2014 27th International Conference on VLSI Design and 2014 13th International Conference on Embedded Systems

Hardware Implementation of Real-Time, High Performance, RCE-NN based Face Recognition System

Santu Sardar & K. Ananda Babu
Scientist, Defence Research & Development Organization, India
santusardar@anurag.drdo.in

***Abstract*—Hardware implementation of a real-time, highly accurate face recognition system (FRS) is proposed in this correspondence. Face images are acquired from a CMOS sensor camera connected to Field Programmable Gate Array (FPGA) based reconfigurable hardware board using CamLink interface. We used contrast limited adaptive histogram equalization (CLAHE) for image contrast enhancement, discrete wavelet transform (DWT) to remove variable illumination & select appropriate subband and principal component analysis (PCA) with 35 principal components which is optimized for performance and speed. Finally, Restricted Coulomb Energy (RCE) based neural network (NN) classifier is used for face recognition. We have implemented the RCE based NN in FPGA and thus utilized the inherent parallelism effectively which is not possible with NN software implementation. The performance of our implementation is superior than face recognition software and hardware implementations, which are targeted to achieve higher recognition accuracy at faster rate using minimum computational resources. Our system recognizes a single image in real-time i.e. within 18 ms corresponding to 37 frames per second image capture. We have verified our proposed system with multiple standard face databases as well as using our own face data repository.**

***keywords* - FRS, CMOS, FPGA, CamLink, CLAHE, DWT, PCA, RCE, NN.**

I. Introduction

User authentication is becoming increasingly popular due to the security control requirement in identity verification, access control, and surveillance applications. Face recognition, among other conventional biometric authentication techniques is most suitable alternative because it is non-intrusive and economic with low cost cameras and embedded systems. Over the past few years, extensive research works on various aspects of face recognition by human and machines have been conducted by psychophysicists, neuroscientist and engineering scientists [1]. Automatic face recognition by embedded platform can be divided into two main approaches i.e. content-based and face-based [2]. In content-based approach, recognition depends on the relationship between human facial features such as eyes, mouth, nose, profile silhouettes and face boundary [3]. Every human face has similar facial features; a small deviation in the extraction may introduce a large classification error. Face-based approach attempts to capture and define the face as a whole [4]. The face is taken as a two-dimensional pattern of intensity variation. In this approach, face is matched with its underlying statistical regularities. Principal Component Analysis (PCA) ([4], [5]) is proved to be an effective face-based approach.

Sirovich and Kirby first proposed using Karhunen- Loeve (KL) transform to represent human faces [11]. They proposed the idea of eigenfaces. Turk and Pentland developed a face recognition system using PCA [12]. But common PCA-based methods suffer from two limitations, namely, poor discriminatory power and large computational load. The result in [12] shows that 3-level wavelet has a good performance in face recognition applications. This method works on lower resolution, 16 x 16, instead of the original image resolution of 128 x 128, thus reducing calculation complexity. Moreover, experimental results demonstrate that, applying PCA on DWT sub-image gives better recognition accuracy and discriminatory power than applying PCA on the original image.

The FRS algorithms selection, analysis and design parameter finalization was carried out in our hardware-software co-design implementation, published in [6]. Here, we focus on hardware implementation of the same to attain faster recognition time with less design resources. We have used CLAHE for adaptive local contrast enhancement for improving the local contrast of input image. The effectiveness of this module is verified in [6].

Restricted Coulomb Energy (RCE) ([7], [8]) can be used for a wide range of applications [9] primarily because it can learn any regular pattern and its training is faster than that of traditional multi-layer perceptron network. This faster learning speed comes from the fact that the neurons are connected in parallel [10]. But, software implementation of RCE based NN classifier has the problem of long training and recognition time due to the sequential nature of processing. So, feature vectors are classified using RCE based classifier implemented in FPGA where the neurons are operating in parallel, which resulted in less recognition time per image. In this case, the recognition time is independent of the number of neurons.

