ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology
Volume 1, Issue 6, August 2012

ADVANCED HARDWARE ARCHITECTUREFOR ADABOOSTBASEDREALTIMEOBJECT DETECTION

Polani Devi Vara Prasad, Janakani Devi Vara Prasad, Malijeddi Murali

***Abstract*—In this paper, we present a work of new hardware architecture to a Flexible Parallel Hardware Architecture for AdaBoost-Based Real-Time Object Detection. Here, we anticipate that optimizations in terms of power consumption will significantly improve the architecture. Because of optimization of architecture, we get a new hardware, which consumes less amount of power than previous hardware architecture.Real-time object detection is becoming necessary for a wide number of applications related to computer vision and image processing, security, bioinformatics, and several other areas. Existing software implementations of object detection algorithms are constrained in small-sized images and rely on favorable conditions in the image frame to achieve real-time detection frame rates. Efforts to design hardware architectures have yielded encouraging results, yet are mostly directed towards a single application, targeting specific operating environments. Consequently, there is a need for hardware architectures capable of detecting several objects in large image frames, and which can be used under several object detection scenarios.In this work, we present a generic, flexible parallel architecture, which is suitable for all ranges of object detection applications and image sizes. The architecture implements the AdaBoost-based detection algorithm, which is considered one of the most efficient object detection algorithms. Through both field-programmable gate array emulation and large-scale implementation, and register transfer level synthesis and simulation, we illustrate that the architecture can detect objects in large images (up to 1024× 768 pixels) with frame rates that can vary between 64–139 fps for various applications and input image frame sizes.**

Index Terms—Object detection, systolic arrays, VLSI.

***Polani Devi Vara Prasad**, M.Tech VLSI 2'nd Year Student, (Electronics and Communications Engineering), Andhra University/ Sanketika Vidya Parishad Engineering College. Vizianagaram, India, +919492814240.*

***Janakani Devi Vara Prasad**, Assistant Professor, Electronics and Communications Engineering, Andhra University/Sanketika Vidya Parishad Engineering College, Visakhapatnam, India, +919985373908.,*

***Malijeddi Murali**, Professor & HOD, Electronics and Communications Engineering, Andhra University/Sanketika Vidya Parishad Engineering College, Visakhapatnam, India, +919440149970.*

I. INTRODUCTION

Object detection in video and images is an important operation in several embedded applications, such as computer vision and image processing applications, bioinformatics, security, and artificial intelligence. Object detection involves the extraction of information from an image (or a sequence of) frames, processing of the information, and determining whether the information contains a particular object and its exact location in the image. This process is computationally intensive, and several attempts have been made to design hardware-based object detection algorithms, especially in the context of embedded and real time systems [1]-[5], [8]-[12], [16], [18]-[24], and [28], [29]. This is particularly emphasized in safety-critical applications such as search-and-rescue operations, biomedical applications (such as laparoscopic surgeries), surveillance of critical infrastructure, and several other applications. The majority of the proposed works target field-programmable gate- array (FPGA) implementations; additionally, they are either application-specific or operate on images of relatively small sizes in order to achieve real-time response [8], [16], [17]. As such, generic, real-time object detection hardware architecture, independent of image sizes and types of objects, can potentially benefit several applications, and most importantly, provide the foundations for further post detection applications such as object recognition.

There are several algorithms used to perform detection, each of which has its own advantages and disadvantages. This paper presents a generic architecture based on the object detection framework presented by Viola and Jones [6] where they utilize the AdaBoost learning algorithm introduced by Freund and Schapire [7], [13]. The proposed architecture extends our preliminary work proposed initially in [8]. In this work, we extend the implementation of the AdaBoost detection framework by several algorithm-driven design optimizations, focus on the general object detection problem, and address the image size limitations. We also expand our evaluation strategy to include both an FPGA implementation for the purposes of validation of

