Pattern Recognition 45 (2012) 2573–2583

Contents lists available at SciVerse ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Pedestrian detection in images via cascaded L1-norm minimization learning method

Ran Xu [a], Jianbin Jiao [a,*], Baochang Zhang [b], Qixiang Ye [a]

[a] Graduate University of Chinese Academy of Sciences, Beijing 100049, China
[b] Science and Technology on Aircraft Control Laboratory, ASEE, Beihang University, Beijing 100191, China

ARTICLE INFO

Article history:
Received 14 December 2010
Received in revised form
4 January 2012
Accepted 8 January 2012
Available online 20 January 2012

Keywords:
Pedestrian detection
L1-norm minimization
Feature selection
Cascaded classifier

ABSTRACT

A new cascaded L1-norm minimization learning (CLML) method for pedestrian detection in images is proposed in this paper. The proposed CLML method, which is designed from the perspective of Vapnic's theory in the statistical learning, integrates feature selection with classifier construction via solving meaningful optimization models. The method incorporates three stages: weak classifier learning, strong classifier learning and the cascaded classifier construction. In the weak classifier learning, the L1-norm minimization learning (LML) and min–max penalty function model are presented. In the strong classifier learning, an integer programming optimization model is built, equaling the reformulation of LML in the integer space. Finally, a cascade of LML classifiers is constructed to promote detection speed. During the classifier learning and pedestrian detection, Histograms of Oriented Gradients of variable-sized blocks (v-HOG) are used as feature descriptors. Experimental results on the INRIA and SDL human datasets show that the proposed method achieves a higher performance and speed than the state-of-the-art methods.

1. Introduction

Detecting objects in images and videos is one of the fundamental tasks of pattern recognition and computer vision. It has many important applications in robot vision, visual surveillance, image retrieval and driving assistant systems, etc. Pedestrian detection is regarded as one of the most difficult and typical problems in object detection owing to the various appearance and pose of a human body together with the cluttered background under different illuminations.

Extracting more effective features and developing more powerful learning algorithms (classifiers) have always been the pursuits of researchers for pedestrian detection. In this paper, we focus on developing a powerful feature selection and classification method in a comprehensive way.

In the field of statistical learning, VC-dimension is one of the core concepts, which measures the generalization capability of a classification function set. However, it sometimes is hard to quantitatively measure the VC-dimension. Consequently, the other optimization principles are chosen to replace VC-dimension. In the classifier construction, the margin maximization is the state-of-the-art method, which is in fact as an alternative of pursuing VC-dimension minimization [1,2]. SVMs and Adaboost are the state-of-the-art classifiers for pedestrian detection. One of the most important reasons for the success of SVMs and Adaboost is that both methods aim to directly or indirectly maximizing the margin. In this paper, we make an attempt to design the classifiers in the light of pursuing a smaller VC-dimension.

We design a cascaded classifier by incorporating the principle of L1-norm minimization into the test error upper bound. The intuition that we adopt L1-norm comes from the successful application of L1-norm in the fields of face recognition [5], human detection [19,25] and compressive sensing of signals [3,4,22] in recent years. More importantly, in the field of signal processing, L1-norm minimization can be considered as an approximately optimal implementation of the L0-norm minimization [3]. In addition, the early studies of L1-norm are also presented in [37–39]. In [37], the object function in optimization model is comprised of the L1-norm term of the point average violation and an appended term. The appended term is expressed as the number of nonzero elements of the weight vector (the L0-norm of the weight vector), and is approximated by the concave exponential function. It is an earlier prototype using L1-norm to measure the violation degree of points. It adopts the concave exponential function to facilitate to solve the model, which is different from our object function of the weak classifier. In [38], the authors propose L1-norm support vector machine and introduce an efficient

* Correspondence to: Graduate University of Chinese Academy of Sciences, Room 307, Union Building, No. 19A, Yuquan Road, Shijing Shan District, Beijing 100049, China. Tel.: +86 10 88256968.
E-mail address: jiaojb@gucas.ac.cn (J. Jiao).

0031-3203/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2012.01.004

approach to solve it. In [39], L1-norm support vector machine is formulated by the unconstrained convex differentiable minimization, which is solved by applying a Newton method. These works in [38,39] present the L1-norm support vector and mainly research how to solve L1-norm support vector machine. Both of these optimization models accord with our weak classifiers. We more focus on using the sparseness of the L1-norm to select features and simultaneously construct classifiers for human detection.

During the construction of our classifiers, the weak classifiers are learned by the L1-norm minimization principle. The min–max penalty function is employed to determine the appropriate thresholds for the weak classifiers. After obtaining the weak classifiers, we utilize the integer programming optimization model to select the minimal number of them to construct a strong classifier and simultaneously select the most compact features. The cascade mechanism is employed to achieve high detection speed, in which the number of cascade is added until expected performance is met. The final classifier inherits the advantages of both cascade and L1-norm minimization learning method, and obtains higher performance on classification accuracy and efficiency, which are validated on two pedestrian detection datasets.

The contributions of our work are summarized as follows:

1) Construct a classifier from the perspective of the upper bound of error via VC-dimension.
   Both weak and strong classifiers are constructed to pursue a smaller error upper bound via VC-dimension. The weak classifiers are trained via L1-norm minimization learning and the strong ones are constructed by the integer programming optimization. The integer programming can be viewed as a special case of L1-norm in integer space. The relationship between the L1-norm minimization and VC-dimension is explained. The result is that the L1-norm minimization can contribute to a smaller upper bound of error via VC-dimension and then improve the generalization ability of the classifiers.
2) Fuse feature selection with classifier construction for pedestrian detection in a new way.
   Feature selection and the strong classifier construction are achieved simultaneously. Inspired by weighted voting principle stemming from the Adaboost method, we utilize a linear weighted combination of weak classifiers to construct a strong classifier. The difference between our method and the Adaboost lies in two aspects. Firstly, the way to select features is different. Feature selection of the Adaboost is performed with an iterative greedy strategy while ours is with the integer programming. The integer programming can find out the optimal combination of features. Moreover, this way can reduce redundancy and contributes to efficiency. Secondly, our approach to determine the thresholds of weak classifiers solved by the min–max function is simple and flexible, which is also different from that of the Adaboost.

The rest of this paper is organized as follows. Related work is introduced in Section 2. The feature representation of pedestrian is described in Section 3. The CLML method for pedestrian detection is presented in Section 4. The experiments are presented in Section 5 with conclusions in Section 6.

2. Related work

Two main processing steps are utilized in a typical pedestrian detection algorithm. One step is feature representation during which the descriptor is extracted to represent the human body, and the other is classification model with which the extracted descriptors of a region are used to detect whether the region contains a human body.

