Fast Human Detection by Boosting Histograms of Oriented Gradients

Hui-Xing Jia, Yu-Jin Zhang
(Department of Electronic Engineering, Tsinghua University, Beijing 100084, China)
（jiahx03@mails.tsinghua.edu.cn, zhang-yj@mail.tsinghua.edu.cn）

Abstract

In this paper, a novel real-time human detection system based on Viola's face detection framework and Histograms of Oriented Gradients (HOG) features is presented. Each bin of the histogram is treated as a feature and used as the basic building element of the cascade classifier. The system keeps both the discriminative power of HOG features for human detection and the real-time property of Viola's face detection framework. Experiments on DaimlerChrysler pedestrian benchmark data set and INRIA human database demonstrate that this framework is more powerful than Viola's object detection framework on human detection.

1. Introduction

Human detection in images has becoming an increasingly important research area in both computer vision and pattern recognition community because of its potential applications in video surveillance, driving assistance system and content-based image retrieval. However, human detection is a challenging problem due to the various appearances caused by different clothes and articulations of body parts, as well as varying backgrounds.

In this paper, we present a novel real-time human detection system by integrating Viola's famous object detection framework [1] and Histograms of Oriented Gradients (HOG) features [2]. We treat each bin of the histogram as an individual feature and build the whole system using the new feature pool. Each feature in our system can be evaluated in 8 look-ups with the help of integral images, so it costs about the same time as a haar feature. By substituting the Haar features with the HOG features, our system keeps the speed advantage of Viola's object detection framework as well as the discriminative power of HOG features on human detection. Experiments demonstrate that our system achieves a better accuracy at nearly the same speed as original haar features for human detection.

The rest of this paper is organized as follows: section 2 reviews related work; section 3 gives the architecture of our human detection framework; section 4 details the definition and evaluation of our HOG feature pool; section 5 demonstrates the experimental results and finally section 6 concludes the paper.

2. Previous work

Despite all the difficulties on human detection, a lot of work has been done recent years. Previous methods differ in three perspectives: first, they may use different features such as edge, haar features and gradient orientation features; second, they may use different classifiers such as Nearest Neighbor, Neutral Network, SVM and Adaboost; third, they may treat the image region as a whole or detect each part first and then combine them by these parts' geometrically configurations. We classify previous methods into three categories based on the features they use.

Edge features have been used in earlier work. Gavrila and Philomin [3] uses edge template as the feature and compare edge images to a template dataset using the chamfer distance. This method has been experimented on a vehicle of DaimlerChrysler. Edge feature is affected by background clutter greatly and not very robust.

Haar features have been used successfully in face detection and also adopted by a lot of researchers for human detection. Oren et al. [4] combine over-complete haar features and SVM classifiers to detect pedestrians. Mohan et al. [5] extend Oren's work using a cascade SVM to detect human component first and then vote for a human. Viola et al. [6] extend the haar features to capture spatial-temporal information for moving-human detection in surveillance system but adopt the boosted cascade classifier.

Recently, gradient orientation features such as SIFT descriptor [7] and HOG descriptors [2] have attracted more attention. Shashua et al. [8] manually divide the human into 13 regions and compute SIFT-like features of each region, then combine these features using Adaboost to train the classifier and detect pedestrians on a moving

0-7695-2929-1/07 $25.00 © 2007 IEEE
DOI 10.1109/ICIG.2007.53

IEEE computer society

vehicle. Dalal and Triggs [2] propose HOG features as human representation, which achieve amazing good results combined with SVM classifiers. Later they extend their approach to detect humans in video streams using oriented histograms of flow and appearance [9]. Zhu et al. [10] integrate the cascade-of-rejecters approach with HOG features to speed up Dalal's method greatly, using linear SVM as weak classifier. There are also other systems that use gradient orientation features but adopt a parts-based approach that aims at dealing with the great variability in appearance due to body articulation or occlusion. For example, Mikolajczyk et al. [11] represent human parts as co-occurrences of local orientation features. Their system proceeds by detecting features, then parts and eventually humans are detected based on assemblies of the parts.

