Development of real time Face detection system using Haar like features and Adaboost algorithm

A.S.M Shihavuddin[1], Mir Mohammad Nazmul Arefin[2], Mir Nahidul Ambia[3], Shah Ahsanul Haque[4] and Tanvir Ahammad[2]

[1]MSc Student, VIBOT 3rd generation and Lecturer (On leave), EEE Department, IUT,
[2]Lecturer, Electrical and Electronic Engineering Department, Stamford University, Dhaka, Bangladesh
[3]Lecturer, Electrical and Electronic Engineering Department, Eastern University, Dhaka, Bangladesh
[4]Lecturer, Electrical and Electronic Engineering Department, International Islamic University Chittagong (IIUC), Bangladesh

Summary

Human face detection is an active area of research covering several disciplines such as image processing, pattern recognition and computer vision. This paper describes a face detection framework that is capable of processing input images pretty swiftly while achieving high detection rates. The existing methods for face detection can be divided into image based methods and feature based methods. The developed system is intermediary of these two, using a hybrid method comprising boosting algorithm and a hyper plane to train a classifier which is capable of processing images rapidly while having high detection rates. Using the response of simple Haar-based features used by Viola and Jones [1], AdaBoost algorithm and an additional hyper plane classifier, the presented face detection system is developed. This system is further modified by some intuitive noble heuristics. A set of experiments in the domain of face detection is presented. The system yields face detection performance comparable to the best previous systems

***Keywords*:**
Adaboost algorithm, Haar like features, false positive, false negative.

1. Introduction

An ideal face detection system is considered as to be able to identify and locate all faces regardless of their positions, scale, orientation, lightning, expressions and so on. Due to the large intra-class variations in facial appearances, face detection has been a challenging problem in the field of computer vision.

Evidently, face detection is the first step in any automated system which solves the above problems and a robust and effective face detector system is essential. Face detection can be performed based on several different cues: skin colour (for faces in color images), motion (for faces in videos), facial/head shape and facial appearances, or a combination of them. However, detecting faces in black and white, still images with unconstrained, complex backgrounds is a complicated task. So far learning-based approaches have been most effective and have therefore attracted much attention the last years. Recently, Viola and Jones [18], [19] introduced an impressive face detection system capable of detecting frontal-view faces in real time. The desirable properties are partly attributed to the used AdaBoost learning algorithm. AdaBoost, evolved from adaptive boosting, rapidly made impact in the machine learning community when it was presented by Freund and Schapire [2] about 10 years ago. The AdaBoost algorithm sequentially constructs a classified as a linear combination of “weak” classifiers [11], [12]. More recently attention has shifted to a refinement of the original Discrete AdaBoost. One such example is the Real AdaBoost algorithm, by Schapire and Singer [8], [9], [13], which incorporates a measure of confidences to the predictions of each weak classifier.

Automatic face detection is a complex problem which consists in detecting one or many faces in an image or video sequence. The difficulty resides in the fact that faces are non rigid objects. Face appearance may vary between two different persons but also between two photographs of the same person, depending on the lightning conditions, the emotional state of the subject and pose. That is why so many methods have been developed during last few years. Each method is developed in a particular context. These numerous methods clustered into two main approaches: image based methods and feature-based methods. The first one use classifiers trained statically with a given example set. Then the classifier is scanned through the whole image. The other approach consists in detecting particular face features as eyes, nose, etc.

The presented detection system in this paper uses bit of both of this approach. The developed system uses Haar-Like features models of five different templates and uses AdaBoost optimal discrete classifier to select the best combination of weak classifier with corresponding coefficient to create the strong classifier with stronger accuracy. To reduce the computation complexity features are initially created using meaning full heuristics developing a database storing all the feature information’s (size, position, type, threshold and sign). This boosting system can use this information learning very quickly to

Manuscript received January 5, 2010
Manuscript revised January 20, 2010

select the weak classifiers. To access the detection system, 0-1 error and exponential error have been calculated on the both training sets and the test sets. The block diagram graphically illustrates the whole system (Fig.2) for selecting the decision boundary for each feature; the maximum distances between the cumulative distribution of the face and non face variable are used. In addition a hyper plane classifier is also included to further strengthen the prior decision. In this report, the developed face detection system is briefly explained in the next coming sections. The performance of the system on the real images is shown in the later sections.

