Cascaded L1-norm Minimization Learning (CLML) Classifier for Human Detection

Ran Xu[1], Baochang Zhang[2], Qixiang Ye[1], Jianbin Jiao[1,+]

[1]Graduate University of Chinese Academy of Sciences, Beijing, China

[2]School of Automation Science and Electrical Engineering, Beihang University, Beijing, China

+Corresponding Author: Fax: +86-10-88256278, Email: jiaojb@gucas.ac.cn

Abstract

This paper proposes a new learning method, which integrates feature selection with classifier construction for human detection via solving three optimization models. Firstly, the method trains a series of weak-classifiers by the proposed L1-norm Minimization Learning (LML) and min-max penalty function models. Secondly, the proposed method selects the weak-classifiers by using the integer optimization model to construct a strong classifier. The L1-norm minimization and integer optimization models aim to find the minimal VC-dimension for weak and strong classifiers respectively. Finally, the method constructs a cascade of LML (CLML) classifier to reach higher detection rates and efficiency. Histograms of Oriented Gradients features of variable-size blocks (v-HOG) are employed as human representation to verify the proposed method. Experiments conducted on INRIA human test set show more superior detection rates and speed than state-of-the-art methods.

1. Introduction

Human detection in images is an open problem owing to the various appearance and pose of the human body, together with cluttered background under different illumination. A robust solution to this problem has extensive applications, including surveillance, image retrieval and some assistant systems etc.

Extracting more effective features and developing more powerful learning algorithms (classifiers) are the pursuits of researchers for human detection. The goal of this paper includes two aspects 1) Building a classifier from the perspective of VC-dimension minimization. 2) Fusing feature selection with classifier construction for human detection.

In the classifier construction, the margin maximization as an alternative of VC-dimension minimization [1,2] is the state-of-the-art method to minimize the error bound in the test procedure. Support Vector Machine (SVM) [1] employs support vectors and kernel trick to maximize the margins of different classes. The Adaboost [2] is another way to get the margin maximization which incorporates the feature selection in building a strong classifier. In the filed of statistic learning, however, the VC-dimension is the key concept. Obviously, neither SVM nor Adaboost can give a direct way to minimize the VC-dimension, even if they are able to provide effective solutions on expectation risk minimization based on the hypothesis that the classifiers are designed in the light of the margin maximization.

We design a cascaded classifier by incorporating the principle of L1-norm minimization into VC-dimension minimization. In the field of signal processing, L1-norm minimization can be considered as an approximately optimal implementation of the L0-norm minimization [3]. In the procedure of the classifier construction, the weak classifiers are built based on the L1-norm minimization principle and the min-max optimization penalty function which determines the appropriate thresholds for the weak classifiers. The intuition that we adopt L1-norm to learn weak classifiers comes from the successful application of L1 -norm in the fields of face recognition and compressive sensing of signals [4-5] in recent years. Compared with the weak classifier in Adaboost, ours is much stronger and also more suitable to deal with the histogram features. After obtaining the weak classifiers, we utilize the integer programming optimization model to select the minimum number of them to construct a stronger classifier, simultaneously selecting the most compact features. To achieve high detection accuracy and speed, we employ the cascade mechanism to detect human in images . Cascades will be added until predefined quality can be met. The final classifier inherits the advantages of both cascade and L1-norm Minimization Learning (LML) method, and obtains higher performance on classification accuracy and efficiency. This is validated on human detection.

The rest of this paper is organized as follows. In section 2, we review the related work of human detection. The feature representation of human is described in section 3. In section 4, the CLML method is presented. The experiments are presented in section 5 with conclusions in section 6.

2. Related work

In recent work, various features are widely proposed for pedestrian representation, including Haar-like [7], HOG [8], variable-size HOG (v-HOG) [9], Gabor [10], COV [11], LBP [12], HOG-LBP [13], Edgelet [14], Shapelet [20] etc.

Several methods have been employed/developed for feature selection and classification for human detection. In [6], the classification is actually based on the match with Chamfer distance. In [7], Adaboost cascade is applied to feature selection and classification. In [11], the authors firstly transform the features into tangent space of Riemannian manifolds, and then use the cascade of Logitboost for classification. In [15-18], the probability and reasoning methods are proposed for human detection. In [8, 12, 19] linear or kernel SVMs are employed for classification. In [9], the authors use linear-SVM as weak classifiers and then build an Adaboost cascade mechanism for human detection. Munder *et al.* have carried out an experimental study on pedestrian classification, and they conclude that SVMs perform best, and the Adaboost cascade approach achieves the comparable performance at much lower computational costs for human classification [21].

