Face Detection Using The Embedded System

Yong Jun Lee*, Dong Hwan Ko, Seung Min Song and Hoon Kang
School of Electrical and Electronic Engineering, Chung-Ang University
221 Heukseok-Dong, Dongjak-Gu, Seoul 156-756, Korea
Tel. +82-2-816-8234, Fax. +82-2-816-8234
e-mail : yj3628@robot.cau.ac.kr
Keyword: Face Detection, Adaboost, Embedded System

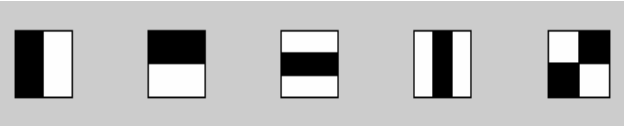
Abstract: This paper based on the face detection framework proposed by Viola and Jones 2001. There are three key contributions. The first is the introduction of a new image representation called the "Integral Image" which allows the features used by oue detector to be computed very quickly. The second is a learning algorithm, based on AdaBoost, which selects a small number of critical visual features and yields extremely efficient classifier. The third is embedded system. This converts a non-embedded system into a embedded system.

1 Introduction

In this paper, we propose a face detection algorithm using embedded system. Generally, face detection algorithms are degraded by changing illumination in an image. To avoid this, we use the Adaboost algorithm for face detection. Adaboost is a very rapid and robust algorithm. We exploit the intel pxa 320 type's board as the target board of embedded system.

2 Harr Basis Function

To acquire the features of given face images, we use the Haar basis function which is based on rectangular features[2],[4]. An example of rectangular feature is shown in Fig1.



<Figure 1> two, three, four Rectangle feature

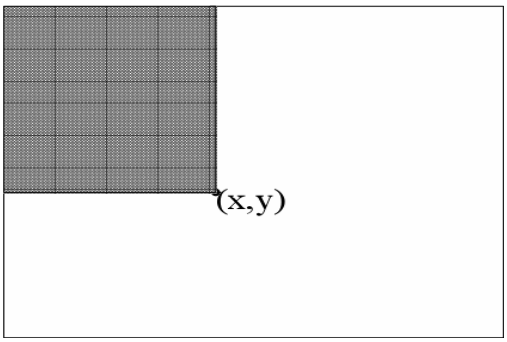
Basically, those regions have the same size and shape. The value of a two-rectangle feature is the difference between the sum of the pixels with in two rectangular regions. A three-rectangle feature computes the sum within two outside rectangles subtracted from the sum in a center rectangle. Finally a four-rectangle feature computes the difference between diagonal pairs of rectangle. Given that the base resolution of the detector is 24x24, the set of rectangle features is 45,396.

3 Integral Image

Rectangle Features can be computed very rapidly using an Intermediate representation for image which we call the integral image. As a result, The integral image very rapidly computes the features which use a harr basis function. The location x, y of The integral image contains the sum of the pixels. Therefore, The values of the integral image are followed this equation:

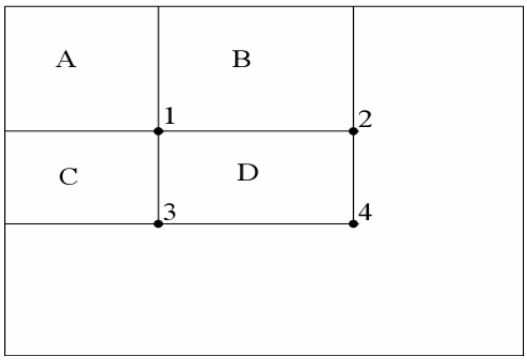
$$ii(x,y) = \sum_{x' \le x,\, y' \le y} i(x', y')$$

Where ii(x,y) is the integral and $i(x',y')$ is the original image.



<Figure2> The value of the integral image at point (x,y) is the sum of all pixels

The value of the special region using the integral image is computed this method. For example, The sum of pixels within D regions can be computed this method. The sum of pixels within D regions can be computed with four array references. The value of the integral image at location 1 is sum of the pixels in rectangle A. The value at location 2 is A + B, at location 3 is A + C, and at location 4 is A + B + C + D. Consequently, the sum within D can be computed as 4 + 1 – (2 + 3).



