Rapid Object Detection of Boosted Classifier for Analysis of Detection Cascades

Shrikant Nirmalkar ,Bhupendra puri Goswami, Prof. Rekhram Janghel

Deptt. Of Computer Science & Engineering, DIMAT, Raipur,
Chhattisgarh, Raipur - India

shrikant.nirmalkar@gmail.com, bhupendra1goswami@yahoo.co.in, rrj.iiit@gmail.com

***Abstract*-- In this paper we analysis the two approach: the first approach is a novel set of rotated haar-like features is introduced. These novel features significantly enrich the simple features and can also be calculated efficiently. With these new rotated features our sample face detector shows off on average a 10% lower false alarm rate at a given hit rate. The second approach is a through analysis of different boosting algorithms (namely Discrete, Real and Gentle Adaboost) and weak classifiers on the detection performance and computational complexity.**

I. INTRODUCTION

In this paper we extends object detection framework in two important approach: Firstly, the basic and over-complete set of haar-like features is extended by an efficient set of 45° rotated features, which add additional domain-knowledge to the learning framework and which is otherwise hard to learn. These novel features can be computed rapidly at all scales in constant time. Secondly, we empirically show that Gentle Adaboost outperforms with respect to object detection accuracy and computational complexity Discrete and Real Adaboost.

II. FEATURES

Let us assume that the basic unit for testing for the presence of an object is a window of pixels. A rectangle is specified by the tuple $r = (x, y, w, h, \alpha)$ with $0 \leq x, \; x + w \leq W$ $0 \leq y, \; y + h \leq H$ $x, y \geq 0$ $w, h > 0$ $\alpha \in \{0^\circ, 45^\circ\}$, and its pixel sum is denoted by . Two examples of such rectangles are given in

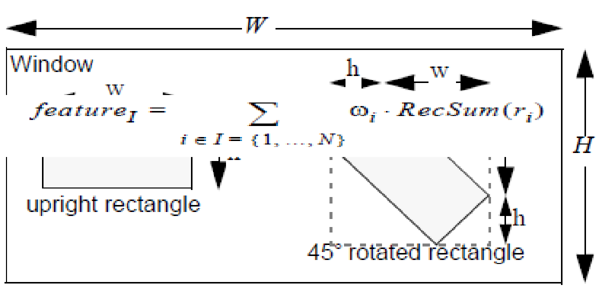


Figure 1. Our raw feature set is then the set of all possible features of the form

where the weights , the rectangles , and N are arbitrarily chosen. This raw feature set is (almost) infinitely large. For practical reasons, it is reduced as follows:

Fig 1: Example of an upright and 45° rotated rectangle.

1. Only weighted combinations of pixel sums of two rectangles are considered.
2. The weights have opposite signs, and are used to compensate for the difference in area size between the two rectangles.
3. The features mimic haar -like features and early features of the human visual pathway such as center-surround and directional responses.

These restrictions lead us to the 14 feature prototypes shown in Figure 2: Four edge features, eight line features, and two center-surround features, and a special diagonal line feature. These prototypes are scaled independently in vertical and horizontal direction in order to generate a rich, over-complete set of features. Note that the line features can be calculated by two rectangles only. Hereto it is assumed that the first rectangle encompasses the black and white rectangle and the second rectangle represents the black area.

In our experiments the additional features significantly enhanced the expressional power of the learning system and consequently improved the performance of the object detection system. This is especially true if the object under detection exhibit diagonal structures such as it is the case for many brand logos.

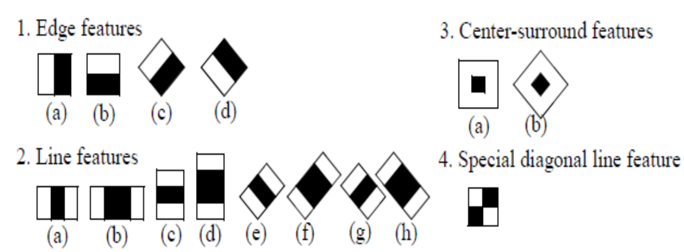


Fig 2: Feature prototypes of simple haar-like and center-surround features. Black areas have negative and white areas positive weights.

A. Number of Features.

The number of features derived from each prototype is quite large and differs from prototype to prototype and can be calculated as follows. Let and be the maximum scaling factors in x and y direction. An upright feature of size wxh then generates $XY(W+1-w(X+1)/2)(H+2-h(Y+1)/2)$ features for an image of size WxH, while a rotated feature generates $XY(W+1-z(X+1)/2)(H+1- z(Y+1)/2)$ with $z=w+h$. The number of features for a window size of 24x24 totals to 117,941.