All Rights Reserved © 2012 IJARCET

ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology
Volume 1, Issue 6, August 2012

our architecture, as well as an application-specific integrated circuit (ASIC) implementation, for which we evaluate based on three different object detection case studies. The architecture proposed in this work is based on a massively parallel systolic computation of the classification engine using a systolic array implementation which yields extremely high detection frames per second (fps). The architecture is designed in such a way as to boost parallel computation of the classifiers used in the algorithm, and parallelize integral image computation, reducing the frequency of off-chip memory access. To make the architecture scalable in terms of image sizes, we utilize an image pyramid generation module in conjunction with the systolic array. As the array elements are modular and simple, and communication is regular and predetermined, the architecture is highly scalable and can operate on high frequency. The designer can select all the appropriate design parameters with the targeted operating environment in mind, without affecting the real-time constraints. The designer can also choose the operating frequency (with power constraints in mind), the array size (with area constraints in mind), and image size (with targeted application specifications in mind). The architecture is flexible as well in terms of input image size; the maximum input image size depends on the silicon budget available, however smaller images may easily be processed by the system as the input image size can be loaded as a parameter. Moreover, the architecture can support different training sets and different training set formats.

The architecture is evaluated by verifying its operation on a Xilinx Virtex II Pro FPGA, and by synthesizing and implementing the architecture using Synopsys Design Compiler and a commercial CMOS 65-nm cell library.

II. PARALLEL ARCHITECTURE

Face detection is a very important application in the field of machine vision. In this paper, we present a scalable parallel architecture which performs face detection using the AdaBoost algorithm. Experimental results show that the proposed architecture can detect faces with the same accuracy as the software implementation, on real-time video at a frame rate of 52 frames per second.

It is evident that we need to provide parallel data access to the integral and integral squared images. As such, we propose the use of a grid array processor, as the structure of our architecture. The array is used as memory to store the computation data and as datatransfer unit, to aid in accessing the integral image in parallel.

Essentially the system consists of three major components: the collection and data transfer units (CDTUs), the multiplication and evaluation units (MEUs) and the control units (CUs). The CDTUs serve as data storage elements for the integral image. In ourarchitecture, the size of the array depends on the size of the image. An image size of mxn pixels requires an mxn array of CDTUs.

While this might seem that the architecture requires excessive resources, techniques such as image scaling can be used to reduce the input image frame and thus keep the size of the array within reasonable limits. Additionally, image partitioning can be used in conjunction with image scaling, to produce image sizes to fit the hardware budget. A floorplan of the system is shown in Figure 1, illustrating the location of each unit and the data movement across the system. Due to space limitations, we omit the detailed description of each unit;

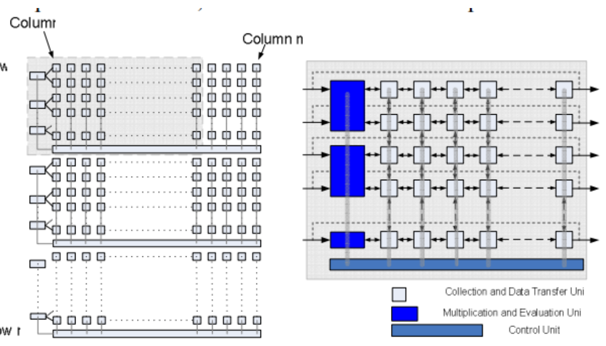

Columr
Column n
Row
Row r
Collection and Data Transfer Uni
Multiplication and Evaluation Uni
Control Unit

Figure (1) Architecture Floatplane

The operation essentially is partitioned into the following stages: configuration, computation of integral and integral squared images, and computation of image variance, computation ofrectangles per feature, feature computation, stage evaluation and image evaluation. When a feature size increases (i.e. when a search window size increases) the computation is essentially treated as a new one. When the image has been searched at all search window sizes, the system is ready for the next image frame. In each case, all three units collaborate to perform the computation. Incoming pixels stream in the processor in parallel along all rows of the processor and are shifted in row-wise every cycle. First, the integral image is computed. The computation consists of horizontal and vertical shifts and additions. Incoming pixels are shifted inside the array on each row. Depending on the current pixel column, each of the computation units performs one of three operations; it either adds the incoming pixel value into the stored sum, or propagates the incoming value to the next-in-raw processing element while, either shifting and adding in the vertical dimension (downwards) the accumulated sum or simply doing nothing in the vertical dimension. The entire computation takes 2 * [(m + (m-1) + (n-1)] cycles, for an input image of n rows by m columns. Next, the rectangle computation happens. For each rectangle in each feature, each corner point is shifted towards the collection point. The points move one at a time, but in parallel for all rectangles in the array. At each collection point, the point is either added or subtracted to anaccumulated sum, with the rectangle value computed when allpoints of each rectangle arrive at the collection point. Each point requires dx +dy cycles to reach the collection point, where dx anddy are the offset coordinates of the point with respect to the upper left corner of the search window. When finished collecting the rectangle sums for a single feature, the collected sums are stored in the CDTU that represents the starting corner for each feature. Next, all the collected sums are then shifted

