J Electr Eng Technol.2015; 10(3): 1227-1232
http://dx.doi.org/10.5370/JEET.2015.10.3.1227

ISSN(Print) 1975-0102
ISSN(Online) 2093-7423

Cascade Selective Window for Fast and Accurate Object Detection

Shu Zhang†, Yong Cai* and Mei Xie*

Abstract – Several works help make sliding window object detection fast, nevertheless, computational demands remain prohibitive for numerous applications. This paper proposes a fast object detection method based on three strategies: cascade classifier, selective window search and fast feature extraction. Experimental results show that the proposed method outperforms the compared methods and achieves both high detection precision and low computation cost. Our approach runs at 17ms per frame on 640×480 images while attaining state-of-the-art accuracy.

Keywords: Object detection, Cascade, Adaboost, Selective window search

1. Introduction

Object detection is a fundamental problem for many computer vision tasks, e.g. surveillance, traffic analysis, clinical diagnosis, face recognition, and robotics. Substantial progress have been made on object detection for the past few years, scaling up to thousands of object categories and obtaining industry-level performance [1, 2]. However, the existing methods remain time consuming for many practical applications [3], which is caused by evaluating a large number of windows in the sliding window search framework [4]. In addition, sophisticated features and classifiers would further decrease detection speed [1].

Notable works for increasing detection speed can be broadly classified into three categories: cascade classifier [2, 5], selective window search [6] and fast feature extraction [7]. Cascade classifier first proposed in [5] effectively saves the detection time by rejecting many true negatives in the early stages of the cascades. Then, some improvement work [2] was done to increase detection precision and speed. However, the existing cascade approaches are still suffering from time-consuming training.

The second category, i.e. selective window search, speed up detection by avoiding the useless search over non-object regions. In [8], the authors proposed an efficient window search using a branch and bound technique. However this method has strict requirements over the classifier score that are not met by most of the existing classifiers. Additionally, several works [6, 9, 10] search objects using coarse-to-fine strategy. For example, Gualdi [6] searched the image toward the area where the target objects are more likely to be found in an iterative manner. Successful detections at coarse resolutions yield to refined searches at finer resolutions. Nevertheless, the speed-up by only using the selective window search strategy is not obvious.

Improving the feature extraction is another efficient work to speed up the detection. Viola and Jones [5] introduced integral images for fast feature computation, but the simple feature was also verified to decrease the detection precision. Recently, channel feature computed by approximate algorithm [7] achieved state-of-the-art performance with the fastest in the literature. However, the high-dimensional channel feature would increase the computational cost in evaluating each window.

To overcome the aforementioned limitations of existing methods, this paper proposes a cascade selective window method (CSW) for fast object detection in terms of three aspects: First, high-dimensional image channel feature is compressed by a sparse projection matrix, which reduces the evaluation time of classifier. Second, this work uses a generalization of the cascade architecture to design a soft cascade SVM classifier, which generates a detection performance comparable to that of the best published ones