2. Developed System

AdaBoost is an efficient boosting algorithm which combines simple statistical learners while reducing significantly not only the training error but also the more elusive generalization error. There is no need to have any a priori knowledge about face structure. The most representative features will automatically be selected during the learning. In this presented work, discrete Adaboost is being used to do the boosting of the learning [11], [12], [13], [14]. This project uses five kinds of Haar-like features (Fig 1). The value of a two-rectangle template is the difference between the sums of the pixels within two rectangular regions. The regions have the same size and shape and are horizontally or vertically adjacent. Two types of (vertical and horizontal) ‘three-rectangle feature’ compute the sum within two outside rectangles subtracted from the sum in a center rectangle. Finally, a four-rectangle feature computes the difference between diagonal pairs of rectangles. The total number of possible features can calculate using the following features.

Figure 1: Collection of used Haar Wavelets

Total number of possible features = X.Y (W+1 - w. (X+1) / 2) (H+1 - h X+1) / 2);

Here, X, Y is the windows dimension on which features will be created. For this project it is taken as 24 X 24. For the five templates the total almost 162000 features might be created. Most of them are not really good enough to be created at the first place, as only 2 X 1 size 2 rectangle features. Thus using some heuristics and closing the upper and lower bound, the totally 56000 features are created. For all the these created features a data base named feature value matrix (Fig 2) created by saving its size, position and also the corresponding values given for all the training instances. Using the values of all the training sets of an individual feature threshold is calculated from the random face sets values and random non face set values.

	windows[1]	windows[2]	windows[3]	windows[...]	windows[10,000]
features[1]	-1.098038	-0.07843137	-1.780392		
features[2]	-3.207843	-0.3215686	-2.619609		
features[3]	-3.835294	-1.352941	-2.494117		
features[...]					
features[3000]	9.917649	-9.917548	8.345098		
features[3001]	-1.560785	-0.1254902	-2.737255		
features[...]	-4.262746	-0.4901961	-3.47451		
features[6000]	-5.109803	-2.039216	-2.286276		
features[6001]					
features[...]					
features[9000]					
features[9001]					
features[...]					

Figure 2: feature value matrix for the classifier selection

The threshold for the feature is calculated using the probability density function (PDF) (Fig3.A) of the face and non face variables. From that the corresponding face and non face PDFs, the respective cumulative distribution function (CDF) (Fig3.B) is derived and the value of the variable at which these threshold have the maximum Euclidian distance is considered as the threshold value. The Adaboost learner uses all this features and corresponding threshold to test the training image and makes decision in form of the label train. From the label train the training system measures percentage of error made by all the features and selects that feature which gives the minimum error as a weak classifier. Then from the corresponding error of that weak feature, respective coefficient is calculated and weight is updated of the sets on which it made wrong decision. The weight of all the training sets is normalized each time any change occurred. This process is repeated until the desired number of weak classifier is being selected. All the information about the selected features is kept in the feature matrix (Fig 4). The system uses the following described Adaboost classifier.