3. The framework

Our framework is the same as Viola's famous face detection framework except the feature evaluation part. For face detection, the Haar feature is more appropriate because the grey pattern of face is very obvious. For example, the eyes are always darker than their surrounding areas. But for human detection, we can only rely on the shape of the human, so HOG feature is more appropriate. The opinion has been proved in the work of Dalal et al. [2] and Zhu et al. [10]. But in their work, each block is treated as a building element, which can not fit very well with Viola's face detection framework. In order to keep Viola's speed advantage, we build our human detection system in a much lower level. Each feature is defined by its owing block position, its cell position and the orientation bin. The new feature can be evaluated in 8 look-ups using integral images. We extract the HOG features of multi-scale blocks and use the new feature pool to build the system, so the speed advantage of the whole system is kept.

The architecture of the system is shown in Figure 1. It consists of a cascade of stage classifiers whose detection rate is very high and false alarm rate is medium. By a cascade of such classifiers, the detection rate of the whole system is high and the false alarm rate is extremely low. For example, if the detection rate and false alarm rate is 0.995 and 0.5 for each stage, the detection rate of a 20 stage system is about 90% while the false alarm rate is only 10^{-6}. Most importantly, most of the background patches will be discarded in earlier stages of the cascade, increasing the detection rate greatly. The stage classifier itself consists of an ensemble of Classification and Regression Tree (CART) as weak classifiers combined by Adaboost. During training, the threshold of each stage classifier is adjusted and the number of weak classifiers is increased until the hit rate and false alarm rate meet predefined values. The CART classifier is trained aggressively, i.e., the leaf of the tree that decreases the error most will be split and corresponding feature and threshold will be saved as the parameters of the node. The node number of each CART can be used as a meta variable to control the complex of the whole system.

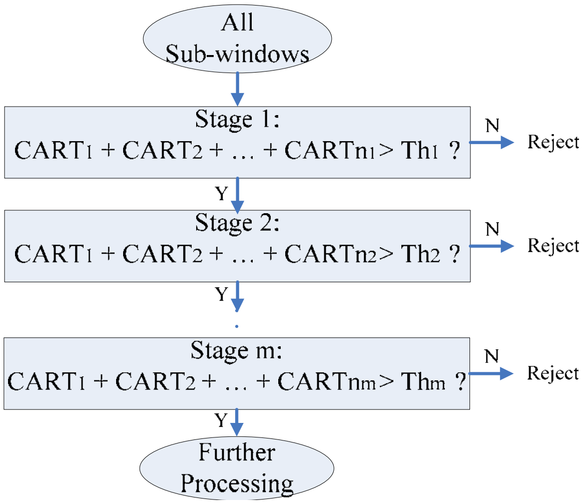

All
Sub-windows
Stage 1:
CART1 + CART2 + … + CARTn1> Th1 ?
N
Reject
Y
Stage 2:
CART1 + CART2 + … + CARTn2> Th2 ?
N
Reject
Y
Stage m:
CART1 + CART2 + … + CARTnm> Thm ?
N
Reject
Y
Further
Processing

Figure 1. **System architecture**

4. The HOG feature pool

The main purpose of using features instead of raw pixels as the input to a learning algorithm is to encode knowledge about the domain, which is difficult to learn from raw and finite set of input data. Take human detection for example, if we use raw pixel value as input, then the in-class variation caused by illumination will make the classification task very hard. From our knowledge, human shape is invariant to illumination, so we can use HOG features to encode shape information. By treating each bin of HOG features as a feature, we create a new feature pool of discriminative features for human detection used by Adaboost algorithm and cascade training.

The speed of feature evaluation is also an important aspect because most object detection algorithms slide a fixed-size window at all scales over the input image and need to evaluate several thousands of image patches. Each feature in our feature pool can be computed at any position and any scale in the same constant time, only 8 look-ups is needed, which enables the real time properties of our human detection system.

4.1. Feature definition

Each feature is defined by its cell position $C\left(x_c, y_c, w_c, h_c\right)$, the parent block position

$B\left(x_b, y_b, w_b, h_b\right)$ and the orientation bin number k, so each feature f is denoted by $f\left(C, B, k\right)$. The gradients at the point $\left(x, y\right)$ of image I can be found by convolving gradient operator with the image [2]:

$$G_x\left(x, y\right) = \left[-1\,0\,1\right] * I\left(x, y\right), \tag{1}$$

and

$$G_y\left(x, y\right) = \left[-1\,0\,1\right]^T * I\left(x, y\right). \tag{2}$$

The strength of the gradient at the point $\left(x, y\right)$ is

$$G\left(x, y\right) = \sqrt{G_x\left(x, y\right)^2 + G_y\left(x, y\right)^2}. \tag{3}$$