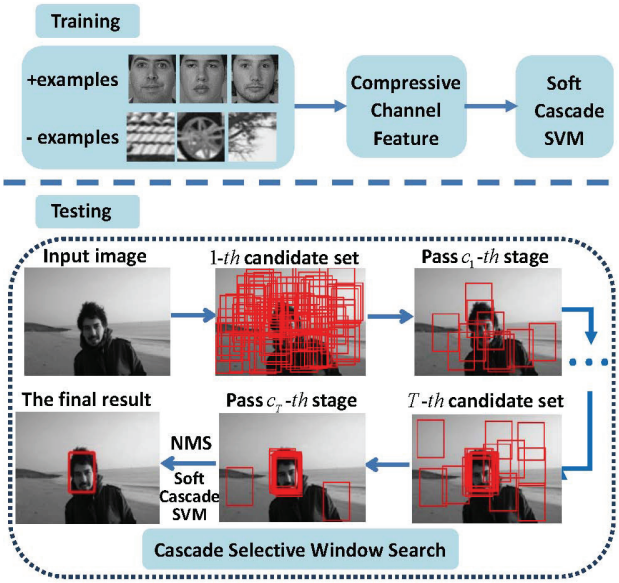

Fig. 1. The flowchart of the proposed detection algorithm

† Corresponding Author: School of Electronic Engineering, University of Electronic Science and Technology of China, China. (jlu zhangshu@163.com)
* School of Electronic Engineering, University of Electronic Science and Technology of China, China. (caiiong@163.com, xiemei@ee.uestc.edu.cn)
Received: June 23, 2014; Accepted: November 27, 2014

[2] while allowing for faster training. Third, this work proposes a coarse-to-fine window search method, which is introduced into soft cascade SVM classifier to further increase detection speed. Fig. 1 shows the flowchart of the proposed detection algorithm.

2. Cascade Selective Window Method

2.1 Compressive channel features

Given an input image window, several channels with the same dimensions are first computed by [7] (See Fig. 2). Sum over each rectangular channel region serves as a first-order feature and can be computed efficiently using integral images [5]. Then all of these first-order features are concatenated to form a high dimensional feature vector $v=(v(1),...,v(h))^T \in \mathbb{R}^h$. This paper intends to use a random measurement matrix $A \in \mathbb{R}^{k\times h}$ to project $v \in \mathbb{R}^h$ onto a vector $x \in \mathbb{R}^k$ in a low dimensional space, namely $x = Av$. The random matrix A needs to be computed only once off-line and remains fixed throughout the detection process.

The work in [11] proved that if v is compressive (such as audio or image) and the random matrix A satisfies the restricted isometry property, v can be reconstructed with minimum error from x with high probability. This theoretical support enables us to classify the high-dimensional features via its low-dimensional random projections. A typical measurement matrix satisfying the restricted isometry property is the random Gaussian matrix $A \in \mathbb{R}^{k\times h}(a_{i,j} \sim N(0,1))$. However, as the matrix is dense, the memory and computational loads are still high when h is large. To solve this problem, a very sparse random measurement matrix [12] is applied in this paper to approximate random Gaussian matrix, where the entries is defined as:

$$a_{i,j} = \frac{\sqrt{h}}{2} \times \begin{cases} 1 & with\ probability\ \ \frac{2}{h} \\ 0 & with probability\ \ 1-\frac{4}{h} \\ -1 & with probability\ \ \frac{2}{h} \end{cases} \tag{1}$$

$$\frac{\sqrt{h}}{2} \times \begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 & \cdots & 1 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 1 & 0 & 0 & \cdots & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} v(1) \\ v(2) \\ \vdots \\ v(h) \end{bmatrix} = \begin{bmatrix} x(1) \\ x(2) \\ \vdots \\ x(k) \end{bmatrix}$$

$$A \in \mathbb{R}^{k\times h} \quad x(i) = \sum_j a_{ij} v(j)$$

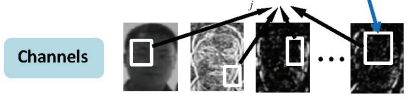

Fig. 2. Illustration of compressive channel features

As the dimensionality h is very large, many entries in the matrix are zeros. As shown in Fig. 2, only the nonzero entries in A and the corresponding first-order features are involved in computation, so computational cost is dramatically reduced.