FPGAs have turned out to be the best option in flexible [11] digital signal processing hardware [12] where they were often applied as configurable logic cells. Complex real-time signal processing functions can be realized due to high clock speeds, Digital Signal Processing (DSP) slices and huge gate densities provided by latest FPGA devices, which are optimized for high-performance logic and DSP with low power serial connectivity. We have selected Virtex-6 SX475T FPGA for our

1063-9667/14 $31.00 © 2014 IEEE
DOI 10.1109/VLSID.2014.37

implementation [13] which is optimized for high-speed DSP and communication applications. Finally we compare accuracy and time performance of our implementation with other face recognition systems in software [14], DSP [15] and FPGA ([16], [17]).

Rest of the paper is organized as follows. Section II describes the proposed face recognition system. Architecture implementation details of the FRS is given in section III. Section IV describes the results of the hardware implementation of FRS and the last section gives the conclusion.

II. Proposed Face Recognition System

Figure 1 depicts the block diagram of the Neural Network Based Face recognition system. The system comprises of a 1 Mega-Pixel Resolution CMOS Sensor configured for frame rate 37fps. The Camera module has an inbuilt FPGA based Real-time image processing module for pre-processing and is used for face detection in our case. The 1024x1024 image from the CMOS sensor is resized to 128x128 resolution image and passed on to the Reconfigurable Embedded Hardware Platform through the CamLink interface. We propose CLAHE as image pre-processing method because this module increases the performance accuracy of the system by contrast enhancing the original images and is evident from the demonstrated results. For comprehensive description of the all the functional modules of FRS can be found in [6]. After that Daubechies-4 two dimensional DWT is performed for multi resolution analysis of the face images, remove variable illumination and select appropriate features/sub-band of the face images. Then PCA is performed for dimension reduction and feature extraction of the face images. This block gives us the feature vectors which is fed to the NN classifier for training, testing and recognition. NN is in the form of Register Transfer Logic (RTL) core implemented in Virtex-6 FPGA. The final facial verification logic is carried out in the Virtex-6 FPGA and the results of the analysis are passed to the on-chip MicroBlaze processor. The results are then captured by the Graphical User Interface (GUI) on host system for presentation to the user through the Ethernet port. The GUI also provides interface to the embedded processor functions and diagnostics supported by the embedded firmware. In addition to supporting the above image processing modules the Hardware platform uses MicroBlaze processor of Virtex-6 FPGA to run a monitor program and firmware to support board diagnosis, embedded processor functions, bitstream Configuration and Control of the various sub-modules on the board.

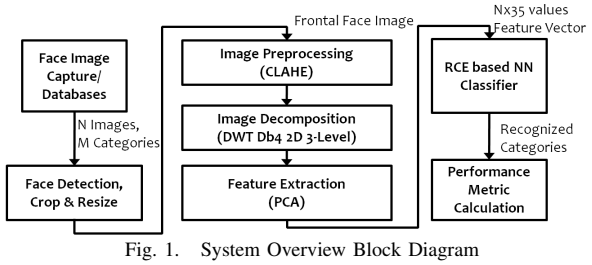

Fig. 1. System Overview Block Diagram

III. Architecture Implementation of FRS

The hardware and software architectures, FRS algorithms details and the software design are described below.

A. Hardware Architecture

The image from the CMOS sensor is resized using on camera processing and passed on to the reconfigurable Embedded Hardware Platform through the CamLink interface. The Serial data on the CamLink is de-serialized using a de-serializer and fed to Virtex-6 FPGA. The Virtex-6 FPGA hosts the image processing pipeline which includes the contrast enhancement, image decomposition and projection to the eigen-space. The eigen-face projection vector is then passed to face recognition classifier. The outputs from the classifiers are then presented in the GUI along with the input face via on board 10/100/1000 mbps Ethernet interface. The board is populated with 512Mbit of FPGA NOR memory to store the configuration images of the on-board FPGA and another 512Mbit memory to store the MicroBlaze firmware image. The Hardware block diagram is shown in figure 2.