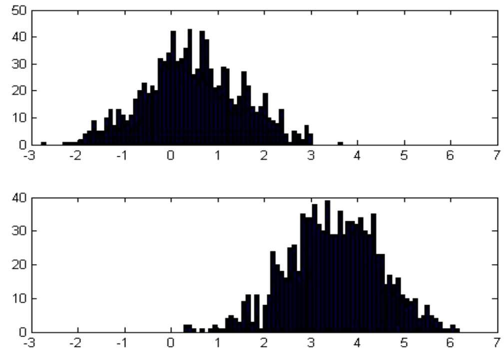


Figure 3A: A PDF of the face and clutter values

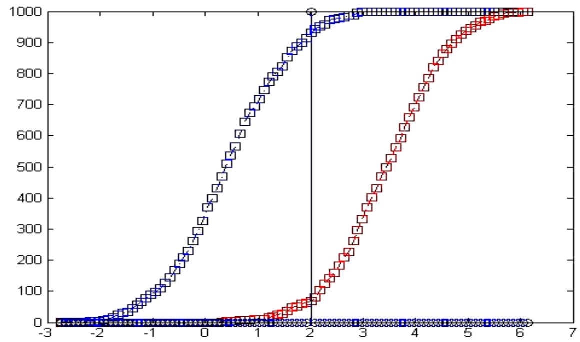


Figure 3.B: Threshold calculation at maximum distances CDF

Features

	features[1]	features[2]			features[18900]
threshold	?	?			?
X1	2	2			21
X2	4	4			23
Y1	2	2			19
Y2	6	8			23
sign	?	?			?
distance	?	?			?

Fig4: feature matrix

2.1 Real AdaBoost

1. Start with weights $W_i = 1/N$, i = 1, 2, … … , N.
2. Repeat for m = 1, 2, … … , M :
   (a) Fit the classifier to obtain a class probability estimate $Pm(x) = P_w(y=1/x) \in [0,1]$ using weights W_i on the training data.
   (b) Set $f_m(x) = ½ \log P_m(x)/(1-P_m(x)) \in R$
   (c) Set $W_i = W_i \exp[-Y_i f_m(X_i)]$, i = 1, 2, … , N and renormalize so that $\sum_i W_i = 1$.
3. Output the classifier, sign ([ ])

The classifier created by this way is tested in two major types. One is on face/clutter classification and the other is on face detector. In the first case, the performance is accessed by the 0-1 loss and exponential loss curve for both the training sets and the test sets. The detection rate, false positive error and false negative error are also calculated to check the performance. In the 2nd method the classifier is directly implemented on a real image and examined visually how it detects the faces.

The overall system block diagram is given as following Fig5

2.2 Hyper plane

Let $a_1, a_2, \ldots, a_n$ be scalars not all equal to 0. Then the set

S consisting of all vectors, $X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$

In R^n such that $a_1x_1 + a_2x_2 + \ldots + a_nx_n = c$

For C a constant is a subspace of Rn called a hyperplane. More generally, a hyperplane is any codimension-1 vector subspace of a vector space. Equivalently, a hyperplane V in a vector space, W is any subspace such that W / V is one-dimensional. Equivalently, a hyperplane is the linear transformation kernel of any nonzero linear map from the vector space to the underlying field.

