LETTER

Partial Derivative Guidance for Weak Classifier Mining in Pedestrian Detection

Chang LIU[†a)], ***Student Member***, **Guijin WANG**[†b)], ***Member***, **Chunxiao LIU**[†c)], ***and*** **Xinggang LIN**[†d)], ***Nonmembers***

SUMMARY Boosting over weak classifiers is widely used in pedestrian detection. As the number of weak classifiers is large, researchers always use a sampling method over weak classifiers before training. The sampling makes the boosting process harder to reach the fixed target. In this paper, we propose a partial derivative guidance for weak classifier mining method which can be used in conjunction with a boosting algorithm. Using weak classifier mining method makes the sampling less degraded in the performance. It has the same effect as testing more weak classifiers while using acceptable time. Experiments demonstrate that our algorithm can process quicker than [1] algorithm in both training and testing, without any performance decrease. The proposed algorithms is easily extending to any other boosting algorithms using a window-scanning style and HOG-like features.

key words: *pedestrian detection, partial derivative, classifier mining, HOG, boosting*

1. Introduction

Pedestrian detection is important for many applications in fields of computer vision such as visual surveillance, image retrieval and driver assistance system. But at the same time, pedestrian detection is still a challenging task because of the variations in appearance, articulation, posture and illumination condition.

Among various algorithms of pedestrian detection, the boosting-from-weak-classifier ones are probably the most popular scheme. These algorithms combine local image features into a strong classifier, which is then applied to all possible sub-windows in the input images to detect pedestrians. Viola et al. [2] proposed an algorithm using Haar wavelet features with the Adaboost and cascade training framework [3]. Dalal and Triggs [4] presented a new feature called Histogram of Oriented Gradient (HOG), which is notably more effective than Haar wavelet in pedestrian detection. Zhu et al. [5] combined HOG feature with the cascade structure, thereby reduced scanning detection time significantly. Sabzmeydani and Mori [6] described a mid-level feature set called shapelet, trained by two levels of Adaboost. Tuzel et al. suggested utilizing covariance matrices as object descriptors and a learning algorithm on Riemannian manifolds. Lin et al. [7] introduced a multiple instance feature to mitigate feature misalignment problem.

Meanwhile, some researchers have attempted to improve the time efficiency of pedestrian detection algorithms. The combination of boosting and cascade structure is a widely used approach. Most efficient works [2], [3], [5]–[7] depend on this structure.

However, in every stage of cascade training, the evaluation of each weak classifier is very time consuming. Considering the training time efficiency, researchers always adopt certain sampling on the feature pool in practical. In comparison with using all features in feature pool, using sampling will result in a little worse performance. [5] The researchers took into account the unacceptable long training time and tolerate the disparity. However, it is practical to search better weak-classifier and make the disparity smaller under the acceptable training time. There are already some successful works in feature selection of online boosting [8], in which a gradient-based feature selection approach is proposed in pedestrian tracking.

We propose a weak classifier mining algorithm in picking weak classifiers which comes out with better result than traditional sampling method. Since diamond search algorithm [9] is useful in finding the minimum residual in the neighborhood and is widely used in video coding, we use it in finding the minimum false alarm rate in the neighborhood of feature space. The proposed weak classifier mining is steered by partial derivatives in feature space. We also propose a practical computing method for these partial derivatives which will notably reduce time consumption in efficient mining of the weak classifiers and can be easily extended to any other boosting algorithm that has a window-scanning style and HOG-like features. Corresponding experiments confirmed that the result after using weak classifier mining is close to the ideal result of evaluating every block in each stage.

2. Typical Training Structure of Cascade Boosting HOG

In a typical pedestrian detector training approach via boosting HOG, numerous HOG blocks with different sizes and positions are included into feature pool. By using weighted samples, every feature can be trained as a weak classifier. In other word, each weak classifier is associated with one feature. In every stage, weak classifiers are continually collected to form a strong classifier.

For every weak classifier, the output is

Manuscript received January 27, 2011.
Manuscript revised March 29, 2011.