B. Fast Feature Computation.

All features can be computed very fast in constant time for any size by means of two auxiliary images. For upright rectangles the auxiliary image is the *Summed Area Table* $SAT(x, y)$ $SAT(x, y)$ is defined as the sum of the pixels of the upright rectangle ranging from the top left corner at (0,0) to the bottom right corner at (x,y)

$$SAT(x,y) = \sum_{x' \le x,\, y' \le y} I(x',y') .$$

It can be calculated with one pass over all pixels from left to right and top to bottom by means of $SAT(x, y) = SAT(x, y-1) + SAT(x-1, y) + I(x, y) - SAT(x-1, y-1)$ with $SAT(-1, y) = SAT(x, -1) = SAT(-1, -1) = 0$. From this the pixel sum of any upright rectangle $r = (x, y, w, h, 0)$ can be determined

$$\begin{aligned} RecSum(r) &= SAT(x-1, y-1) + SAT(x+w-1, y+h-1) \\ &- SAT(x-1, y+h-1) - SAT(x+w-1, y-1) \end{aligned}$$

For 45° rotated rectangles the auxiliary image is the *Rotated Summed Area Table* $RSAT(x, y)$. It is defined as the sum of the pixels of a 45° rotated rectangle with the bottom most corner at (x,y) and extending upwards till the boundaries of the image

$$RSAT(x,y) = \sum_{y' \le y,\, y' \le y - |x - x'|} I(x',y') .$$

It can be calculated also in one pass from left to right and top to bottom over all pixels by

$$RSAT(x, y) = RSAT(x-1, y-1) + RSAT(x+1, y-1) (-RSATx, y-2) + I(x, y) + I(x, y-1)$$

With $RSAT(-1,y)=RSAT(x,-1)=RSAT(x,-2)=RSAT(-1,-1)=RSAT(-1,-2)=0$. From this the pixel sum of any rotated rectangle $r = (x, y, w, h, 45^\circ)$ can be determined by 4 table lookups:

$$\begin{aligned} RecSum(r) &= RSAT(x-h+w, y+w+h-1) + RSAT(x, y-1) \\ &- RSAT(x-h, y+h-1) - RSAT(x+w, y+w-1) \end{aligned}$$

C. Fast Lighting Correction.

The special properties of the haar-like features also enable fast contrast stretching of the form $I(x, y) = (I(x, y) - \mu) / (c\sigma)$ $c \in R_+$. μ can easily be determined by means of *SAT*(x,y). Computing , however, involves the sum of squared pixels. It can easily be derived by calculating a second set of *SAT* and *RSAT* auxiliary images for $I^2(x, y)$ .Then, calculating σ for any window requires only 4 additional table lookups.

III. (STAGE) CLASSIFIER

We use boosting as our basic classifier. Boosting is a powerful learning concept. It combines the performance of many "weak" classifiers to produce a powerful 'committee'. A weak classifier is only required to be better than chance, and thus can be very simple and computationally inexpensive. Many of them smartly combined, however, result in a strong classifier, which often outperforms most 'monolithic' strong classifiers 'such as SVMs and Neural Networks. Different variants of boosting are known such as Discrete Adaboost, Real AdaBoost, and Gentle AdaBoost . All of them are identical with respect to computational complexity from a classification perspective, but differ in their learning algorithm. All three are investigated in our experimental results

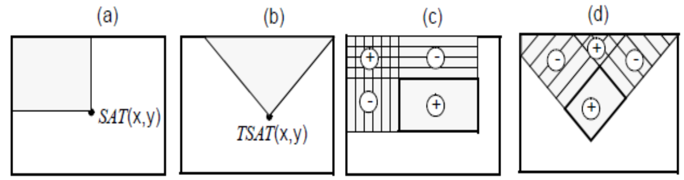


Fig3: (a) *Summed Area Table* (*SAT*) and (b) *Rotated Summed Area Table* (*RSAT*). Calculation scheme of the pixel sum of upright (c) and rotated (d) rectangles.

Learning is based on N training examples $(x_1, y_1), \cdots, (x_N, y_N)$ with $x \in \mathbb{R}^k$ and $y_i \in \{-1, 1\}$. x_i is a K-component vector. Each component encodes a feature relevant for the learning task at hand. The desired two-class output is encoded as –1 and +1. In the case of object detection, the input component x_i is one haar-like feature. An output of +1 and -1 indicates whether the input pattern does contain a complete instance of the object class of interest.