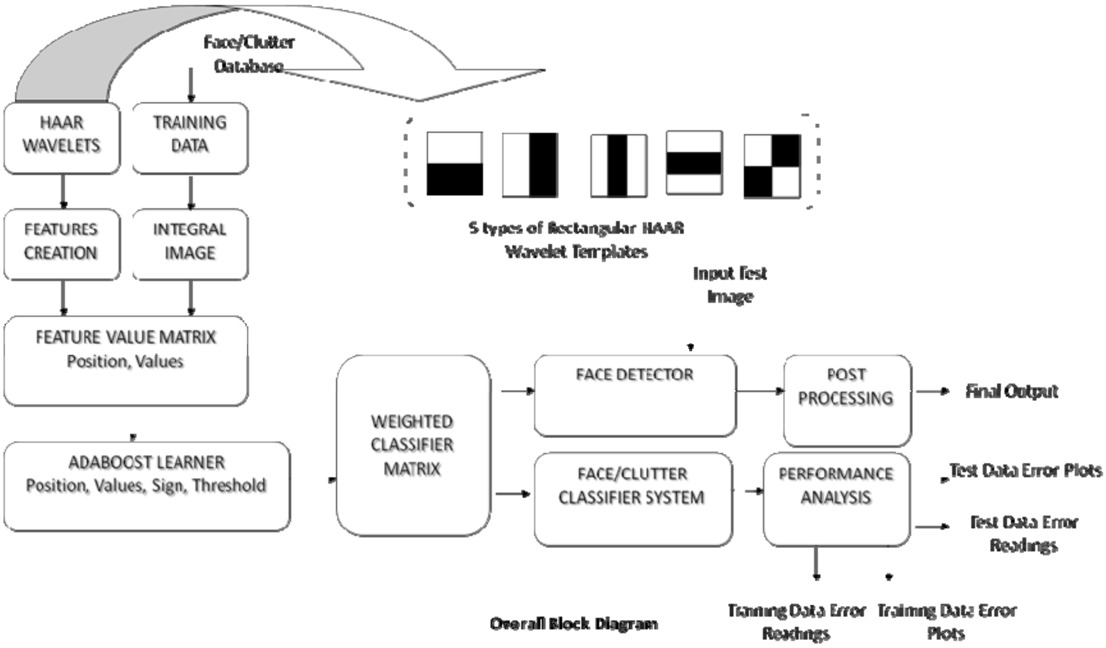


Fig5. Overall block diagram of the system

2.3 Face detector

The detector system uses the learned strong classifier to detect the given samples. In the face/clutter classifier the system, using the same window examines all the selected main weak classifiers on the test set and them combines the decision using the corresponding classifier's coefficients. The final decision is either face (1) or clutter (-1). The classifier performance can be accessed using the 0-1 risk loss and exponential loss. In presented work, Bayes optimal classifier is selected. For the system used,

$$0-1\,risk\ loss = \frac{1}{Total\ number\ of\ test} * Total\ miss\ classification$$

$$Exponential\ loss = 2 * \sqrt{\left(\frac{1-\text{Error Probability}}{\text{Error Probability}}\right)}$$

The detection system is further evaluated on the false positive error and false negative error on the both training and test sets. In this report, false positive error is defined as detection rate of faces where there is none and False Negative means rate of misses to detect a face where there was one. The face detection on real image is done by scanning the whole image window by window having different window size. The base size of the window is 24 X 24. For the bigger faces this window is increased at bigger sixe by scaling all the related features accordingly. During scanning the same face is highly probable of being detected several times. The post processing system cleans up the repetitions of the image or false detection (Fig 6). In the post processing energy of confidence level of the decision is used to find out the best face position and consider all the closest neighbor detections as repetitions. These neighbors are removed from the detection list. The maximum distance of these neighbors (considered as repetition) is

Maximum distance of closer
Neighbors = $\sqrt{2}$ (0.5*Current Window size -1)

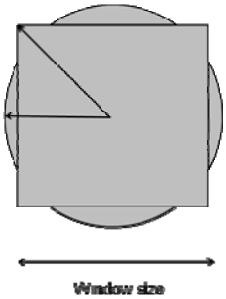


Fig 6: false detection removal (Maximum rejection distance)

The performance can be evaluated visually. The controlling parameter such as window size, Maximum neighbor distance, scanning steps can be adjusted to improve the performance. Some heuristics are also used to intuitively reduce some of the false detection. The more performance achieved, the more computational time required. The tradeoff between these two will give the optimal solution depending on different test inputs and end goal.

3. Results and Evaluation

The developed detection system is first tested as classifier on test images similar to training images. The result can be seen from the table below (Table 1). Performance is mainly accessed on the O-1 Risk loss and the exponential loss. The exponential loss bounds the loss of the developed system. For the face detection experiment, the images used are mainly containing several frontal faces. The result is accessed on the basis of the detection power of faces and also on occurrence of false face detections.

3.1 Testing the face/clutter classifier

In this presented work, the developed system used two different classifier using 4000 and 10000 training sets. The both are showing relatively good performance. The first one (Classifier A) is tested on the 7832 faces and 8000 clutter images. The 2nd (Classifier B) is tested on the 4832 faces and 6000 clutter images.