<Figure3> The sum of pixels within rectangle D: 4 + 1 – (2 + 3)

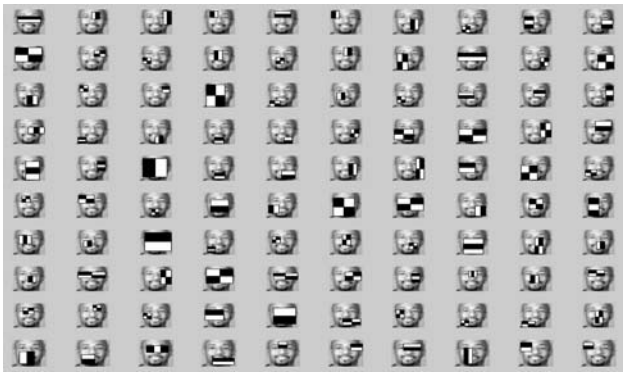
4 Adaboost

We use the Adaboost algorithm as a face detection algorithm. This algorithm classifies face images using simple features. Those simple features are acquired to use the Haar basis function which is based on rectangle features. After finding

adequate face features using the Adaboost algorithm, a strong classifier can be obtained by combining those weak classifiers. The learning procedure, the error of the selected weak classifier is set up in inverse proportion to weight. Therefore, next learning weak classifier classifies better than current learning weak classifier.
Figure 4 shows selected features by Adaboost. The Table 1 shows a process of AdaBoost [1-3],[5-7].

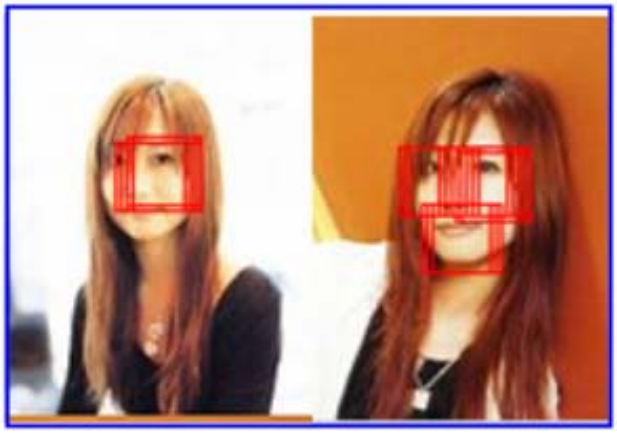
Step1	Given example images $(x_1, y_1),...,(x_n, y_n)$ where $y_i = 0,1$ for negative and positive examples respectively.
Step 2	Initialize weights $\omega_{1,i} = \frac{1}{2m}, \frac{1}{2l}$ $for\ y_i = 0,1$, respectively , where m and l are the number of negatives and positive respectively.
Step3	For t=1,…, T 1. Normalize the weights $\omega_{t,i} = \frac{\omega_{t,i}}{\sum_j^n \omega_{t,j}}$ So that ω_t is a distribution 2. For each , j, train a classifier h_j which is restricted to using a single feature. The error is evaluated with respect to $\omega_t, \varepsilon_j = \sum_i \omega_t \mid h_j(x_i) - y_i \mid$ $h_j(x) = \begin{cases} 1 & if\ p_j f_j(x) < p_j \theta_j \\ 0 & otherwise \end{cases}$ Feature (f_j), threshold(θ_j), Parity(p_j) 3. Choose the classifier, h_t , with the lowest error ε_t 4. update the weights: $\omega_{t+1,i} = \omega_{t,i} \beta_t^{1-e_i}$ Where $e_i = 0$ if example x_i is classified correctly, $e_i = 1$ otherwise, and $\beta_t = \frac{\varepsilon_t}{1-\varepsilon_t}$
Step4	The final strong classifier is: $h(x) = \begin{cases} 1 & \sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\ 0 & otherwise \end{cases}$ Where $\alpha_t = \log \frac{1}{\beta_t}$

<Table 1> The boosting algorithm of Adaboost



<Figure4> The selected feature by adaboost

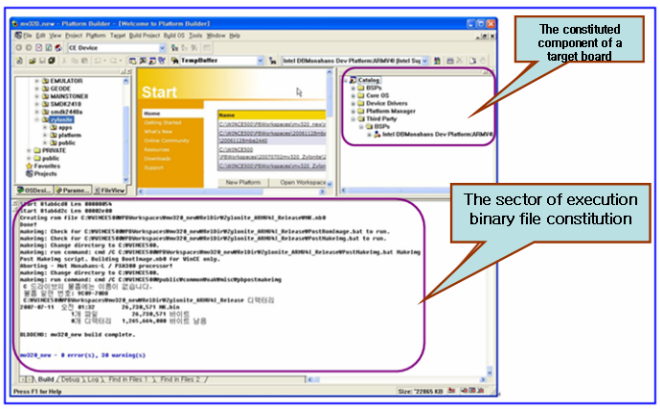
Figure 5 shows an example of the Adaboost result.