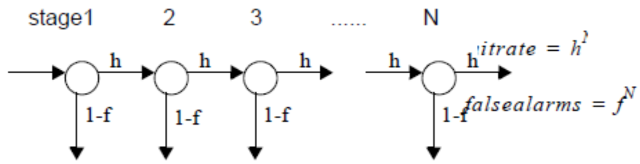
A. Cascade of Classifiers

A cascade of classifiers is a degenerated decision tree where at each stage a classifier is trained to detect almost all objects of interest (frontal faces in our example) while rejecting a certain fraction of the non-object patterns . For instance, in our case each stage was trained to eliminated 50% of the non-face patterns while falsely eliminating only 0.1% of the frontal face patterns; 20 stages were trained. Assuming that our test set is representative for the learning task, we can expect a false alarm rate about $0.5^{20} \approx 9.6e-07$ and a hit rate about . Each stage was trained using one out of the three Boosting variants. Boosting can learn a strong classifier based on a (large) set of weak classifiers by re-weighting the training samples. Weak classifiers are only required to be slightly better than chance. Our set of weak classifiers are all classifiers which use one feature from our feature pool in combination with a simple binary thresholding decision or which are small CART trees with up to 4 features. At each round of boosting, the feature-based classifier is added that best classifies the weighted training samples. With increasing stage number the number of weak classifiers, which are needed to achieve the desired false alarm rate at the given hit rate, increases.

V. EXPERIMENTAL RESULTS

All experiments were performed on the complete CMU Frontal Face Test Set of 130 grayscale pictures with 510 frontal faces . A hit was declared if and only if

- the Euclidian distance between the center of a detected and actual face was less than 30% of the width of the actual face as well as
- the width (i.e., size) of the detected face was within ±50% of the actual face width.



Input pattern classified as a non –object

Fig4: Casecade of classifier with N stages. At each stage a classifier is trained to achieve a hit rate of h and a false alarm rate of f.

Every detected face, which was not a hit, was counted as a false alarm. Hit rates are reported in percent, while the false alarms are specified by their absolute numbers in order to make the results comparable with related work on the CMU Frontal Face Test set. Except otherwise noted 5000 positive frontal face patterns and 3000 negative patterns filtered by stage 0 to *n*-1 were used to train stage *n* of the cascade classifier. The 5000 positive frontal face patterns were derived from 1000 original face patterns by random rotation about ±10 degree, random scaling about ±10%, random mirroring and random shifting up to ±1 pixel. Each stage was trained to reject about half of the negative patterns, while correctly accepting 99.9% of the face patterns. A fully trained cascade consisted of 20 stages.

During detection, a sliding window was moved pixel by pixel over the picture at each scale. Starting with the original scale, the features were enlarged by 10% and 20%, respectively (i.e., representing a rescale factor of 1.1 and 1.2, respectively) until exceeding the size of the picture in at least one dimension. Often multiple faces are detect at near by location and scale at an actual face location. Therefore, multiple nearby detection results were merged. Receiver Operating Curves (ROCs) were constructed by varying the required number of detected faces per actual face before merging into a single detection result.

A. Feature Scaling.

Any multi-scale image search requires either rescaling of the picture or the features. One of the advantage of the Haar-like features is that they can easily be rescaled. Independent of the scale each feature requires only a fixed number of look-ups in the sum and squared sum auxiliary images. These look-ups are performed relative to the top left corner and must be at integral positions. Obviously, by fractional rescaling the new correct positions become fractional. A plain vanilla solution is to round all relative look-up positions to the nearest integer position. However, performance may degrade significantly, since the ratio between the two areas of a feature may have changed significantly compared to the area ratio at training due to rounding. One solution is to correct the weights of the different rectangle sums so that the original area ratio between them for a given haar-like feature is the same as it was at the original size.

B .Comparison Between Different Boosting Algorithms.

We compared three different boosting algorithms: Discrete Adaboost, Real Adaboost, and Gentle Adaboost. Three 20-stage cascade classifiers were trained with the respective boosting algorithm using the basic feature set (i.e., features 1a, 1b, 2a, 2c, and 4a of Figure 2) and stumps as the weak classifiers. As can be seen from Figure 5, Gentle Adaboost outperformed the other two boosting algorithm, despite the fact that it needed on average fewer features. For instance, at a an absolute false alarm rate of 10 on the CMU test set, RAB detected only 75.4% and DAB only 79.5% of all frontal faces, while GAB achieved 82.7% at a rescale factor of 1.1. Also, the smaller rescaling factor of 1.1 was very beneficial if a very low false alarm rate at high detection performance had to be achieved. At 10 false alarms on the CMU test set, GAB improved from 68.8% detection rate with rescaling factor of 1.2 to 82.7% at a rescaling factor of 1.1. Table 1 shows in the second column (nsplit =1) the average number of features needed to be evaluated for