All Rights Reserved © 2012 IJARCET

leftwards towards the MEUs, one sum at a time per MEU. From left to right, eventually all sums arrive in each MEU, where the rectangle sums are used with the training data of each feature, in order to evaluate the feature. The feature result is shifted in a toroidal fashion to the CDTU on the far right of the grid, to continue the computation. Eventually, when all feature results are computed, they are stored back into the CDTUs in the grid and the computation resumes with the next feature. When all stages complete for a single scale, the flagged locations contain a face. If the scale computed is the last one, the computation ends, and each search window with a face still has its flag bit set inside the representing CDTU. Each location that contains a face is shifted to the right and outside of the grid array,to the output of the processor for the host application to proceed.

Face detection is widely used in various computer vision applications, such as face recognition, content-based image retrieval, video surveillance, and human-computer interface. However, detecting faces in real-time is challenging since the human face is a dynamic object which has huge varietyof instances. Many different approaches to face detection have been conducted in the past decades. Generally,statistical face detection methods, such as SVM and adaboost,are widely used because of their robustness and computational efficiency. However, practical use of pure software-based face detection remains difficult. This is because conventional face detection algorithms are carriedout by repeated downscaling and searching for all possibleface candidates until the downscaled image is smaller than the processing window. This feature computation consumes almost all of the computing power of conventional processors and requires frequent memory access. As a result, deployment and real-time processing of facedetection in an embedded environment becomes difficult.For this reason, several ap proaches to face detectionhardware implementations ha ve been conducted using FPGAs and reconfigurable multi-processor platforms . In particular, FPGA-based systems show significant improvements compared to previous systems, as a result of the rapid development of pr ogrammable devices in recentyearsproposed an FPGA-based haarclassifier face detection algorithm accelerator and implemented it with a Xilinx Virtex-5 LX110T FPGA. Theproposed system can process 256×192 images at 37 fps(frames per second) with 1-classifier and 98 fps with 16-classifiers.We developed a cost-effective facedetection system based on a low-cost FPGA prototype board from Altera (DE2 board). They proposed an area efficient modular architecture for the Vi ola-Jones face detector with 320×240 video streams and a minimum processing rate of 30frames/sec. One person designed FPGA hardware architecture for high frame rate face detection using featurecascade classifiers. The proposed architecture is verified using the Xilinx Virtex-2 Pro 30 platform, achieving 143 fpsfor an image size of 640×480 pixels using a single scan window when running at 126 MHz frequency. Another one proposed a unified stream ing architecture for real-time face detection and gender classification and implemented iton the Xilinx Virtex-4 FX12 FPGA. The system processes 320×240 images and runs at about 52 and 175 frames/sec in the worst and typical cases, respectively.Although considerable progress has been made, face detection systems developed so far can only partially satisfyrequirements for accuracy, sp eed, and area-efficiency.Systems have a relatively high FAR (false acceptance rate)caused by a small number of classifiers ,employ anunrealistically high pixel offset of 10 and processimages which are of relatively small size . In thispaper, we propose a dedicated hardware architecture for areal-time face detection algorithm that proposed. By employing FCM in the post-processing step,the proposed method successfully reduced the FAR toaround one-tenth that of the existing cascade adaboost etector while maintaining a comparable detection rate. We designed an algorithm and implemented the entire procedure used to detect the face in to a single FPGA, includingpyramid scaling, LBP (local binary pattern) transform, multi-stage classifier cascade, and FCM-based post-processing. Byeliminating the asymmetry between image capturing andface detection through fully parallelized processing of pyramid images, the sequential bottleneck caused byrepetitive feature evaluation is removed.

III. FPGAHardwareArchitecture

FACE Detection is the Process of finding all possible faces a given image or a video sequence.More precisely,face detection has to determ inethelocations andsizes of human faces.It is the essential first step toward many advanced computer vision, biometrics recognition and multimedia applications; such as face tracking, face recognition, and video surveillance.Due to factors such as scale, rotation, illumination variation,face detection involve smany research challenges. How to detect different size faces, howto berobust to illumination variation, how to achieve high detection rate with low false detection rate?These are only a few of the challenges a face detection algorithm needs to consider.