[†]The authors are with the Department of Electronic Engineering, Tsinghua University, Beijing, 100084 China.
a) E-mail: chang-liu06@mails.tsinghua.edu.cn
b) E-mail: wangguijin@tsinghua.edu.cn
c) E-mail: lcx08@mails.tsinghua.edu.cn
d) E-mail: xglin@tsinghua.edu.cn

DOI: 10.1587/transinf.E94.D.1721

$$f(x_i; p; l) = \frac{2}{\pi} arctg(\beta^T h_p(x_i) - t) \quad (1)$$

where $\{x_i\}$ is the dataset with K positive and negative samples in total. $p = \{l_x, l_y, c_x, c_y\}$ is the feature parameters of block, and $l = \{\beta, t\}$ is the parameter of linear classifier.

A strong classifier is the sum of several different weak classifiers.

$$SC(x_i) = \sum_{j=1}^{J} f(x_i; p_j; l_j) \quad (2)$$

where J is the weak classifiers number in that stage.

The form of cascade classifier is several strong classifiers in series.

$$CC(x_i) = \begin{cases} 1 & SC_s(x_i) > 0 \; for \; every \\ & stage \; s = 1 \ldots S \\ -1 & otherwise \end{cases} \quad (3)$$

where S is the stage number.

3. Partial Derivatives in Feature Space

3.1 Feature Space of HOG Block

For a typical HOG block, we use four parameter to describe it, l_x, l_y, c_x, c_y, which represent the x-coordinate and y-coordinate of the block center and the width and height of the cell (half of the corresponding block size) (Fig. 1). We denote the HOG block feature space by $\{l_x, l_y, c_x, c_y\}$. It is easy to prove that these four figures are independent in forming a feature space of HOG block.

3.2 Partial Derivatives in Feature Space

In our proposal, we seek to find a best block for current boosting stage in the neighborhood of feature space. For best block, we mean the block that minimizes the mean square error (MSE).

$$\varepsilon = \sum_{i=1}^{K} w_i (f(x_i; p; l) - y_i)^2 \quad (4)$$

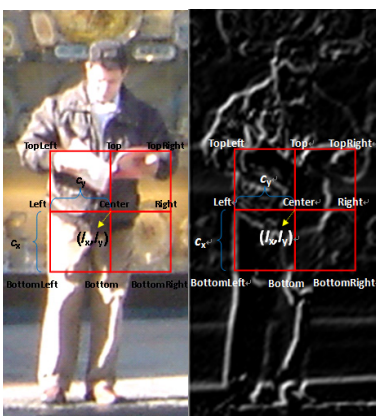

Fig. 1 The parametrization and edge points of a block.

where w_i is the current weight of all the K samples. $\{y_i\}$ is the positive or negative flag. Taking the derivative with respect to p gives

$$\frac{d\varepsilon}{dp} = \sum_{i=1}^{K} 2w_i (f(x_i; p; l) - y_i) \frac{df_i}{dp} \quad (5)$$

where $\frac{df_i}{dp} = [\frac{\partial f_i}{\partial l_x}, \frac{\partial f_i}{\partial l_y}, \frac{\partial f_i}{\partial c_x}, \frac{\partial f_i}{\partial c_y}]$. Based on the theory of partial derivatives, we have

$$\frac{\partial f_i}{\partial l_x} = \frac{2}{\pi} \frac{\beta^T \frac{\partial h_p(x_i)}{\partial l_x}}{1 + (\beta^T h_p(x_i) - t)^2} \quad (6)$$

The $h_p(x_i)$ in Eq. (6) is the HOG bins computed from the sample $\{x_i\}$ in the special block determined by parameter p. The other partial derivatives, $\frac{\partial f_i}{\partial l_y}$, $\frac{\partial f_i}{\partial c_x}$ and $\frac{\partial f_i}{\partial c_y}$ can be computed in the same manner. In practice, the feature is computed by integral image $\overline{x}_{i,j}$. Since there are nine bins feature in 4 different cells in the block, they are calculated as follows