In the aspect of feature representation, various features are proposed to represent a human body. Some shape clues [6,26] draw more attention. Complex human shape models are learned from the shape contour examples modeled by discrete and continuous representation methods [26]. Non-adaptive Haar-like wavelet features based on the local intensity differences have been proposed by Papageorgiou and Poggio [27], which are further improved by other researchers [7,18,28,35]. In [7,35], the over completed Haar-like wavelet features are utilized to represent a face and a pedestrian at various locations on different scales. Later, some adaptive features considering the particular configuration of spatial constraints are proposed by Munder and Gavrila [20], and Szarvas et al. [31]. A typical one of such features is the local receptive fields [20] simulating the neural structures of human visual cortex [32]. The well-known dense histograms of oriented gradients (HOG) descriptors in [8] are proposed to capture the local contours of a pedestrian. The HOG descriptors of each block are computed on a fixed scale at a fixed location to save computational cost. Finally, many variants of HOG are presented in [9,12,14]. These descriptors based on gradient orientations are extracted on variable-size blocks and different locations. Results from their reports are better than the original HOG descriptors. In [29], some color clues are captured as the descriptors of objects. Tuzel et al. [10] utilize covariance (COV) features as the pedestrian descriptors. A local image region is represented by the covariance matrix of point descriptor which consists of intensity, location, derivatives, etc. Mu et al. [11] propose the improved LBP to represent human by considering geometrical position and frequency information. In [12], the authors combine the HOG with LBP descriptors to characterize the local and global clues of a human body. Moreover, local clues are implemented to handle the occlusion problem. In [13], edgelet features consisting of silhouette oriented features are introduced as human part descriptors. All part descriptors are combined to form a human model. In [33], high-dimensional descriptors containing edge-based features with texture and color are utilized to represent the human body.

After obtaining feature representation, some methods have been employed and developed for feature selection and classification for pedestrian detection. In [33], the authors employ partial least squares (PLS) to perform feature dimension reduction, and then use SVMs to classify pedestrians. In [6], the statistical field model is utilized to characterize the shape variation of pedestrians and classify pedestrians. In [7], the authors propose cascade mechanism and use Adaboost cascade to select features and make a classification. Since then, the cascade structure has been widely used to detect objects. In [40], the paper proposes to integrate cascade structure with multiple instance learning (MIL) in a modified "min–max and L1-norm" framework to detect the diseased structure in medical images.

In our most recent work [19], a linear classifier for human detection based on L1-norm minimization is proposed. Although it can perform better than some existing methods, however, it still cannot attain a higher detection rate. Our later work [25] has been extended and a cascaded LML are designed to perform feature selection of blocks and obtain better performance for human detection. In [10], the authors firstly transform the features into tangent space of Riemannian manifolds, and then use a cascade of Logitboost for classification. In [8,11,18,30] linear or kernel SVM is employed for classification. In [9], the authors use linear SVM to form weak classifiers and then build an Adaboost cascade mechanism for pedestrian detection. A multilayer neural network has been introduced into pedestrian detection using adaptive local receptive field features [31]. Regarding the specificity and difficulty of a pedestrian detection problem,

many pedestrian detection approaches [13,15–18,41–43] propose to break down the appearance of the human body into parts. Furthermore, additive combinations of classifiers are utilized to detect pedestrians [36]. In addition, Lin et al. [34] combine local part-based and global template-based schemes to detect pedestrians via Bayesian framework. In [41], the pictorial structures are proposed to partition a human body into head, torso, leg area, etc. Andriluka et al. [42] use pictorial structures to detect humans and estimate the pose of humans. Later, Felzenszwalb et al. [43] combine the global coarse detection with local pictorial structure for human detection and obtain a better result.

In terms of the work mentioned above, the classification methods for pedestrian detection generally can be divided into two categories. One is probability and reasoning method such as Bayesian Reasoning and the other is deterministic methods such as template matching, Neural Networks, Adaboost and SVMs, etc. Template matching methods employ some rules (distance and so on) to measure the similarity of feature vectors in a feature space. Neural Networks has many extensions according to different network structures. Most of them evaluate the optimal decision boundary by minimizing an error criterion with regard to some network parameters. In contrast to Neural Networks, SVMs [1] do not minimize the error metric but maximize the margin of a linear decision hyper plane. To cope with the samples not distinguished by a linear classifier, SVMs employ the Kernel-theory [30] to project feature vectors into a high-dimension space, where all samples can be discriminated by a linear classifier. Adaboost [2] is another way to get the margin maximization which constructs a strong classifier using a linear weighted combination of weak classifiers. Simultaneously, it can perform the task of feature selection via iteratively adjusting weights of the samples, and the process can be considered as a greedy strategy.

Munder et al. have carried out an experimental study on pedestrian classification, and they conclude that SVMs perform best, and the cascaded Adaboost approach achieves the comparable performance at much lower computational costs for pedestrian classification [20]. It can be seen that SVMs and Adaboost are both state-of-the-art classifiers for pedestrian detection. However, SVMs cannot appropriately select features in the detection procedure, although its performance is generally better than other classifiers. Compared with SVMs, the speed of cascaded Adaboost can be much higher, while the choice of thresholds for its weak classifiers is a bit trivial due to the observation of the feature distribution of a large number of samples. This paper provides a new viewpoint to develop a method integrating feature selection with classification for pedestrian detection.

3. Pedestrian representation

Dalal and Triggs [8] propose the Histogram of Oriented Gradients (HOG) descriptors in a fixed size and fixed position blocks to represent a human body. The success of HOG descriptors lies in its adopting statistical information of gradients to characterize the local contour of a pedestrian. However, Zhu et al. [9] consider fixed-size HOG blocks miss some global clues. Therefore, they use variable-size HOG (v-HOG) blocks on various scales to capture more information and report better results. In this paper, we employ v-HOG blocks to extract feature descriptors as a pedestrian representation. For the v-HOG feature extraction of a 64×128 sample image, the ratio of the width to the height of a block is either 1:1, 1:2 or 2:1. Blocks range in size from 12×12 to 64×128. Each block consists of 2×2 cells and the size of each cell is 8×8 pixels. Gradient orientations of pixels in a cell are accumulated to discrete 9 histogram bins. A 36 dimensional vector concatenating gradient orientation histograms of four cells in a block is extracted. More details of the feature extraction procedure can refer to [9]. In Fig. 1, we illustrate v-HOG features extracted from blocks of different sizes at different locations.

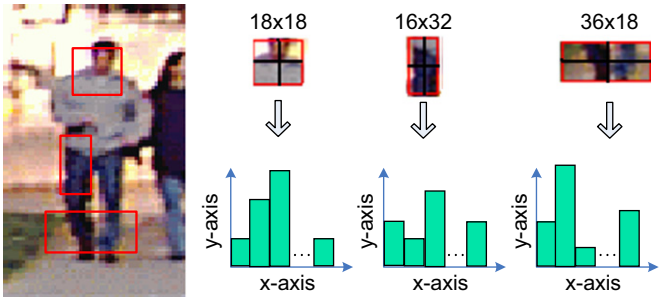

Fig. 1. Three v-HOG feature blocks with different sizes at different locations. The x-axis denotes the dimension of the feature vector and the y-axis denotes the value of the feature vector.

4. The proposed learning method

The proposed method is based on an L1-norm minimization learning (LML) framework, which is used to build weak classifiers. The strong classifiers are achieved using the integer programming optimization method.