Facedetection techniques have been developed for many years and significant progress has been proposed in research literature. Most of the face detection methods focus ondetecting frontal faces with good lighting conditions.According to Yang's survey,face detection methods can be categorized into four types: knowledge-based, feature in variant,template matching and appearance-based.

Knowledge-based methods utilize human-coded rules to model facial feature such as the presence of two symmetric eyes, Feature invariant methods try to find facial features which are invariant to pose, lighting condition or rotation.Examples includes skin colors, edges and shapes. Template matching methods calculate the correlation between test images and pre-selected facial templates .The lastcategory, appearance-based, adopts machine learning technique to extract discriminative features from apre-labeled training set.The Eigenface is the most fundamental method of this category. Recent proposed face detection algorithms include support vector machines neuralnetworks ,statistical classifiers and AdaBoost face detection

All Rights Reserved © 2012 IJARCET

[1].Thestate-of-the-art face detection algorithms interms of both performance and speed are the approaches using different flavors of AdaBoost-type training methods for building a cascade of weak classifiers that will yield stronger classifiers for face detection. Usingthis approachwith theintegral image representation for evaluating the weak classifiers fast, can yield a face-detection speed of nearly real-time (15-25fps). For an image with 384x288 pixel resolution, are ported speed of 15 fps can be achieved using PIll 700MHz[.However, this is based on occupying all CPU computational power for this task only and face detection is just the first pre-processing step in many face recognition application applications. More CPU computational power should be reserved for further processing allocated for pose correction, feature extraction and recognition.

Furthermore, with the progressing development ofnewCCD image sensors, higher resolution images can becaptured with higher frame-rates which have many applications in computer vision andsurveillance..Although onec an suggest to simply downscale input images to smaller size and obtain the desired processing speed, this will sacrifice the detection rate of small faces as one can detect faces at larger stand-offs compared to low-resolution video.faces with size of 20x20 pixels, applying any downscale operation will scalefaces sizeto smaller than 20x20pixels,thus make a face detection algorithm targeting sub-window size of 20x20 fail finding these detections. Unless the algorithm up samples to locate smaller faces, but then poor and noisy images will be difficult to detect low-resolution faces. Therefore, arapidface detection architecture with out sacrificing any detection hitor miss-rateis needed.

The contribution of this paper is that, based on a variation of the AdaBoost trained facedetectiontraining algorithm,it introduces an ovelfast EllWarchitecture, includingaspeciaRAMstructureandapipedregistermodulewhi challowsfastface detection feasible to extremelyhigh framerates which allow it to tack lehigh resolution imagea pplications.Intheremaining of this paper, SectionII briefly reviewsAdaBoostbasedface detection algorithm andrelated work found in literature both in software implementations and previous hardware designs. Details of the proposed architecture are given in Section III. Section IVpresentsexperimental results and validates the performance of our architecture. Section Vsummarizes our work and outline spot ential further improvements.

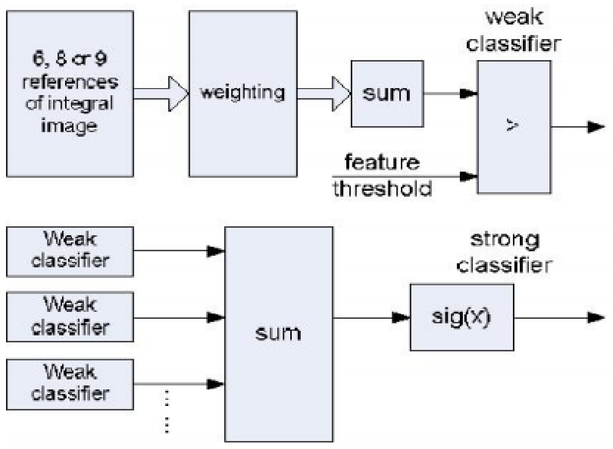
IV. PERFORMANCE EVALUATION

To evaluate the performance of the proposed implementation, we designed and verified the architecture using VHDL and Xilinx. We then synthesized the architecture using a commercial 90nm library and targeting a 500 MHz clock cycle. We use the Intel Open Computer Vision Library (CV) for the training data. We evaluated our design using sample images which contain various numbers and sizes of faces. Our synthesized design indicates that the experimental architecture consumes an area of approximately 115mm2. We computed the average number of cycles per test frame and obtained a rough estimate of 52 frames per second.

