International Journal of Modern Engineering Research (IJMER)
www.ijmer.com Vol. 2, Issue. 5, Sep.-Oct. 2012 pp-3556-3560 ISSN: 2249-6645

Face Detection System On AdaBoost Algorithm Using Haar Classifiers

M.Gopi Krishna, A. Srinivasulu,
[1]*M. Tech Student, Department of Electronics (E.C.E), Vaagdevi Institute of Tech & Science, Peddasettipalli (Village), Proddatur (M.D), Kadapa (Dist), Andhra Pradesh A.P, India.*
[2]*M. Tech, Assistant Professor, Department of Electronics (E.C.E), Vaagdevi Institute of Tech & Science, Peddasettipalli (Village), Proddatur (M.D). Kadapa(Dist), Andhra Pradesh A.P, India.*

Abstract: *This paper presents an architecture for face detection based system on AdaBoost algorithm using Haar features. We describe here design techniques including image scaling, integral image generation, pipelined processing as well as classifier, and parallel processing multiple classifiers to accelerate the processing speed of the face detection system. Also we discuss the optimization of the proposed architecture which can be scalable for configurable devices with variable resources. The proposed architecture for face detection has been designed using Verilog HDL and implemented in Modelsim. Its performance has been measured and compared with an equivalent hardware implementation. We show about 35 time's increase of system performance over the equivalent hardware implementation.*

Keywords: *AdaBoost; architecture; face detection; Haar classifier; image processing; real-time.*

I. INTRODUCTION

Face detection in image sequence has been an active research area in the computer vision field in recent years due to its potential applications such as monitoring and surveillance [1], human computer interfaces [2], smart rooms [3], intelligent robots [4], and biomedical image analysis [5]. Face detection is based on identifying and locating a human face in images regardless of size, position, and condition Simple features such as color, motion, and texture are used for the face detection in early researches. However, these methods break down easily because of the complexity of the real world.

Face detection proposed by Viola and Jones [6] is most popular among the face detection approaches based on statistic methods. Although real-time face detection is possible using high performance computers, the resources of the system tend to be monopolized by face detection.

Therefore, this work has resulted in the development of a real-time face detection system employing an FPGA implemented system designed by Verilog HDL. Its performance has been measured and compared with an equivalent software implementation.

This paper is organized as follows: In Section II, we explain the face detection algorithm. In Section III, we describe the architecture, designed with Verilog HDL, of a face detection system using block diagrams. In Section IV, we show the implementation of the real-time face detection system in an FPGA and measure the corresponding performance. Finally, we conclude in Section V.

II. FACE DETECTION ALGORITHM

The face detection algorithm proposed by Viola and Jones is used as the basis of our design. The face detection algorithm looks for specific Haar features of a human face. When one of these features is found, the algorithm allows the face candidate to pass to the next stage of detection. A face candidate is a rectangular section of the original image called a sub-window. Generally these sub-windows have a fixed size (typically 24×24 pixels).

This sub-window is often scaled in order to obtain a variety of different size faces. The algorithm scans the entire image with this window and denotes each respective section a face candidate [6].

A. Integral Image

The integral image is defined as the summation of the pixel values of the original image. The value at any location (x, y) of the integral image is the sum of the image's pixels above and to the left of location (x, y). "Fig. 1" illustrates the integral image generation.

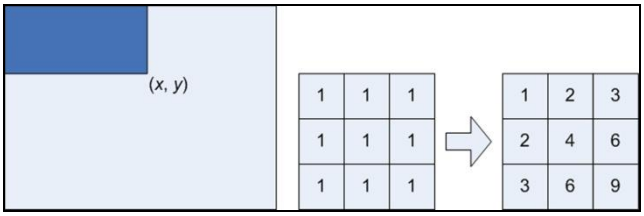


Figure 1. Integral image generation. The shaded region represents the sum of the pixels up to position (*x*, *y*) of the image. It shows a 3×3 image and its integral image representation.

B. Haar Features

Haar features are composed of either two or three rectangles. Face candidates are scanned and searched for Haar features of the current stage.The weights are constants generated by the learning algorithm. There are a variety of forms of features as seen below in "Fig. 2".

Figure 2. Examples of Haar features. Areas of white and black regions are multiplied by their respective weights and then summed in order to get the Haar feature value.