4.1. L1-norm minimization learning framework

A general linear classifier $y=w^Tx+\theta$ can be viewed as a decision hyper plane, where w is a normal vector (weight vector) of the decision hyper plane with θ as a threshold. All linear hyper planes (linear classifiers) comprise a set called the linear function set. In our work, we calculate the normal vector of a linear classifier via L1-norm minimization learning (LML). The framework is shown as

$$\min \|w\|_1 \quad \text{s.t.} \quad \text{constrains of } w \tag{1}$$

where $w \in R^n$ and $\|w\|_1 = \sum_{j=1}^{n} |w^j| \cdot | \cdot |$ denotes the absolute value operator. L1-norm minimization is the approximately optimal solution of L0-norm minimization which aims to find a normal vector having the fewest nonzero components, which is also called sparseness. It should be mentioned that the L1-norm minimization can lead to minimize the upper bound of test error when the training is given. The upper bound on the test error of classifiers is given by

$$R_{train} + \sqrt{\frac{h(\log(2N/h)+1) - \log(\eta/4)}{N}} \tag{2}$$

where R_{train} is the training error, h is the VC-dimension of a set of classification functions. It can be seen that the error upper bound consists of the training error and a function of VC-dimension. A smaller VC-dimension can result in a smaller error upper bound when the training error is given. N is the size of the training set (restriction: this formula is valid when the VC-dimension h is smaller than N).

The minimization of w given by L1-norm is to get the fewest nonzero components of w, and this aims to pursue the dimension n of w as low as possible. Correspondingly, it equals to make the weight vector w sparse. All k-sparse weight vectors and thresholds comprise a sparse linear function set. It has been shown a sparse linear function set has a smaller VC-dimension than linear function set in [44]. Therefore, the L1-norm minimization for the weight vector of the linear classifier can result in a smaller VC-dimension.

In this paper, the forms of weak-classifiers and strong classifiers are both linear ones, which are obtained by employing the L1-norm minimization and integer programming. Furthermore, R_{train} in our training procedure is boundary, guaranteed by constraints in integer programming. Therefore, our method pursues a smaller VC-dimension and then minimizes the error upper bound [1]. For more details of the relationship between the VC-dimension and sparseness, please refer to [44] which have given the rigorous bound of VC-dimension of a k-sparse linear function.

4.2. Weak classifier construction

4.2.1. Learning the normal vector of a weak classifier

The weak classifier in the linear style is learned by Model I based on the LML framework.

Model I:

$$\min_{w_k,\xi_i} \quad \|w_k\|_1 + C_1 \sum_{i=1}^{N} \xi_i \tag{3}$$

$$\text{s.t.} \begin{cases} y_i \cdot h_k(x_i) \geq \alpha - \xi_i \\ h_k(x_i) = w_k^T x_i \\ \xi_i \geq 0 \end{cases} \tag{4}$$

In Eq. (3), w_k is the normal vector of the kth weak classifier. ξ_i is used to measure misclassification degree of the ith training sample. N is the number of all training samples. C_1 is a predefined parameter to balance the minimization of the misclassification degree and L1-norm of the normal vector. x_i represents a 36 dimensional v-HOG feature vector of the ith sample, and y_i is the class label of the sample. α is a fixed and predefined parameter to guarantee the separability of the training samples. C_1 combined ξ_i with the constraints Eq. (4) ensures that certain percent of the training samples can be correctly classified. The larger C_1 is, the smaller the sum of ξ_i should be, which means fewer misclassified samples. C_1 is assigned a larger value than 50.0 empirically.

It is known that L1-norm is not differentiable, which makes Model I difficult to be solved directly. There is, however, a simple and relatively common transformation that allows this problem to be solved effectively. Details converting the optimization model refer to [14] and the Interior Point methods solving the optimization model consult [21].

4.2.2. Threshold determination

In a weak classifier construction procedure, after obtaining its normal vector, we need to determine a threshold. To meet a high detection rate, it is not always available to adopt the value of $\alpha - \xi_i$ of Eq. (4). Furthermore, it is exhausting to balance each threshold based on the feature distribution of positives and negatives as in Adaboost. We build a min–max penalty function model to solve a threshold in terms of the Game Theory. Our goal is to get the threshold which can balance the misclassification between positives and negatives best.

Model II:

$$\min_{\theta_k} \left(r_1 \left(\sum_{pos=1}^{PN} \max\{0, \theta_k - h_k(x_{pos})\} \right) + r_2 \left(\sum_{neg=1}^{NN} \max\{0, h_k(x_{neg}) - \theta_k\} \right) \right) \tag{5}$$

where $\theta_k \in R$ is the threshold of the kth weak classifier, which is the only variable in Model II, x_{pos} denotes feature vector of the positives and x_{neg} of the negatives. PN is the number of positives and NN is negatives. $N = PN + NN$. Using the gained normal vector w_k, we can get the value of $h_k(x)$ via computing the inner-product of the feature vectors of the training samples and the normal vector. Function

$$\max\{0,t\} = \begin{cases} 0 & \text{if } t < 0 \\ t & \text{otherwise} \end{cases} \qquad r_1 \in R,\ r_2 \in R$$

are penalty factors.

We explain the meaning of this model as follows. The determination of the threshold is a dynamic process in which the positives and the negatives participate. $\max\{0,t\}$ is the maximum misclassification degree of both positives and negatives. The positives pursue a lower threshold θ_k to make $\theta_k - h_k(x_{pos}) < 0$ in order to minimize the misclassification degree $\max\{0, \theta_k - h_k(x_{pos})\}$. On the contrary, the negatives endeavor for a higher threshold to minimize the maximum misclassification degree of negatives.

In real application, pedestrian in still image is a rare-event because the number of pedestrian patches is much less than non-pedestrian's. Furthermore, the numbers of pedestrian and non-pedestrian in training samples are also unbalanced. Therefore, the optimization model employs two penalty factors r_1, r_2 between positives and negatives to balance the asymmetry. How to set the values of these two factors is described in the experiments (Section 5.1).

The optimization Model II is an unconstrained convex programming and can be converted into linear programming [21]. After solving the Models I and II, we obtain a weak classifier

$$g_k(x) = sign(h_k(x) - \theta_k) = sign(w_k^T x - \theta_k) \tag{6}$$

where $sign(\cdot)$ is a sign function.

4.3. Strong classifier construction

With respect to the computation cost and redundancy existing in feature representation, it is unadvisable to make all weak classifiers contribute to the final strong classifier, which is consistent with the principle of building a classifier as 'Many can be better than all' [23]. We employ a global integer programming method to form a strong classifier:

$$G(x) = \begin{cases} 1 & \sum_{k=1}^{M} a_k(\lambda_k g_k(x) - 0.5) \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

where $a_k = \log[(1-\varepsilon_k)/\varepsilon_k]$ is the weight of the kth weak classifier and ε_k is the training error rate. In Eq. (7), λ_k is a 0/1 binary variable. $\lambda_k = 0$ means that the kth weak classifier is not selected and $\lambda_k = 1$ selected. On the basis of these definitions, we construct Model III to solve λ_k.

Model III:

$$\min_{\lambda_k,\eta_i} \quad \sum_{k=1}^{M} \lambda_k + C_2 \sum_{i=1}^{N} \eta_i$$

$$\text{s.t.} \quad \begin{cases} A\lambda \leq r_3 \eta \\ \sum_{i \in pos} \eta_i \leq \sigma_1 PN \\ \sum_{i \in neg} \eta_i \leq \sigma_2 NN \end{cases} \tag{8}$$

where η_i is also a 0/1 binary variable which corresponds to the ith training sample. If the ith sample can be correctly classified by the combination of selected weak classifiers, then $\eta_i = 0$. Otherwise $\eta_i = 1$. M is the number of weak classifiers and C_2 is a predefined factor.