The orientation of the edge at the point $\left(x, y\right)$ is

$$\theta\left(x, y\right) = \arctan\left(\frac{G_y\left(x, y\right)}{G_x\left(x, y\right)}\right). \tag{4}$$

We divide the orientation range $\left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$ into K bins and denote the value of k_{th} bin to be

$$\psi_k\left(x, y\right) = \begin{cases} G\left(x, y\right) & \text{if } \theta\left(x, y\right) \in bin_k \\ 0 & otherwise \end{cases}. \tag{5}$$

Then the feature value is defined as

$$f\left(C, B, k\right) = \frac{\sum_{(x,y)\in C} \psi_k\left(x, y\right) + \varepsilon}{\sum_{(x,y)\in B} G\left(x, y\right) + \varepsilon}. \tag{6}$$

4.2. Feature evaluation

All the features can be computed very fast and in constant time by 8 lookups with the help of K+1 auxiliary integral images

$$\begin{aligned} IG_k\left(x, y\right) &= \sum_{x' \le x, y' \le y} \psi_k\left(x, y\right), \quad k = 1, \ldots, K \\ IG\left(x, y\right) &= \sum_{x' \le x, y' \le y} G\left(x, y\right) \end{aligned}, \tag{7}$$

Then

$$\begin{aligned} &\sum_{(x,y)\in C} \psi_k\left(x, y\right) \\ &= IG_k\left(x_c - 1, y_c - 1\right) + IG_k\left(x_c + w_c - 1, y_c + h_c - 1\right) \\ &\quad - IG_k\left(x_c - 1, y_c + h_c - 1\right) - IG_k\left(x_c + w_c - 1, y_c - 1\right) \cdot \\ &\sum_{(x,y)\in B} \theta\left(x, y\right) \\ &= IG\left(x_b - 1, y_b - 1\right) + IG\left(x_b + w_b - 1, y_b + h_b - 1\right) \\ &\quad - IG\left(x_b - 1, y_b + h_b - 1\right) - IG\left(x_b + w_b - 1, y_b - 1\right) \end{aligned} \tag{8}$$

So each feature $f\left(C, B, k\right)$ can be evaluated in 8 lookups.

4.3. Feature pool build

If we do not put any constraint on the relative position and size of the cell and the block, the feature number will be too large. Inspired by the extended haar feature definition of Lineart and Maydt [12], we only consider the relative position of the cell and the block shown in Figure 2. The black rectangle represents the position of the cell while the whole large rectangle denotes the position of the block. The ration between the width and the height of the block is 1:2, 1:1 and 2:1. For a predefined training sample size, the feature template can be moved at a predefined stride step *st* and enlarged at a scale step *sc*. The minimum of the block is denoted by $w_{\min}$ and $h_{\min}$. The maximum of the block is just as large as the size of the training sample. The number of the feature pool is very large, for example if $w_{\min}$=8, $h_{\min}$=8, *st*=0.5, *sc*=1.2, *K*=9, and the training sample size is 18×36, the whole feature pool contains more than ten thousands features.

We should note here that some similar feature pools have been defined in previous work. For example, the first line of Figure 2 is similar to the dominate orientation feature pool proposed by Levi and Weiss [13]. The second line of Figure 2 is similar to the feature set of Dalal's [2] and Zhu et al. [10]. However, the feature pool defined here is much richer than those previous feature sets because we can add any new feature pattern here. The definition of this new feature pool is inspired the definition of extended haar features. Like the extended haar features, we can also add some rotated patterns, but they have not been included in the implementation right now.

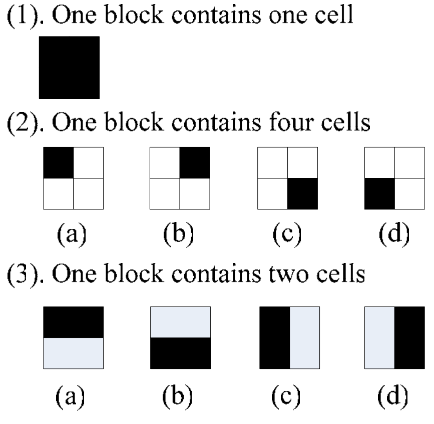

(1). One block contains one cell
(2). One block contains four cells
(a)
(b)
(c)
(d)
(3). One block contains two cells
(a)
(b)
(c)
(d)

Figure 2. **HOG feature templates**

5. Experiments

There are some striking differences in the classification and detection performance of different systems reported in the literature. The variations come from different training data sets and different test criteria. In order to make objective and fair comparisons, we use two publicly available databases to compare the classification and detection performance of different systems.