Each Haar feature has a value that is calculated by taking the area of each rectangle, multiplying each by their

respective weights, and then summing the results. The area of each rectangle is easily found using the integral image. The coordinate of the any corner of a rectangle can be used to get the sum of all the pixels above and to the left of that location using the integral image Since L1 is subtracted off twice it must be added back on to get the correct area of the rectangle. The area of the rectangle R, denoted as the rectangle integral, can be computed as follows using the locations of the integral image: L4-L3-L2+L1.

B. Haar Feature Classifier

A Haar feature classifier uses the rectangle integral to calculate the value of a feature. The Haar feature classifier multiplies the weight of each rectangle by its area and the results are added together. Several Haar feature classifiers compose a stage. A stage comparator sums all the Haar feature classifier results in a stage and compares this summation with a stage threshold.

The threshold is also a constant obtained from the AdaBoost algorithm. Each stage does not have a set number of Haar features. For example, Viola and Jones' data set used 2 features in the first stage and 10 in the second. All together they used a total of 38 stages and 6060 features [6]. Our data set is based on the OpenCV data set which used 22 stages and 2135 features in total.

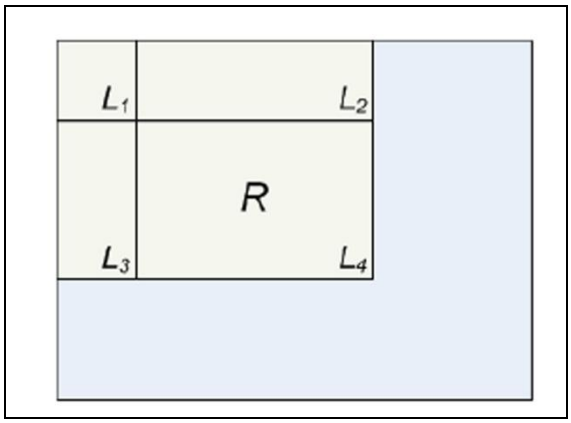


Figure 3. Calculating the area of a rectangle R is done using the corner of the rectangle: L4-L3-L2+L1.

C. Cascade

The Viola and Jones face detection algorithm eliminates face candidates quickly using a cascade of stages. The cascade eliminates candidates by making stricter requirements in each stage with later stages being much more difficult for a candidate to pass. Candidates exit the cascade if they pass all stages or fail any stage. A face is detected if a candidate passes all stages. This process is shown in "Fig. 4".

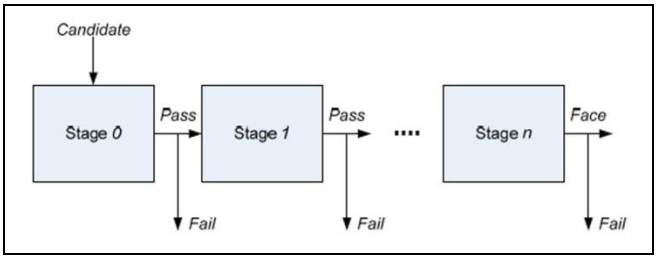


Figure 4. Cascade of stages. Candidate must pass all stages in the cascade to be concluded as a face.

III. IMPLEMENTATION

A. System Overview

We proposed architecture for a real-time face detection system. "Fig. 5" shows the overview of the proposed architecture for face detection. It consists of five modules: variant pose, illumination condition, Facial Expression, Occulsion, Uncontrolled Background, display. Face Detection systems are not only detected faces on uniform environment. In reality, Peoples are always located on complex background with different texture and object. These 'thing' are the major factors to affect the performance of face detection system

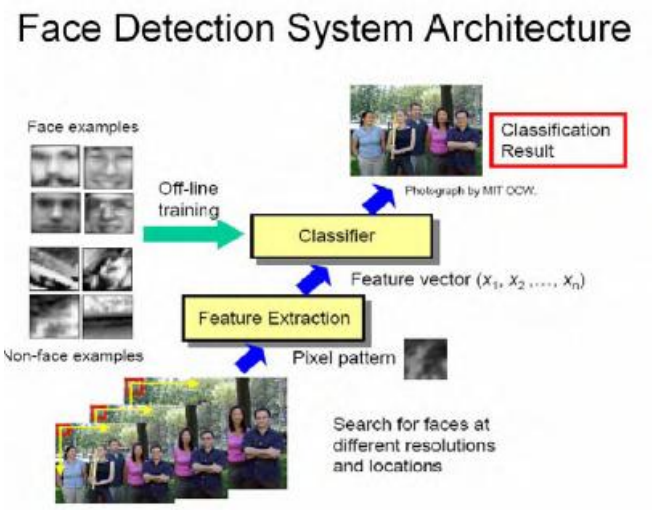


Figure 5. Block diagram of proposed face detection system.

