Journal of Applied Sciences

ISSN 1812-5654

Journal of Applied Sciences 11 (16): 2984-2992, 2011
ISSN 1812-5654 / DOI: 10.3923/jas.2011.2984.2992
© 2011 Asian Network for Scientific Information

Hardware Approach of ANN Based Iris Recognition for Real-time Biometric Identification

[1]Mamun Bin Ibne Reaz, [1]Md. Syedul Amin, [1]Fazida Hanim Hashim and [2]Khandaker Asaduzzaman
[1]Department of Electrical, Electronic and Systems Engineering, Universiti Kebangsaan Malaysia, 43600, UKM, Bangi, Selangor, Malaysia
[2]Plasma Research Laboratory, Research School of Physical Sciences and Engineering, Australian National University, Oliphant Building 60, Mills Road, Canberra 0200, Australia

Abstract: Artificial Neural Networks (ANN) are increasingly applied to biometric identification because neural nets have been shown to be technologically powerful and flexible, ideally suited to perform identification analysis. Therefore, it demands the development of a new processing structure that allows efficient hardware implementation of the neural networks mechanism. This research presents the ANN based iris recognition for biometric identification modeled by the very high speed integrated circuit Hardware Description Language (VHDL) to ease the description, verification, simulation and hardware realization of this kind of systems. The project is divided into two processes which are image processing and recognition. Image processing was performed by using Matlab where back propagation was used for recognition. The iris recognition architecture comprises of three layers: Input layer with three neurons, hidden layer with two neurons and output layer with one neuron. Sigmoid transfer function is used for both hidden layer and output layer neurons. Neuron of each layer is modeled individually using VHDL. Functional simulations were commenced to verify the functionality and performance of the individual modules and the system. Iris vector from captured human iris has been used to validate the effectiveness of the model. An accuracy of 88.6% is achieved in recognizing the sample of 100 data of irises.

Key words: Artificial intelligence, iris recognition, VHDL, neural network, image processing

INTRODUCTION

The biometric identification which merges technical and fundamental analysis utilizing artificial intelligence tools and synthesizes technical and fundamental data within an analytical framework, resulting in better recognition capabilities. Among different artificial intelligence tools, neural networks are increasingly applied to biometric identification because neural nets have been shown to be technologically powerful and flexible, ideally suited to performing identification analysis.

Multi-features and Multi-stages RBF Neural Network Classifier with Fuzzy Integral system (Haddadnia *et al*., 2002) is very efficient and has well-suited rules for biometric identification but a drawback that the system is more complex. Component-based approaches have shown promising results in various object detection and recognition tasks such as face detection (Schneiderman and Kanade, 2000) and face recognition (Brunelli and Poggio, 1993) but in terms of speed, hardware simplicity and applicability it is not suited. Rules extractions approach (Kane and Milgram, 1994) is less relevant in biometric identification and involved complex algorithms.

In today's security industry, possession has become an encoded card, knowledge equates knowing one's Personal Identification Number (PIN) and personal recognition has given way to physiological and behavioral characteristics known as biometrics. Computer vision-based techniques that recognize human features such as faces, fingerprints, palms and eyes have many applications in surveillance and security. In Biometric identification system all relies upon forms of random variation among persons. The more complex the randomness the better, because more dimensions of independent variation produce signatures having greater uniqueness (Chenhong and Zhaoyang, 2005).

Biometrics offers new solutions that are user friendly and yet secure access. It works by inspecting biological features of human beings that distinguish one from another. The critical attributes for any measure are the

Corresponding Author: Mamun Bin Ibne Reaz, Department of Electrical, Electronic and Systems Engineering, Universiti Kebangsaan Malaysia, 43600, UKM, Bangi, Selangor, Malaysia
Tel: +603-89216311 Fax: +603-89216146