For comparisons of classification performance, we use the pedestrian benchmark data set proposed by Muder and Gavrila [14]. The content of the dataset is shown in Table 1. We use similar performance analysis procedure as in [14]: for each system, three different classifiers are trained, each by selecting one out of the three training sets. Testing the three classifiers on the two test sets yields six different ROC curves, i.e., six different detection rates for each possible number of false positives. Then the mean detection rate and corresponding 0.95 confidence interval is computed. We use only one training set for each classifier instead of two in [14] to make the six ROCs more independent. The higher of the ROC curve, the better of the system.

For comparisons of detection performance, we use the database and test criteria proposed by Dalal and Triggs [2]. The content of the database are shown in Table 2. For each system, one classifier is trained using the training set and the Detection Error Tradeoff (DET), i.e. miss rate versus FPPW curve on the test set is computed. The lower of the DET curve, the better of the system. In the following sections, we will propose two experiments. The first experiment demonstrates the effect of two important parameters of our system. The second experiment compares our system with previous systems.

5.1. The effect of different parameters

In this experiment we test the effect of two important parameters on our HOG-Adaboost-Cascade system: bin number K and the number of nodes of CART. The default values of the two parameters are 9 and 1 respectively. Other parameters are $w_{\min}$=4, $h_{\min}$=4, st=0.5, sc=1.2. When changing one parameter, the other parameters take the default value.

When training the two cascade systems, the minimum hit rate and maximum false alarm rate is 0.99 and 0.6 for each stage, totally 16 stages are trained; Gentle Adaboost are used because of its numerical stability; 4800 positive samples are taken from one positive sample set out of three; 5000 negative samples for the first stage of the cascade are generated randomly from one of the additional non-ped images training sets; negative images for subsequent stages are generated from the same additional non-ped images by bootstrapping using trained classifier. Points of each ROC are got by changing the number of stages used. When compute final ROC with confidence interval, spline interpolation is used when necessary. The results are shown in Figure 3.

From Figure 3 (a) we can see when K increase from 3 to 6, the ROC becomes better. However, when it changes from 6 to 9, the ROC only increase very little at the low false positive side. So there is no necessary to increase K. From Figure 3 (b), we can see that when the node number is 1, i.e. stump classifier is best.

5.2. Comparisons with previous systems

In order to build an efficient human detector, our HOG-Adaboost-Cascade system absorbs the advantages of Viola's Haar-Adaboost-Cascade system and Dalal's HOG-SVM-Bootstrapping system, so we compare the classification and detection performance of the three systems. For viola's system, we use the implementation in OpenCV. For Dalal's system we use their implementation in Linux.

First, we compare their classification performance using DaimlerChrysler pedestrian data sets. When training our HOG-Adaboost-Cascade system, we use the default parameters in section 5.1. When training Viola's Haar-Adaboost-Cascade system, we take the same parameters as HOG-Adaboost-Cascade system for classifier architecture and all the features except the rotated. When training Dalal's HOG-SVM-Bootstrapping system, we first train an initial classifier use both the positive and negative samples of each training set shown in Table 1 and then get more negative samples by bootstrapping from corresponding add-on non-ped images. Since our template is only 18x36, so the minimum cell is changed to 4x4 instead of 8x8 in Dalal's origin implementation. The results are shown in Figure 4(a). Second, we compare their detection accuracy and speed using the INRIA human database. For the two cascade systems, two 16 stage classifiers are trained, the minimum hit rate and max false alarm rate for each stage is 0.995 and 0.5. Then we test the three systems on the test set. The DET curves are shown in Figure 4 (b).Some results are shown in Figure 5. The time cost of the three systems when processing a 320x240 image can be seen in Table 3.