B. Architecture for Face Detection

1) Variant Pose

Variant pose is occurred because of peoples not always orient to camera.

The image sync signal and the color image data are transferred from the image interface module. The image cropper crops the images based on the sync signals.

2) Illumination Condition

Different lighting and the quality of camera directly affect the quality of face. Sometimes it can be varied greater than facial expression and occlusion.

3) Facial Expression

The integral image generation requires substantial computation. A general purpose computer of Von Neumann architecture has to access image memory at least width×height times to get the value of each pixel when it processes an image with width×height pixels. For the incoming pixel where the coordinate is (x, y), the image line buffer controller.

4) Occulsion

Face detection not only deals with different faces, however, it need deal with any optional object. E.g. Hairstyle, sunglasses are the example of occlusion in face detection.

C. Integral Image

For the incoming pixel where the coordinate is (x, y), the image line buffer controller performs operations such as in "(1)", where n is the image window row size, p(x, y) is the incoming pixel value, and L(x, y) represents each pixel in the image line buffer.

$$L(x, y-k) = L(x, y-(k-1)), \text{where } 1 \le k \le n-2 \quad (1)$$

$$L(x, y-k) = p(x, y), \text{where } k = 0$$

The image window buffer stores pixel values moving from the image line buffer and its controller generates control signals for moving and storing the pixel values.

$$I(i-k, j) = I(i-(k-1), j), \text{where } 1 \le k \le m-1 \quad (2)$$

$$I(i, j-l) = L(x, y-(l-1)), \text{where } 1 \le l \le n-1$$

$$I(i-k, j-l) = p(i, j) = p(x, y), \text{where } k = l = 0,$$

$$\text{when } k+l = n-1,\ 1 \le k \le n-1,\ 0 \le l \le n-2,\ m = 2n,$$

$$I(i-k, j-l) = I(i-(k-1), j-l) + I(i-(k-1), j-(l+1))$$

For the incoming pixel with coordinate (x, y), the image window buffer controller performs operation as in "(2)" where n and m are the row and column size of the image window buffer, respectively. p(i, j) is the incoming pixel value in the image window buffer; p(x, y) is the incoming pixel value; I(i, j) represents each of the pixels in the image window buffer; and L(x, y) represents each of the pixels in the image line buffer.

$$II(s-u, t-v) = \quad (3)$$
$$II(s-u, t-v) + I(i-k, j-l) - I(i-(2n-1), j-l)$$
where $0 \le u \le n-1, 0 \le v \le n-1, n-1 \le k \le 2n-2,$
$0 \le l \le n-1$

Since pixels of an integral image window buffer are stored in registers, it is possible to access all integral pixels in the integral image window buffer simultaneously to perform the Haar feature classification.

"Fig. 6" shows all of the actions in the proposed architecture to generate the integral image. For every image from the frame grabber module, the integral image window buffer is calculated to perform the feature classification using the integral image.

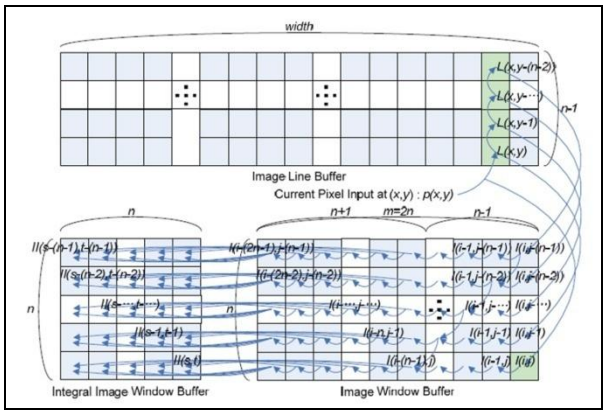


Figure 6. Architecture for generating integral image window.