number of degrees-of-freedom of variation in the chosen index across the human population, determines uniqueness, its immutability over time and its immunity to intervention and the computational prospects for efficiently encoding and reliably recognizing the identifying pattern. Iris has all above unique features and compare to the other biometrics it could be the best. It has key advantages in terms of speed, simplicity, accuracy and applicability (Yan *et al*., 2007). The Iris patterns possess a high degree of randomness. It can reveal about 266 independent degrees-of-freedom of textural variation across individuals (Daugman, 2005). It is stable over time and visible in a face. Thus it also impossible of surgically modifying it without unacceptable risk to vision and its physiological response to light provides a natural test against artifice. This internal organ which is protected by a transparent layer of cornea, can be imaged adequately at distances of up to about a meter (depending on the camera) (Jarjes *et al*., 2010). In iris recognition, the probability of finding two people with identical iris pattern is almost zero (Wang *et al*., 2007). Moreover, the spatial patterns which apparent in the human iris are highly distinctive to an individual (Wildes, 1997). There are several researches on iris recognition, especially usage of localization algorithm for iris recognition is worth mentioning (Mohammed *et al*., 2009; Yahya and Nordin, 2008).

The Field-Programmable Gate Arrays (FPGA) offers a potential alternative to speed up the hardware realization (Coussy *et al*., 2009; Marufuzzaman *et al*., 2010; Reaz *et al*., 2007). From the Perspective of Computer-aided Design, FPGA comes with the merits of lower cost, higher density and shorter design cycle (Choong *et al*., 2005; Akter *et al*., 2008). It comprises a wide variety of building blocks. Each block consists of programmable look-up table and storage registers, where interconnections among these blocks are programmed through the hardware description language (Reaz *et al*., 2004, 2003, 2005a). This programmability and simplicity of FPGA made it favorable for prototyping digital system. FPGA allows the users to easily and inexpensively realize their own logic networks in hardware. FPGA also allows modifying the algorithm easily and the design time frame for the hardware becomes shorter by using FPGA (Choong *et al*., 2006; Ibrahimy *et al*., 2006).

In this study, a unified framework for FPGA realization of neural network based iris recognition system is designed by means of using a standard Hardware Description Language VHDL where iris vector from captured human iris has been used to validate the effectiveness of the model. The use of VHDL for modeling is especially appealing since it provides a formal description of the system and allows the use of specific description styles to cover the different abstraction levels (architectural, register transfer and logic level) employed in the design (Pang *et al*., 2006; Reaz *et al*., 2006). In the computation of method, the problem is first divided into small pieces, each can be seen as a submodule in VHDL. Following the software verification of each submodule, the synthesis is then activated. It performs the translations of hardware description language code into an equivalent netlist of digital cells. The synthesis helps integrate the design work and provides a higher feasibility to explore a far wider range of architectural alternative (Reaz *et al*., 2007, 2005b). The method provides a systematic approach for hardware realization, facilitating the rapid prototyping of the fuzzy based subway train braking system.

IMAGE PROCESSING

The objective of digital image processing is for improving the pictorial information for human interpretation and processing of scene data for autonomous machine perception. An iris image which had been captured contains not only the region of interest (iris) but also some ‘unuseful’ parts. Moreover, a change in the camera-to-eye distance may result in the possible variation in the size of the same iris. In addition, the brightness is not uniformly distributed because of non-uniform illumination. Before the process of feature extraction of the iris, the image needs to be preprocessed to localize the iris, normalize the iris and reduce the influence of the factors mentioned above (Bhattacharyya *et al*., 2009).

The iris images of Ms. Lee Peik Shyan were used shown in Fig. 1, which is in bitmap image file (bmp) format. Imread function in Matlab was used to read the file which displays each pixel of the iris images.

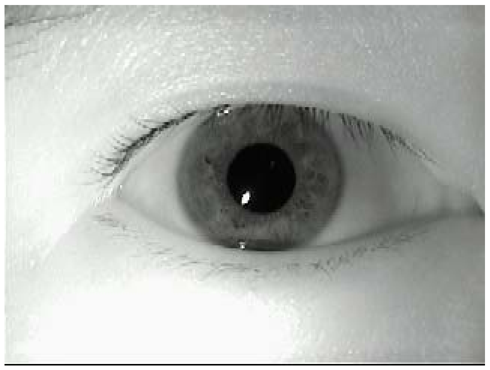


Fig. 1: Iris image of Ms. Lee Peik Shyan

The histogram stress was used to enhance the contrast of the image as it does not change the probability of the pixel but change the level of the pixel.

