International Journal of Engineering Research in Electronic and Communication Engineering (IJERECE) Vol 3, Issue 5, May 2016

Optimized Face Detection on FPGA

[1]Sanmati Kuthale, [2] Sadashiva V Chakrasali [3] A.R.Priyarenjini
[1][2][3] Dept. of Electronics & Communication
M. S. Ramaiah Institute of Technology, Bangalore, India
[1] sanmatiak55@gmail.com, [2] sadashiva.c@msrit.edu [3] arp@msrit.edu

***Abstract*— This paper gives the hardware implementation of face detection on FPGA using HAAR features. The design consisting of integral image generation which is used to compute the HAAR features at a faster rate, has been illustrated. The classifiers are built using the Ada Boost algorithm which selects a minimum number of critical HAAR features from a very large set. Also, parallel processing classifiers increase the speed of the face detection system. The described detection architecture has been designed using Verilog HDL and implemented on Xilinx vertex-5 FPGA which shows optimization in terms of area and speed.**

***Index Terms*— Ada Boost, HAAR classifier, FPGA, Face detection, Real time.**

I. INTRODUCTION

The arena is turning into more and more protection conscious; human beings are seeking out new procedures for safety which might be more reliable and proper. Regarding this, dependable individual authentication strategies play a great role in everyday actions. In security systems, legitimate users must be allowed for access with better accuracy. At the same time, the unauthorized clients should be denied. Examples of such applications include e-commerce, biomedical image analysis, smart rooms etc. Determination of an individual's characteristics is one of the critical tasks in any identity supervision system. Sometimes a unibiometric system fails to authenticate the legitimate users due to lack of information. This is solved by using multibiometrics system consisting of many algorithms such as face recognition, speech recognition, iris recognition etc. Due to its increasing security level the multibiometrics system has got more importance in recent years.

Face recognition [1] is one among the multimodal system, in that face detection is the first stage and is very important. The existing literature on face detection in real time gives only software implementation but some of the papers have presented hardware design. N.Vijaykrishnam et.al [2] presented the performance of face detection, based on neural network in an Application Specific Integrated Circuits (ASIC). However, VLSI technology is not reconfigurable which means, if we want to change the small function in the design then the entire architecture needs to be changed. So it requires large development time and cost. Yang et al. [3] implemented face detection using cyclone II FPGA with low detection rate. Gao et al. [4] presented HAAR feature classifier based face detection in which each stage consist of 16 classifiers and it is implemented using Vertex-5 FPGA (XC5VLX155-2 FF1153) because of large design size and also there is no explanation about experimental results and analysis of the implemented system. Junguk Cho et al.[5] implemented the face detection system using the same algorithm as used by Gao et al and also discussed the experimental result.

In this paper, the discussion of feasible hardware architecture for face detection system in real time using Viola and Jones method to enhance the speed of detection system, has been illustrated. The proposed architecture generates an integral image window, which is used to extract the HAAR features, the generation of integral image is different as compare to the design present in the previous papers, this causes the reduction in the number of slice Look-Up Tables (LUT), which intern reduces the area and then feature classification has been done in parallel using classifiers to find whether a given image sub-window is face or not. Therefore, this proposed work of a real-time face detection system is designed using Verilog HDL and implemented on Xilinx vertex-5(XC5VLX155-2 FF1153) FPGA.

The paper is prepared as follows Section II describes the algorithm used for face detection. Section III presents the proposed face detection block diagram. Section IV describes experimental results and concluded in Section V.

II. FACE DETECTION ALGORITHM

The location of a face in an image used by Viola and Jones is used in our proposed design. In this work all the possible features are extracted using integral image window for different faces and average of those features are stored in

memory location. The algorithm uses cascade structure to eliminate the non-face candidates at a faster rate. Each stage consists of a lot of HAAR features and is classified by classifier. The output of all the classifiers is given to stage comparator, it sums all the outputs and compare with threshold value to decide whether the stage is passed or not. The image sub-window is considered as face if it passes all the stages. The following sections give the detailed information about these terms.

A. Integral Image

Integral image value at any location (x, y) is defined as the sum of pixels which are present above and to the left of that location including with the pixel present at $(x, y)^{th}$ location. Figure 1. Illustrates how to find the integral image window for the given image.