It can be seen that SVMs and Adaboost are the state-of-the-art classifiers for human detection. However, SVM can not perform efficiently in the detection procedure, although its performance is generally better than the other classifiers. Compared with SVMs, the performance of Adaboost can be much faster, but it is usually based on weak classifiers which need an exhausting procedure to form a strong classifier. The determination of thresholds of weak classifiers is a little time consuming procedure due to the observation of histogram distribution of huge amount of samples. Most importantly, neither of these two methods can give a direct way to minimize the VC-dimension of classification function. In this paper, we provide a new viewpoint to develop a more powerful feature selection and classification method in a comprehensive way.

3. Human Representation

Dalal & Triggs [8] propose the Histogram of Oriented Gradients (HOG) features on fixed-size blocks to represent human body. However, Zhu *et al.* [9] consider fixed-size HOG blocks miss some global cues. Therefore, they use variable-size HOG (v-HOG) blocks to capture more information and obtain better results.

In this paper, we employ v-HOG blocks as the features among which the most informative blocks are selected by our proposed learning method. For v-HOG feature extraction, in a 64x128 human sample image, there are about 4097 blocks extracted by varying scales and locations. Blocks have width/height ratios of 1.0, 0.5 or 2.0 and their sizes range from 12x12 to 64x128. In each block, a 36-dimension concatenated histogram vector can be extracted, which comprises histograms of gradient orientation projected into 9 discrete bins in 2x2 sub-regions. Details of the feature extraction procedure can refer to [9].

4. The Proposed Learning Method

The proposed method is based on a new L1-norm minimization learning (LML) framework, which is directly used to build a weak classifier. The strong classifiers are achieved by using the integer programming optimization method which can also be interpreted as one special kind of LML in the integer space.

4.1. L1-norm Minimization Learning Framework

A general linear classifier $y = w^T x$ can be viewed as a decision hyper plane, where w is the normal vector of the decision hyper plane with x as the test sample. Such a hyper plane has two elements, the normal vector and the threshold. In our work, we calculate the normal vector of a linear classifier via L1-norm minimization learning (LML). The framework is shown as:

$$\begin{aligned} &\min \ \|w\|_1 \\ &s.t. \ constrains \text{ of } w \end{aligned} \tag{1}$$

where $w \in R^n$ and $\|w\|_1 = \sum_{j=1}^{n} |w^j|$. 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 emphasized that the L1-norm minimization can minimize the VC-dimension of a linear classifier. VC-dimension is one of the core concepts of the statistical learning theory, as it is related to a probabilistic upper bound on the test error of a classifier given by:

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

where R_{train} is the training error usually given in training procedure, h is the VC-dimension of the classifier, and N is the size of the training set (restriction: this formula is valid when the VC-dimension h is smaller than N). The VC-dimension of a linear classifier $y = w^T x$ is $h = n + 1$, where n is the dimension of w. The minimization of w given by L1-norm is to get the fewest nonzero components of w, and this aims to minimize the VC-dimension of the linear classifier. In this paper, the forms of weak-classifiers and strong classifiers are linear ones, which are obtained by employing the L1-norm minimization and integer programming viewed as a special case of LML in the integer space. Therefore, our method directly pursues to minimize VC-dimension and upper error bound [1].

4.2. Building Weak Classifier

The weak classifier is built based on each 36-sized v-HOG feature vector (given in section 3).

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:

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

$$s.t. \begin{cases} y_i \cdot h_{w_k}(x_i) \geq \alpha - \xi_i, \ i=1,\dots,(PN+NN) \\ \xi_i \geq 0, \qquad i=1,\dots\dots,(PN+NN) \end{cases} \tag{4}$$

where $h_{w_k}(x_i) = w_k^T \cdot x_i$. In Eq.(3), $w_k \in R^{36}$ is the normal vector of the kth weak classifier and w_k^j is the jth dimension of the vector. ξ_i is used to measure misclassification degree of the ith training sample. PN and NN are the number of the positives and the negatives respectively. C_1 is a predefined parameter to balance the minimization of the misclassification degree and L1-norm of the normal vector. $x_i \in R^{36}$ represents the feature vector of ith sample, and $y_i \in \{-1,1\}$ is the class label of the sample. α is a predefined parameter to guarantee the separability of the training samples.