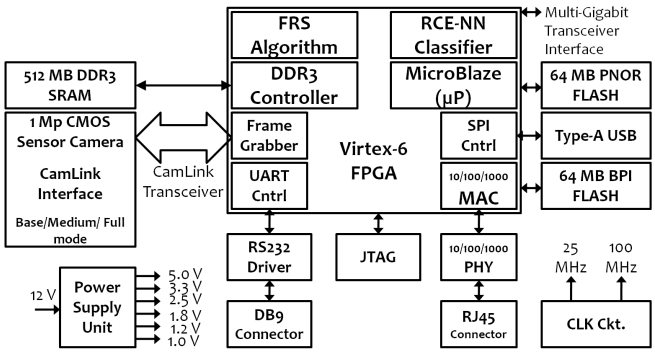

Fig. 2. System Hardware Architecture Block Diagram

The smart camera module, which is used to captures the face in steady state, feature a start-up or control switch to select the recognition. It is interfaced with the embedded hardware platform through 10 meter Camlink cable with both end terminated by 26 pin Mini D Ribbon (MDR) plug connector. CamLink consists of a transmitter, a receiver and is used to transfer digital data at a clock speed of 85 MHz. The transmitter converts 28 bits of CMOS/TTL data into four linear variable differential signal (LVDS) data streams. The data is sampled and transmitted with every cycle of the transmit clock. The receiver converts the LVDS data streams back into 28 bits of CMOS/TTL data. Using the transmit clock, 28 bits of TTL data is transmitted at 595 Mbps per LVDS channel. With four LVDS data channels the total data throughput is 2.04Gbit/s. It supports Basic, Medium, and Full CamLink specification, thus allowing up to approximately 800-900 MB/sec transfers at 85 MHz.

B. Software Architecture

1) Drivers for the Interfaces: The Tri-Mode Ethernet driver enables higher layer software (application) to communicate

to the TEMAC. The device can be configured for two major modes of operation i.e. FIFO direct, or scatter gather DMA (SGDMA). Each mode has their own frame transfer API. The Reduced-GMII (RGMII) is an alternative to GMII/MII. RGMII achieves a 50-percent reduction in the pin count, achieved by the use of double-data-rate (DDR) flip-flops and can carry traffic at 10/100/1000 Mbps. We have used RGMII for our implementation. Apart from these, we used Xilinx UART driver for our Serial Peripheral Interface (SPI). Flash driver enables higher layer software to communicate with the parallel Flash devices. This driver provide few functions like flash initialization, read, write, Erase, Reset, Lock and unlocking of the parallel flash device.

2) Web Server: On this system, the Web server is running HTTP 1.1. A file system, built using the LibXil Memory File System (MFS) library, stores the files for the Web page. The server receives requests at port 80. Every request is processed, and replies are sent by the server to the client. The Web server design running on the soft processor uses the lwIP (lightweight IP) TCP/IP stack, the Xilinx Microkernel (XMK) operating system, and the Xil MFS memory file system library. The web server running on the Hardware platform acts as server and the Host PC acts as the client. Client make requests to a server by sending messages and servers respond to the client by acting on each request and returning results. Client can able to operate in three modes of operations: Display Mode, Training Mode, Configuration mode.

3) FRS Algorithms Description: The top level block diagram of the FRS algorithm, shown in figure 3, consists of the following blocks: Block1: Frame buffers or Frame grabber, Block2: Image equalization, Block3: Image decomposition, Block4: Matrix Decomposition or Dot product computation.

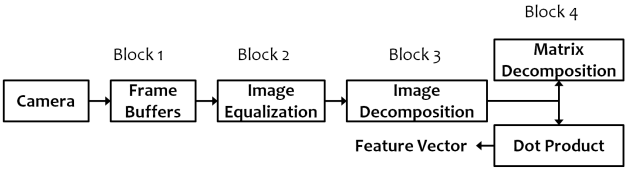

Fig. 3. FRS Pipeline Block Diagram

Frame Grabber: The input to this block is real world scene and output is 8/12 bit image. We have considered five frame buffers. This block gets a 128x128 pixel image input from a CamLink interface. Implementation assumes that only one camera will be connected to this board. It basically contains a CamLink wrapper and five 2-port memories of size 128x1024 (representing 8 bits per pixel). Each CamLink port is used to transfer 8-bit per pixel. Pixel data from CamLink interface is accumulated until 128 pixels are received, so that all 128 pixels are written in the memory in a single clock. Upon filling one frame, it is read by next block in the pipeline. The second frame data is filled in the second memory and this continues till the fifth frame buffer. Once the fifth frame is filled, this process gets repeated from first frame.