Table 1: Performance assessment table of classifier A & B

Name of the checking	***Classifier A (trained on 4000 sets)***	***Classifier B (trained on 10000 sets)***
The False Positive Errors in Training	1%	15.35%
The False Negative Errors in Training	0%	1.92%
The 0-1 Risk loss in Training	0.005	0.078889
The Exponential in Training	0.14107	0.53913
The False Positive Errors in Testing	3.1125%	16.8333
The False Negative Errors Testing	4.3412%	1.5728
The 0-1 Risk loss in Testing	0.037203	0.10026
The Exponential Error in Testing	0.37852	0.60069

The result clearly shows a good detection power of the classifier. The classifier A has less 0-1 Risk losses than the Classifier B. Also the same result for the exponential loss in comparison with A & B. Here single decision boundary is used for each feature which reduces the chance of over fitting. Still for classifier A, the 0-1 Risk loss increased by 3.5% in the test sets, which is 2% for the Classifier B. It shows the Classifier A is more over fitted that classifier B. But in overall performance Classifier A is better. In comparison with this system developed by other, the performance of both A & B is satisfactory.

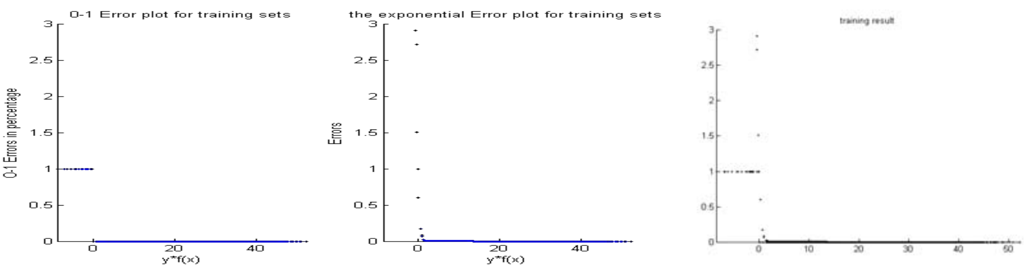


Fig7: The 0-1 Risk loss curve and Exponential curve of the training sets for Classifier A

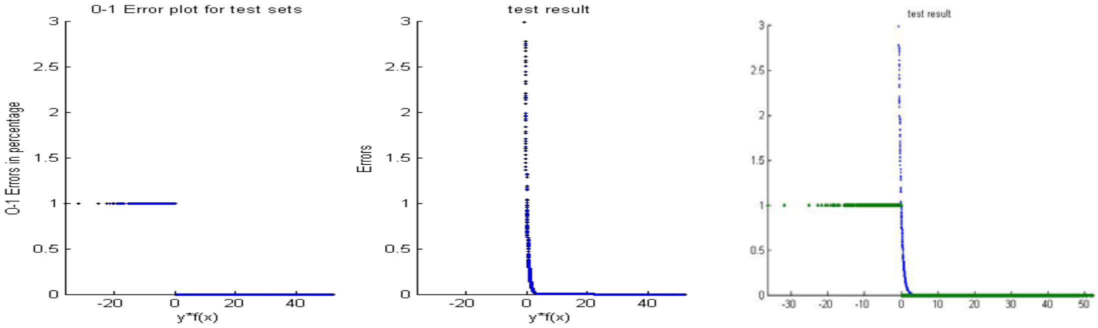


Fig8: The 0-1 Risk loss curve and Exponential curve of the test sets for Classifier A

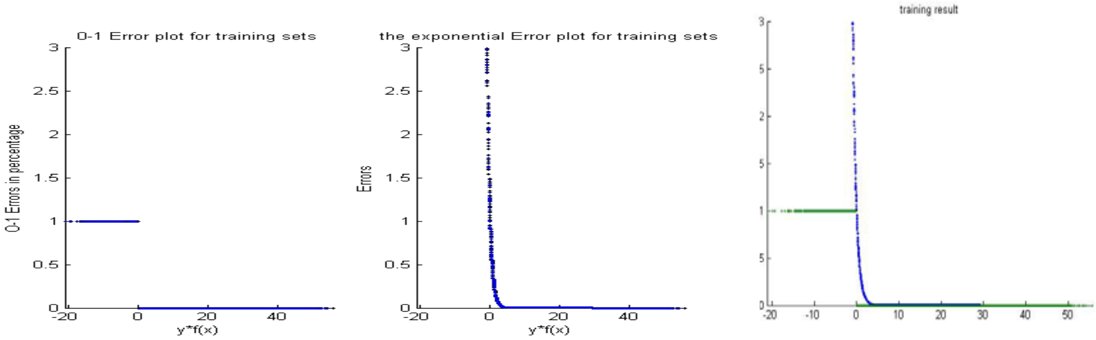


Fig9: The 0-1 Risk loss curve and Exponential curve of the training sets for Classifier B