$$\begin{aligned} h_p(x_{i,j}) &= \overline{x}_{i,j}(TopLeft(p)) + \overline{x}_{i,j}(Center(p)) \\ &\quad - \overline{x}_{i,j}(Top(p)) - \overline{x}_{i,j}(Left(p)) \\ h_p(x_{i,j+b}) &= \overline{x}_{i,j}(Top(p)) + \overline{x}_{i,j}(Right(p)) \\ &\quad - \overline{x}_{i,j}(TopRight(p)) - \overline{x}_{i,j}(Center(p)) \\ h_p(x_{i,j+2b}) &= \overline{x}_{i,j}(Left(p)) + \overline{x}_{i,j}(Bottom(p)) \\ &\quad - \overline{x}_{i,j}(Center(p)) - \overline{x}_{i,j}(BottomLeft(p)) \\ h_p(x_{i,j+3b}) &= \overline{x}_{i,j}(Center(p)) + \overline{x}_{i,j}(BottomRight(p)) \\ &\quad - \overline{x}_{i,j}(Right(p)) - \overline{x}_{i,j}(Bottom(p)) \end{aligned} \quad (7)$$

where $i \in [1, K]$ and $j \in [1, b]$. As we use the discrete differentiation formula as

$$\frac{\partial \overline{x}_{i,j}}{\partial x}|_{(x_0,y_0)} = \frac{1}{2}[\overline{x}_{i,j}(x_0 + 1, y_0) - \overline{x}_{i,j}(x_0 - 1, y_0)] \quad (8)$$

We can calculate the partial derivatives in Eq. (6) as

$$\begin{aligned} &\frac{\partial h_p(x_{i,j})}{\partial l_x} \\ &= +\frac{1}{2}[\overline{x}_{i,j}(TopLeft(p) - (1,0)) + \overline{x}_{i,j}(Left(p) + (1,0)) \\ &\quad - \overline{x}_{i,j}(TopLeft(p) + (1,0)) - \overline{x}_{i,j}(Left(p) - (1,0))] \\ &\quad + \frac{1}{2}[\overline{x}_{i,j}(Top(p) - (1,0)) + \overline{x}_{i,j}(Center(p) + (1,0)) \\ &\quad - \overline{x}_{i,j}(Top(p) + (1,0)) - \overline{x}_{i,j}(Center(p) - (1,0))] \end{aligned} \quad (9)$$

The homologous partial derivatives c_x, l_y and c_y can be calculated similarly. After involving these in Eq. (6), we can practically calculate the partial derivatives in feature space.

4. 4D Diamond Search Algorithm

We use the proposed 4D diamond search algorithm to find the best (or probably best) block in adjacent feature space. Of course by traversing all the adjacent blocks and training the corresponding SVM classifier, we may directly find the

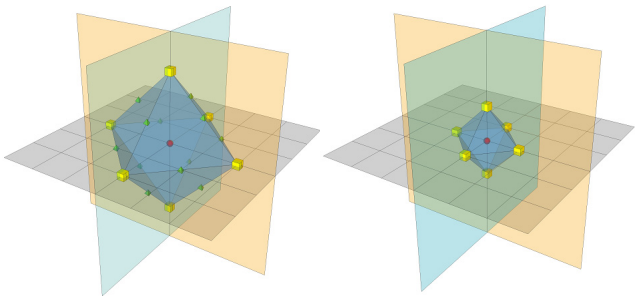

Fig. 2 4D large diamond search pattern and small diamond search pattern. (Schematic figures is 3D)

best block with its SVM parameters. But such method is unaffordably time consuming. The 4D DS algorithm could significantly reduce the time consumption in training SVM classifier. This dramatic decline in searching time makes selecting best block in neighborhood feasible. The proposed 4D DS algorithm employs two search patterns as illustrated in Fig. 2. The first pattern, called large diamond search pattern (LDSP), comprises 33 points from which 32 points surround the center one to compose a 4D diamond shape. Among all the 32 blocks selected, only one block needs to be calculated with SVM training. The second search pattern is small diamond search pattern (SDSP), it consists of 9 points in feature space. In the searching procedure of the 4D DS algorithm, LDSP is repeatedly used until LDSP can not improve the MSE ε for all the weighted samples. The search pattern is then switched from LDSP to SDSP and SDSP is used only once. Among the 8 points in SDSP, we select 4 of them to train SVM classifier. After taking the center point into concern, the best block is then chosen with its SVM parameters in the measurement of minimum MSE ε.

