LI-008

Feature Selection By AdaBoost For SVM-Based Face Detection

Duy Dinh LE [†]　　Shin'ichi SATOH [† ‡]

Abstract

In this paper, we present a three-stage method to speed up a SVM-based face detection system. In this proposed system, a large number of simple non-face patterns are rejected quickly by two first stage cascaded classifiers using flexible sizes of analyzed windows while the last stage uses a non linear SVM classifier to robustly classify complex 24x24 pixel patterns as either faces or non-faces. For all stage classifiers, an optimal subset of over-complete Haar wavelet feature set selected by AdaBoost learning is used to achieve both fast and high detection rate. Experimental results show that our system can achieve comparable results to state of the art face detection systems.

1. Introduction

Face detection is one of the most active research areas in computer science because of many interesting applications in fields such as security, multimedia retrieval, human computer interaction,... A fast and robust face system detection will be necessary and useful for such video applications as [8].

In a face detection system, the major factor affecting to computation time is the number of analyzed windows. For example, if using a 24x24 pixel window to scan at every location and scale with the scale factor of 1.25 over a 320x240 pixel image, the number of windows processing is about 158,685 windows while the number of windows containing face-like patterns is very small. Using any robust classifiers like Support Vector Machines (SVMs) ([4]) or Neural Networks ([7]) can lead useless consumption time for easily distinguish background patterns.

To deal with this problem, a combination of simple-to-complex classifiers is proposed ([6, 9, 4]) in which fast and simple classifiers are used as filters at earliest stages to reject quickly almost background patterns and a slower yet more accurate classifier is used for final classification. There are two methods for this complexity reduction. The first is only the reduction complexity of classifiers. For example, in [6], the complexity of non linear SVM classifiers in early stages is reduced by the number of support vectors, or in [4], linear SVM classifiers and feature reduction are used. The other is both reductions in feature vector size and complexity of classifiers. For example, in [9], almost background patterns are rejected only by average of 10-feature evaluation and complexity of classifiers is measured by the number of combined weak classifiers in the strong classifier learned by AdaBoost.

Our system also takes the same structure of simple-to-complex classifiers approach to take advantages for fast and robust performance. However, it is distinguished from previous systems by two things:

- The first is that a new layer is added to estimate face candidate regions by using a larger window size and step size. Specifically, we use 36x36 pixel window based classifier with the step size of 12 pixels to estimate candidate face regions. The idea of using larger window and step size was used in [7] but here we take the advantages of combination of Haar wavelet features and AdaBoost learning for fast evaluation to improve the speed.
- Recall that training the full efficient face detection system by AdaBoost as in [9] can take several weeks because in later stages when negative examples (false positives of previous layer classifiers) are too complex and similar to face samples, convergence speed will be very slow. Therefore, we replace later stages in the cascade classifier by a non linear SVM classifier learned on Haar wavelet feature sub space to reduce both the training and evaluation time while maintaining the comparable performance.

This work aim is to keep a balance between the rapid and efficient background rejection method and the robust classification performance of support vector machine classifiers. Experimental results show that significant computation time is devoted to hard classified patterns.

The outline of the paper is as follows: In section 2, we describe our face detection system. Haar wavelet features used with AdaBoost learning to build a classifier and Support Vector Machines are presented in section 3 and section 4, respectively. Section 5 contains experimental results and section 6 concludes the paper.

2. Description of The Proposed Face Detection System

The proposed face detection system consists of three stages that classify a 24x24 pixel window as either a face or non-face. To detect faces of different sizes and locations, we apply the detector at every location and scale in the input image with scale factor of 1.25. An outline of the system is showed in figure 1.

The first stage of the system is a flexible classifier used to estimate face candidate positions in a 36x36 pixel input window. This classifier is expected to be invariant to translations of original 24x24 pixel face window. Figure 2 shows some face examples used to train this stage classifier. If a 36x36 window is detected as existence of some face, 12x12=144 likely face positions are collected and passed to the next stage.

The second stage is a 24x24 window based classifier that is used to explore face candidate locations returned from the previous stage and try to filter as many

[†]The Graduate University for Advanced Studies
[‡]National Institute of Informatics

non-face patterns as possible before passing hard patterns to the final stage classifier.