Image Equalization (CLAHE): 128X128 image is divided into 32 tiles, each tile is 32x16 pixel block. We have finalized the parameters for CLAHE implementation in accordance with [6]. For each of the 32 tiles all the below operations are performed:
Histogram: Runs on a tile which is 32x16 pixels, simultaneously 8 tiles are processed. Processed histogram function is stored in a memory with dimension 256x10; hence 32 such single port memories are needed to store 32 histogram functions. And the original 128x128 image is also stored in a single port memory 128x1024 which will be used for processing during bilinear interpolation and Wavelet decomposition.
Redistribution Histogram: Redistributing of the histogram is done as per the algorithm, and the updated histograms are overwritten in the 32 histogram memories.
Cumulative histogram calculation: Cumulative histogram function is applied on the redistributed histograms and is updated and overwritten in the 32 histogram memories.
Bilinear interpolation (BI) & Remapping pixel: Interpolation equations are applied on the original image which is stored in the single port memory 128x1024 and the calculated 32 histograms to get the interpolated 128x128 image. The single port memory 128x1024 is updated with the interpolated image.

Following are the sub blocks in Image equalization as shown in figure 4.

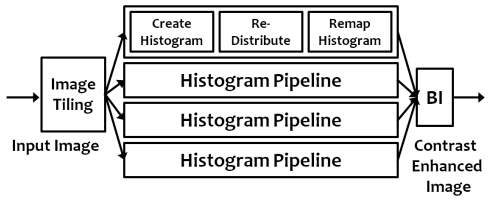

Fig. 4. Image Equalization Block Diagram

Image Decomposition with 2D-DWT: The Daubechies4 wavelet from Daubechies wavelet family is selected as it offers high number of vanishing moments that is suitable for image decomposition application. Poly-phase architecture with Distributed Arithmetic (DA) was used for this implementation as proposed by the authors in [14]. The Lifting and Lattice implementations require fewer computations than conventional poly-phase implementation. However, it can be made more efficient for long filters by incorporating techniques like Distributed Arithmetic (DA). DA is a powerful technique to reduce the size of a parallel hardware multiply-accumulate, suited to FPGA designs. Besides, it can economize the restricted memories available on FPGA (BRAM) and enhance operation speed. Also, the lattice structure cannot be used for all linear phase filters and imposes restrictions on length of filters. In poly-phase architecture, the input signal and filter coefficients are split into odd and even samples. The filters with $G0_{even}$ and $G0_{odd}$ are half as long as G0, as they are obtained by splitting G0. Since, the even and odd terms are filtered separately, by the even and odd coefficients of the filters, the filters can operate in parallel, improving the efficiency. In the RTL design, DWT Core and its communication with

Test Bench/External Memory is shown in figure 5. Due to scalability of the core, with minor change in the control logic, this core can be used for any level of decomposition [14].

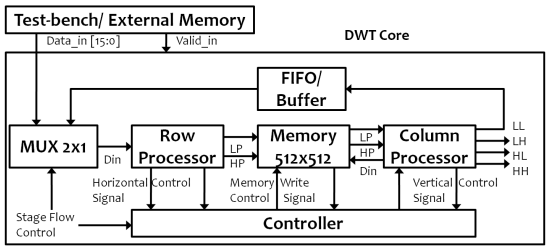

Fig. 5. DWT and its communication with Test Bench/External Memory

Matrix Decomposition-PCA: The DWT output data set is very large in size. It can be reduced to lower dimensional data set before feeding it to the neural network classifier for computational efficiency and quick training of the network. For multi-dimensional scaling (MDS), many methods are available viz. Principal Component Analysys (PCA), linear discriminant analysis (LDA) and independent component analysis (ICA). We have chosen PCA for this purpose [18].