In Eq.(3), we minimize the L1-norm of the normal vector, and the sum of the mis-classification degree ($\sum_{i=1}^{PN+NN} \xi_i$) of training samples. C_1 combined with the constraints Eq.(4) ensures that certain percent of the training samples can be correctly classified. The larger C_1 is, the smaller $\sum_{i=1}^{PN+NN} \xi_i$ should be, which means fewer misclassified samples. Since we focus on the minimization of the misclassification rate in the learning procedure, C_1 is assigned a larger value than 50.0.

As is known, 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. We introduce vectors, $u \in R^{36}, v \in R^{36}$ and make the substitution $w_k = u - v$, $u \geq 0$, $v \geq 0$. These relationships are satisfied by $u^j = (w_k^j)_+$ and $v^j = (-w_k^j)_+$, $j = 1,2\dots 36$, j denotes dimension of feature vector, and $(\cdot)_+$ denotes the *positive-part* operator defined as $(w_k^j)_+ = \max\{0, w_k^j\}$ .We have $\| w_k \|_1 = I^T u + I^T v$, where $I = [1,1,1,\dots 1,1]^T$ is a 36 -dimension unit vector. Therefore, Model I can be rewritten as the following disciplined convex programming in Model II.

Model II:

$$\min \quad I^T u + I^T v + C_1 \sum_{i=1}^{PN+NN} \xi_i$$

$$s.t. \begin{cases} y_i \cdot (u - v)^T x_i \geq \alpha - \xi_i \\ \xi_i \geq 0 \qquad i=1,\cdots PN+NN \\ u \geq 0 \\ v \geq 0 \end{cases} \tag{5}$$

where u and v are two new variables of the model. The optimization Model I shown in Eq.(3)-(4) is equivalent to Model II, which can be solved by using the Interior Point method [22].

4.2.2. Threshold Determination

After computing the normal vector of the weak classifier, we need to determine a threshold. To meet the high detection rate, it is not always available to adopt the value of $\alpha - \xi_i$ of Eq.(4), since 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 obtain the thresholds in terms of the Game Theory for an individual weak classifier. The target is to attain the threshold θ_k which can balance the misclassification between positives and negatives best.

Model III :

$$\begin{aligned} \min_{\theta_k} \quad & (r_1 (\sum_{pos=1}^{PN} \max\{0, \theta_k - h_{w_k}(x_{pos})\}) \\ & + r_2 (\sum_{neg=1}^{TN} \max\{0, h_{w_k}(x_{neg}) - \theta_k\})) \end{aligned} \tag{6}$$

where $\theta_k \in R$ is a threshold of the kth weak classifier, which is the variable in Model III. x_{pos} denotes feature vector of the positives and x_{neg} of the negatives. PN and NN denotes the numbers of positives and negatives respectively. $h_{w_k}(x) = w_k^T \cdot x$ has the same meaning as above description. Function $\max\{0,t\}$ guarantees a non-negative result. $r_1 \in R$, $r_2 \in R$ are penalty factors.

We explain the meaning of this model from the perspective of the Game Theory. There are two parties in the game. $\max\{0,t\}$ can be regarded as the maximum misclassification degree of both parties. The positive party pursues a lower threshold θ_k to make $\theta_k - h_{w_k}(x_{pos}) < 0$ to minimize the maximum misclassification degree represented by $\max\{0, \theta_k - h_{w_k}(x_{pos})\}$ .In other words, the positive party pursues a lower threshold θ_k aiming to minimize the maximum misclassification degree. On the contrary, the negative party endeavors for a higher threshold to minimize the maximum misclassification degree of negatives.

In real application, human in still image is a rare-event because the amount of human patches is much less than the amount of non-human. Furthermore, the amounts of human/non-human in training samples are also unbalanced. Therefore, the optimization model employs two penalty factors between positives and negatives to balance the asymmetry contained in the dataset. If we appreciate the misclassification engendered by positives, we will set the penalty factor r_1 higher, and on the contrary set r_2 higher. How to set the values of these two factors is described in the experiments (section 5.2).