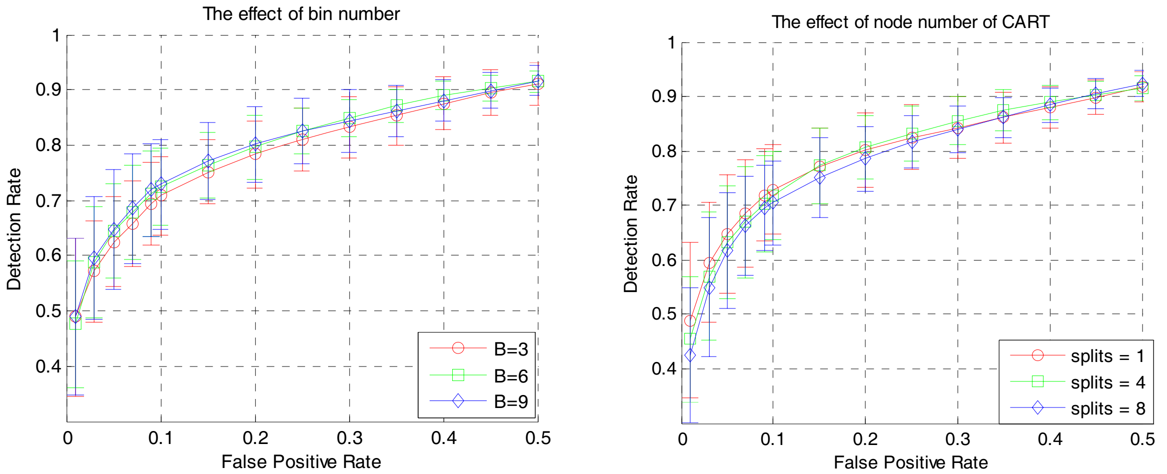
From Figure 4 and Table 3, we can see that our system achieve much better detection rate at comparable speed as Viola's system. However, our system is inferior to Dalal's system; this is because we simplify the feature evaluation part too much. Since most of the processing time is spent in the computation of the auxiliary integral images, so there is very little time difference when using sparse scan and dense scan. This explains why our system is much slower than Viola's when using sparse scan and comparable when using dense scan.

Table 1. **DaimlerChrysler pedestrian data set [14]**

	#Data sets	Pedestrian Labels Per Set	Pedestrian Examples Per Set	Non-Pedestrian Examples Per Set	Additional Non-Ped Images
Training Sets	3	800	4800	5000	1200
Test Sets	2	800	4800	5000	0

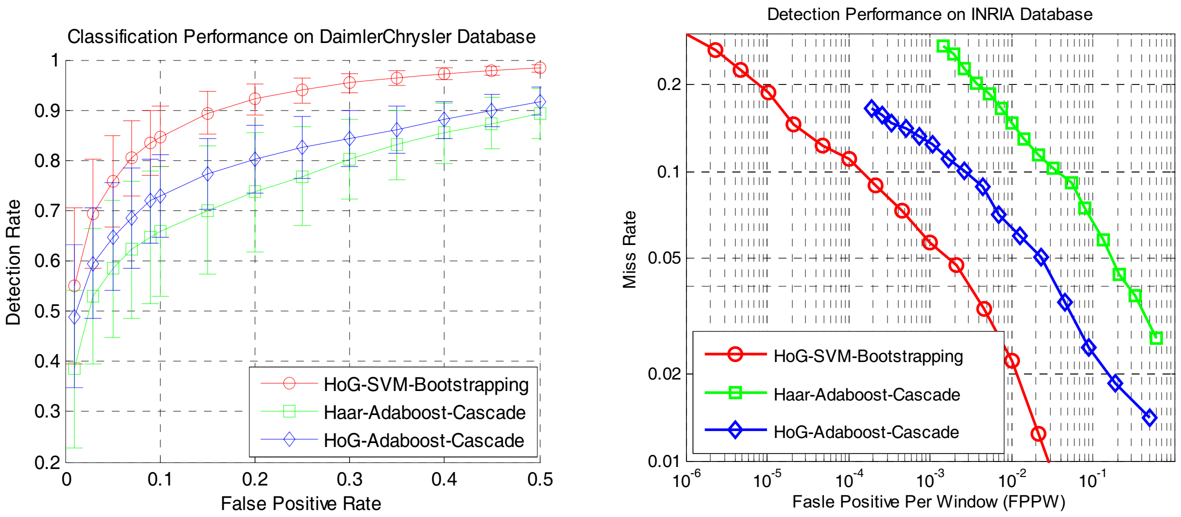
Table 2. **INRIA human data set [2]**

	#Data sets	Human Labels Per Set	Human Examples Per Set	Non-Human Examples Per Set	Additional Non-Human Images
Training Set	1	1208	2416	0	1218
Test Set	1	566	1132	0	453