There are mainly three methods for implementation of PCA: Power method, QR algorithm and Jacobi method. Jacobi method is based on the principle of Eigne Value Decomposition (EVD). Jacobi method may be again either classical, cyclic, threshold and parallel [19]. The Power method is simplest but computes single eigen value with largest value which is not suitable for our implementation. The performance of QR algorithm is very poor because it needs to calculate the QR decomposition at each iteration step. Parallel Jacobi Method can facilitate parallel processing by efficient choice of rotation sets [19]. We have selected Parallel Jacobi Method for our system. The block diagram of the system is shown in figure 6.

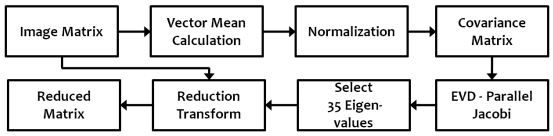

Fig. 6. PCA Implementation using Eigen value Decomposition

We have considered 35 principal components by experimentation in the software implementation explained in [6].

Dot Product: 256 D vectors are read from Wavelet decomposition module, and a dot product is done with 35 Eigen vectors each 256 elements, as of now Eigen vectors are hard coded data. The Dot products of the 35 elements are calculated by multiplying Row of Matrix A with the 35 Columns of the Matrix B individually resulting in Matrix C with size 1x35. As the maximum available size of the row and column of any matrix operation in Xilinx ISE13.3 is 32, the two input matrices are re-ordered. The input Matrix 1x256 A has 256 elements arranged in 8 Rows and 32 Columns, calling the resultant matrix as Matrix A_i. The second input 256x35 matrix is split into 35 32x8 matrices. Each 35 matrix represents each column of 256x35 matrix. E.g. first 32x8 matrix (Matrix B_0) will arrange the 256 elements of first column of 256x35 matrix. The matrix multiplication between Matrix A_i and Matrix B_0 gives the resultant 8x8 output Matrix C_0. All the diagonal elements of the Matrix C_0 are added to get the first element of the 35 elements of the final output Matrix C. Similarly, the matrix multiplication between Matrix A_i and Matrix B_1 gives the resultant 8x8 output Matrix C_1. All the diagonal elements of the Matrix C_1 are added to get the second element of the 35 elements of the final output matrix C. The same is done for all the 35 elements of the final output Matrix C [1x35].

Neural Network Classifier: The NN classifier is a Restricted Coulomb Energy (RCE) based classifier. The performance of this classifier is compatible to available latest Neural Network Devices [10]. In the work [6], superior performance of RCE based classifier over other existing approaches is established and advantages of this architecture for hardware implementation is described. It has 64 Neurons operating in Parallel and it can learn and recognize patterns of up to 256 bytes. The core has expandable functionality i.e. we can instantiate more number of neurons in the soft RTL core as and when the application demands. Figure 7(a) shows the simplified functional blocks of the NN functional core. During training phase, the results of each of the neuron are examined by all other neurons to adjust the influence field. During recognition phase, the results of Neuron are read on the Neuron Bus. Distance as calculated by each of the firing neurons is sorted in ascending order with corresponding category as recognized, unrecognized or recognized with uncertainty. Figure 7(b) shows the simplified architecture of a Neuron. Neuron signature block contains the signature of the pattern, for whose recognition, the neuron has been committed. Feature vector is weighed against the stored signature and distance is calculated in one of the two ways depending on the selected Norm, L_1 or L_{sup}. We have selected L_1 Norm for our application [6]. Distance comparison block compares the distance with Active Influence Field (AIF) of Neuron which in turn determines the recognition status.

IV. Results

Hardware Platform: The developed FRS platform is shown in figure 8 and the feature summary of face recognition system is given in table I.

Resource Utilization & Power consumption: The resource utilization and power consumption of FRS algorithm and the FRS system with neural network core containing 64 neurons are shown in table II. The FRS algorithm contains only the FRS pipeline, whereas the FRS system contains the FRS pipeline and other submodules and the firmware over microblaze processor. Power distribution design of this board can take max. 100watts of power load i.e. 12V @8.5Amp maximum. We have taken reference of Xilinx power measurement excel sheet to calculate static power consumption of the FPGA.