To facilitate the writing and concise expression, the forms of vectors and some notations are introduced. Let $\lambda = [\lambda_1, \ldots \lambda_k, \ldots \lambda_M, 1]^T$ and $\eta = [\eta_1, \ldots \eta_{PN} \ldots \eta_N]^T$. A is a matrix formulation of which the ith row is the ith constraint conducted by the ith

sample, as

$$(A_{\cdot i}) = \begin{cases} (-a_1 g_1(x_i), \ldots, -a_k g_k(x_i), \ldots, -a_M g_M(x_i), 0.5(a_1 + \cdots a_M)) & \text{if} \quad x_i \in \text{positives} \\ (a_1 g_1(x_i), \ldots, a_k g_k(x_i), \ldots, a_M g_M(x_i), -0.5(a_1 + \cdots a_M)) & \text{if} \quad x_i \in \text{negatives} \end{cases} \tag{9}$$

$r_3 \in R$ is a predefined slack factor which ensures the constraint equations $A\lambda \le r_3\eta$ to have a feasible solution. In other words, the classification result of the ith sample can range in $[0,r_3]$ instead of $[0,1]$. $A\lambda \le r_3\eta$ is actually the reformulation of Eq. (7) by substituting 0 into the vector $r_3\eta$. $\sigma_1$is the maximum misclassified rate of positives, and σ_2 is the maximum false positive rate (see Section 4.4). $\sigma_1 PN$ and $\sigma_2 NN$ are the upper bounds of the number of misclassified positives and negatives, respectively.

On the condition of meeting the constraints, the objective function in Model III aims to pursue nonzero components in λ, η as minimal as possible. C_2 is a predefined factor. Since the vectors of λ,η are positive and their components can just be 0 or 1, $\sum_{k=1}^{M} \lambda_k + C_2 \sum_{j=1}^{N} \eta_j$is equivalent to $\sum_{k=1}^{M} |\lambda_k| + C_2 \sum_{j=1}^{N} |\eta_j|$. It can be further transformed into $\|\lambda\|_1 + C_2\|\eta\|_1 - 1$ in integer space. $\|\lambda\|_1$and $\|\eta\|_1$, respectively, are the L1-norm of the vector λ and η in the integer space. This minimization of objective function in Model III can be considered as a special case of LML in the integer space.

The optimization Model III is a typical 0/1 integer programming problem, in which objective function and constraints are linear. We use the improved the Branch and Bound algorithm [21] to solve this problem. The basic idea of this algorithm initially converts the integer program into many sub problems (branches) and then compares the solutions of all branches.

We add some practical requirements and restrictions into this program (Section A.1). In the first several cascades, we hope the number of weak classifiers is as few as possible and we restrict the total number of them to 10 to decrease the search space. As the number of cascade increases, the number of weak classifiers is gradually increased and the restriction is gradually relaxed. Although we use the idea of branch and bound algorithm, the sub problems of this model are no longer the linear program. We improve the branch aiming at the specific constraints.

Here, we first fix the value η to solve λ, and then change η to form several branches during using the branch and bound method. According to the last two constraints in model, we can estimate at most how many η_i is one. Then, we rank the misclassification degree of each feature and select the top hardest $\sigma_1 PN$ positives and $\sigma_2 NN$ negatives. The corresponding η_i of these samples are set one and others are zero, and then solve λ. This is the first branch. The other branches can be obtained via changing the value η_i. The cascade requires the most of positives should be correctly classified. Therefore we only circularly reduce the number $\sigma_1 PN$ and do not change the number $\sigma_2 NN$ unless the last levels of cascade.

An initial solution first chooses the weak classifier with the highest classification rate and assign its λ_k to 1. Then, it searches other complementary classifiers (which can correctly classify the misclassified sample by the first selected classifier) until the constraints are met. At this time, the value of object function is chosen as a lower-bound. Then it repeats this process and compares the object function of other sub branches with the lower-bound. If the value of object function is bigger than the lower-bound, then abandon this sub branch. Otherwise, a solution of this branch is obtained. Then, the optimal solution can be obtained by comparing all left sub branches.

4.4. Training the CLML classifier

The cascade mechanism is adopted, which is a classic technique to promote detection speed. In each level of the cascade, some weak classifiers are selected to form a strong classifier. The strong classifier is implemented using the integer programming model. The convergence of cascade mechanism can refer to [7]. In the training procedure, the final detection rate and false positive rate decrease as the number of cascade increases. Therefore, we need to choose the number of cascade and balance the detection rate and false positive rate. The requirement, a minimal detection rate is 0.998 and the maximum false positive is no more than 0.3, is met in each cascade stage. In accordance with the Model III, we set $\sigma_1 = 1 - 0.998 = 0.002$, $\sigma_2 < 0.3$. The training procedure of CLML is as follows:

Algorithm 1. The cascaded L1-norm minimization learning (CLML) method

Input: the minimal detection rate, maximum acceptable false positive rate in tth level of the cascade
  POS: set of positives
  NEG: set of negatives
  F_{target}: target overall false positive rate
  f_t: false positive rate in tth level of cascade
  D_t is detection rate in the tth level of cascade
Initialize: $t=0, F_0=1.0, D_0=1.0$
While $F_t > F_{target}$
  –Train weak classifiers using POS and NEG samples, compute normal vectors and thresholds
  $-t=t+1$, $\Delta_t=0$
  $-\sigma_2=0.7-\Delta_t$
    if there is no solution for Model III
      increase $\Delta_t=\Delta_t+0.1$
    else
      1. Solve integer programming Model III
      2. Evaluate Pos and Neg by the current strong classifier
      3. Compute f_t under this threshold
–End
$-F_{t+1}=F_t \times f_t$
$-D_{t+1}=D_t \times (1-\sigma_1)$
–NEG $\leftarrow \emptyset$
–Evaluate the current cascaded detector on the negatives, i.e. images without human and add misclassified samples into set NEG
End
Output: A t-level cascade strong classifiers

In this paper, it takes 15 days to achieve certain detection accuracy. At the beginning of training, it needs more time. Then, it needs less time in last cascades as the number of both blocks and negatives decreases. The platform is on Pentium IV 3.0 GHZ CPU and 2.0 G MEMORY with the matlab program and three PC are used to perform the parallel computation. Although the computational cost of global selection is a bit expensive compared with the local one in the training procedure, the training process is offline process. It only needs to store the weight vectors and thresholds of the selected weak classifiers in each cascade. Thus, it does not have an effect on the detection process. To speed up the training, we will continue to improve the solving method and realize it in C++ language in future.

4.5. Discussion

Compared with L2-norm used in SVMs, L1-norm is effective for achieving sparseness which appreciates the differences of feature vector. L2-norm emphasizes more on "average variation", which means each element of a vector varies almost equally, cannot result in sparseness. Hence, L1-norm is adopted in this work.

To be mentioned, the sparse features selected by L1-norm can alleviate the occlusion and variations of views problems to some extent. The view variation of training samples cause the difference of features in the corresponding positions, but the different views may have some common clues. Here, the concept "common features" does not mean the total same, but denotes the almost similar response. For example, some common features of the front view sample can be obtained through removing the different features and remaining the common features from nearly front view. It can also be considered the different features of two views are occluded by non-person objects and only keep the common features. The sparsity can be viewed as a feature selection process. It can learn these common features as the important and principle components, when the training samples deriving from the different views are used. In this sense, the sparsity has the tolerance to view variation to some degree.