These two first stages are cascaded classifiers similar to the proposed system in [9]. Specifically, it consists of an optimal subset of the over-complete Haar wavelet feature set which is selected to build each layer classifier by AdaBoost and combination of layer classifiers into a cascade structure of classifiers. By using the cascade structure, the complexity of classifiers is adapted corresponding to the difficulty of input patterns. With step size of 12 and fast estimation of cascaded classifiers, the speed of rejection will be increased significantly.

The last stage classifier is a non linear SVM classifier. The features that have been selected by AdaBoost learning in the second stage classifier are reused as pattern representation. Explicitly, for each input 24x24 window, these feature values are evaluated and normalized to be between 0 and 1 to form a feature vector for the RBF kernel based SVM classifier. In our experiments, only 200 features are used and hence it is faster than using pixel-based representation.

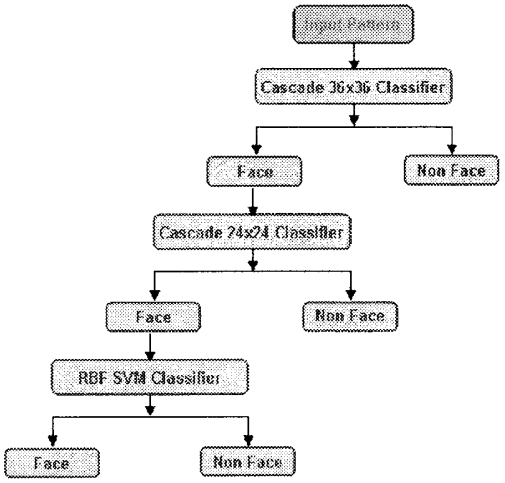

Figure 1: Three-stage system for face detection

Figure 2: Face patterns are used for training the 36x36 window based classifier

3. Haar Wavelet Features and AdaBoost Learning

3.1 Haar Wavelet Features

In [5], Haar wavelets are used as a kind of feature representation for object detection. Features are responses of wavelet filters that are applied to each input window at each scale and location. Due to the large number of wavelet coefficients (1,734 coefficients for the face class), an optimal subset of 37 coefficients is selected manually for computational reduction when working with SVM classifiers. In [9], similar yet more complex Haar wavelet features selected by a variant AdaBoost learning method are also used very efficient in quick rejecting background patterns.

In our system, we use the same feature set proposed in [9]. Specifically, they consist of three kinds of features modelled from adjacent rectangles with the same size and shape. The feature value is defined as the difference of sum of the pixels within rectangles.

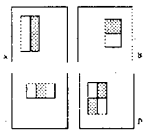

Figure 3: Kinds of rectangle features

By using integral image definition, these feature rectangle values can be computed very fast. The integral image at location (x, y) is defined as $ii(x, y) = \sum_{x'<=x, y'<=y} i(x', y')$ where $ii(x, y)$ is the integral image and $i(x, y)$ is the original image. In practice, $ii(x, y)$ can be computed simply by using the following recurrent function: $ii(x, y) = ii(x, y-1) + ii(x-1, y) + i(x, y) - ii(x-1, y-1)$ and sum the pixels within a rectangle can be computed from four integral image values of its vertices, for example, $Sum(D) = 1 + 4 - (2 + 3)$.

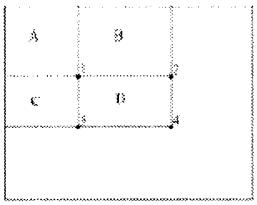

Figure 4: Evaluation of the sum of the pixel within a rectangle

3.2 AdaBoost Learning

The aim of boosting is to improve the classification performance of any given simple learning algorithm [3]. Given T weak classifiers $h_t(x)$ learned through T round of boosting, the strong classifier is formed by a linear combination: $H(x) = \sum_{t=1}^{T} \alpha_t h_t(x)$ where α_i are coefficients found in the boosting process.