Algorithm Verification: We have compared the performance of the proposed FRS with published approaches as

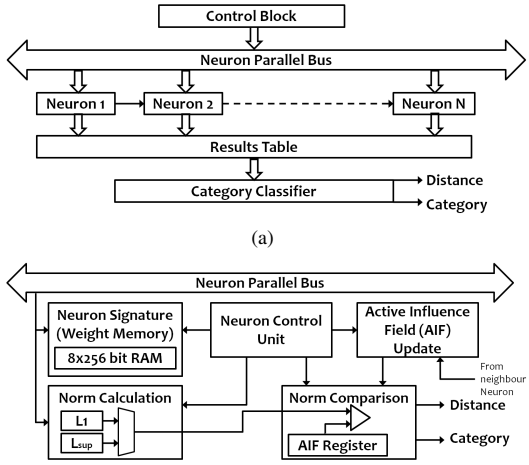

Fig. 7. (a) Functional block diagram of the Neural Network core, (b) Simplified Architecture of a Neuron

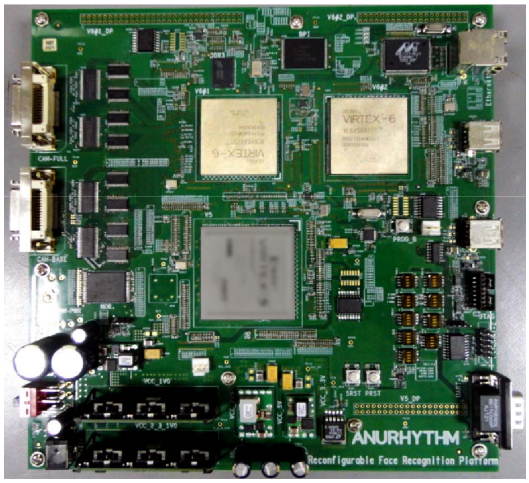

Fig. 8. FRS Hardware Platform

given in [6]. The results of our software and hardware implementations are provided in III. From the table, we can observe that, DWT coefficients are only 1.02% off from their exact Golden reference values. PCA values are 0.15% off and CLAHE output values are 0.39% off from their exact Golden reference values.

The summary of our FRS implementation is given in table IV. We verified the performance of our proposed FRS with multiple standard face databases namely Yale, CroppedYale, AT&T, GeorgiaTech, FERET, BioID, Indian face, CalTech etc. [20] apart from our own developed face databases captured using CMOS sensor camera. We achieved an average accuracy of 94%. It can recognize a single image in 18 ms. The recognition time is irrespective of the database under test because the images are resized to 128x128 before giving as input to CLAHE module. We targeted 27 ms recognition time for real time application corresponding to 37 fps frame rate which is achieved successfully. We set the configuration mode in Byte Peripheral Interface (BPI) mode so that, at board

TABLE I
FRS SYSTEM FEATURES

Device/Board	Static Power (Watt)
Board Size	8.5 x 8.5 Inch
PCB Thickness	2.2 mm
No. of Layers	16
Material Used	EMC827, High Tg(170), RoHS compliant
Image capture	1-Mp CMOS sensor camera @37fps
Camera Interface	CamLink
Processing Element	Virtex-6 SX475T
On Board Memory	DDR3-512 MB
Configuration Memory (NOR)	64 MB
Image Processing Cores	Frame buffer, CLAHE, DWT, PCA
Pattern Classifier	RCE-NN in FPGA

TABLE II
FPGA RESOURCE UTILIZATIONS

Model	FRS Algorithm	FRS System
Device	XC6VSX475TFF1156-1	XC6VSX475TFF1156-1
Tool	Xilinx ISE 13.4	Xilinx ISE 13.4
Slices	39432 of 74400 (53%)	50881 of 74400 (68%)
Block RAM	1526 of 3192 (48%)	2712 of 3192 (84%)
Clock	80 MHz	62.5 MHz
PCA Generate time	21 ns	21 ns
Static Power	4.264 Watt	4.262 Watt
Dynamic Power	1.5 Watt	5.8 Watt
Total Power	5.764 Watt	10.062 Watt

power-up the FPGA will be programmed automatically.