The optimization Model III is an unconstrained convex programming and can be converted into linear programming [22]. After solving the Model II and III, we obtain a weak classifier:

$$g_k(x) = h_w(x) - \theta_k = w_k^T x - \theta_k \quad (7)$$

4.3. Strong Classifier Construction

Inspired by weighted voting principle deriving from the bagging method, we utilize linear weighted combination of weak classifiers to construct a strong classifier. Each 36-dimension v-HOG feature vector (given in section 3) corresponds to a weak classifier. Therefore, the number of weak classifiers is the same as the number of v-HOG blocks, about 4097. 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’ [24].

We employ a global integer programming method to select a minimal number of weak classifiers, and get a strong classifier $G(x)$ which is formulated as

$$G(x) = \begin{cases} 1 & \sum_{k=1}^{N} \lambda_k a_k (sign(g_k(x) - 0.5)) \geq 0 \\ -1 & otherwise \end{cases} \quad (8)$$

where $$sign(g_k(x)) = \begin{cases} 1 & if \quad g_k(x) \geq 0 \\ 0 & otherwise \end{cases} \quad (9)$$

$a_k = \log \frac{1-\varepsilon_k}{\varepsilon_k}$, it is the weight of the kth weak classifier. ε_k is the training error rate. In Eq. (8), λ_k assigned to the kth weak classifier is a 0/1 binary variable and $\lambda_k = 0$ denotes that the kth weak classifier is not selected, and $\lambda_k = 1$ denotes that it is selected. Similarly, we assign a 0/1 binary variable η_j to a training sample x_j. If sample x_j can be correctly classified by the combination of selected weak classifiers, then $\eta_j = 0$, and otherwise, $\eta_j = 1$. It should be noted that, the strong classifier is a kind of linear classifier with λ as the normal vector. On the basis of these definitions, we construct Model IV to select the most compact weak classifiers in the condition that most samples can be correctly classified. Model IV is described as:

Model IV:

$$\min_{\lambda_k, \eta_j} \sum_{k=1}^{N} \lambda_k + C_2 \sum_{j=1}^{PN+TN} \eta_j$$

$$s.t. \begin{cases} A\tilde{\lambda} \leq r_3 \eta \\ \sum_{j=1}^{PN} \eta_j \leq \beta_{pos} \\ \sum_{j=PN+1}^{TN} \eta_j \leq \beta_{neg} \end{cases} \quad (10)$$

where $\sum_{k=1}^{N} \lambda_k + C_2 \sum_{j=1}^{PN+TN} \eta_j = \|\lambda_k\|_1 + C_2\|\eta_k\|_1$. Since the elements of λ, η are positive and their components can just be 0 or 1, this model can be considered as a special case of LML in the integer space. $\lambda = [\lambda_1, \cdots \lambda_k, \cdots \lambda_N]^T$ is a vector composed of binary variables corresponding to all weak classifiers. The sum number of weak classifiers is N, as same as the number of v-HOG blocks. $\eta = [\eta_1, \cdots \eta_{PN}, \eta_{PN+1} \cdots \eta_{PN+TN}]^T$ is the vector of binary variables corresponding to training samples. $\tilde{\lambda}$ is the homogeneous vector of λ, $\tilde{\lambda} \in R^{N+1}$, $\tilde{\lambda} = [\lambda_1, \cdots \lambda_k, \cdots \lambda_N, 1]^T$, C_2 is a predefined factor and $C_2 > 1.0$ denotes that we emphasize more on the minimizing misclassified samples than the number of selected weak classifiers. A is defined as

$$A = \begin{pmatrix} -a_1 g_1(x_1), \cdots, -a_k g_k(x_1), \cdots\cdots\cdots, -a_N g_N(x_1), 0.5(a_1 + \cdots a_N) \\ \vdots \\ -a_1 g_1(x_{PN}), \cdots, -a_k g_k(x_{PN}), \cdots\cdots, -a_N g_N(x_{PN}), 0.5(a_1 + \cdots a_N) \\ a_1 g_1(x_{PN+1}), \cdots, a_k g_k(x_{PN+1}), \cdots, a_N g_N(x_{PN+1}), -0.5(a_1 + \cdots a_N) \\ \vdots \\ a_1 g_1(x_{PN+NN}), \cdots, a_k g_k(x_{PN+NN}) \cdots, a_N g_N(x_{PN+NN}), -0.5(a_1 + \cdots a_N) \end{pmatrix}$$

and $A\tilde{\lambda} \leq r_3 \eta$ is actually the reformulation of the Eq.(10). $r_3 \in R$ is a predefined slack factor which ensures the constraint $-\sum_{k=1}^{N} \lambda_k a_k g_k(x_j) + 0.5(a_1 + \cdots a_N)$ of the jth sample can range from $[0, r_3]$ instead of [0,1]. β_{pos}, β_{neg} are the upper-bound of the number of misclassified positives and negatives respectively. $\beta_{pos} = \sigma_1 PN$, $\beta_{neg} = \sigma_2 NN$, where σ_1 is the minimum acceptable detection rate, and σ_2 is the maximum false positive rate.