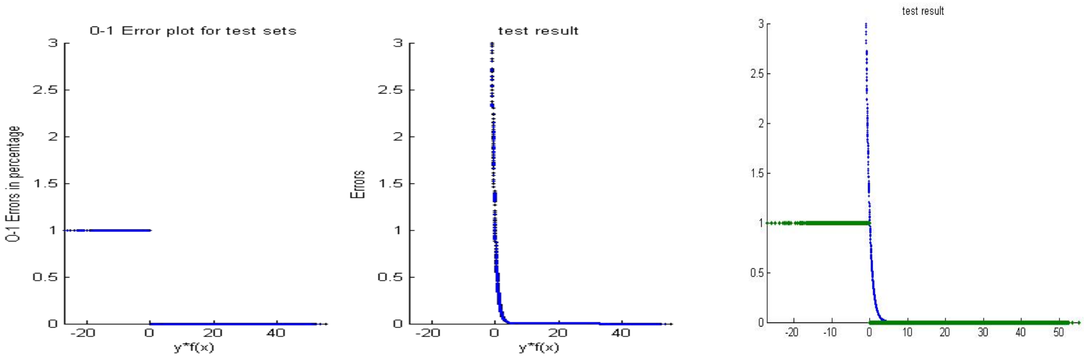


Fig10. The 0-1 Risk loss curve and Exponential curve on the test sets for Classifier B

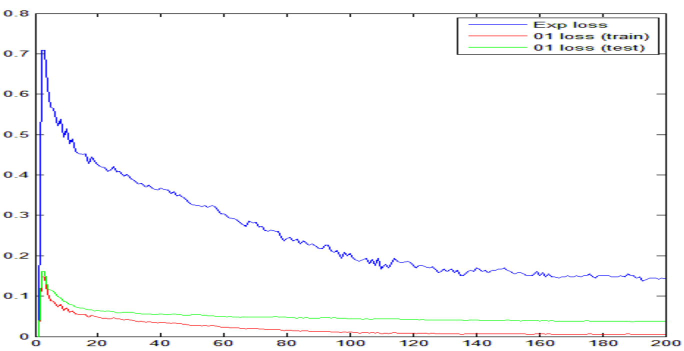
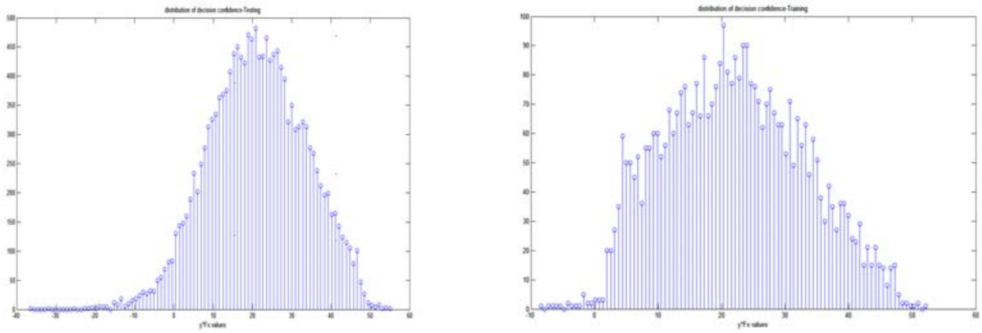


Fig11: Performance of the system Vs the number of classifiers

The number of the selection of the weak classifier is bounded up to 200 as the 0-1 risk loss and also the exponential loss is almost constant at close to 200 classifiers.