2.2 Soft cascade SVM

To begin with, a linear SVM model is learned by using compressive channel features of training samples, as shown in formula (2).

$$f(x) = \beta + \sum_i \alpha_i \langle x, x_i \rangle \tag{2}$$

where α_i denotes the learned weight of each training samples, β is the learned bias. x_i , x denote the feature vectors of i-th training sample and test sample respectively. Let $x_i(j)$, $x(j)$ denote the j-th dimension feature of x_i and x , Eq. (2) can be transformed as below:

$$f(x) = \beta + \sum_{j=1}^{k} x(j) \sum_i \alpha_i x_i(j) = \beta + \sum_{j=1}^{k} \omega_j x(j) \tag{3}$$

where $\omega_j = \sum_i \alpha_i x_i(j)$ is the $j\text{-}th$ dimension feature's

Algorithm 1. Post-training process of soft cascade SVM

Input:

Positive sample set Pos and negative sample set Neg

Feature's weight: $\{\omega_j\}_{j=1}^K$

Set $D_0 = \varnothing$

For $(n = 1,...,k-1)$

$nneg_{opt} = 0$

For $(m = 1,...,k)$

If $(m \notin D_{n-1})$

Define a temporary classifier :

$$f_n^*(x) = \sum_{j\in D_{n-1}} \omega_j x(j) + \omega_m x(m)$$

Rejection threshold :

$$r_n = \min_{x\in Pos} f_n^*(x)$$

The number of negative samples which is smaller than r_n

$$nneg = card\{x \mid f_n^*(x) < r_n, x \in Neg\}$$

If $(nneg > nneg_{opt})$

$$m_{opt} = m,\ nneg_{opt} = nneg$$

End

End

End

$D_n = D_{n-1} \cup m_{opt}$

Obtain the $n\text{-}th$ stage classifier:

$$f_n(x) = \sum_{j\in D_n} w_j x(j)$$

End

Output:

Soft cascade SVM $f_n(x)_{n=1}^k$, rejection threshold $\{r_n\}_{n=1}^k$

Where $f_k(x) = f(x)$, $r_k = 0$

weight.

Based on linear SVM model, this work proposes a post-training process for each stage of cascade (as shown in Algorithm 1). Firstly, from all the dimensions of feature, this work selects the most discriminative one m_{opt} to construct the first stage of cascade $f_1(x) = \sum_{j \in D_1} w_j x(j)$. The rejection threshold of the first stage is defined as the minimum response of all the positive samples, i.e. $r_1 = \min_{x \in Pos} f_1(x)$. Compared with weak classifier using any other dimensions (except m_{opt}), $f_1(x)$ removes the most negatives, while lets all the positives pass to the next stage. Then this work selects the optimal dimension of remaining ones, just as the selection in the first stage. The second stage is obtained by adding the optimal one to the first stage. Finally, the entire soft cascade SVM is obtained by repeating the above process until all the dimensions of feature is selected. Note that the last stage is the original SVM classifier.

Compared with former cascade structure which imposes a severe requirement on training multiple individual classifiers, our method only trains one linear SVM model followed by a fast post-training. Therefore, soft cascade SVM spends less time than existing cascade classifier [2, 5] on training.

2.3 Cascade selective window search

Intuitively, detection speed can be further increased by introducing selective window search strategy into cascade. Based on this motivation, this paper proposes a cascade selective window search strategy which alternates between estimating object probability density function (PDF) using sampled windows' object possibility and drawing new windows from the object PDF. Within the proposed search strategy, a window is defined as a 2D vector $l = (l_x, l_y)$, being coordinates of the window center. l is also considered as a random vector, and its state space comprises all possible locations of image. Given a window l, we define an object possibility on the *i-th* stage of soft cascade SVM as:

$$L_i(l) = \begin{cases} 1 & l \text{ pass the i-th stage} \\ 0 & otherwise \end{cases} \quad (4)$$

The main process of the proposed search strategy is shown in Algorithm 2. In the *j-th* loop, candidate windows Q are obtained by combining sampled windows drawn from object PDF $q_{j-1}(l)$ and reserved ones S_{j-1} in the previous stage (step 1). Then the candidate windows which pass the stage c_j are reserved as S_j and used to approximate the observational density function $p_j(l \mid S_j)$ by Gaussian kernel density estimation (step 2). The new object PDF $q_j(l)$ is linearly combined with the uniform distribution to the observational density function $p_j(l \mid S_j)$ (step 3). Adding an uniform distribution on $p_j(l \mid S_j)$ enable the algorithm to still have opportunity to detect objects that are missed in the previous stage.