The optimization Model IV is a typical 0/1 integer programming problem, in which objective function and constraints are linear. So far, there have been many available algorithms and softwares to solve the above problem. We use the Branch and Bound algorithm [22]. The basic idea of the algorithm is to convert integer programming into linear programming by using relaxation and then solve linear programming. Depending on the decision for the solution of linear programming, it separates the problem into sub-problems and utilizes fathoming to further solve the sub-problems. If there is no solution for the linear programming, it indicates that there is no solution for the integer programming. In this case, we adjust σ_2 to a lower value.

The procedure of constructing a strong classifier is summarized in Algorithm 1 as follows.

Algorithm 1. Learning a strong classifier

·Given training samples $(x_1, y_1), \cdots (x_N, y_N)$,where $y_i \in \{-1,1\}$ is the class label for negatives and positives.

· For $k = 1, \cdots, N$

For each feature vector (from a v-HOG block), train a weak classifier $g_k(x)$

1. Learn the normal vector w_k of hyper plane using L1-minimization Model II;
2. Determine the appropriate threshold θ_k for each weak classifier by solving min-max penalty function Model III;
3. Compute weights of weak classifier a_k .
4. Obtain the weak classification function $g_k(x) = w_k^T x - \theta_k$.

·End

· Select the best of weak classifiers to construct the strong classifier by solving the integer programming Model IV, using the branch and bound method to obtain the vector λ .

·The final strong classifier is

$$G(x) = \begin{cases} 1 & \sum_{k=1}^{N} \lambda_k a_k (\mathrm{sign}(g_k(x) - 0.5) \geq 0 \\ -1 & otherwise \end{cases} \quad (11)$$

4.4. Training the CLML classifier

To speed up detection, we construct cascade rejecters. In each level of the cascade, sequences of weak classifiers are selected to form the strong classifier. The strong classifier is implemented by using the integer programming model. The requirements, a minimum detection rate is 0.998 and the maximum false positive is no more than 0.3, are met in each stage. In accordance with the Model IV, we set the parameter: $\sigma_1 = 0.998$, $\sigma_2 < 0.3$ .The training procedure of CLML is as follows:

Algorithm2. The Cascaded L1-norm Minimization Learning (CLML) method

---

·**Input:** the minimum detection rate σ_1 , maximum acceptable false positive rate σ_2 in *t*th 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 *t*th level of cascade.

D_t is detection rate in the *t*th 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 IV

increase $\Delta_t = \Delta_t + 0.1$

else

1.Solve integer programming Model IV.

2. Evaluate Pos and Neg by current strong classifier

3. Compute f_t under this threshold

–End

$-F_{t+1} = F_t \times f_t$

$-D_{t+1} = D_t \times \sigma_1$

–NEG $\leftarrow \varnothing$

–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

4.5. Discussion

This paper proposed a new classifier based on L1-norm

Minimization scheme, which are theoretically different from SVM and Adaboost and can directly minimize the VC-dimension. Compared with L2-norm used in SVM, L1-norm is effective for achieving sparseness which appreciates the local feature. While L2-norm emphasizes more on "average variation", which means each component of a vector varies almost equally, instead of sparseness. Hence L2-norm is not suitable for feature selection.

The proposed method is faster than the Linear-SVM and Kernel-SVM during the detection. The Linear-SVM employs the inner-product between the test sample and the support vectors. Instead, our weak classifier only projects the test sample on the normal vector with 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.

The strong classifier is the core part of this paper. It adopts integer programming to construct the strong classifier, which is different from the iteration method utilized in Adaboost. The integer programming can be interpreted as a special case of the L1-norm Minimization in the integer space to select the minimal weak-classifiers. The strong classifier of CLML in each cascade is actually obtained based on the VC-dimension minimization. The extensive experiment results in the following section validate that our method outperforms the Adaboost method.