(a) (b)

Fig12: Decision confidence evaluation of both training (a) and testing sets (b)

From the figure 12 confidence level of the right and wrong decision for both training and test sets are evident. For the training set the wrong decisions are made with lower confidence. But in the testing set this values is a bit higher. Overall the right confidence level is very good.

3.2 Testing the face detector

The face detection is done on real images by a scanning system. Initially an image created from the training images is tested and perfect result is achieved, meaning all the faces were detected (Fig.13A). An image having faces of different size is being tested. The detector successfully detects all the faces (Fig.13B). When given a more bigger but simple image the performance was still satisfactory (Fig.13C). But for the complex image (Fig 13D), the detector detected all the faces with addition of some false faces which were difficult to remove by post processing system. Finally the exam image was tested and the result given is as the Fig.13E. The overall performance was satisfactory compared with the other developed system. This entire test was done using the classifier A.

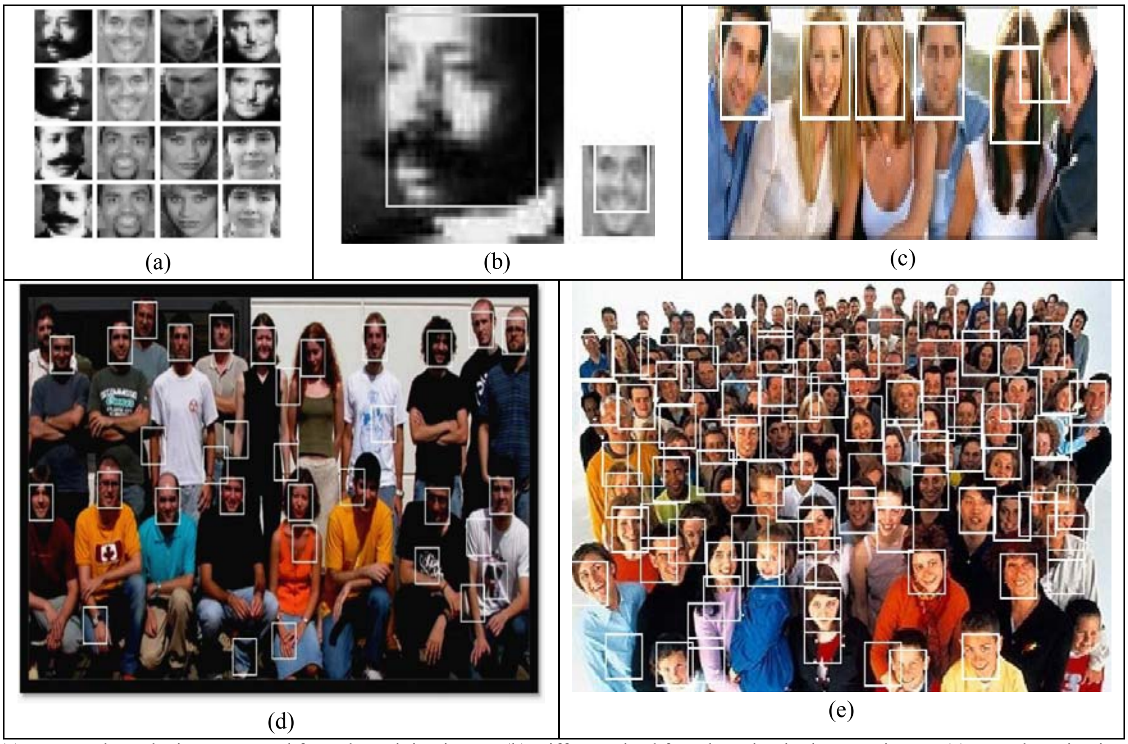


Fig13: (a) Test result on the image created from the training image, (b) Different sized face detection in the same image, (c) Face detection in a simple image, (d) Detection result in a complex image, (e) Result of detection in the exam image

From the results, it can be accessed that the performance is satisfactory, still remains many options to improve. The use of cascading system in the classifier, increased number of clutters in the training sets, good selection of training sets, more precise calculation of threshold will definitely improve the performance of the system. The inclusion of cascading by selecting same types of features in single cascade and using them heuristically to get the better performance can be one possible modification. The hybrid classifier combining classifiers created using different methodology might work better.