A Haar classifier consists of two or three rectangles and their weight values, feature threshold value, and left and right values. Each rectangle presents four points using the coordinates (x, y) of most left and up point, width w, and height h as shown in "Fig. 7".

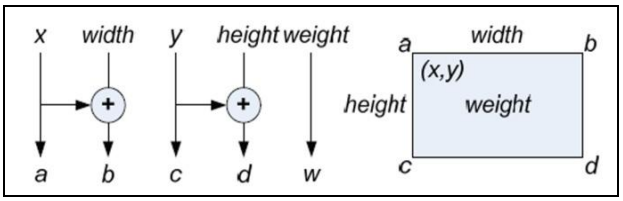


Figure 7. Rectangle calculation of Haar feature classifier.

The integral pixel value of each rectangle can be calculated using these points from the integral image window buffer as shown in "Fig. 8".

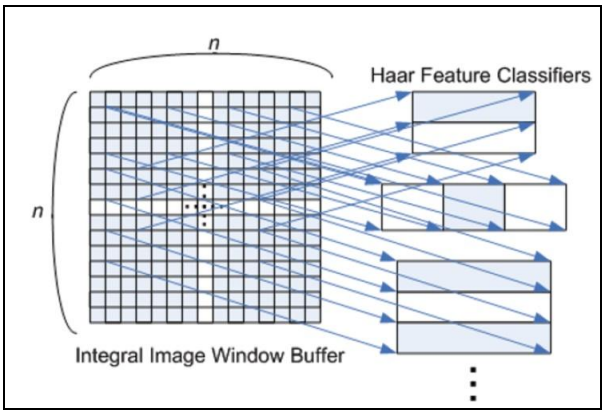


Figure 8. Simultaneous access to integral image window in order to calculate integral image of Haar feature classifiers.

Four points of the rectangles of the Haar feature classifer are calculated by the method as shown in "Fig. 7". The integral image values of Haar classifier are obtained from the integral image window buffer as shown in "Fig. 8". Integral image value of each rectangle multiplies with its weight.

Display:

In the display module, the Digital Visual Interface (DVI) specification is applied to display the image sequence to the LCD monitor through a DVI transmitter in the DVI interface module. This module generates the sync signals and image data for the DVI transmitter.

Implementation

The proposed architecture for face detection has been designed using Verilog HDL and implemented in MODEL-SIM Altera 6.3 . We use the Haar feature training data from OpenCV to detect the frontal human faces based on the Viola and Jones algorithm. are 20x20, that includes a total of 22 stages, 2135 Haar classifiers, and 4630 Haar features.

1. Preprocessing

"System input is color images which included images of human faces or not, output is the human faces which is extracted from original images. In order to get the better result of detection, pre-processing is essential.

Gray scale Conversion

For getting to reduce the information of images, image should be done a converting to grayscale. Each color images (RGB images) are composed of 3 channels to present red, green and blue components in RGB space.

1. Given example images (R1,G1,B1),….(Rn,Gn,Bn) where R, G, B are the value of red, green and blue respectively and 'n' is total number of pixel in given image.

2. The new grayscale images has pixel from G1,…..Gn, where using formula is as follows: 0.21R + 0.71G + 0.07 B = G . Unlike averages method, this form is considering the ratio because of human perception.

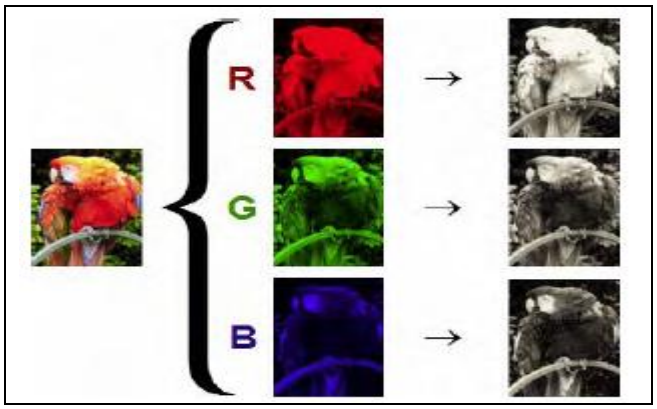
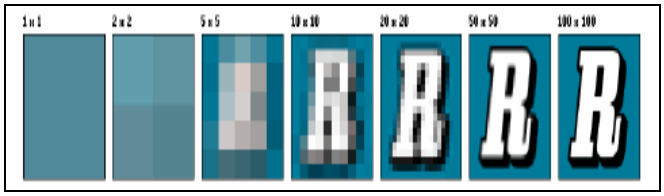