We adopt integer programming to construct the strong classifier. The way of selecting features and determining threshold are different from Adaboost. Adaboost employs re-sampling principle to adjust the weight of each weak classifier, and then adopts greedy strategy iteratively to select features. Instead, our method tries to select features in a new way. Our method aims to globally select the minimal weak classifiers, instead of locally iterative selecting the features. Of course, the computational cost of this selection is a bit expensive compared with the local one used by Adaboost in the training procedure. However, the higher computational cost during training is worthy if we can alleviate the burden of detection.

Although it has been proven that the error upper bound of Adaboost exists in a probabilistic framework, it may not always obtain the global optimal "Bag-of-features" owing to its feature selection strategy. In other words, Adaboost may become unstable, and hence select more redundant feature to achieve the acceptable performance. From training samples arranged in XOR-like layout, we give a comparison between our method and Adaboost in Section A.2 of appendix. From that, we can see our method is dedicated to obtaining the compact features and more suitable for detection problem.

5. Experiments

There are about 2400 training positives from MIT and SDL [24] and about 4900 negatives from INRIA datasets. We use 1000 positives and 2000 negatives in the first cascades. More negatives will be added in the cascaded training process to ensure the ratio of the positives to negatives. In Fig. 2, some training samples are displayed.

Fig. 2. Training positives and negatives.

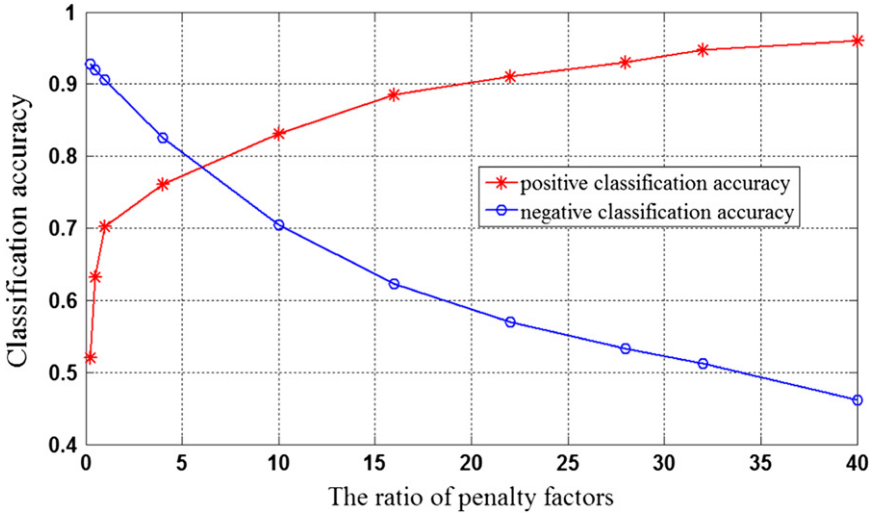

Fig. 3. Classification accuracy with different r_1/r_2 ratios.

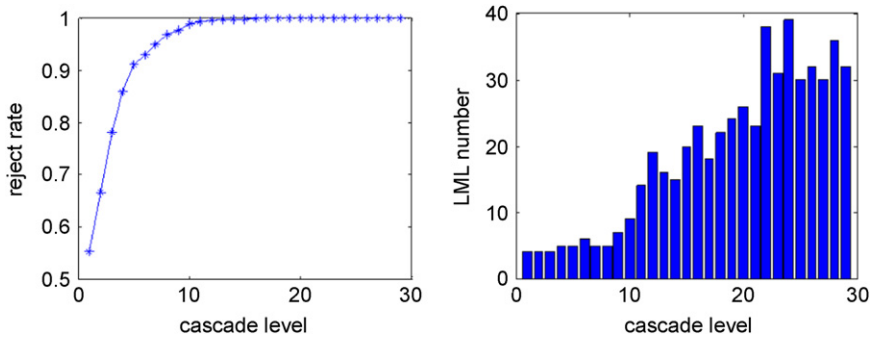

Fig. 4. Left: the accumulated rejection rate over all cascade levels. Right: the number of LML weak classifiers at each level.

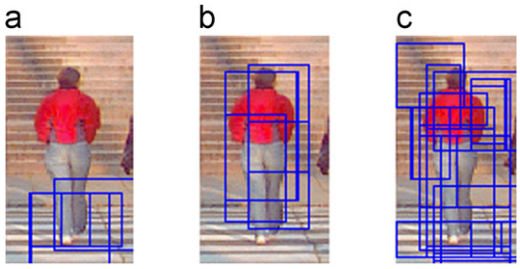

Fig. 5. Feature blocks: (a) blocks selected in the first level, (b) in the second level and (c) in 15th level of the cascade.

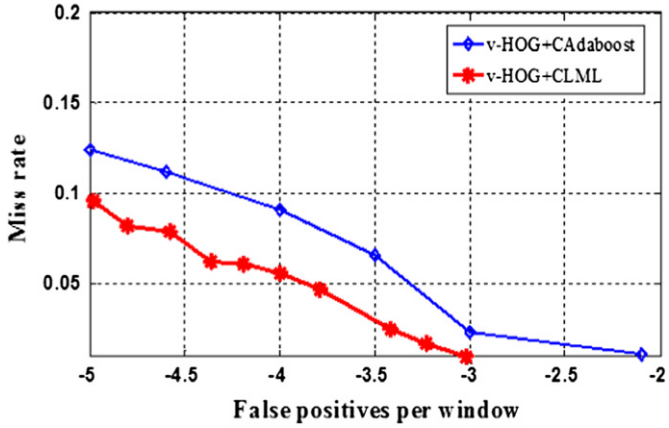

Fig. 6. The comparison of our method with Zhu's via same v-HOG features on INRIA test set.

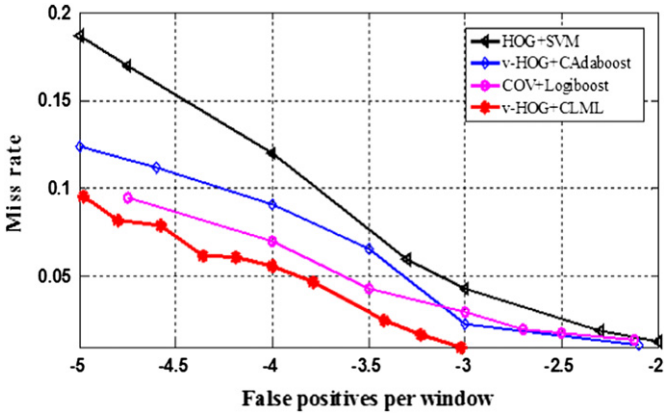

Fig. 7. The comparison of our method with the state of arts on INRIA test set.