Algorithm 2. The process of cascade selective window search

Initialize:

Initialize object PDF $q_0(l)$ as uniform distribution

Initialize array $\{c_j\}_{j=1}^{T}$ and set $S_0 = \varnothing$

For $(j = 1,...,T)$

step 1 Obtain candidate set Q:

$Q = \{ N_j$ samples drawn from $q_{j-1}(l) \} \cup S_{j-1}$

step 2 Compute the observational density function:

$$p_j(l \mid S_j) = \frac{1}{num(S_j)} \sum_{l_i \in S_j} Norm(l_i, \Sigma)$$

$$S_j = \{ l_i \mid L_{c_j}(l_i) = 1,\ l_i \in Q \}$$

step 3 Compute the new object PDF:

$$q_j(l) = (1 - \alpha) U(l) + \alpha p_j(l \mid S_j)$$

End

step 4 For each window $l_i \in S_T$, compute the c_T*-th* stage classifier response, and retain only local maximum windows.

Output:

Obtain final object detection result by using the entire soft cascade SVM to classify the reserved samples.

The above process is iterated for T times ($T = 3$ in the experiment). The sampled windows that pass the stage c_T ($c_T = 60$ in the experiment) and have a locally maximum response in its neighborhood (5×5) are retained (step 4). Final detection result is obtained by judging whether the reserved windows can pass the entire soft cascade SVM. Note that multi-scale object detection can be achieved by employing cascade selective window search on each image scale.

3. Experimental Results

We apply the proposed approach to face detection and car detection. This section will show evaluation results on public datasets and the detection speed of the proposed approach. The accuracy of object detection is measured in terms of the PASCAL criterion [1]. The experiments are conducted on 2.2 GHz Intel Core 2 Duo processor Windows platform with 2GB of RAM. Note that the proposed approach is not limited to face detection and car detection. It can be applied to detect many other object categories without large deformation, such as pedestrian detection and palm detection.

3.1 Evaluation of detection accuracy

In face detection experiment, linear SVM is learned using L1-regularized L2-loss SVM tool [13]. The initial

training set consists of 8625 frontal upright faces rescaled to a resolution of 50×36, as well as 20000 non-face windows. New bootstrapped non-face windows are continually added during training. The training result is a linear SVM classifier consisting of 2479 features. Then a soft cascade SVM is learned as described in Section 2.2.

Fig. 3(a) and (b) depict the precision-recall curves for CSW and the comparison cascade-based methods on two idealized datasets (BioID and Caltech). The experimental results show that the three soft cascade methods achieve almost the same detection precision, and outperform the hard cascade Adaboost. To provide more practice testing, we select the ESOGU dataset, whose images contain faces appearing at a wide range of image positions and scales, and also complex backgrounds. Experimental result on ESOGU dataset is shown in Fig. 3(c). It can be seen that detection accuracy of CSW is the highest, followed by soft cascade SVM and soft cascade Adaboost, and that of hard cascade Adaboost is the worst. CSW achieves 93.5% detection precision at 95% recall rate, exceeding the other two soft cascade methods by about 1%. It can be concluded that: (1) Hard cascade classifier has the flaw that valuable information is discarded at each stage. Soft cascade classifier addresses the problem and obtains higher detection accuracy. (2) Compared with sliding window search, selective window search which captures less windows in non-object area can effectively suppress false positive, (3) Soft cascade SVM has comparable performance as soft cascade Adaboost.