Figure 9 : RGB to Gray Scale Conversion

Image resizing

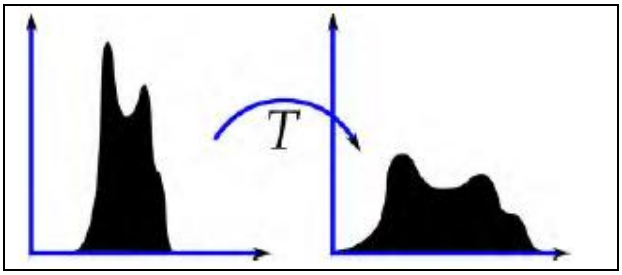
Images are synthesized by numerous of pixel which is the small unit in the imageFor example, '0' is white and '255' is black in gray scale images.

Image has 3000 pixels in width and 2000 pixels in height which means it has 3000 x 2000 = 6,000,000 pixels or 6 megapixels. If the image has been resized into 1000 pixels in width and 600 pixels in height, image only has 0.6 megapixels. At least system only uses 1/10 timing to handle it.

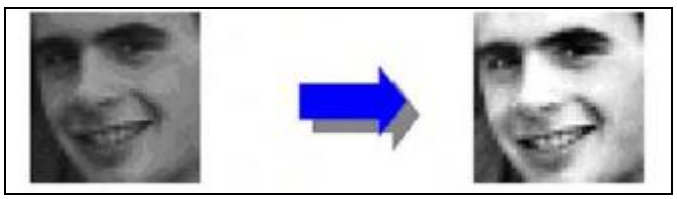


3. Histogram Equalization

Histogram equalization is a statistical method of images processing. It works as a statistical histogram of color distribution of the average scattered in the histogram The change of the histogram after perform histogram equalization.



In the above chart, the shape of graph has been widened which is the meaning of average scattered in the histogram. This method usually increases the contrast of the input imagesIn face detection system, The left-hand side of below images is resized grayscale images


Example of the process of histogram equalization

Algorithms of Histogram equalization:

1. Grayscale images has Xn pixels with i represent a value of gray level in each pixel. The following chart is represent the relationship between probability of occurrence and the value of each pixel:

Chart of Probability Density Function (PDF) And

$$\sum_{i=0}^{255} p(x_i) = 1$$

Px is being historgram of images and normalized to [0,1]

2. Let us define the cumulative distribution function as follows:

$$F(n) = \sum_{i=0}^{n} p(x_i) \qquad n = 0,1,2...255$$

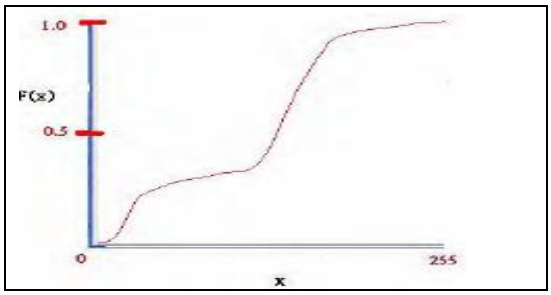


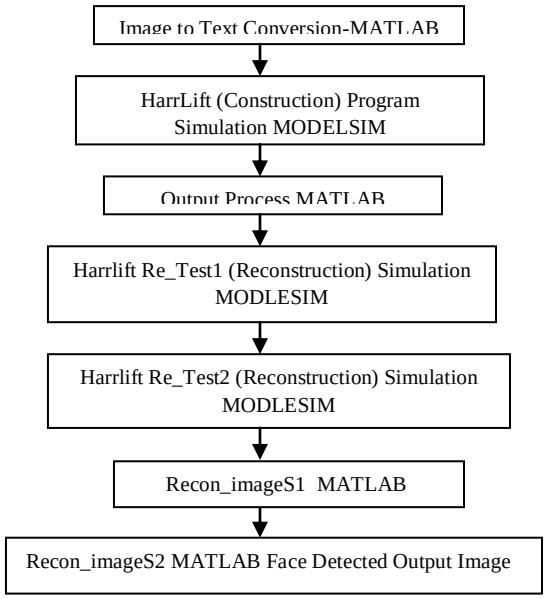
Chart of Cumulative distribution function