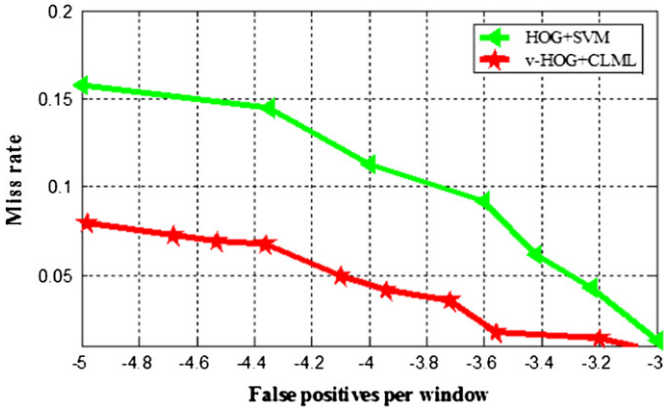

Fig. 8. The comparison of our method with HOG+SVM on SDL test set.

We evaluate our algorithm via the challenging the INRIA test set of 288 images [8] and the SDL test set of 140 images [24]. In the INRIA test set, pedestrians are mostly in standing posture, while it covers more diverse body poses and cluttered backgrounds. In the SDL test set, pedestrians are almost situated in multi-view appearance and crowded environment. Furthermore, some images including some pedestrians from lateral views are collected. Although the chosen training positives are mostly from front view, the trained model benefited from sparseness can handle some multi-views and occlusion cases, demonstrated by the experiments.

5.1. Parameter analysis

To deal with the variability of appearance, illumination conditions and background, the normalization of both the feature vectors and the normal vectors of weak classifiers are carried out. The feature vector of each block and the normal vector are normalized, respectively, as follows:

$$x_i = x_i \Big/ \sqrt{\sum_{j=1}^{36} x_{ij} + \varepsilon} \quad w_k = w_k \Big/ \sqrt{\sum_{j=1}^{36} w_k^j + \varsigma} \tag{10}$$

where x_i denotes a feature block of the ith sample, and x_{ij} is the jth dimension component in the feature block of the ith sample. w_k denotes the normal vector of the kth weak classifier, and w_k^j is the jth dimension component of the kth weak classifier. ε and ς are small disturbance numbers (1.0 in our experiments).

There are several important parameters when learning the weak classifiers. In Model II, when we determine the threshold of an individual weak classifier, the penalty factors r_1 and r_2 have important effects on the classification accuracy of positives and

negatives. In Section 4.2.2, the penalty factors affecting the threshold should be different. In general, r_1 of positives should be larger than r_2 to guarantee that most of the positives remain for the next level training. However, in regard of classification accuracy of negatives, we cannot increase r_1 too much. It is appropriate to set the ratio of r_1/r_2 a value among [1,40]. In Fig. 3, we illustrate the influence of the ratio on the positive and negative classification accuracy in the first cascade level.

5.2. Evaluation and comparison

In Fig. 4, we present the results of the cascade classifier. It can be seen that 6 cascades are enough to reject 95% of the negatives. As we discussed in Section 4.5, the computational time of LML weak classifier is less than Dalal's [8] and Zhu's [9]. The proposed method is faster than the Linear-SVM and Kernel-SVM during the detection. The SVM employs the inner-product of the test sample

Fig. 9. Detection examples. Detected false positives are marked with white rectangle of dash line, and missed positives are marked with black rectangle of dash line.

and the support vectors. Instead, our weak classifier only projects the test sample on the normal vector at a low computation cost to classify objects. It should be noted that the cascade scheme of CLML method can also perform efficiently. Therefore, our classifier is much faster than a SVM classifier.

We can get the most compact feature blocks by our method. Fig. 5(a) shows the best four blocks which are different from [9], in which blocks in the 36×80 pixels size are considered the best. However, in the first level, our best block size is 40×40 pixels in the leg parts of human body. In the second level, our best block size is 36×80 pixels covering the contour of human body. The difference between our blocks and the blocks selected by Zhu's method is mainly due to the different learning methods. Our method adopts L1-norm which emphasizes more on the difference among features.

We compare our classifier with [9] on the INRIA human test set using miss rate tradeoff False Positives Per Window (FPPW) on a log scale in Fig. 6. Points on curves in Fig. 6 are obtained from different cascade levels. Miss rate and False Positives Per Window (FPPW) are defined as follows:

$$MissRate = \frac{\#\text{Missed positive detections}}{\#\text{Total positives}},$$

$$FPPW = \frac{\#\text{False positive detections}}{\#\text{Total image windows}}$$

Although the same v-HOG features are extracted as pedestrian representation by the two compared methods, the way of feature selection and classifier construction is different. Zhu's method utilizes SVM to build weak classifiers and uses Adaboost method to construct a strong classifier. Our method proposes LML principle and uses integer programming to make a classification. From Fig. 6, it can be seen that the proposed classifier performs better than Zhu's method.

We compare our method with the state-of-the-art methods on the INRIA test set, including HOG+SVM method [8], v-HOG+CAdaboost [9] and the COV+Logitboost method [10], which is shown in Fig. 7 on a log scale. We implement the method [8] by the open source codes of HOG and LibSVM and the results accord with their reports. The curves of the methods [9,10] are obtained from their reported results. As shown in Fig. 7, our method reaches a much better performance than the HOG-based results on the INRIA dataset. Comparing with others at the FPPW rate of 10^{-5}, our method achieves 9% miss rate, which is about 8% lower than the HOG+SVM method, about 3% lower than Zhu's and about 1% lower than Tuzel' method which use different pedestrian representations (COV features) from ours.

In addition, we compare our method and the HOG+SVM method [8] on the SDL test set to validate robustness of our method to view variations, which is shown in Fig. 8 on a log scale. The SDL test set is also challenging owing to the view variation of pedestrians. It is unreasonable and unfair to compare our method against the above methods except the HOG+SVM on the SDL test set because we cannot obtain the exact expression of codes and the optimal parameters in other methods. As shown in Fig. 8, the obvious difference between our method and HOG+SVM can also embody the superior performance of our method on the SDL test set. At the FPPW rate of 10^{-4}, our method achieves 4% miss rate, which is about 7% lower than the HOG+SVM method.

In Fig. 9, we show some detection examples from multiple detection scales on two test sets. In Fig. 9(d), all pedestrians are detected, although they are overlooked. In Fig. 9(e) most of the pedestrians except the rightmost person are correctly located whether or not they are occluded or in multi-posture, since that person is too close to the image boundary. The statue in the up-left side of picture is detected, since it is very similar to a pedestrian. In Fig. 9(f), the pedestrians are detected, and especially the pedestrian in black jacket occluded can be found correctly. In Fig. 9(g), four children are correctly located although they have posture variation and a child is missed. From Fig. 9(h) to (k), all pedestrians are detected correctly in spite of variations of posture and view. However, in Fig. 9(i), there is a false positive window owing to the effects of trunks.

6. Conclusions

Pedestrian detection is a rapidly evolving topic in pattern recognition and some state-of-the-art classification methods are employed in this topic. In this paper, we propose a new learning and classification method, which is superior to the state-of-the-art ones for pedestrian detection. The method aims to select more informative and compact features and simultaneously to construct classifiers via solving L1-norm minimization and integer programming optimization models. The method will pursue to a smaller upper bound of test error and improve detection performance. Considering the detection efficiency problem, a cascaded mechanism is employed to construct the final classifier.

Features in each block selected by the L1-norm minimization criterion emphasize the principle difference between positives and negatives, which are sparse to some extent. The combination of feature blocks in a cascade is performed by integer programming, which can achieve the compact and sparse blocks for pedestrian. Therefore, both features in each block and blocks selected by our method are sparse, and are insensitive to view and occlusion validated by experiments.