In [9], each weak classifier h_j is associated with a feature f_j and a threshold θ_j such that the number of incorrect classified examples corresponding to this weak classifier is minimized:

$$h_j(x) = \begin{cases} 1 & if\ p_j f_j(x) < p_j \theta_j \\ 0 & \text{otherwise} \end{cases}$$

where polarity p_j indicates the direction of the inequality sign. Each round of boosting, the best weak classifier h_t with the lowest error ϵ_t is chosen. The error of each weak classifier is measured with respect to the set of weights over each examples of the training set $\epsilon_j = \sum_{i=1}^{N} |h_j(x_i) - y_i|$ where w_i and y_i are the weight and the label of the training example x_i respectively.

After each round, these weights are updated such that the weak learner will focus much more on the hard examples in the next round.

3.3 Cascade of classifiers

The main idea of building a cascade of classifiers is to reduce computation time by giving different treatments to different kinds of input windows depending on their complexity as showed in figure 5. Only input windows that have passed through all layers of the cascade are classified as faces. With this flexible structure, easily distinguish non-face patterns like homogeneous texture ones can be simply rejected by simple one-feature classifier as in figure 6.

Training cascaded classifiers that can achieve both good detection rate and less computation time is quite complex because higher detection rate requires more features but more features are correspondent to more time to evaluate. To simplify, the detection rate goal and the false positive rate goal for each layer are set beforehand.

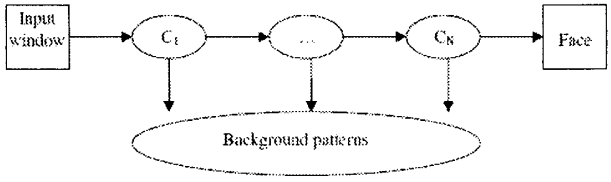

Figure 5: A cascade structure of simple-to-complex classifiers

Figure 6: A simple feature is used to reject simple background patterns

4. Support Vector Machines

Support Vector Machines (SVMs) is a statistical learning method based on the Structure Risk Minimization principle that has been showed very efficient in pattern recognition applications ([1]). In the binary classification case, the objective of SVMs is to find a best separating hyperplane with maximum margin. The form of SVM classifiers is:

$y = sign(\sum_{i=1}^{N} y_i \alpha_i K(x, x_i) + b)$

where: x is the d-dimensional vector of an observation example, $y \in \{-1, +1\}$ is a class label, x_i is the vector of the i^{th} training example. N is the number of training examples, $K(x, x_i)$ is a kernel function. $\alpha = \{\alpha_1, \alpha_2, ..., \alpha_n\}$ is learned by solving the following quadratic programming problem: $minQ(\alpha) = -\sum_{i=1}^{N} \alpha_i + \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j K(x_i, x_j)$ subject to $\sum_{i=1}^{N} \alpha_i y_i = 0$ and $0 \leq \alpha_i \leq C, \forall i$. C is a predefined parameter which is a trade off between wide margin and a small number of margin failures. All the x_i corresponding to non zero α_i are called support vectors.

5. Experiments

5.1 Training Data Set

For training, we collected 5,000 face patterns with the size of 24x24 on the Internet like [9]. Non-face patterns are generated randomly at different locations and scales from more than 700 images containing non faces collected with various subjects such as rocks, trees, buildings, scenery, flowers,... Totally, about 500,000 non-face patterns are used for training our system. To train 36x36 pixel window based classifier, 8,000 face patterns are generated from 24x24 pixel face patterns above by translating randomly a 24x24 pixel window within the 36x36 pixel window. Some samples are showed in figure 2.

In training 24x24 cascade classifier, the same 5,000 face patterns are used for all layers while non-face patterns of subsequent layer classifiers are false positives collected by the partial cascade on the non face images. Non face patterns to train the first layer classifier in one cascade are selected randomly. For each layer classifier, a total 5,000 non face patterns are used for training. In these two first stages, we fixed the detection rate to 99.9% for every layer while the number of layers and the number of features in each layer are controlled to trade off between speed and detection performance through empirical experiments. Note that these two first stages work as filters in the overall system, so the training is more flexible and easier than that of [9].