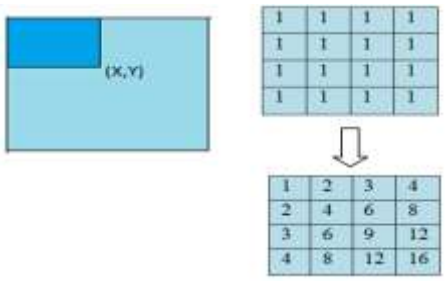
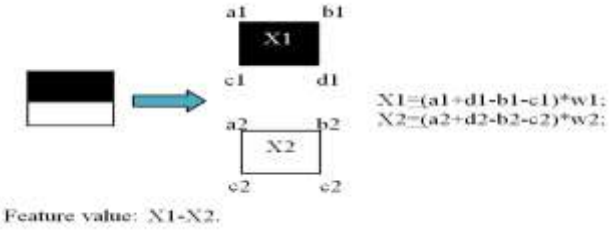


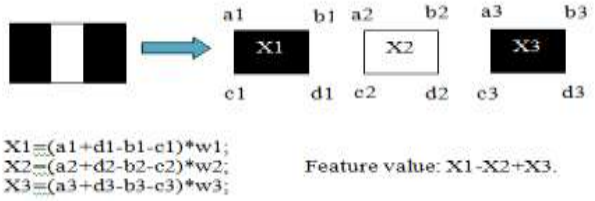
Figure 1. Generation of integral image for (4*4) image window

B. HAAR Features

Feature 1.



Feature 2.



Feature 3.

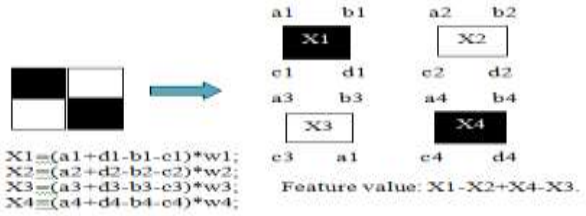


Figure 2. Example of HAAR features.

All human faces have some related features such as the eye region is brighter than the upper cheeks and the nose bridge region is brighter than the eye. This information is used to construct certain features known as HAAR features. These features are used to construct many stages and number of features in each stage is not restricted. Faces are scanned and searched for HAAR features of the current stage which consist of two or three rectangles. Figure 2 shows the considered HAAR features. Each HAAR feature has a value that is computed by multiplying weight with the area of each rectangle separately and then summing the results. Weights are calculated from the AdaBoost [6] algorithm. Using integral image, the area of the rectangle is computed at faster rate. Figure 3. Shows the procedure for calculating the area of rectangle.

C. HAAR Feature Classifier

HAAR classifier is used to construct the cascade structure, shown in Figure 4. This has been done by classifying the HAAR features using training set of AdaBoost.

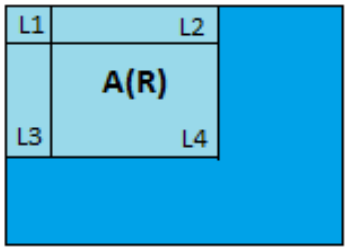


Figure 3. Calculating the area of rectangle i.e

A(R) =L1+L4-L2-L3

D. Cascade.

The Viola and Jones based Face detection system uses cascade structure shown in Figure 4 which is used to eliminate the testing sub-window at a faster rate by making stricter requirements at the later stages so that it is very difficult to pass all the stages for this sub-window, if it meets all the requirements present in all the stages then it is considered as face otherwise it is rejected in any intermediate stage.

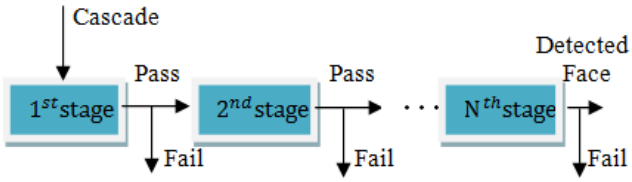


Figure 4. Cascade of stages

III. IMPLEMENTATION OF FACE DETECTION SYSTEM

A. Integral Image Generation

The computation of integral image window has been done using equation (1) and is derived from the block diagram shown in Figure 5.

$$ii(x,y) = \sum_{x' \le x, y' \le y} i(x', y') \qquad (1)$$

Where $i(x,y)$ is the input image pixel value at location (x,y) and $ii(x,y)$ is the value of integral image window at the same location (x,y).

