-Up Award at the IEEE International Conference on Computer Vision and Pattern Recognition, the Most Popular Scientist Award in 2007, and the Superior Invention Award from MELCO in 2008 and 2009.

Larry Davis (F'98) received the Ph.D. degree from the University of Maryland, College Park, in 1976.

He is currently a Professor and the Chair of the Department of Computer Science, University of Maryland, and a Professor with the Institute for Advanced Computer Studies (UMIACS). He is a former Director of UMIACS and the former Head of the Computer Vision Laboratory. From 1977 to 1981, he was an Assistant Professor of computer science with the University of Texas, Austin. In 1981, he returned to the University of Maryland as an Associate Professor. He has authored more than 200 papers on computer vision and high-performance computing. His current research interests include visual surveillance, particularly the modeling and recognition of human movement and activity.

Prof. Davis is a Fellow of the International Association for Pattern Recognition. He is currently serving on the Information Science and Technology committee of the Defense Advanced Research Projects Agency.