Car detector is learned as well as face detection experiment does. The positive samples come from the MIT car datasets, and the total number of negative samples is about 80000. Moreover, we manually choose 500 testing images from the TME Motorway dataset, which is

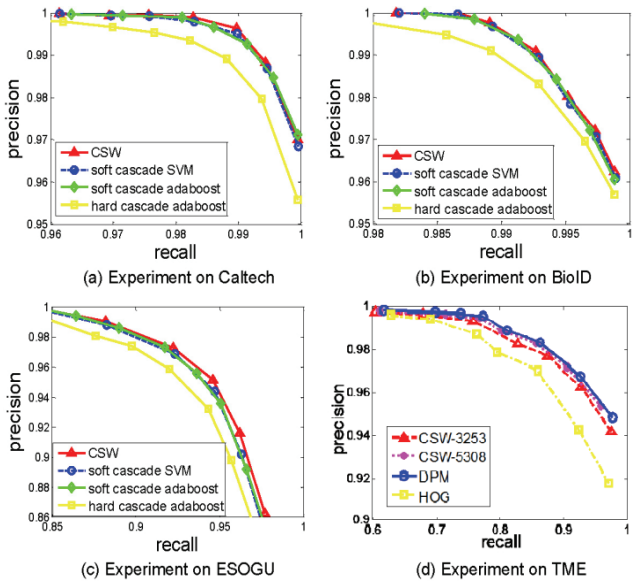

Fig. 3. Precision-Recall curves for CSW and several comparison detection methods on four object datasets.

Fig. 4. Detection results of CSW in case of occlusion, multi-object, rotation and varying illumination.

composed of 28 clips for a total of approximately 27 minutes with vehicle annotation. Fig. 3(d) shows the detection performance of CSW and two baseline methods. It can be seen that detection accuracy of CSW is higher than that of HOG method [14], and is a little bit lower than that of DPM [1]. Specifically, CSW (3253 features) achieves 96.5% precision at 92% recall rate, compared to a 94% precision for HOG and a 97% precision for DPM. When CSW method increases feature dimension (up to 5308 features), it can obtain almost the same detection accuracy as DPM. Fig. 4 shows some detection results of CSW. Obviously, our method can obtain satisfying detection results in case of occlusion, multi-object, rotation and varying illumination.

3.2 Running time

Table 1 summarizes the average running time of different methods for face detection (the resolution of test image is 640×480). SVM denotes the original SVM detector using compressive channel features. Experiments show that detection speed of soft cascade SVM is higher than hard cascade Adaboost and soft cascade Adaboost. This speedup is caused by the fact that soft cascade SVM employ fewer features (2479 features) than soft cascade Adaboost (5120 features) and hard cascade Adaboost (6061 features). Moreover, CSW further increase detection speed by introducing selective window search into cascade. Specifically, CSW only cost about 17ms to detect face in image with 640×480. What's more important, the proposed method not only achieves higher detection speed,

Table 1. The average running time of different methods for face detection

Method	Training Time	Detection Time
Hard cascade Adaboost [5]	>72 hours	65ms
Soft cascade Adaboost [2]	≈ 2 hours	38ms
SVM	≈ 10 minutes	>1s
Soft cascade SVM	≈ 20 minutes	25ms
CSW	≈ 20 minutes	17ms

Table 2. The car detection time of different methods

Method	Detection Time
CSW-3253	32ms
Soft cascade SVM-3253	53ms
DPM [1]	>1s
HOG [11]	>1s

but also costs much less time to learn classifier than [2, 5].

The car detection time of different methods is shown in Table 2. Note that CSW (3253 features) taking about 32ms to detect cars in one image with 1024×768 is the fastest method in the experiment. Soft cascade SVM is slightly slower than CSW. But it can still detect cars in real-time. HOG and DPM which spend more than 1s per image are far slower than ours. In sum, the proposed method is much more competitive because of its outstanding detection speed, although its detection accuracy is a little bit lower than that of DPM.

4. Conclusion

This paper proposes a cascade selective window method for fast object detection. The main advantages of CSW include: (1) The training complexity of cascade classifier is greatly reduced. (2) CSW significantly increases detection speed by combining well the strengths of cascade and selective window search strategy. Experimental results on face and car datasets show that the computational efficiency and detection precision of the proposed method is superior to the compared method.