(a) (b)

Figure 3. **Effect of different parameters. (a) Effect of bin number. (b) Effect of node number of CART.**



(a) (b)

Figure 4. **Comparisons with previous systems. (a) Classification performance on DaimlerChrysler Pedestrian Database. (b)Detection performance on INRIA Human Database**

Table 3. **Speed comparisons of three different systems**

	Sparse scan (800 windows per image)	Dense scan (12800 windows per image)
HOG-SVM-Bootstrapping	300ms	5sec
Haar-Adaboost-Cascade	5ms	32ms
HOG-Adaboost-Cascade	29ms	51ms

6. Conclusion

In this paper, we present a novel real-time human detection system based on Viola's face detection framework and the HOG feature pool. The system keeps the discriminate power of HOG features and the real-time properties of Viola's face detection framework. Besides human, the detection framework can also be used to detect other objects such as vehicles. Future research includes the integration of Haar feature and HOG feature into a framework for generic object detection and the detection of human in video streams.

Acknowledgements

This work has been supported by the Grants NNSF-60573148 and SRFDP-20060003102.

References

[1] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. *CVPR*, 2001.

[2] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. *CVPR*, 2005.

[3] D. Gavrila and V. Philomin. Real-time object detection for "smart" vehicles. *ICCV*, 1999.

[4] M. Oren, C. Papageorgiou, P. Sinha, E. Osuna and T. Poggio. Pedestrian Detection Using Wavelet Templates. *CVPR*, 1997.

[5] A.Mohan, C. Papageorgiou, T. Poggio, and T. Poggio. Example-based object detection in images by components. *PAMI*, 23(4):349–361, 2001.

[6] P. Viola, M. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. *ICCV*, 2003.

[7] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[8] A. Shashua, Y. Gdalyahu, and G. Hayon. Pedestrian detection for driving assistance systems: Single-frame classification and system level performance. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, 2004.

[9] N. Dalal, B. Triggs and C. Schimid. Human Detection Using Oriented Histograms of Flow and Appearance. *ECCV*, 2006.

[10] Q. Zhu, S. Avidan, M. C. Yeh, and K. T. Cheng. Fast Human Detection Using a Cascade of Histograms of Oriented Gradients. *CVPR*, 2006.

[11] K. Mikolajczyk, C. Schmid, and A. Zisserman. Human detection based on a probabilistic assembly of robust part detectors. *ECCV,* 2004.

[12] R. Lienhart and J. Maydt. An Extended Set of Haar-like Features for Rapid Object Detection. *ICIP*, 2002.

[13] K. Levi and Y. Weiss. Learning Object Detection from a Small Number of Examples: the Importance of Good Features. *CVPR*, 2003.

[14] S. Munder and D. M. Gavrila. An Experimental Study on Pedestrian Classification. *PAMI*, 28(11):1863-1868, 2006

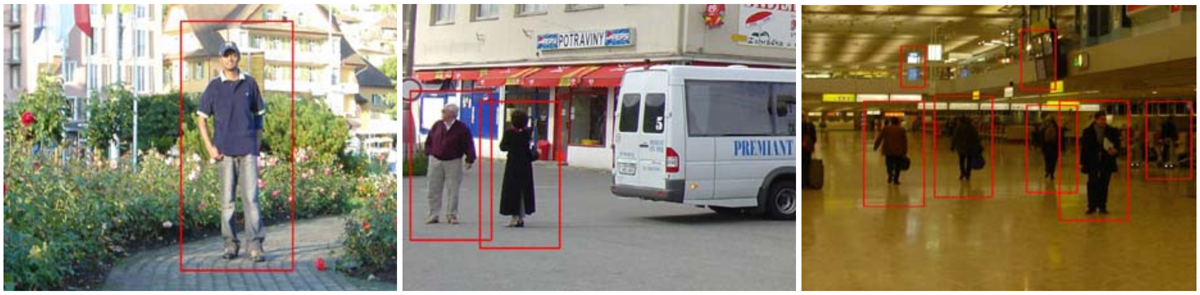


Figure 5. **Some of the detection results of INRIA human database**