Face Detection Using Look-Up Table Based Gentle AdaBoost

Cem Demirkır and Bülent Sankur

Boğaziçi University, Electrical-Electronic Engineering Department,
80815 Bebek, İstanbul
{cemd,sankur}@boun.edu.tr
http://busim.ee.boun.edu.tr

Abstract. In this work, we propose a face detection method based on the Gentle AdaBoost algorithm which is used for construction of binary tree structured strong classifiers. Gentle AdaBoost algorithm update values are constructed by using the difference of the conditional class probabilities for the given value of Haar features proposed by [1]. By using this approach, a classifier which can model image classes that have high degree of in-class variations can be constructed and the number of required Haar features can be reduced.

1 Introduction

Classical object detection schemes necessitate complex and computationally heavy classifiers for face detection in gray level images. Since the face detector is applied at each location and scale it requires significant computational power. However most of the locations in the searched scene do not contain any face and in fact the odds of finding a face at any one location is very small. Thus most of the non-face blocks can be eliminated with very simple classifiers. To find an efficient alternative to this exhaustive search approach, a rejection based classification approach is used to eliminate rapidly non-face regions in an image. Based on such a rejection strategy, Viola and Jones [1] used cascaded non-face rejection stages with low face elimination rate. Their algorithm consisting of simple classifiers are based on easily computable Haar features yield good detection with low false alarm rate. The required number of Haar features using their approach depends on the target false alarm rate. Liehart [6] made performance comparison of boosting algorithms using haar feature based binary-output weak classifiers.

Various extensions of this detector structure have been proposed in [2], [3] and [4]. For example Wu [4] has proposed a multi-view face detector using Real AdaBoost confidence-rated Look-Up-Table (LUT) classifiers to detect faces under rotation and pose variations. Our work is in the same line as the Viola-Jones scheme. The contribution consists in the use of a simple real valued Gentle AdaBoost (GAB) algorithm procedure to construct cascaded classifier structure. The GAB approach helps to reduce the required number of Haar features vis-a-vis the boosting approach that instead uses binary output weak classifiers. Using

T. Kanade, A. Jain, and N.K. Ratha (Eds.): AVBPA 2005, LNCS 3546, pp. 339–345, 2005.

LUT based confidence values for each Haar feature the information wasted in the binary weak classifier DAB approach can be utilized in the cascaded classifier training. We also used classifier output propagation as proposed in [4] to further reduce the number of features.

The rest of the paper is organized as follows: in Section 2 we define the GAB procedure. In Section 3, we define the GAB based Haar feature selection and strong classifier construction. In Section 4, we show we can the use of previous classifier output in the following classifier construction. In Section 5 we give experiments and results, and finally in Section 6 the conclusions.

2 GAB Algorithm

Boosting is a classification methodology which applies sequentially reweighted versions of the input data to a classifier algorithm, and taking a weighted majority vote of sequence classifiers thereby produced. At each application of the classification algorithm to the reweighted input data, classification algorithm finds an additional classifier $f_m(x)$ at stage m. GAB algorithm is a modified version of the Real AdaBoost (RAB) algorithm and it is defined in Figure 1. The main difference between GAB and RAB is the way it uses the estimates of the weighted class probabilities to update the weak classifier functions, $f_m(x)$.In GAB the update is given by $f_m(x) = P_w(y = 1|x) - P_w(y = -1|x)$, while in the RAB algorithm is given by half the log-ratio $f_m(x) = \frac{1}{2} log \frac{P_w(y=1|x)}{P_w(y=-1|x)}$ [5]. Log-ratios can be numerically unstable, leading to very large update values, while the update in GAB lies in the range [-1,1]. This more conservative algorithm has classification performance similar to RAB algorithm, and outperforms it both especially when stability problems arise.

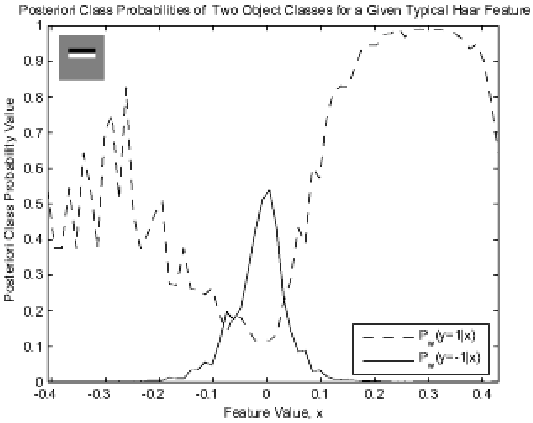
GAB Algorithm

1. Start with weights $w_i = 1/N, i = 1, ..., N, F(x) = 0$.
2. Repeat for $m = 1, 2, \ldots, M$.
   (a) Estimate $f_m(x)$ by weighted least squares fitting of y to x.
   (b) Update $F(x) \leftarrow F(x) + f_m(x)$
   (c) Set $w_i \leftarrow w_i \exp[-y_i.f_m(x_i)], i = 1, 2, \ldots, N$, and renormalize so that $\sum_{\mathrm{i}} w_i = 1$.
3. Output the classifier sign $[F(x)] = \text{sign}\left[\sum_{m=1}^{M} f_m(x)\right]$.