Performance metric comparison of our implementation with the published results in the literature in terms of accuracy is described in [6]. This paper provides complete comparison of different FRS algorithms, which shows that the proposed method is better than the published face recognition implementations given in ([15]–[17], [22]–[24]) for the above mentioned face databases. In [15], face classification requires 40ms recognition time and in [17], recognition time required using PCA is 110ms. On the other hand, our implementation requires 18ms recognition time. Also in comparison to [25] proposed method shows better results. For Yale and ORL databases, the recognition accuracy achievable are 91% and 93.3% respectively [16]. But we achieved 93% and 96% accuracy for the corresponding databases. Also traditional methods like Support Vector Machine (SVM) suffers with the limitation of large memory requirement and more computational time while dealing with large datasets [16] and hence may not be suitable for real-time face recognition applications.

Figure 9 shows the recognition window of the GUI. During training, images are either captured from the camera or taken from static face database & test-case name is provided. The face images are pre-processed and the corresponding feature vectors are input to the NN classifier in FPGA. The window will show the status after completion of training. During recognition, we have to provide a test image & it will be submitted to the hardware platform for recognition. After the operation is complete, the GUI displays the recognition status.

TABLE III
FRS ALGORITHM RTL INTEGRATION VERIFICATION RESULTS

Module	Min. Difference	Max. Difference	% Variation
Image Capture	0	0	0.00%
CLAHE	-1	1	0.39%
DWT	2.622	2.726	1.02%
PCA	-0.39	0.37	0.15%
Dot product	-1.68	1.65	0.65%

TABLE IV
PERFORMANCE OF NEURON CLUSTER CORE BASED FRS SYSTEM

Database	Images	Accuracy	Recognition time/imafe
YaleDB	400	96%	18 ms
ANURAG (own database)	180	94%	18 ms
Georgia-Tech.	200	93%	18 ms
AT&T (ORL)	400	93%	18 ms

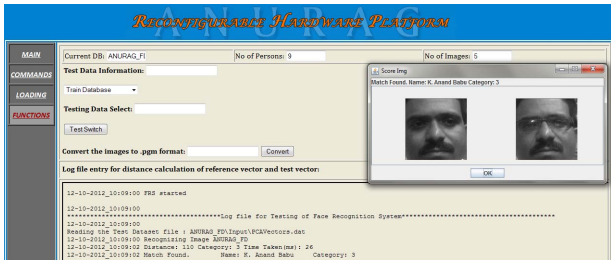

Fig. 9. FRS GUI: Recognition Window

V. CONCLUSION

Real-time, improved face recognition system design & its implementation on FPGA based embedded platform is presented in this paper. The face images are captured using CMOS sensor camera connected to FPGA board via CamLink. Image pre-processing techniques namely CLAHE, DWT & PCA are implemented for local contrast enhancement, image decomposition & dimensionality reduction respectively. The 35 feature vectors derived from PCA module are used to categorize and recognize the face using RCE based classifier. We have fully utilized the parallelism of the NN architecture by implementing this RCE based NN in FPGA itself. Due to this implementation, the system achieved an average recognition accuracy of 94% for multiple image datasets and recognizes a single image in real-time i.e. within 18 ms. The superiority of our proposed system comprising of pre-processing, feature extraction and classification modules is established with other published implementations.

In future, we can implement more robust face detection algorithm in the FPGA on FRS board for better face detection and recognition performance, mandatory for video-camera based recognition. Our proposed system can be verified against disguised & occluded images. By incorporating suitable image pre-processing techniques and appropriate interfaces, the system can be extended to use for secure access control, attendance monitoring & network security applications.

REFERENCES

[1] P. Sinha, B. Balas, Y. Ostrovsky, and R. Russell, "Face recognition by humans: Nineteen results all computer vision researchers should know about," *Proceedings of the IEEE*, vol. 94, no. 11, 2006.

[2] R. Chellappa, C. L. Wilson, and S. Sirohey, "Human and machine recognition of faces: a survey," *Proceedings of the IEEE*, vol. 83, no. 5, pp. 705–740, 1995.

[3] F. Goudail, E. Lange, T. Iwamoto, K. Kyuma, and N. Otsu, "Face recognition system using local autocorrelations and multiscale integration," *IEEE Trans. PAMI*, vol. 18, no. 10, pp. 1024–1028, 1996.