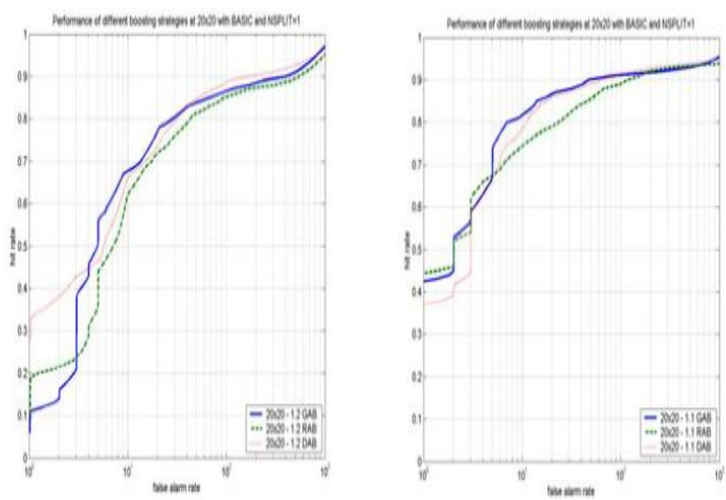


Figure 5: Performance comparison between identically trained cascades with 3 different boosting algorithms using the basic feature set and stumps as weak classifiers.

background patterns by the different classifiers. As can be seen GAB is not only the best, but also the fastest classifier. Therefore, we only investigate a rescale scaling factor 1.1 and GAB in the subsequent experiments

Table 1: Avg. # of features evaluated per background pattern at a pattern size of 20x20

NSPLIT	1	2	3	4
DAB	45.09	44.43	31.86	44.86
GAB	30.99	36.03	28.58	35.40
RAB	26.28	33.16	26.73	35.71

.

C. Input Pattern Size

Many different input pattern sizes have been reported in related work on face detection ranging from 16x16 up to 32x32. However, none of them have systematically investigated the effect of the input pattern size on detection performance. As our experiments show for faces an input pattern size of 20x20 achieves the highest hit rate at an absolute false alarms between 5 and 100 on the CMU Frontal Face Test Set . Only for less than 5 false alarms, an input pattern size of 24x24 worked better.

D. Basic vs. Extended Haar-like Features.

Two face detection systems were trained: One with the basic and one with the extended haar-like feature set. On average the false alarm rate was about 10% lower for the extended haar-like feature set at comparable hit rates. . At the same time the computational complexity was comparable. The average number of features evaluation per patch was about 31. These results suggest that although the larger haar-like feature set usually complicates learning, it was more than paid of by the added domain knowledge. In principle, the center surround feature would have been sufficient to approximate all other features, however, it is in general hard for any machine learning algorithm to learn joint behavior in a reliable way.

CONCLUSION

Our experimental results suggest that 20x20 is the optimal input pattern size for frontal face detection. In addition, they show that Gentle Adaboost outperforms Discrete and Real Adaboost. Logitboot could not be used due to convergence problem on later stages in the cascade training. It is also beneficial not just to use the simplest of all tree classifiers, i.e., stumps, as the basis for the weak classifiers, but representationally more powerful classifiers such as small CART trees, which can model second and/or third order dependencies. We also introduced an extended set of haar-like features. Although frontal faces exhibit little diagonal structures, the 45 degree rotated features increased the accuracy. In practice, the have observed that the rotated features can boost detection performance if the object under detection exhibit some diagonal structures such as many brand logos.

REFERENCES

[1] Stan Z. Li, Long Zhu, ZhenQiu Zhang, Andrew Blake, HongJiang Zhang, and Harry Shum. Statistical Learning of Multi-View Face Detection. In Proceedings of The 7th European Conference on Computer Vision. Copenhagen, Denmark. May, 2002.

[2] A. Mohan, C. Papageorgiou, T. Poggio. Example-based object detection in images by components. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 23, No. 4, pp.349 -361, April 2001.

[3] Y. Freund and R. E. Schapire. Experiments with a new boosting algorithm. In Machine Learning: Proceedings of the Thirteenth International Conference, Morgan Kauman, San Francisco, pp. 148-156, 1996.

[4] Paul Viola and Michael J. Jones. Rapid Object Detection using a Boosted Cascade of Simple Features. IEEE CVPR, 2001.

[5] H. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. In IEEE Patt. Anal. Mach. Intell., Vol. 20, pp. 22-38, 1998.

[6] Rainer Lienhart and Jochen Maydt. An Extended Set of Haar-like Features for Rapid Object Detection. IEEE ICIP 2002, Vol. 1, pp. 900-903, Sep. 2002.