5. Experiments

There are about 2400 training positives from MIT and SDL [25] for front view and about 4900 negatives from INRIA datasets. More negatives will be added in the training process.

We evaluate our algorithm via the challenging INRIA test set of 288 images [8]. In this test set, humans are mostly in standing position, while it covers more diverse body poses and cluttered backgrounds. Although the chosen training positives are mostly from front view, the trained model can handle multi-posture and occlusion cases, demonstrated by experiments.

5.1. Normalization

To deal with the variability of appearance, illumination conditions and background, the normalization of feature and normal vectors of weak classifier are carried out. The 36-dimension feature vector of each block and the normal vector are normalized respectively as follow:

$$x_i^k = \frac{x_i^k}{\sqrt{\sum_{j=1}^{36} x_{ij}^k + \varepsilon}} \qquad w^k = \frac{w^k}{\sqrt{\sum_{j=1}^{36} w_j^k + \varsigma}} \qquad (12)$$

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

5.2. Parameter Analysis

There are several important parameters when learning the weak classifiers. In Model III, 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 impacting the threshold should be different. Generally, r_1 of positives should be larger than r_2 to guarantee that most of the positives remain for the next level training. However, regarding of classification accuracy of negatives, we can not increase r_1 too much. It is appropriate to set the ratio r_1 / r_2 in [1, 40]. In Figure 1, we illustrate the influence of the ratio on positive and negative classification accuracy in the first cascade level.

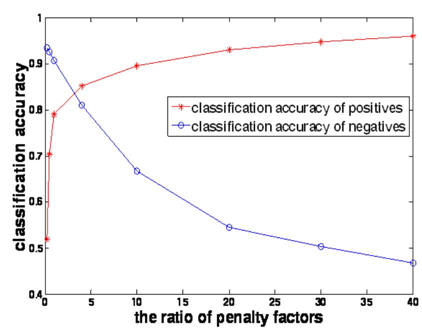


Figure 1. Classification accuracy with different r_1 / r_2 ratio.

5.3. Evaluation and Comparison

In Figure 2, we present the results of the cascade classifier. It can be seen that about 6 cascades are enough to reject 95% of the negatives. The classification for a detection window needs to calculate 3.9 feature blocks averagely, which is less than that of Zhu's method. In addition, as we discussed in section 4.5, the computational time of LML weak classifier is less than Dalal's [8] and Zhu's [9]. Therefore, our detection speed is higher than theirs.

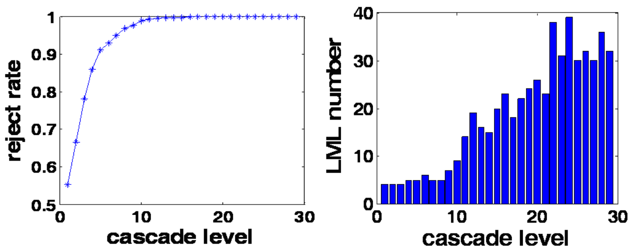
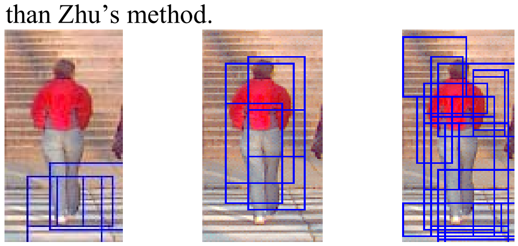


Figure 2. Left: the accumulated rejection rate over all cascade levels. Rigth: the number of LML weak classifiers at each level.

We can get the most compact feature blocks by our method. Figure3 (a) shows the best four blocks which are different from Zhu's, in whose paper blocks with size 36×80 pixels are considered the best. However, in the first level, our best block size is 40x40 pixels in the leg parts of human body. In the second level, our best block size is 36x80 pixels covering the contour of human body as Zhu reported. 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 variety and locality of features. The detection error curve (Figure 4) demonstrates our method is better than theirs.

We compare our classifier with Zhu's method (SVM classifiers and Adaboost cascade) [9] using miss rate tradeoff False Positives Per Window (FPPW) on a log scale. The same v-HOG features are employed as human representation in the two methods. The miss rates on false positives per window (FPPW) are shown in Figure 4. Points on curves are obtained from different cascade levels. It can be seen that the proposed classifier performs much better than Zhu's method.