The concepts and techniques introduced in this paper include the detailed construction principles of optimization models and analysis of representation of pedestrian patterns, the L1-norm minimization learning, the applications of integer programming and min–max function model. Detailed experimental results are reported with comparisons to several state-of-the-art methods, confirming that the proposed method has superior performance and is robust to view and occlusion problems in pedestrian detection.

At present, the proposed method is applied to pedestrian detection. In the future, we will extend our method to other objects e.g. vehicles, etc. and the multi-class object detection task.

Conflict of interest

None.

Acknowledgment

The authors would like to thank the associate editor, the anonymous reviewers and Dr. Jinzhu Jia for their constructive comments. This work is supported by the National Basic Research Program of China (973 Program) with nos. 2011CB706900, 2010CB731800 and the National Science Foundation of China with nos. 60872143 and 61039003.

Appendix A

A.1. Integer program procedure

Input: cascade level, c_2, V, g_k, a_k, σ_1, σ_2, PN, NN where $k=1,2\ldots,M$.

Step 1: Initialization:

1.1 Rank all feature blocks according to their classification error rates, and select the top hardest $\sigma_1 PN$ positives and $\sigma_2 NN$ negatives.

1.2 $\eta_i=1$for these samples, and $\eta_i=0$ for others.

Step 2: If the Cascade level is lower than 6, add a constraint $\sum_{k=1}^{M}\lambda_k \le 10$ to **Model III**.

Else gradually relax the constraint as $\sum_{k=1}^{M}\lambda_k \le V$, where $V>10$.

Step 3: Loop to adjust the values of η_i and form some branches.

3.1. Given η_i, loop to solve the integer program and form a combination of weak classifiers.

3.1.1. Select a weak classifier and assign its λ_k to 1, and record it as the used label.

Loop: $i=1,2,\ldots$.

Label the misclassified samples by previous selected classifiers.

Choose complementary classifiers with ability to correctly classify the misclassified samples as

possible as they can.

Check constraint and compare objective value

If $\sum_{k=1}^{M}\lambda_k \le V$

If other constraints are met, record the objective value and return to 3.1.1

Else record its label and return to 3.2.1

Else set Inf as the objective value, record its label and return to 3.1

Search the possible combinations in this branch and continually compare lower bound, to decide whether cutting this sub branch, and then return to 3.

Step 4: Search all of branches

If constraints are met, select the minimum objective value, then output the combination of λ_k.

Else relax σ_1, σ_2, return to Step 1.

A.2. Illustrative XOR example

In this section, we describe the difference between Adaboost and our method for a XOR problem. In a two-dimensional space, there are four samples to be classified in Fig. 10, i.e.

$$\begin{Bmatrix} (x_1=(+1,+1),y_1=+1) \\ (x_2=(-1,-1),y_2=+1) \\ (x_3=(-1,+1),y_3=-1) \\ (x_4=(+1,-1),y_4=-1) \end{Bmatrix}$$

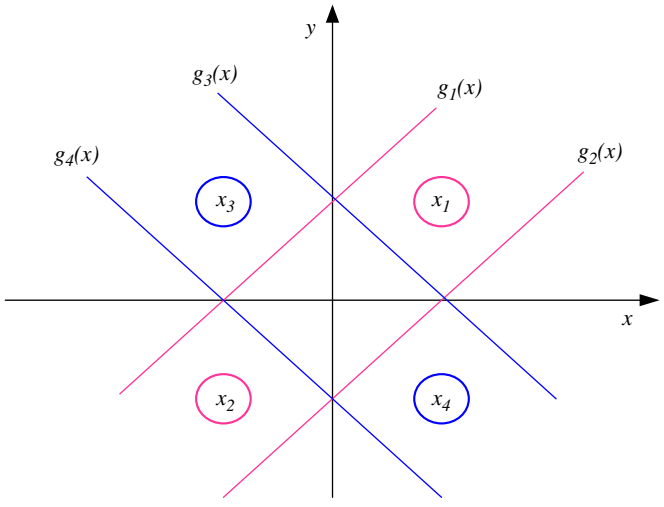

Fig. 10. Instances arranged in XOR layout and its decision functions. The instances with the same label have the same color.

For simplicity, suppose we have four weak classifiers (decision functions):

$$g_1(x)=\begin{cases}1 & \text{if } x+1>0\\ -1 & \text{otherwise}\end{cases} \quad g_2(x)=\begin{cases}1 & \text{if } x-1>0\\ -1 & \text{otherwise}\end{cases}$$

$$g_3(x)=\begin{cases}1 & \text{if } -x+1>0\\ -1 & \text{otherwise}\end{cases} \quad g_4(x)=\begin{cases}1 & \text{if } -x-1>0\\ -1 & \text{otherwise}\end{cases}$$

It can be seen that arbitrary two parallel functions will be the optimal solution of this problem. Of course, if all functions will be chosen, then the problem can also be solved. The Adaboost may become unstable, owing to its randomness. It will pick a classifier with minimal error from all functions in one iterative process. Suppose $g_1(x)$ is chosen in the first iteration, then sample x_4 is falsely classified. The weight of x_4 becomes larger comparing with the other three samples after the weights being normalized. At this time, there are three classifiers with the same weights and errors. According to randomness, if it chooses $g_2(x)$, then the error constraints have been met and the optimal solution has obtained. However, if it chooses $g_3(x)$, unfortunately, it must select all classifiers to assure the samples to be correctly classified. Instead, our method globally selects the minimal number of functions to construct the final strong classifier. It can choose two parallel functions as the optimal solution from the global view. Therefore, it can select the minimal number of features and reaches the optimal solution.

References

[1] Christopher J.C. Burges, A tutorial on support vector machines for pattern recognition, Data Mining and Knowledge Discovery 2 (2) (1998) 121–167.
[2] M. Collins, R.E. Schapire, Y. Singer, Logistic regression, AdaBoost and Bregman distances, Machine Learning 48 (1–3) (2002) 253–285.
[3] D. Donoho, For most large underdetermined systems of linear equations the minimal L1-norm near solution approximates the sparsest solution, Communications on Pure and Applied Mathematics 59 (6) (2006) 797–829.
[4] K. Huang, SAviyente, sparse representation for signal classification, advances in neural information processing systems, in: Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems, 2007.
[5] A.Y. Yang, J. Wright, Y. Ma, S.S. Sastry, Robust face recognition via sparse representation, IEEE Transactions on Pattern Analysis and Machine Intelligence 31 (2) (2009).
[6] Ying Wu, Ting Yu, A field model for human detection and tracking, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (5) (2006) 753–765.
[7] P. Viola, M. Jones, Robust real-time object detection, International Journal of Computer Vision 57 (2) (2001) 137–154.
[8] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, 2005, pp. 886–893.
[9] Q. Zhu, S. Avidan, M.C. Yeh, K.T. Cheng, Fast human detection using a cascade of histograms of oriented gradients, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, 2006, pp. 1491–1498.
[10] O. Tuzel, F. Porikli, P. Meer, Human detection via classification on riemannian manifolds, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2007, pp.1–8.
[11] Y. Mu, S. Yan, Y. Liu, T. Huang, B. Zhou, Discriminative local binary patterns for human detection in personal album, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, vol. 23–28, 2008, pp. 1–8.
[12] Xiaoyu Wang, Tony X. Han, Shuicheng Yan, An HOG–LBP human detector with partial occlusion handling, in: Proceedings of the IEEE International Conference on Computer Vision, Kyoto, 2009.
[13] B. Wu, R. Nevatia, Detection of multiple, partially occluded humans in a single image by Bayesian combination of edgelet part detectors, in: Proceedings of the IEEE International Conference on Computer Vision, 2005.
[14] L. Zhang, B. Wu, R. Nevatia, Detection and tracking of multiple humans with extensive pose articulation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2007.
[15] S. Ioffe, D.A. Forsyth., Probabilistic methods for finding people, International Journal of Computer Vision 43 (1) (2001) 45–68.
[16] B. Leibe, E. Seemann, B. Schiele, Pedestrian detection in crowded scenes, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, 2005, pp. 878–885.
[17] D. Vinay, J. Neumann, V. Ramesh, L.S. Davis, Bilattice-based logical reasoning for human detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2007.