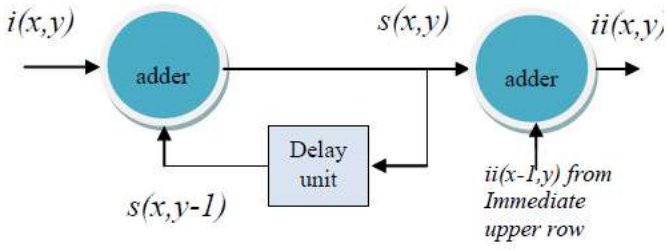


Figure 5. Block diagram for integral image generation.

The following modified equations are used to calculate the integral image for 2*2 matrixes in proposed design without performing image line buffer and image window buffer [5], which causes the reduction in the number of LUT's.

$$ii(0,0) = i(0,0) \qquad (2)$$

$$ii(0,1) = i(0,0) + i(0,1) \qquad (3)$$

$$ii(1,0) = i(0,0) + i(1,0) \qquad (4)$$

$$ii(1,1) = ii(0,1) + i(1,0) + i(1,1) \qquad (5)$$

The following equations are used to reduce the number of addition operations.

$$s(x,y) = i(x,y) + ii(x,y-1)$$

$$ii(x,y) = ii(x-1,y) + S(x,y-1)$$

The design of integral image generation for (256*256) is different in proposed system. The computation of integral values for first 256 i.e. 0 to 255 is performed, and these results are used instead of using input data for computation of next row i.e. 256 to 511 and so on. This results in the reduction of number of LUT slices shown in Table 1, hence improved the speed.

B. HAAR Features And Classifiers.

The considered HAAR features are shown in Figure 2. The size of each HAAR features are changed further so total number of HAAR features are 6402 for each (20*20) sub-window. Each HAAR feature has its own threshold and weight value obtained from AdaBoost [6] algorithm using MATLAB. These HAAR features are classified and stored in 21 stages to increase the speed of detection system using Classifier. Figure 6. Shows how to calculate the HAAR feature for rectangular window using width, height, weight. The integral version of each (20*20) input sub-window is given to the classifier shown in Figure 9 Consisting of 21 stages. This sub-window is given to next stage if it satisfies the requirement for both feature threshold and stage threshold else it is rejected at the current stage and considered as not a face. If this sub-window passes all the 21 stages then it is considered as face. The (X, Y) positions for all the sub-windows which are detected as faces are calculated from the Verilog output, and these positions are used to draw the square boxes on face using MATLAB shown in Figure 12.

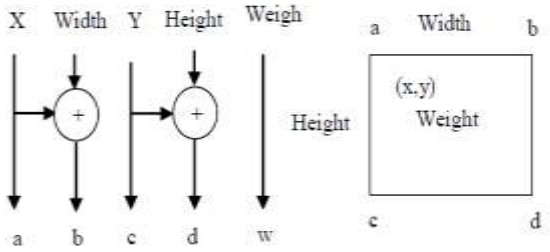


Figure 6. HAAR feature calculation of rectangular window

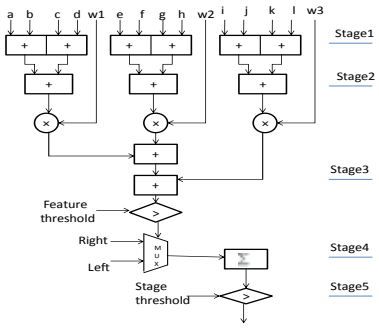


Figure 7. Architecture for performing the HAAR feature classifier.

IV. EXPERIMENTAL RESULTS

The results of the study are compared with Junguk et al.[5] work and it is examined. The implemented architecture is programmed to use in field programmable gate arrays (FPGAs).

For the shown (Figure 8) the face detection architecture inputs are address_i (4:0), address_j (4:0), input_image (7:0), IN_SCALE (3:0), pos_x (9:0), pos_y (8:0), CLK, CLK3x, enable, in_valid, out_ready, read_write, reset, and the outputs will be Detected, Failed, in_ready, out_valid, shift.

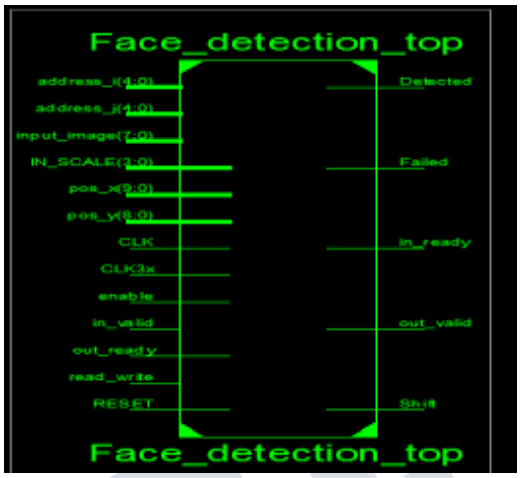


Figure 8. shows the face detection schematic.