<Figure5> Adaboost result

5 The Porting Of Embedded System

An embedded system and a non-embedded system has a difference of an environment. Therefore, A proper translation process essentially is required. It follows that:
1) An Wince(Platform Builder) installation in a PC(non-embedded system)
2) The constituted component installation of a target board and execution binary file constitution in a platform builder.
Figure 6 shows the constituted component of a target board and the sector of execution binary file constitution(NK.bin, Nk.nb0).

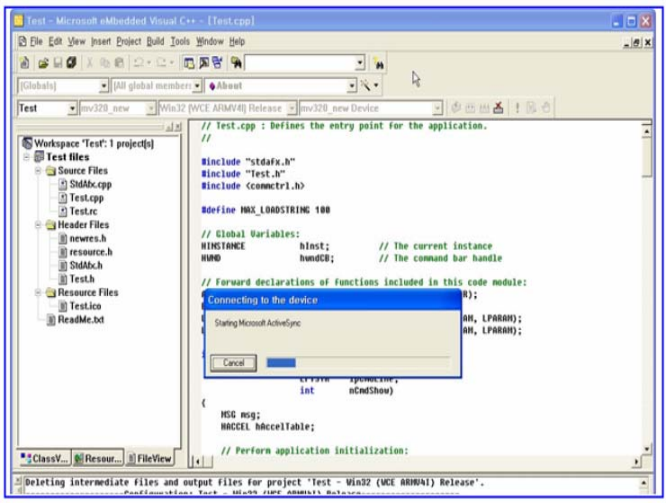


<Figure 6> The constituted component and execution binary file

3) The sdk setting/installation of a target board in a platform builder. Figure 7 shows composed sdk file of the target board.

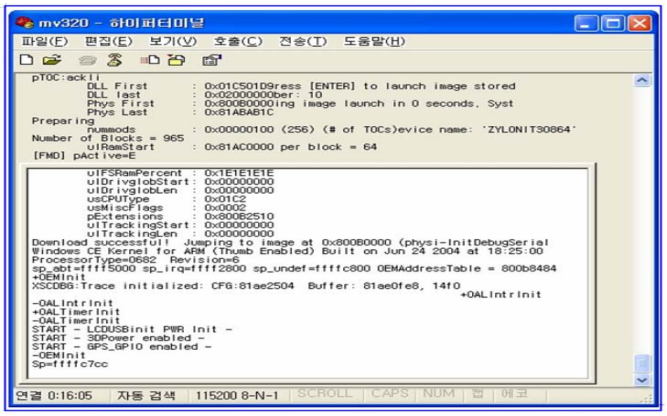
<Figure 7> composed sdk file

4) The development tools(EVC4.0) setting. Figure 8 shows an application test in an EVC4.0
Development Tools: WINCE (Platform Builder) and EVC(Embedded Visual C++), etc…



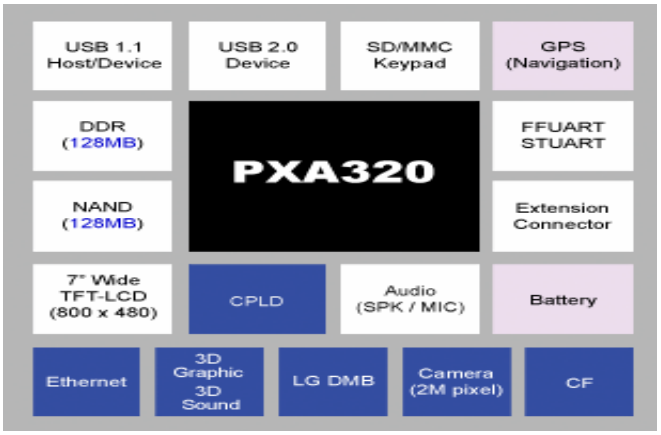
<Figure 8> EVC4.0 test

5) The communication setting between a pc and a target board and boot-loader setting.
6) The execution binary file setting in A Target board(Ethernet or usb communication use). Figure 9 shows NK.bin file downloading within a target board