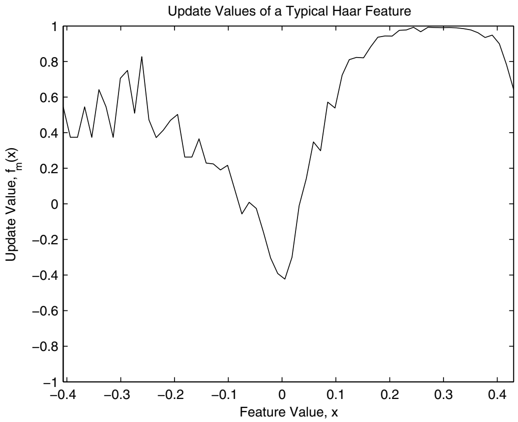
Fig. 1. The GAB algorithm allows for the estimator $f_m(x)$ to range over real numbers

3 GAB Based Haar Feature Selection

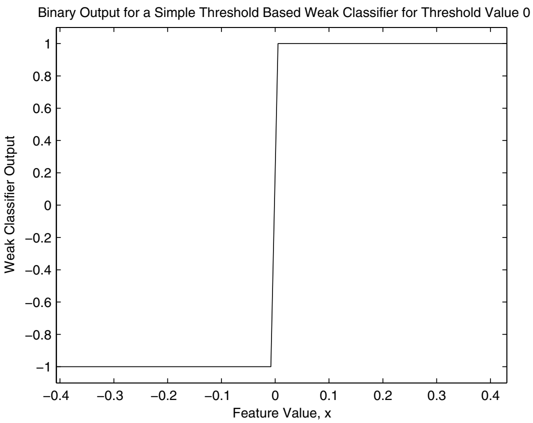
In the original algorithm the simple weak classifiers are built by simply comparing the Haar features to a threshold and thereby producing binary outputs. The feature and its threshold are selected to yield the minimum error at every stage.



(a) Likelihood values of a typical chosen Haar feature by GAB algorithm for two object classes. The corresponding Haar feature is shown in the upper left of the figure



(b) GAB update values $f_m(x) = P_w(y = 1|x) - P_w(y = -1|x)$



(c) Binary output of the classifier in the case a simple threshold based weak classifier is used

Fig. 2. Comparison of GAB update scheme (Fig. a,b) to the simple binary output scheme (Fig. c)

In contrast, in GAB based feature selection mechanism the weak classifiers are not forced to yield the binary outputs, instead they give the values of the update functions $f_m(x)$ at finite number of samples. The comparison of the update values for the binary case and GAB-based case is shown for a typical chosen Haar feature in Figure 2. We use the GAB training algorithm described in Figure 1 to construct a strong stage classifier using confidence values $f_m(x)$ for each feature. Under equal prior probabilities for the object classes, the update $f_m(x)$ is given by

$$
\begin{aligned}
f_m(x) &= P_w(y=1|x) - P_w(y=-1|x) \\
&= \frac{P_w(x|y=1)P(y=1) - P_w(x|y=-1)P(y=-1)}{P_w(x|y=1)P(y=1) + P_w(x|y=-1)P(y=-1)} \\
&= \frac{P_w(x|y=1) - P_w(x|y=-1)}{P_w(x|y=1) + P_w(x|y=-1)}
\end{aligned} \tag{1}
$$

where $P_w(x|y=\pm1)$ are the likelihood values computed by using histograms of feature values x for two different object hypothese. Histogram bins are updated by summing the sample weights of the training set. The subscript w denotes the likelihood values with respect to updated sample weights at each boosting round.

A stage of Haar feature classifier construction using GAB

1. Start with weights $w_i = 1/2p$ and $1/2l$ where p and l are the number of positive and negatives class samples.
2. Repeat for $m = 1, 2, \ldots, M$.
   (a) For each Haar feature j, $f_m(x) = P_w(y=1|x) - P_w(y=-1|x)$ using only the feature j values.
   (b) Choose the best feature confidence set of values $f_m(x)$ giving the minimum weighted error $e_m = E_w\left[1_{(y_i \neq sign[f_m(x_i)])}\right]$ for all feature j.
   (c) Update $F(x) \leftarrow F(x) + f_m(x)$
   (d) Set $w_i \leftarrow w_i \exp\left[-y_i.f_m(x_i)\right], i = 1, 2, \ldots, N$, and renormalize so that $\sum_{\mathrm{i}} w_i = 1$.
3. Output the classifier $\operatorname{sign}[F(x)] = \operatorname{sign}\left[\sum_{m=1}^{M} f_m(x)\right]$.