4.1 Large Diamond Search Pattern

There are 32 integral points on the 4D diamond shape in the LDSP. If all 32 HOG blocks are calculated with SVM training, the time complexity is unaffordable during several LDSP steps in a searching process. So we employ the partial derivatives in feature space to guide the diamond search. Firstly, we calculate the partial derivatives in feature space in Eq. (5), and get their absolute value: $\left|\frac{d\varepsilon}{dl_x}\right| \left|\frac{d\varepsilon}{dl_y}\right| \left|\frac{d\varepsilon}{dc_x}\right| \left|\frac{d\varepsilon}{dc_y}\right|$. Secondly, we select the first and second parameters with largest partial derivatives as p_1 and p_2 ($\in p$)

$$p_1 = \arg\max_{p} \left|\frac{d\varepsilon}{dp}\right| \tag{10}$$

$$p_2 = \arg\max_{p\setminus\{p_1\}} \left|\frac{d\varepsilon}{dp}\right| \tag{11}$$

- If $\left|\frac{d\varepsilon}{dp_1}\right| > (\sqrt{2}+1)\left|\frac{d\varepsilon}{dp_2}\right|$, the step of this LDSP is set as double unit vector in p_1 axis. And also, the direction is decided by the sign of $\frac{d\varepsilon}{dp_1}$. The destination point is one of the vertex of 4D diamond. For instance, $\left|\frac{d\varepsilon}{dl_y}\right|$ is several times more than the other three absolute value and $\frac{d\varepsilon}{dl_y}$ is negative. The LDSP step is set as $(0, -2, 0, 0)$ in feature space.
- If $\left|\frac{d\varepsilon}{dp_1}\right| < (\sqrt{2}+1)\left|\frac{d\varepsilon}{dp_2}\right|$, the step is set as the sum of unit vectors in p_1 and p_2 axis. Similarly, the direction of two unit vectors are decided by the sign of $\frac{d\varepsilon}{dp_1}$ and $\frac{d\varepsilon}{dp_2}$. For instance, $\left|\frac{d\varepsilon}{dl_x}\right|$ and $\left|\frac{d\varepsilon}{dc_y}\right|$ are more or less the same while the other two absolute value are smaller. And $\frac{d\varepsilon}{dl_x}$ is negative but $\frac{d\varepsilon}{dc_y}$ is positive. The LDSP step is set as $(-1, 0, 0, 1)$ in feature space.

Finally, the SVM classifier of the block corresponding to the destination feature point is trained and the MSE is compared with that of the center point. If the destination point is better, this point is set as a new center point and a new LDSP will be start. On the other hand, if the destination point is worse than the center point. The search pattern is switched to SDSP.

4.2 Small Diamond Search Pattern

There are 9 points in the SDSP. One is the center point and the others are distributed on the four axes. We use the p_1 which has been calculated in the last LDSP step to judge the axis and direction of candidate point. The SVM classifier of the corresponding block is trained. The point which has least MSE between center point and candidate point is chosen as the final search result of diamond search.

5. Experiments

To evaluate the performance of the proposed algorithm, we implemented it with the support of the Intel OpenCV library and a PC running on a 2.33 GHz Intel CPU. Except for our original weak classifier mining algorithm, other components of training and detection algorithm are implemented according to Pang's work [1] in order to demonstrate the improvement of our algorithm over fundamental methods.

We used the INRIA dataset [4] as our training and testing samples, which includes 2416 positive training samples. For negative samples, we bootstrapped new training samples from the negative training images in the INRIA dataset at the beginning of training each cascade stage. The test dataset contains 1126 calibrated positive samples and 453 negative images. As in Pang's work [1], we chose HOG as feature, linear SVM as the weak classifier, and Adaboost + cascade as the training framework. We sampled 5% features from the feature pool in every training process. Figure 3 compares the weak classifier number involved in every stage of our algorithm with the number of basic algorithm. It can be noticed that the number of weak classifier has about 25% in decrease. In training section, we use the same goal in each level. Thus, we can prove that our algorithm use less weak classifiers to get the same performance. This will speed up the training and detection process about 15% and 10%. Table 1 summarizes the training and detection time comparison.