[18] A. Mohan, C. Papageorgiou, T. Poggio, Example-based object detection in images by components, IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (4) (2001) 349–360.
[19] Ran Xu, Baochang Zhang, Qixiang Ye, Jianbin Jiao, Human detection in images via L1-norm minimization learning, in: IEEE International Conference on Acoustics, Speech and Signal Processing, 2010, pp. 3566–3569.
[20] S. Munder, D. Gavrila., An experimental study on classification, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (11) (2006) 1863–1868.
[21] Michael Juger, Denis Naddef, Computational Combinatorial Optimization: Optimal or Provably Near-Optimal Solutions, Springer Press, 2001.
[22] A.T. Mario, D. Nowak, J. Wright, Gradient projection for sparse reconstruction: application to compressed sensing and other inverse problems, IEEE Selected Topics in Signal Processing 1 (4) (2007) 586–597.
[23] Zhi-Hua Zhou, Jianxin Wu, W. Tang, Ensembling neural networks: many could be better than all, Artificial Intelligence 137 (1–2) (2002) 239–263.
[24] 〈http://coe.gucas.ac.cn/SDL-HomePage/resource.asp〉.
[25] Ran Xu, Baochang Zhang, Qixiang Ye, Jianbin Jiao, Cascaded L1-norm minimization learning (CLML) classifier for human detection, in: Proceedings of the IEEE Confernce on Computer Vision and Pattern Recognition, 2010.
[26] S. Munder, C. Schnörr, D.M. Gavrila, Pedestrian detection and tracking using a mixture of view-based shape-texture models, IEEE Transactions on Intelligent Transportation Systems 9 (2) (2008) 333–343.
[27] C. Papageorgiou, T. Poggio, A trainable system for object detection, International Journal of Computer Vision 38 (2000) 15–33.
[28] H. Shimizu, T. Poggio, Direction estimation of pedestrian from multiple still images, Proceedings of the IEEE Intelligent Vehicles Symposium (2004) 596–600.
[29] J. Alvarez, Th. Gevers, A. Lopez, Learning photometric invariance for object detection, International Journal of Computer Vision 90 (1) (2010) 45–61.
[30] S. Maji, A. Berg, J. Malik, Classification using intersection kernel SVMs is efficient, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2008.
[31] M. Szarvas, A. Yoshizawa, M. Yamamoto, J. Ogata, Pedestrian detection with convolutional neural networks, Proceedings of the IEEE Intelligent Vehicles Symposium (2005) 223–228.
[32] B.E. Goldstein, Sensation and Perception, sixth ed., Wadsworth, 2002.
[33] William Robson Schwartz, Aniruddha Kembhavi, David Harwood, Larry S. Davis, Human detection using partial least squares analysis, in: Proceedings of the IEEE International Conference on Computer Vision, 2009.
[34] Z. Lin, L. Davis, D. Doermann, D. DeMenthon, Hierarchical part-template matching for pedestrian detection and segmentation, in: Proceedings of the IEEE Internationall Conference on Computer Vision, 2007.
[35] P. Viola, M. Jones, D. Snow, Detecting pedestrians using patterns of motion and appearance, International Journal of Computer Vision 63 (2) (2005) 153–161.
[36] S. Maji, A.C. Berg, Max-margin additive classifiers for detection, in: Proceedings of the IEEE International Conference on Computer Vision, 2009.
[37] Paul S. Bradley, O.L. Mangasarian, Feature selection via concave minimization and support vector machines, in: Proceedings of Fifteenth International Conference on Machine Learning, 1998.
[38] J. Zhu, S. Rosset, T. Hastie, R. Tibshirani, 1-Norm support vector machines, Advances in Neural Information Processing Systems (2003) 49–56.
[39] O.L. Mangasarian, Exact 1-norm support vector machines via unconstrained convex differentiable minimization, Journal of Machine Learning Research 7 (2006) 1517–1530.
[40] D.J. Wu, J.B. Bi, K. Boyer, A min–max framework of cascaded classifier with multiple instance learning for computer aided diagnosis, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 1359–1366.
[41] P.F. Felzenszwalb, D.P. Huttenlocher, Pictorial structures for object recognition, International Journal of Computer Vision 61 (1) (2005) 55–79.
[42] M. Andriluka, S. Roth, B. Schiele, Pictorial structures revisited: people detection and articulated pose estimation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 1014–1021.
[43] P.F. Felzenszwalb, R.B. Girshick, D. McAllester, D. Ramanan, Object detection with discriminatively trained part-based models, IEEE Transactions on Pattern Analysis and Machine Intelligence 32 (9) (2009) 1627–1645.
[44] Tyler Neylon, Sparse Solutions for Linear Prediction Problems, Dissertation, 2006.

Ran Xu received her B.S. degree in information and computer science from Wuhan University in 2006. From 2006 to 2008, her major was the applied mathematics in the Graduate University of Chinese Academy of Sciences. From 2006 to 2011, she was a candidate Ph.D. of computer science, in the Graduate University of Chinese Academy of Sciences. Her research interests include image processing, pattern recognition and object detection, etc.

Jianbin Jiao received the B.S., M.S. and Ph.D. degrees in mechanical and electronic engineering from Harbin Institute of Technology of China (HIT), Harbin, in 1989, 1992 and 1995, respectively. From 1997 to 2005, he was an associate professor of HIT. Since 2006, he has been a professor of the Graduate University of Chinese Academy of Sciences, Beijing. His research interests include image processing, pattern recognition, and intelligent surveillance, etc.

Baochang Zhang received the B.S., M.S. and Ph.D. degrees in computer science from the Harbin Institute of Technology, China, in 1999, 2001 and 2006, respectively. From 2006 to 2008, he was a Research Fellow with the Chinese University of Hong Kong and Griffith University, Australia. Currently, he is a lecturer at Beihang University, China. His research interests include pattern recognition, machine learning, face recognition, and wavelets.

Qixiang Ye received his B.S. and M.S. degrees in mechanical and electronic engineering from Harbin Institute of Technology of China (HIT), Harbin, in 1999 and in 2001, respectively. He received his Ph.D. degree from the Institute of Computing Technology, Chinese Academy of Sciences in 2006. Since 2009, he has been an associate professor at the Graduate University of the Chinese Academy of Sciences, Beijing. His research interests include image processing, pattern recognition, and statistic learning, etc.