Acknowledgements

This work was supported by the National Natural Science Foundation of China under Grant No.61271288 and No.61172117.

References

[1] Pedro Felzenszwalb, Ross Girshick, David McAllester and Deva Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Mach. Intel*, vol. 32, no. 9, pp. 1627-1645, 2010.

[2] Lubomir Bourdev and Jonathan Brandt, "Robust object detection via soft cascade," *Computer Vision and Pattern Recognition*, Colorado, America, 2005.

[3] Piotr Dollar, Christian Wojek, Bernt Schiele, and Pietro Perona, "Pedestrian detection: An evaluation of the state of the art," *IEEE Trans. Pattern Anal. Mach. Intel*, vol. 34, no 4, pp. 743-760, 2012.

[4] Nicholas Butko and Javier Movellan, "Optimal scanning for faster object detection," *Computer Vision and Pattern Recognition*, Miami, America, 2009.

[5] Paul Viola and Michael Jones, "Rapid object detection using a boosted cascade of simple features," *Computer Vision and Pattern Recognition*, Kauai Hawaii, 2001.

[6] Giovanni Gualdi, Andrea Prati, and Rita Cucchiara, "A multi-stage pedestrian detection using monolithic classifiers," *Advanced Video and Signal Based Surveillance*, Klagenfurt, Austria, 2011.

[7] Piotr Dollar, Serge Belongie and Pietro Perona, "The fastest pedestrian detector in the west," British Machine Vision Conference, Aberystwyth, UK, 2010.

[8] Christoph Lampert, Matthew Blaschko and Thomas Hofmann, "Efficient subwindow search: A branch and bound framework for object localization," *IEEE Trans. Pattern Anal. Mach. Intel*, vol.31, no.12, pp. 2129-2142, 2009.

[9] Marco Pedersoli, Jordi Gonzàlez, Andrew Bagdano and Juan Villanueva, "Recursive coarse-to-fine localization for fast object detection," *European Conference on Computer Vision*, Heraklion, Crete, Greece , 2010.

[10] Wei Zhang, Gregory Zelinsky and Dimitris Samaras, "Real-time accurate object detection using multiple resolutions," *International Conference on Computer Vision*, Rio de Janeiro, Brazil, 2007.

[11] Richard Baraniuk, Mark Davenport, Ronald DeVore and Michael Wakin, "A simple proof of the restricted isometry property for random matrices," *Constructive Approximation*, vol.28, no.3, pp.253-263, 2008.

[12] Ping Li, Trevor Hastie and Kenneth Church, "Very sparse random projections," *Knowledge Discovery and Data Mining*, New York, USA, 2006.

[13] Rong Fan, Kai Chang, Cho Hsieh, Xiang Wang and Chih Lin, "Liblinear: A library for large linear classification," *Journal of Machine Learning Research*, vol. 9, pp. 1871-1874, 2008.

[14] Navneet Dalal and Bill Triggs, "Histograms of oriented gradients for human detection," *Computer Vision and Pattern Recognition*, SanDiego, USA, 2005.

Shu Zhang He received the MS degree in circuits and systems from JiLin University in 2009. Now he is a PHD candidate in University of Electronic Science and Technology of China. His research interests are in computer vision and especially in object detection, image segmentation, and pattern recognition.

Yong Cai He received the MS degree in signal processing from Xi'an Jiaotong University in 2003. Now he is a PHD candidate in University of Electronic Science and Technology of China. His research interests are in computer vision and especially in pattern recognition and behavior recognition.

Mei Xie She received the MS degree and PhD degree from the University of Electronic Science and Technology of China in 1992 and 1996. She was a postdoctoral research assistant at University of Hong Kong and University of Texas between the years of 1997-1999. Now she is professor in School of Electronic and Engineering, University of Electronic Science and Technology of China. Her researches are concerned with image processing, object recognition and Information system security.