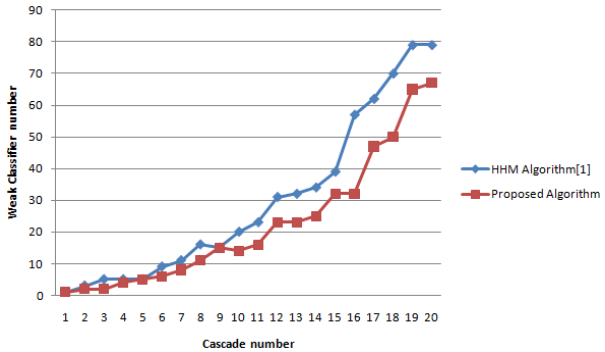

Fig. 3 Weak classifier number comparison.

Table 1 A comparison of training time and detection time on a 320×240 sized image between our algorithm and [1] algorithm.

	Training time	Detection time
HHM (Pang [1])	9.7 day	33 ms
Proposed algorithm	8.2 day	30 ms

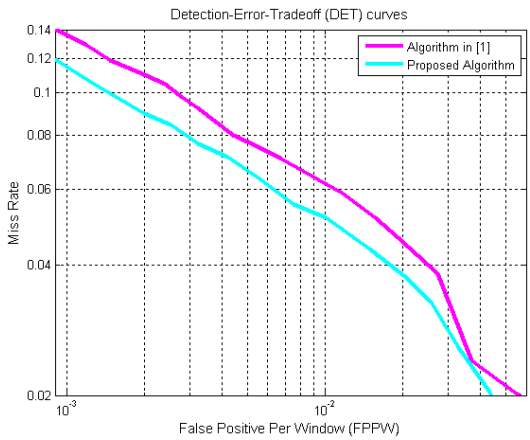

Fig. 4 ROC curve of proposed algorithm together with algorithm in [1].

The comparison demonstrates the advance in boosting results of our algorithm, since weak classifier mining method make the boosting process cover much more HOG features.

Figure 4 provides the ROC curve of proposed algorithm together with algorithm in [1]. Theoretically, our new algorithm only reorganizes the computation structure, without any simplification or approximation. Therefore, in the terminal case, it will not make any difference in performance compared to the reference algorithm. But in limited stage, the performance is different by using the weak classifier mining. Our weak classifier mining method should be easily extended to any other boosting algorithms using a window-scanning style and HOG-like features.

6. Conclusion

In this paper, we propose a partial derivative guidance for weak classifier mining method which can be used together with boosting algorithm. We make the sampling less degraded in the performance by weak classifier mining method. And fewer classifiers involved in the detector make the time consumption less in training and detection.

Experiments demonstrate that the performance of our algorithm keeps the same with fewer weak classifiers.

References

[1] G. Pang, G. Wang, and X. Lin, "Real-time human detection using hierarchical hog matrices," IEICE Trans. Inf. & Syst., vol.E93-D, no.3, pp.658–661, March 2010.
[2] P. Viola, M.J. Jones, and D. Snow, "Detecting pedestrians using patterns of motion and appearance," ICCV, 2003, pp.734–741, 2003.
[3] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," CVPR, 2001, vol.1, pp.I–511–I–518, 2001.
[4] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," CVPR, 2005, vol.1, pp.886–893.
[5] Q. Zhu, M. Yeh, K. Cheng, and S. Avidan, "Fast human detection using a cascade of histograms of oriented gradients," CVPR, 2006, vol.2, pp.1491–1498, 2006.
[6] P. Sabzmeydani and G. Mori, "Detecting pedestrians by learning shapelet features," CVPR, 2007, pp.1–8, 2007.
[7] Z. Lin, G. Hua, and L.S. Davis, "Multiple instance feature for robust part-based object detection," CVPR, 2009, pp.405–412, 2009.
[8] X. Liu and T. Yu, "Gradient feature selection for online boosting," ICCV, 2007, pp.1–8, 2007.
[9] S. Zhu and K. Ma, "A new diamond search algorithm for fast block-matching motion estimation," IEEE Trans. Image Process., vol.9, no.2, pp.287–290, Feb. 2000.