[4] M. Kirby and L. Sirovich, "Application of the karhunen- loeve procedure for the characterization of human faces," *IEEE Trans. PAMI.*, vol. 12, pp. 103–108, 1990.

[5] M. V. Wickerhauser and Large-rank, "Approximate component analysis with wavelets for signal feature discrimination and the inversion of complicated maps," *J. Chemical Information and Computer Sciences*, vol. 34, no. 5, pp. 1036–1046, 1994.

[6] S. Sardar, G. Tewari, and K. A. Babu, "A hardware/software co-design model for face recognition using cognimem neural network chip," *International Conference on Image Information Processing (ICIIP 2011)*, 2011.

[7] N. Kouda and N. Matsui, "On the function approximation in restricted coulomb energy neural network with gaussian radial basis function," 2011.

[8] X. MuMehmet, A. Paul, W. Mohamad, and H. Hassoun, "An rce-based associative memory with application to human face recognition," *Neural Processing Letters*, 2006.

[9] C. Sui, N. M. Kwok, and T. Ren, "A restricted coulomb energy (rce) neural network system for hand image segmentation," *2011 Canadian Conference on Computer and Robot Vision*, 2011.

[10] CogniMem, "http://www.general-vision.com/ product_cm1k.htm."

[11] B. Fawcett, "Fpgas in reconfigurable computing applications," in *WESCON/'95. Conference record. 'Microelectronics Communications Technology Producing Quality Products Mobile and Portable Power Emerging Technologies'*, 1995.

[12] O. Maslennikow and A. Sergiyenko, "Mapping dsp algorithms into fpga," in *Parallel Computing in Electrical Engineering, 2006. PAR ELEC 2006. International Symposium on*, 2006, pp. 208–213.

[13] Xilinx, "www.xilinx.com/products/silicondevices/fpga/virtex-6/sxt.htm."

[14] G. Tewari, S. Sardar, and K. A. Babu, "High-speed & memory efficient 2-d dwt on xilinx spartan3a dsp using scalable polyphase structure with da for jpeg2000 standard," *3rd International Conference on Electronics Computer Technology (ICECT 2011), Kanyakumari, India*, 2011.

[15] A. U. Batur and B. E. Flinchbaugh, "Performance analysis of face recognition algorithms on tms320c64x," *Texas Instruments Application Report-SPRA874*, pp. 1–12, 2002.

[16] L. Lang, F. Xia, and X. Wang, "Study of face recognition algorithm based on proximal support vector machine," in *Intelligent Computation Technology and Automation, 2009. ICICTA '09. Second International Conference on*, vol. 1, 2009, pp. 702–705.

[17] C.-S. Fahn and Y.-T. Lin, "Real-time face recognition techniques used for the interaction between humans and robots," in *Computer Symposium (ICS), 2010 International*, 2010, pp. 234–239.

[18] N. Nallammal and V. Radha, "Performance evaluation of face recognition based on pca, lda, ica and hidden markov model," *Lecture Notes in Computer Science*, vol. 6411, pp. 96–100, 2012.

[19] G. Golub and C. Loan, *Matrix Computations*. The Johns Hopkins University Press, Baltimore and London, 1996.

[20] M. Grqic and K. Delac, "www.face-rec.org/databases."

[21] M. Joo-Er, S. Wu, J. Lu, and H. L. Toh, "Face recognition with radial basis function (rbf) neural networks," *IEEE Trans. Neural Networks*, vol. 13, pp. 697–710, 2002.

[22] S. Lawrence, C. L. Giles, A. C. Tsoi, and A. D. Back, "Face recognition: A convolutional neural-network approach," *IEEE Trans. Neural Networks*, vol. 8, pp. 98–113, 1997.

[23] S. Z. Li and J. Lu, "Face recognition using the nearest feature line method," *IEEE Trans. Neural Networks*, vol. 10, pp. 439–443, 1999.

[24] M. Mazloom and S. Ayat, "Combinational method for face recognition: Wavelet, pca and ann," *IEEE Computer Society*, 2008.

[25] S. Du and R. Ward, "Statistical non-uniform sampling of gabor wavelet coefficients for face recognition," *Proc. of IEEE ICASSP Conference*, vol. 2, pp. 73–76, 2005.