Comparison of the results obtained by Junguk et al. [5] and proposed system (from figure 8) shown in Table 1.

Number of LUT slices / BLOCKS	Junguk et al.[5]	Proposed architecture
Integral Image	18038	3570 (80% reduction in area)
Feature Classifier	18297	13182 (28% reduction in area)
Total Classifier Module	62890	16076 (74% reduction in area)

The above table shows the comparison results for area optimization by reducing the number of LUT (look up-table) slices. From the above Table it is examined that the implemented architecture with Modified integral image generation has optimized the area than Junguk et al. [5] for face detection system, when synthesized on Virtex 5 FPGA (XC5VLX155-2 FF1153).

Feature_extration_classification Project Status			
Project File:	face_detection.xise	Parser Errors:	No Errors
Module Name:	face_detection_top	Implementation State:	Synthesized
Target Device:	xc5vlx155-2ff1153	• Errors:	
Product Version:	ISE 14.7	• Warnings:	
Design Goal:	Balanced	• Routing Results:	
Design Strategy:	Xilinx Default (unlocked)	• Timing Constraints:	
Environment:	System Settings	• Final Timing Score:	

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slice Registers	15289	97280	15%
Number of Slice LUTs	16076	97280	16%
Number of fully used LUT-FF pairs	688	30953	2%
Number of bonded IOBs	66	800	8%
Number of Block RAM/FIFO	4	192	2%
Number of BUFG/BUFGCTRLs	3	32	9%
Number of DSP48Es	3	128	2%

Figure 9. Device utilization for face detection system.

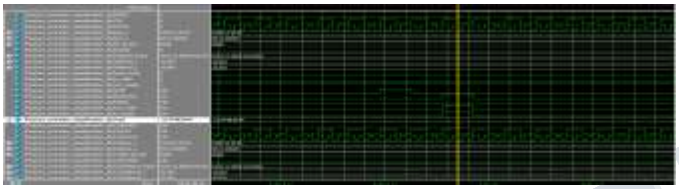


Figure 10. Shows the simulation module output for face detection.

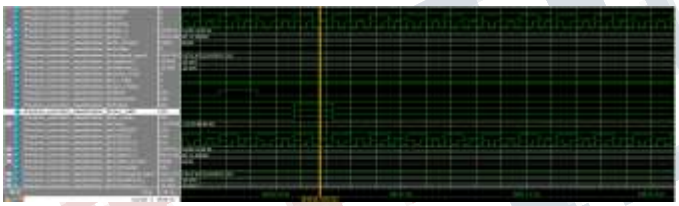


Figure 11. Shows the simulation module output for face not detected

Figure 12. Experimental results for face detection system.

V. CONCLUSION

In this paper area, optimization is carried out on the face detection system. The implemented architecture is programmed to use in field programmable gate arrays (FPGAs). The target FPGA used is vetex-5 FPGA (i.e., XC7A100T, CSG324), which includes DSP48E blocks (performs mathematical operations). The design is implemented at same synthesis and FPGA condition with Junguk et al. [5] Design (synthesized on Virtex-5 FPGA (XC5VLX155-2 FF1153)). From the comparison results for area optimization, it is examined that the implemented architecture exhibit optimized area.

REFERENCES

[1] Janarbek Matai, Ali Irturk and Ryan Kastner "Design and Implementation of an FPGA-based Real-Time Face recognition System".2010.

[2] T. Theocharides, N. Vijaykrishnam, and M. J. Irwin, "*A parallel architecture for hardware face detection*," In Proceedings of IEEE Computer Society Annual Symposium Emerging VLSI Technologies and Architectures, pp. 452-453, 2006.

[3] M. Yang, Y. Wu, J. Crenshaw, B. Augustine, and R.Mareachen, "*Face detection for automatic exposure control in handheld camera*," In Proceedings of IEEE international Conference on Computer Vision System, pp.17,2006.

[4] C. Gao and S. Lu, "*Novel FPGA based HAAR classifier face detection algorithm acceleration*," In proceedings of International Conference on Field Programmable Logic and Applications, 2008.

[5] Junguk Cho, Shahnam Mirzaei, Jason Oberg, Ryan Kastner , " *FPGA-Based Face Detection System Using HAAR Classifiers*" FPGA'09, February 22-24, 2009.

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