This paper can also presented a parallel architecture that performs face detection using the AdaBoost algorithm. The architecture targets both parallel computation and parallel data movement, and is capable of processing on average 52 frames per second.

The proposed hardware architecture is first implemented using VHDL and verified by the Modelsimg simulator.After successful simulation ,we use the XilinxISE® tool for synthesis. Thetarget platform is Xilinx XUP development board with onboard Virtex-I1ProXC2VP30 FPGA,shown in Fig. 11. This FPGA contains 2.44MBitson-chipmemory[13].Based on the synthesis result,the FPGA can operate at a clock speed of 126.8MHz.Hardware resource utilized by face detection core with 52 weak classifiers is shown inTableII(video decoder or other hardware count as extra resources but needed for areal-timedemo).

We also verify our detection on part of MIT+CMU data base[20].The detector with 52 classifiers was tested on 10 images, containing 45 faces, and the preliminary result



Fi2. 10.Blockdia2ramofclassifier.

In previous literature, frame rate is generally derived bythree parameters, the clock frequency of FPGA, the 0parameter which is the number of cycles needed forprocessing one sub-window and another parameter denoted as A which are the number of sub-windows in one frame or one image, T. Frame rate is defined as 0 /(AxT), where(AxT) is the total cycles for one frame. Using the formula,since our architecture can process one sub-window in oneclock cycle, A is there fore equals to 1. For a 640x480 image,there are totally 755699 sub-windows based on window size20 by 20, scale factor 1.25. Hence, we can achieve 132 fps when running at 1OOMHz frequency, which is fast when comparing to 15 fps for 120x120 .and 52 fps at 500 MHz.However, this formula is for ideal case. Overhead from switching sub-window and frames should be taken into consideration which may reduce this speed slightly. Based onthe simulation result, our architecture takes AxT = 876,540clock cycles for one frame, including

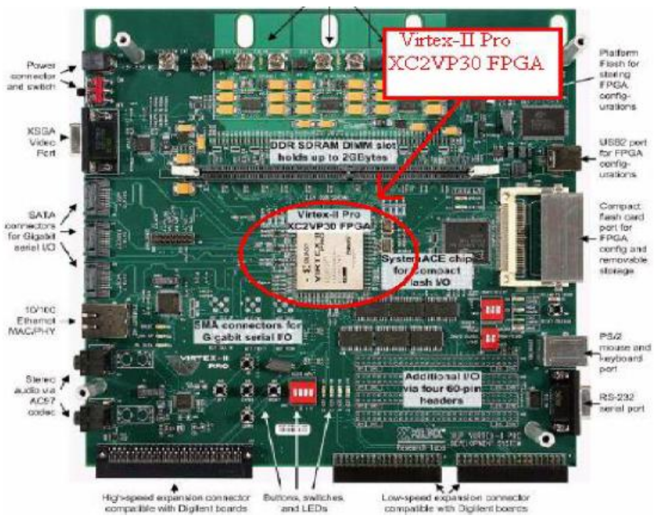
All Rights Reserved © 2012 IJARCET

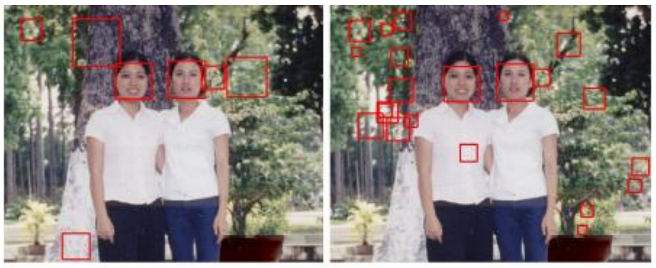
overhead. Therefore,the frame rate is 114 fps g1OOMHz and 143 fps @126MHzat our proposed architecture.