3. Minimum and maximum value are found and applied into following equation to find out the histogram equalization of each pixel:

$$h(v) = \text{round}\left(\frac{cdf(v) - cdf_{min}}{(M \times N) - cdf_{min}} \times (L-1)\right)$$

Where cdf_{min} is the minimum value of CDF, M is the width of image and N is the height of image. L represent a large value of grey level, = 25

SIMULATION FLOW



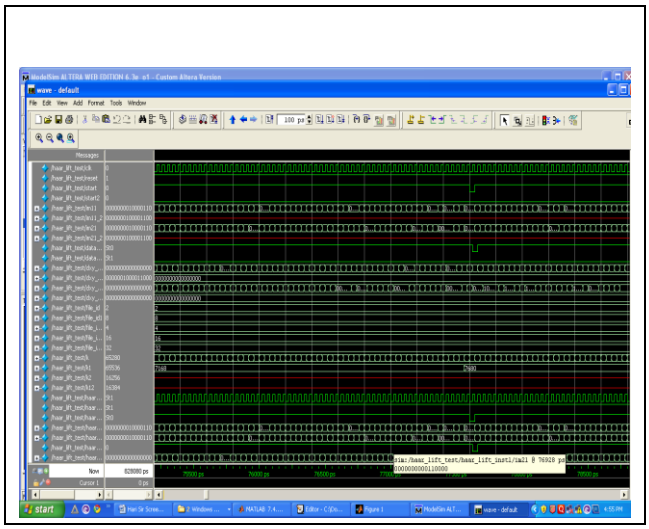


FIGURE: WAVE FORMS IN MODELSIM

IV. EXPERIMENTS/RESULTS

A high frame processing rate and low latency are important for many applications. We measure the performance of the proposed architecture for the face detection system.Face detection system when it is applied to a camera, which produces images consisting of 640×480 pixels at 60 frames per second.

Figure 10.Experimental result of face detection system.

V. CONCLUSION

We present face detection based on the AdaBoost algorithm using Haar features. In our architecture, the scaling image technique is used instead of the scaling sub-window, and the integral image window is generated instead of the integral image contains whole image during one clock cycle.

The Haar classifier is designed using a pipelined scheme, and the triple classifier which three single classifiers processed in parallel is adopted to accelerate the processing speed of the face detection system.

Finally, the proposed architecture is implemented on a Modelsim Altera 6.3 and its performance is measured and compared with an equivalent hardware implementation. We show about 35 times increase of system performance over the equivalent software implementation. We plan to implement more classifiers to improve our design.

We have demonstrated that this face detection, combined with other technologies, can produce effective and powerful applications.

REFERENCES

[1] Z. Guo, H. Liu, Q. Wang and J. Yang, " A Fast Algorithm of Face Detection for Driver Monitoring, " In Proceedings of the Sixth International Conference on Intelligent Systems Design and Applications, vol. 2, pp.267 - 271, 2006.

[2] M. Yang, N. Ahuja, "Face Detection and Gesture Recognition for Human-Computer Interaction," The International Series in Video Computing, vol.1, Springer, 2001.

[3] Z. Zhang, G. Potamianos, M. Liu, T. Huang, "Robust Multi- View Multi-Camera Face Detection inside Smart Rooms Using Spatio-Temporal Dynamic Programming," International Conference on Automatic Face and Gesture Recognition, pp.407-412, 2006.

[4] W. Yun; D. Kim; H. Yoon, "Fast Group Verification System for Intelligent Robot Service," IEEE Transactions on Consumer Electronics, vol.53, no.4, pp.1731-1735, Nov. 2007.

[5] V. Ayala-Ramirez, R. E. Sanchez-Yanez and F. J. Montecillo-Puente "On the Application of Robotic Vision Methods to Biomedical Image Analysis," IFMBE Proceedings of Latin American Congress on Biomedical Engineering, pp.1160-1162, 2007.

[6] P. Viola and M. Jones, "Robust real-time object detection," International Journal of Computer Vision, 57(2), 137-154, 2004.

[7] Y. Freund and R. E. Schapire, "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting," Journal of Computer and System Sciences, no. 55, pp. 119-139, 1997.