Fig. 3. At each iteration AdaBoost finds a set of update values $f_m(x)$ which use the values of the feature corresponding to the minimum error

GAB algorithm chooses the best Haar feature resulting with the minimum weighted error $e_m = E_w\left[1_{(y_i \neq sign[f_m(x_i)])}\right]$ from among all available features. The chosen update values are accumulated in the classifier output $F(x)$ value. The output of the classifier, $F_m(x)$, is thresholded by T_m such that the desired target false alarm rate, f, and detection rate, d , of the classifier is achieved. We also used the approach in [4], where the previous stage classifier output,$F_m(x)$, is inputted to the next stage classifier. According to this, in the training process,

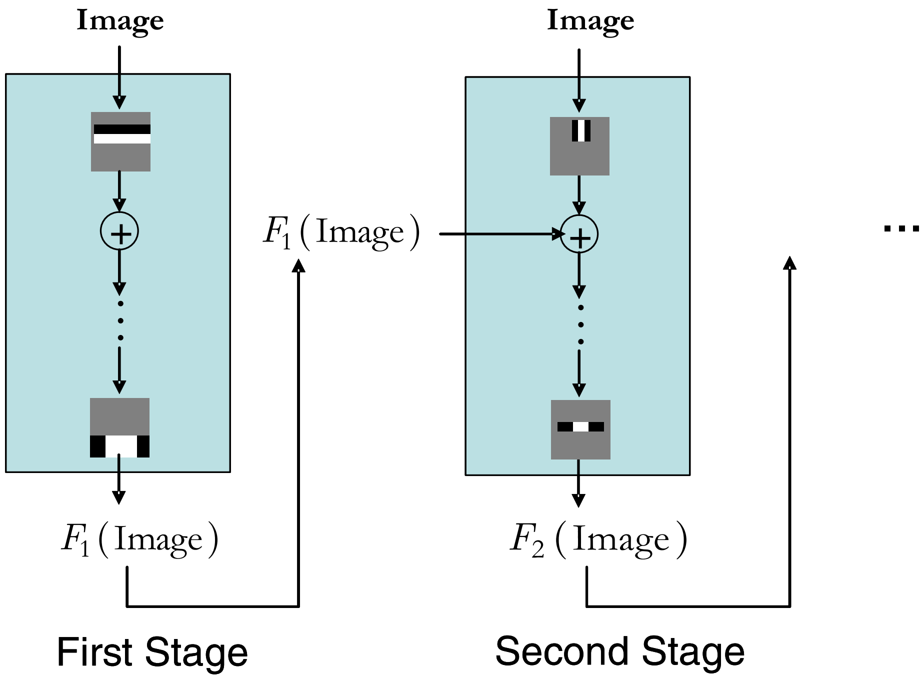


Fig. 4. Propagating the classifier output to the next stage classifier, the previous stage classifier outputs are used to compute the posterior class probabilities of the object classes, $P_w(x|y = 1)$ and $P_w(x|y = -1)$, the update values computed from them is used as an update corresponding to the first feature of the classifier

classifier output values for the training samples of the next stage is used to prepare the posteriori class probabilities of the object classes. Thereby the first update value $f_1(x)$ value of the succeeding classifier is computed by using the histogram values resulting from the classifier outputs of the previous stage. In this sense each previous stage conveys its accumulated information to the next stage classifier. A cascaded classifier is produced by this kind of GAB procedure as illustrated in the Fig 4.

4 Experiments

Figure 5 illustrates a chosen Haar feature, its corresponding posterior class probabilities as a function of feature values. The rightmost figure shows the classifier output as more and more classifiers are added, abscissa indicates the training sample index. Adding more features, samples of two classes can be separated from each other by simply thresholding the final classifier output $F(x)$.