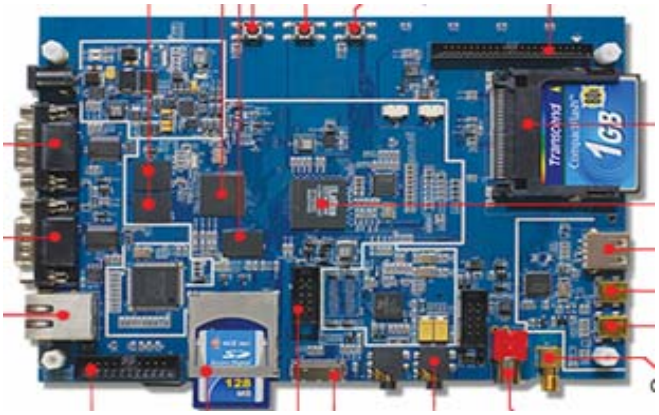


<Figure 9> NK.bin file downloading

7) An application test of target board.
Figure 10 and 11 show the specification of target board and the hardware, respectively.



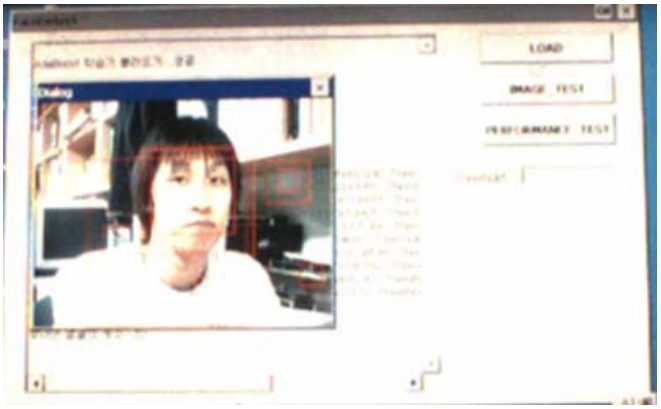
<Figure 10 > A specification of target board



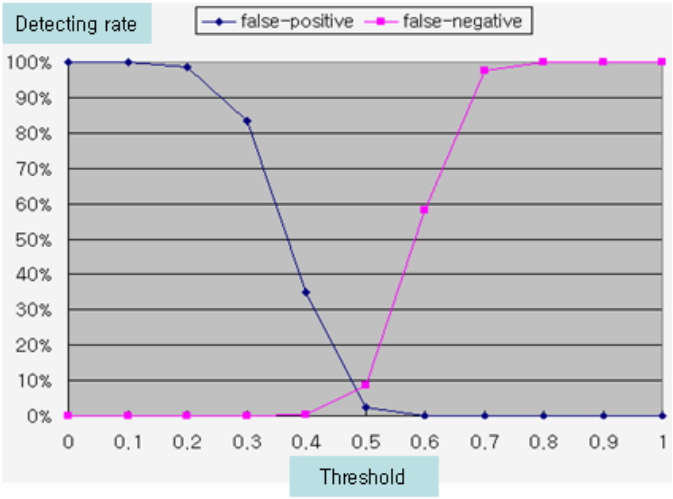
<Figure 11> A hardware of a target board

6 Conclusion

We propose a system of face detection in embedded environment. The difference of extracted detection results of a embedded system and non-embedded system doesn't exist. The difference is constituting the environment and the computing speed.
Figure 12 shows an application of face detection for the embedded system. The characteristics of this application is decided according to the class of prepared target board.
Figure 13 shows detecting rate by threshold.



<Figure 12> An example of an application in a Embedded system



<Figure13> The Detecting rate by threshold

Reference

[1] Yoav Freund and Robert E. Schapire, " A short Introduction to Boosting", Journal of Japanese Society for artifical Intelligence, Sep. 1999

[2] Paul Viola and Michael Jones, "Robust Real-time Object Detection" , Second International Workshop on Statistical and computational Theories of Vision-Modeling Learning Computing and Sampling July 2001

[3] Paul Viola and Michael Jones, "Rapid object detection using a boosted cascade of simple feature", Conference on Computer Vision and Pattern Recognition, 2001

[4] 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

[5] C. Papageorgiou, M. Oren, and T. Poggio. A general framework for objection detection. In International Conference on Computer Vision, 1998.

[6] Robert E. Schapire, Yoav Freud, Peter Bartlet, and Wee sun Lee. Boosting the margin: A new explanation for the effectiveness of voting methods. In proceedings of the Fourteenth International Conference on Machine Learning, 1997.

[7] Yong Ma, Xiaoqing Ding."Robobust Real-time Face Detection Based on Cost-Sensitive AdaBoost Method", 2003 IEEE