5.2 Structure of the system

The first stage cascaded classifier consists of 3 layers which the number of features of each layer is 10, 20, 30, respectively. The second stage cascaded classifier consists of 13 layers which the number of features of first 5 layers is 5, 30, 50, 95, 125. Totally, 2475 features are used for the second stage. Compared with 6061 features used in [9], our system can save a half training time. The final stage SVM classifier takes 200 features of the last layer in the second stage as a feature vector and is trained by LibSVM [2] with RBF kernel where $C = 8, \gamma = 0.015625$. The number of layers used in first two stages is found by emperical experiments that are a trade-off between speed and performance.

5.3 Results

We tested our system on the MIT+CMU frontal face standard test set [7]. This test set consists of 130 images with 507 frontal faces. As for performance showed in table 1, two first stage cascaded classifiers rejects almost 99.88% non face patterns in which the first stage classifier contributes average up to 58.73%. Only 0.12% hard patterns are put into the final stage SVM classifier to make to final decisions. Time for background rejection is about 38% total time of detection while time of classification using SVM is about 62%.

Figure 7 shows our result a little worse than that of other state of the art face detection systems. Figure 8 shows some detection results.

6. Conclusions

We have presented a method to speed up a SVM-based face detection system while maintaining the com-

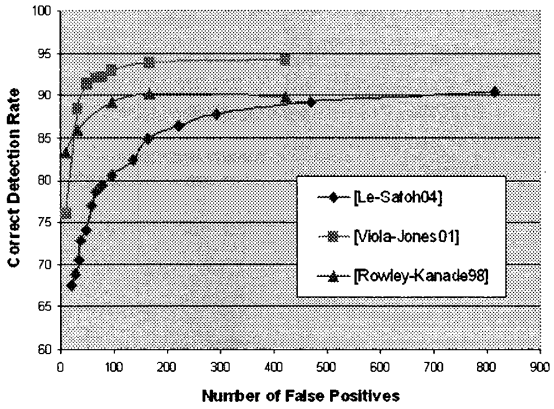

Figure 7: ROC curves of face detection systems

Stages	Rejection rate	Time rate
Stage 1 - AdaBoost	58.73%	0.38%
Stage 2 - AdaBoost	99.72% (99.88%)	37.61%
Stage 3 - SVM	77.23% (99.97)%	62.01%

Table 1: Performance of each stage

parable detection rate. The improvements are employed through two first stage classifiers which quickly filter almost background patterns and most of computation time is devoted for potential face regions. Discriminant Haar wavelet features selected from AdaBoost learning are used for all stage classifier to take advantages from their efficient representation and fast evaluation. Cascade structure of classifiers in two first stage classifiers allows adapting best to various complexities of input patterns. The non linear SVM classifier at the final stage is robust enough to achieve good results.

7. Acknowledgements

The authors would like to thank F. Yamagishi for his helpful technical supports.

References

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

[2] Chih-Chung Chang and Chih-Jen Lin, LIBSVM: A library for support vector machines, 2001. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[3] Yoav Freund and Robert E. Schapire, *A short introduction to boosting*, Journal of Japanese Society for Artificial Intelligence, 14(5):771-780, September, 1999

[4] B. Heisele, T. Serre, S. Prentice, and T. Poggio, *Hierarchical Classification and Feature Reduction for Fast Face Detection with Support Vector Machines.* Pattern Recognition, Vol. 36, No. 9, pp. 2007-2017, 2003.

[5] C. Papageorgiou, *A Trainable System for Object Detection in Images and Video Sequences*, Ph.D. Thesis, EECS, MIT, December 1999

[6] Sami Romdhani, Philip H. S. Torr, Bernhard Schlkopf, Andrew Blake, *Computationally Efficient Face Detection.* In Proc. Intl. Conf. on Computer Vision, pp. 695-700, 2001

[7] H. Rowley, S. Baluja, and T. Kanade, *Neural Network-Based Face Detection*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 20, No. 1, pp. 23-38, January, 1998.

[8] Shin'ichi Satoh, Takeo Kanade, *Name-It: Association of Face and Name in Video.* In Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp 368-373, 1997.

[9] Paul Viola and Michael Jones, *Rapid Object Detection using a Boosted Cascade of Simple Features*, In. Proc. Conference on Computer Vision and Pattern Recognition (CVPR01), pp. 511-518 , 2001.

Figure 8: Detection results of our system on test images of the MIT+CMU test set