Since each classifier is evaluated parallel, increasingnumber of classifiers (for more robust trained face detectors)will not affect our fast face detection speed but will requiremore hardware resource (FPGA gates). Therefore, there is atrade off between of number of features and hardware cost.Right now, we just show that one can implement 52classifiers to verify our architecture and about 100 classifierscan be theoretically added to the current XC2VP30 FPGA.However in practice resources for interfacing to camera anddisplay will take FPGA logic and thus the number ofclassifiers will decrease, thus we require a larger FPGA suchas the XC2VP100 where we can add more than 200classifiers to FPGA to improve the correct detection rate.Additionally, based on our preliminary results, we could havefewer number of classifier by adjust training parameters

TABLE II
RESOURCE REQUIREMENT

Resource Available	Resource Used	Percentage
Slices (13696)	11505	84%
Slice Flip Flops (27392)	7872	28%
4 input LUT (27392)	20901	76%
BRAM (136)	44	32%
GCLKs (16)	1	6%


Virtex-II Pro
XC2VP30 FPGA
Power connector and switch
XSGA Video Port
SATA connectors for Gigabit serial I/O
10/100 Ethernet MAC/PHY
Stereo audio via AC97 codec
Platform Flash for storing FPGA configurations
USB2 port for FPGA configurations
Compact flash card port for FPGA config and removable storage
PS/2 mouse and keyboard port
RS-232 serial port
DDR SDRAM DIMM slot holds up to 2GBytes
Virtex-II Pro XC2VP30 FPGA
SystemACE chip for compact flash I/O
SMA connectors for Gigabit serial I/O
Additional I/O via four 60-pin headers
High-speed expansion connector compatible with Digilent boards
Buttons, switches and LEDs
Low-speed expansion connector compatible with Digilent boards
Fig. 11. Xilinx Virtex-I1 Pro system board



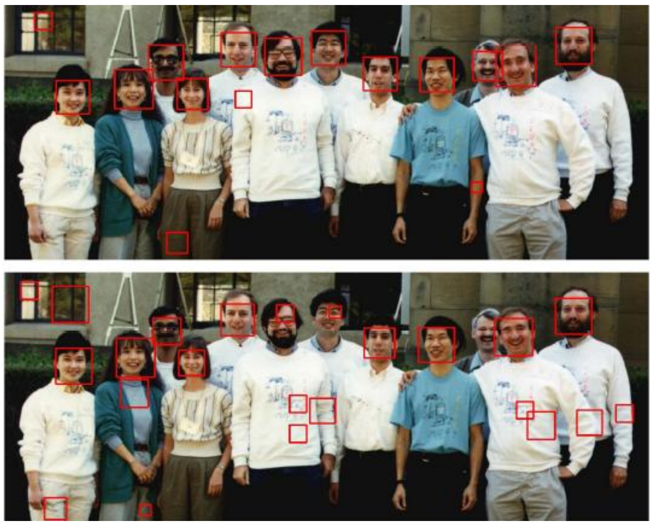
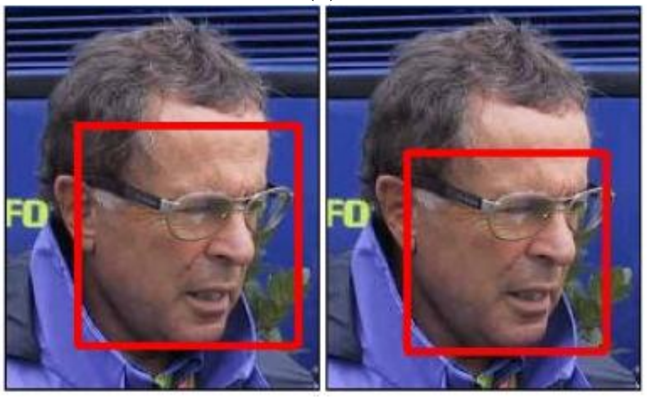


Fig. 12. (top) Original available implementation of Viola-Jones (26 stages, 2609 total features), (bottom) Our re-trained version using only 52 features

(A)



V. CONCLUSION

Object detection is an important step in multiple applications related to computer vision and image processing, and real-time detection is critical in several domains. In this paper, we are presenting a new architecture of flexible parallel for implementation of the AdaBoost object detection algorithm. The architecture combines an image pyramid generation process, along with highly parallel systolic computation, to offer a flexible design that is suitable for several types of applications and budgets.