We implemented GAB algorithm for near frontal face detection in cluttered scenes. For training we used about 10000 aligned face and 6000 non-face images with the size of 24x24 at each stage training. Face images contain near frontal

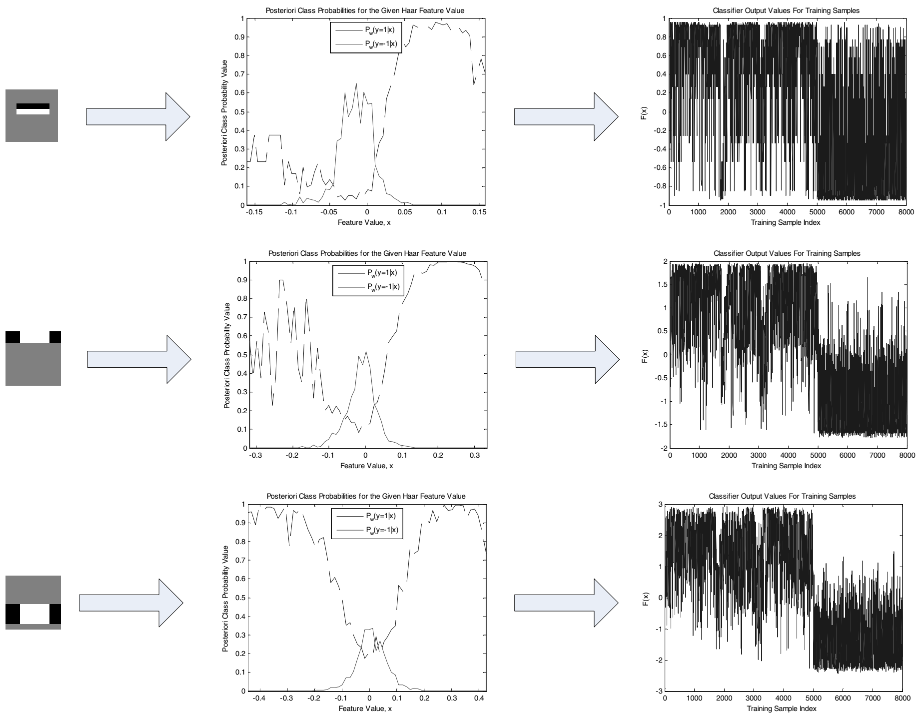


Fig. 5. First column shows the Haar features selected by GAB, second column shows the posterior class probabilities of the object classes, $P_w(x|y=1)$ and $P_w(x|y=-1)$, for the corresponding Haar feature, the last column shows classifier output values $F(x)$ values for the training samples after being updated with each $f_m(x_i)$

faces which are subject to $\pm15°$ inplane rotations and in the range of $\pm30°$ out-of-plane rotations. The number of points to sample the GAB update functions, $f_m(x_i)$, used in our work is 64 and the false alarm probability was chosen as 10^{-6}. These values are stored for each selected feature. In test process, these stored Look-Up Table (LUT) values of the update functions are used to compute the confidence values of each computed feature values. On the same training set, we trained the cascaded structure by using the proposed LUT-based GAB procedure and threshold-based Discrete AdaBoost (DAB) approach . The number of total features produced by GAB based procedure is about 20 percent of the DAB based training case for the same overall target detection and false alarm rate. We tested two methods on the CMU test set containing 503 faces in the 130 images. The number of features and False Alarm (FA)/Detection Rates (DR) are given in Table 1 for the two methods.

As seen from Table 1, not only the LUT-based GAB method performance is higher than the DAB method and also requires much fewer feature as compare to the DAB method, but about one fifth of the number of features the detection performance is even better.

Table 1. Performance and number of feature comparison for the methods. FA : False Alarm, DR : Detection Ratio

Method	Number of FAs/DR	Number of total features
LUT-based GAB	15/85.2%	531
DAB	15/80.1%	2652

5 Conclusions

In this work, we developed a simple and efficient LUT-based GAB training procedure using Haar like features for the near frontal face detection. We tested and compared two methods on the public common test set, CMU test set. This procedure necessitates significantly fewer features with respect to the DAB-based training case for the near frontal face detection problem. We plan to use this approach to implement a rotation invariant multi-view face detection system.

References

1. Viola, P., Jones, M.: Rapid Object Detection Using A Boosted Cascade of Simple Features. IEEE Conference on Computer Vision and Pattern Recognition, (2001).
2. Li, S. Z.: Statistical Learning of Multi View Face Detection. ECCV 2002, Copenhagen, Denmark.
3. Viola, P., Jones, M.: Fast Multi View Face Detection. Technical Report TR2003-96, July 2003, Mitsubishi Electric Research Laboratories.
4. Wu, B., Ai, H., Huang, C.: Fast Rotation Invariant Multi-View Face Detection based on Real Adaboost. AFG'04, May 17-19, 2004, Seoul, Korea.
5. Friedman, J., Hastie, T., Tibshirani, R.: Additive Logistic regression: a Statitical View of Boosting. Technical Report, Standford University, August 17, 1998.
6. R. Liehart, A. Kuranov, V. Pisarevsky, Empirical Analysis of Detection Cascades of Boosted Classifiers for Rapid Object Detection, MRL Tech report, 2002.