6. Conclusion

This paper presents a new approach for face detection which ensures high detection accuracy while reducing computational time. This paper brings together new algorithms, representations, and insights which are quite generic and may well have broader application in computer vision and image processing. The major contribution of this work is to combine adaboost algorithm with the hyper planes concept creating a noble efficient detection system for real time use. The developed system tested on datasets which represents a wide variety of real time situations. This dataset includes faces under a very wide range of conditions including: illumination, scale, pose, and camera variation. Experiments on such a large and complex dataset are difficult and time consuming. Systems which work under these conditions are unlikely to be brittle or limited to a single set of conditions. In the future new templates might be tested, weight can be given more on specific types of error and also a hybrid approach might be taken to use the bests of included methods to work together as a team to give the more efficient and reliable performance.

Reference:

[1] Y. Freund. Boosting a weak learning algorithm by majority. In Proceedings of the Workshop on Computational Learning Theory. Morgan Kaufmann Publishers, 1990.

[2] Y. Freund and R.E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In Proceedings of the Second European Conference on Computational Learning Theory, pages 23–37. Springer-Verlag, 1995.

[3] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: a statistical view of boosting. Technical Report, Dept. of Statistics, Stanford University, July 1998.

[4] ISO/IEC JTC 1/SC 29/WG 11 Moving Picture Experts Group.

[5] Stan Z. Li and ZhenQiu Zhang. FloatBoost learning and statistical face detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 26(9):1112– 1123, 2004.

[6] S.Z. Li, L. Zhu, Z.Q. Zhang, A. Blake, H. Zhang, and H. Shum. Statistical learning of multi-view face detection. In Proceedings of the 7th European Conference on Computer Vision-Part IV, volume 4, pages 67–81. Springer-Verlag, 2002.

[7] R. Lienhart, A. Kuranov, and V. Pisarevsky. Empirical analysis of detection cascades of boosted classifiers for rapid object detection. MRL Technical Report, May 2002.

[8] N. Littlestone and M.K. Warmuth. The weighted majority algorithm. In IEEE Symposium on Foundations of Computer Science, pages 256–261, 1989.

[9] C. Liu. A bayesian discriminating features method for face detection. In IEEE Trans. Pattern Analysis and Machine Intelligence, volume 25, pages 725–740, June 2003.

[10] T. Poggio and K. Sung. Example-based learning for view-based human face detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(1):39–51, 1998.

[11] H.A. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(1):23–38, 1998.

[12] R.E. Schapire, Y. Freund, P. Bartlett, and W.S. Lee. Boosting the margin: a new explanation for the effectiveness of voting methods. In Proc. 14th International Conference on Machine Learning, pages 322–330. Morgan Kaufmann, 1997.

[13] R.E. Schapire and Y. Singer. Improved boosting using confidence-rated predictions. Machine Learning, 37(3):297–336, 1999.

[14] H. Schneiderman and T. Kanade. A statistical method for 3d object detection applied to faces and cars. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), pages 746–751, June 2000.

[15] J. Šochman and J. Matas. AdaBoost with totally corrective updates for fast face detection. In Proceeding of the Sixth IEEE International Conference on Automatic Face and Gesture Recognition, pages 445–450, May 2004.

[16] J. Šochman and J. Matas. Waldboost - learning for time constrained sequential detection. In Proc. of Conference on Computer Vision and Pattern Recognition (CVPR), volume 2, pages 150–157. IEEE Computer Society, June 2005.

[17] V.N. Vapnik and A. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. Theory of Probab. and its Applications, 16(2):264–280, 1971.

[18] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In Proc. IEEE Conference on Computer Vision and Pattern Recognition, pages 511–518, Dec 2001.

[19] P. Viola and M. Jones. Robust real-time object detection. IEEE ICCV Workshop Statistical and Computational Theories of Vision, July 2001.