(a) (b) (c)

Figure 3. Feature blocks. (a) Blocks selected in the first level, (b) in the second level and (c) in 15th level of the cascade.

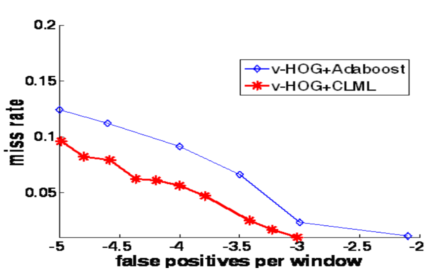


Figure 4. The comparison of our method with Zhu's via same v-HOG features.

To evaluate our human detection algorithm, we compare it with the state of the art, including v-HOG+Adaboost [9], HOG+SVM method [8], and the COV+Logitboost method [11], which is shown in Figure 5 on a log scale. We implement the method [8] by the open source codes of HOG LibSVM, while the curves of the methods [9, 11] are obtained from their reported results. As shown in the figure, our method reaches a much better performance than the HOG-based results on the INRIA dataset. Comparing at the FPPW rate of 10^{-5} our method achieves 9% miss rate, which is about 8% lower than HOG+SVM method, about 3% lower than Zhu's and about 1% lower than Tuzel' method which use different human representation (COV features) from ours.

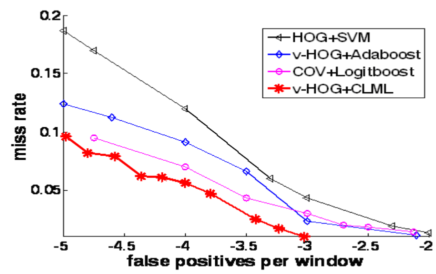
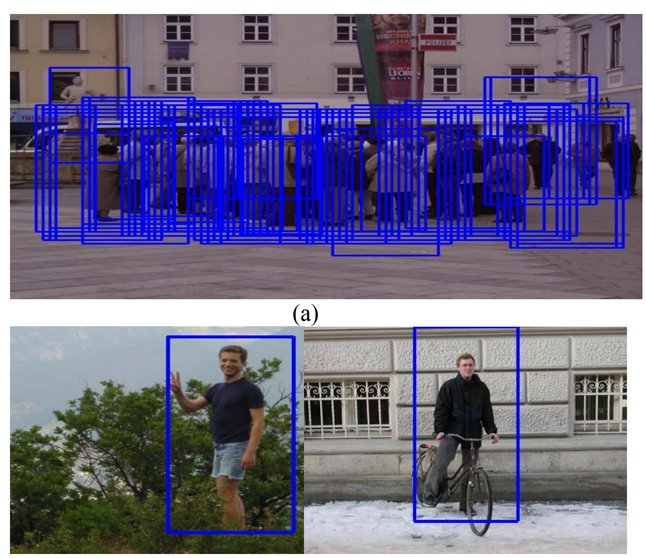


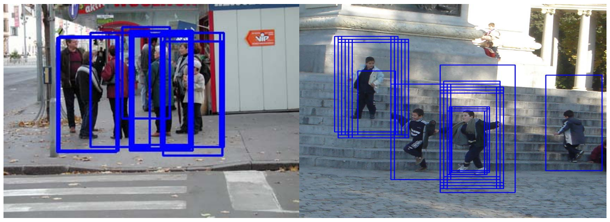
Figure 5 The comparison of our method with the state of arts.

In Figure 6, we show some detection examples without merging results from multiple detection scales. In Figure.6a most of the pedestrians are correctly located whether or not they are occluded or in multi-posture, except that the rightmost person is missed. The statue in the up-left side of picture is detected, since it is very similar to human being. In Figure.6e, four children are correctly located although they have posture variation.



(a)

(b) (c)



(d) (e)

Figure 6. Detection examples.

6. Conclusion and Future Work

In this paper, we propose a new learning method, which integrates feature selection with classifier construction via solving optimization model. The proposed learning strategy based on L1-norm minimization can minimize the VC-dimension of the proposed weak and strong classifiers.

A cascade of L1-norm minimization learning (CLML) classifiers are constructed to detect human in images. Both theoretical and experimental results demonstrate the effectiveness and efficiency of the classifiers. At present, the proposed method is applied to human detection, and in the future, it will be extended to other objects e.g. vehicles etc.