We study and investigate the results for the previous and presented work and compared the results among with previous techniques. Further, we obtaining the results for generating the header file for an input image to use further proceeding of the work. We anticipate that optimizations in terms of power consumption will significantly improve the architecture, leaving this as immediate feature work.

All Rights Reserved © 2012 IJARCET

ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology
Volume 1, Issue 6, August 2012

VI. REFERENCES

[1] l. Dlagnekov and s. Belongie, "Recognizing cars," ucsd cse Tech. Rep. cs2005-083, 2005.

[2] f. Moutarde, b. Stanciulescu, and a. Breheret, "Real-time visual detection of vehicles and pedestrians with new efficient adaBoost features," presented at the Workshop Planning, Perception Nav. for Intel. Veh. (ppniv) Int. Conf. Intel. Robots Syst. (iros), Nice, France, Sep. 2008.

[3] x. Tang, z. Ou, t. Su, and p. Zhao, "Cascade AdaBoost classifiers with stage features optimization for cellular phone embedded face detection system," in Proc. ICNC, 2005, pp. 688-697.

[4] y. Abramson and b. Steux, "Hardware-friendly detection of pedestrians from an on-board camera," presented at the ieee Intel. Veh. Symp. (iv), Parma, Italy, Jun. 2004.

[5] c. Yoon, m. Cheon, e. Kim, m. Park, and h. Lee, "Real-time road sign detection using adaboost and multicandidate," in Proc. 8th Symp. Adv. Intel. Syst. (ISIS), 2007, pp. 953-956.

[6] p. Viola and m. Jones, "Real-time object detection," Int. J. Comput. Vision, vol. 57, no. 2, pp. 137-154, May 2004.

[7] y. Freund and r. e. Schapire, "a short introduction to boosting," J. Japan. Soc. for Artif. Intel, vol. 14, no. 5, pp. 771-780, Sep. 1999.

[8] t. Theocharides, n. Vijaykrishnan, and m. j. Irwin, "a parallel architecture for hardware face detection," in Proc. IEEE Comput. Soc. Annu. Symp. VLSIDes. (ISVLSI), Karlsruhe, Germany, pp. 452–453.

[9] m. Hiromoto, h. Sugano, and r. Miyamoto, "Partially parallel architecture for Adaboost-based detection with Haar-like features," IEEE Trans. Circuits Syst. for Video Technol, vol. 19, no. 1, pp. 41-52, Jan. 2009.

[10] j. Cho, s. Mirzaei, j. Oberg, and r. Kastner, "Fpga-based face detection system using haar classifiers," in Proc. ACM/SIGDA Int. Symp. Field Program. Gate Arrays, New York, 2009, pp. 103-112.

[11] y. Wei, x. Bing, and c. Chareonsak, "fpga implementation of adaboost algorithm for detection of face biometrics," in Proc. IEEE Int. Workshop Biomed. Circuits Syst., 2004, pp. s1/6-17-s1/6-20.

[12] y. Shi, f. Zhao, and z. Zhang, "Hardware implementation of adaboost algorithm and verification," in Proc. 22nd Int. Conf. Adv. Inf. Netw. Appl—Workshops (AINAW), 2008, pp. 343-346.

[13] r. e. Schapire, "The boosting approach to machine learning: An overview," in MSRI Workshop Nonlinear Estimation Classification, 2002, pp. 1134-1227.

[14] T. Theocharides, G. Link, N. Vijaykrishnan, M. J. Irwin, and W. Wolf, "Embedded hardware face detection," presented at the 17th Int. Conf. VLSI Des., Mumbai, India, Jan. 2004.

[15] N. Ranganathan, VLSI Algorithms and Architectures. Los Alamitos, CA: IEEE Computer Society Press, 1993.

[16] R. McCready, "Real-time face detection on a configurable hardware system," presented at the Int. Symp. Field Program. Gate Arrays, Monterey, CA, 2000.

[17] E. Hjelmås and B. K. Low, "Face detection: A survey," Comput. Vision Image Understanding, vol. 83, no. 3, pp. 236-274, Sep. 2001.

[18] Intel Corp., Santa Clara, CA, "Intel OpenCV Library," Jun. 2009 [Online]. Available: http://sourceforge.net/projects/opencvlibrary/files/

[19] A. Price, J. Pyke, D. Ashiri, and T. Cornall, "Real time object detection for an unmanned aerial vehicle using an FPGA based vision system," in Proc. IEEE Int. Conf. Robot. Autom., May 2006, pp. 2854-2859.

[20] [P. Wieslaw, "Vehicle detection algorithm for FPGA based implementation," in Advances in Soft Computing, Computer Recognition System 3. Berlin, Germany: Springer, 2009, vol. 57/2009, pp. 585-592.

[21] M. Kolsch and M. Turk, "Robust hand detection," in Proc. Int. Conf. Autom. Face Gesture Recog., Seoul, Korea, 2004, pp. 614-619.

[22] S. Mahlknecht, R. Oberhammer, and G. Novak, "A real-time image recognition system for tiny autonomous mobile robots," in Proc. 10th IEEE Symp. Real-Time Embed. Technol. Appl., May 2004, pp. 324-330.

[23] V. Lohweg, C. Diederichs, and D. Muller, "Algorithms for hardware-based pattern recognition," EURASIP J. Appl. Signal Process., vol. 12, pp. 1912-1920, 2003.

[24] K. Khattab, J. Dubois, and J. Miteran, "Cascade boosting based object detection from high level description to hardware implementation," EURASIP J. Embed. Syst., vol. 2009, pp. 1687-3955, 2009.

[25] Y. Ming-Hsuan, D. J. Kriegman, and N. Ahuja, "Detecting faces in images: A survey," IEEE Trans. Pattern Anal. Mach. Intel., vol. 24, no. 1, pp. 34-58, Jan. 2002.

[26] C. P. Papageorgiou, M. Oren, and T. Poggio, "A general framework for object detection," in Proc. 6th Int. Conf. Comput. Vision (ICCV), 1998, pp. 555-562.

[27] Xilinx, San Jose, CA, "Xilinx University Program," Jan. 2009. [Online]. Available: http://www.xilinx.com/univ/

[28] H.-C. Lai, M. Savvides, and T. Chen, "Proposed FPGA hardware architecture for high frame rate (> 100 fps) face detection using feature cascade classifiers," in Proc. 1st IEEE Int. Conf. Biometrics: Theory, Appl., Syst., Sep. 2007, pp. 1-6.

[29] Y. Hanai, Y. Hori, J. Nishimura, and T. Kuroda, "A versatile recognition processor employing Haar-like feature and cascaded classifier," in ISSCC, Dig. Tech. Papers, Feb. 2009, pp. 148-149.

[30] [30 Compaq Corp., Palo Alto, CA, "The CACTI Toolset," 2009. [Online]. Available: http://research.compaq.com/wrl/people/jouppi/CACTI. html

[31] V. Kianzad et al, "An Architectural Level Design Methodology for Embedded Face Detection", Proceedings of the International Conference on Hardware/Software Codesign and System Synthesis, New York City, September 2005.

Polani Devi Vara Prasad received the Diploma in Electronics & Communication Engineering from SBTET,Hyderabad and B.Tech in Electronics & Communication Engineering from the Jawaharlal Nehru Technological University Kakinada(JNTUK), Kakinada, in 2006 and 2009 respectively. Currently working toward the M.Tech degree in VLSI form Sankatika Vidya Parishad Engineering College, Visakhapatnam. Fields of interest is VLSI and Communication Systems

Janakani Devi Vara Prasad received the B.E and M.Tech in Electronics & Communication Engineering. Currently working as Assistant Professor in Sanketika Vidya Parishad Engineering College. He was published " CPW FED SQUARE-SLOT ANTENNA FOR 5 GHZ WIRELESS LAN APPLICATION " in 5th "International Conference On Microwaves, Antenna, Propagation And Remote Sensing", which is held in JODHPUR on 19th – 21th December 2009.Again was Published in " Performance Evaluation Of Different Digital Modulation Schemes In AWGN Channel With Antenna Diversity Reception In Mobile Communication" "INTERNATIONAL CONFERENCE ON RESENT ADVANCES IN COMPUTER SCIENCE" .

Malijeedi Murali received the B.Tech in ECE and M.E from Andhra University. He also received MBA degree. Currently working toward the PhD in Communications Systems in Andhra University. He research has published in leading 10 National and 5 International Journals and in Conferences**.**

All Rights Reserved © 2012 IJARCET