7. Acknowledgement

This work is supported by Major State Basic Research Development Program of China (973 Program) with No. 2010CB731800, National Natural Science Foundation of China with No. 60872143, No.60903065.

References

[1] Christopher J.C. Burges, A Tutorial on Support Vector Machines for Pattern Recognition, *Data Mining and Knowledge Discovery,* vol. 2(2), pp.121-167, 1998

[2] M.Collins, R.E. Schapire and Y. Singer, Logistic Regression, AdaBoost and Bregman Distances, *Machine Learning*, vol.48(1-3), pp.253-285, 2002

[3] D. Donoho, For most large underdetermined systems of linear equations the minimal *l*1-norm near solution approximates the sparsest solution, *Comm.on Pure and Applied Math*, vol.59(6),pp.797-829,2006.

[4] K Huang, SAviyente, Sparse representation for signal classification, *Advances in Neural Information Processing Systems, Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems*, 2007

[5] A. Y. Yang, J. Wright, Y. Ma, and S. S. Sastry, Robust Face Recognition via Sparse Representation, *IEEE Transactions on PAMI*, vol.31(2), 2009.

[6] Gavrila, D.M. and Giebel, J. Shape-based pedestrian detection and tracking. *IEEE Intelligent Vehicle Symposium*, vol.1, pp.8-14, 2002.

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

[8] Dalal, N, Triggs, B., Histograms of Oriented Gradients for Human Detection, *IEEE CVPR*, vol.1, pp.886-893, 2005.

[9] Q. Zhu, S. Avidan, M. C. Yeh, and K. T. Cheng. Fast human detection using a cascade of histograms of oriented gradients. *IEEE CVPR*, vol 2, pp.1491-1498,2006.

[10] Serre, T., Wolf L., Bileschi S., Riesenhuber M. and Poggio T., Object Recognition with Cortex-like Mechanisms, *IEEE Transactions on PAMI*, vol.29(3), pp.411-426, 2007.

[11] Tuzel O., Porikli F., Meer P., Pedestrian detection via Classification on Riemannian manifolds, *IEEE Transactions on PAMI,* vol. 30(10), pp.1713-1727, 2008.

[12] Mu Y. Yan S. Liu Y., Huang T., Zhou B., Discriminative Local Binary Patterns for Human Detection in Personal Album, *IEEE CVPR*, issue 23-28, pp.1-8, 2008

[13] Xiaoyu Wang, Tony X Han, Shuicheng Yan., An HOG-LBP Human Detector with Partial Occlusion Handling, *IEEE ICCV*, Kyoto, 2009.

[14] B.Wu, and R. Nevatia. Detection of multiple, partially occluded humans in a single Image by Bayesian combination of Edgelet part detectors. ICCV 2005.

[15] P.F. Felzenszwalb and D.P. Huttenlocher. Pictorial structures for object recognition, *IJCV*, vol. 61(1), pp.55–79, 2005.

[16] S. Ioffe and D.A. Forsyth. Probabilistic methods for finding people, *IJCV*,vol. 43(1), pp.45–68, 2001.

[17] B. Leibe, E. Seemann, and B. Schiele. Pedestrian detection in crowded scenes *IEEE CVPR*, vol. 1, pp.878-885, 2005.

[18] D. Vinay, J. Neumann, V. Ramesh, and L.S. Davis, Bilattice-based logical reasoning for human detection, *IEEE CVPR* , 2007.

[19] A. Mohan, C. Papageorgiou, and T. Poggio, Example-based object detection in images by components, *IEEE Transactions on PAMI*, vol.23(4) pp.349-360, 2001

[20] Sabzmeydani P. and Mori G., Detecting Pedestrians by Learning Shapelet Features, *IEEE CVPR*, 2007.

[21] S. Munder and D. Gavrila. An experimental study on classification. IEEE Trans. PAMI, 28(11):1863–1868, 2006.

[22] Michael Juger, Denis Naddef**,** Computational Combinatorial Optimization: Optimal or Provably Near-Optimal Solutions**,** Springer Press, 2001

[23] 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*, vol.1(4),pp.586-597,2007

[24] Zhi-Hua Zhou, Jianxin Wu, W. Tang, Ensembling neural networks: Many could be better than all. *Artificial Intelligence* . vol.137(1-2), pp. 239-263, 2002

[25] http://coe.gucas.ac.cn/SDL-Homepage/resource.html