The histogram shows that most of the pixel values range from 30 to 90. The very bright pixel is due to the reflection and the very dark pixel is the pixel in the pupil, these pixels have small probability less than 0.002%. We group these entire pixels into one by using Eq. 1:

$$I(i,j)=\begin{cases} x & P[I(i,j)]<0.002 \\ I(i,j) & \text{otherwise} \end{cases} \tag{1}$$

where, x = pixel value with probability ≥0.002 and closest to the pixel value which have probability <0.002.

The histogram stress process was done by using Eq. 2. Figure 2 and 3 show the result of histogram stress by taking into account the grouping:

$$I(i,j)=\left[\frac{I(i,j)-I(i,j)_{\min}}{I(i,j)_{\max}-I(i,j)_{\min}}\right][\max-\min]+\min \tag{2}$$

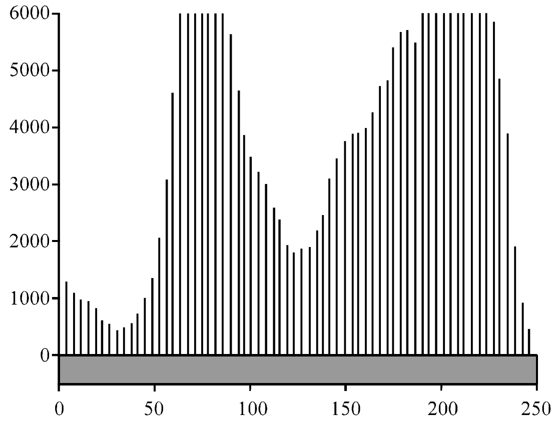


Fig. 2: Result of the histogram stress

where, $I(i,j)_{\max}$ is the largest gray-level value in the image, $I(i,j)_{\min}$ is the smallest gray-level value in the image, max and min correspond to the maximum and the minimum gray level value possible (for 8-bit image).

Referring to the image in Fig. 3 that have been histogram stressed, the pixel outside the iris is much brighter than the iris. The iris has low contrast and the mean of image gray level is more than almost the entire pixel in the iris. We choose the threshold value to be the mean value of the image. Figure 4 shows the image after thresholding.

After thresholding, the problems that remain are to eliminate the eyelash and the pupil. Using statistic analysis, the image is masked convolute with 11×11 mask. The mask value all set to one. The image perform mask convolution with this mask than threshold with a parameter. The parameter for the threshold set 64 to remain the outer layer pixel not to be eliminated.

The detection of iris was done by comparing with a moving circle and change the radius slowly until the radiuses match the iris radius. The result will return the center coordinate and the radius of the circle. Here the image was masked with a radius-changing circle. The pixel on the mask are taken and summed. The black pixel has the value of 0; so perfect circles have the sum of 0. We set that if the mask have more than 3% white pixel then it is not a circle. The percentage is small to avoid the mask to take the eyelid or eyelash which are near the iris. Using the center coordinate and the radius, the iris was reconstructed from the original image by taking the pixel locating in the circle found from the threshold image. Figure 5 shows the reconstructed iris.

The cropping method was used to extract the useful image information and construct it into a smaller image. The iris in the image can be extracted into a smaller image because most of the white pixels are not useful. The initial coordinate or the place where to start and the size of the

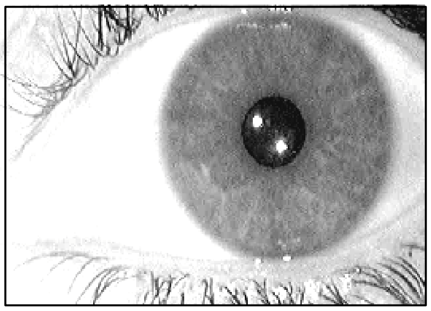


Fig. 3: Image after histogram stress

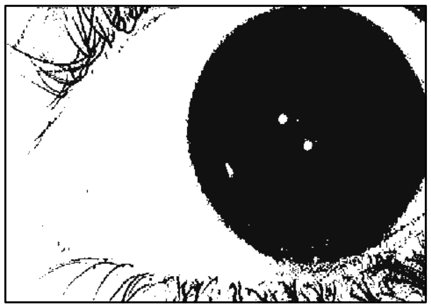


Fig. 4: Thresholded image

square or rectangle was found by using Eq. 3 and 4, to crop the image using imcrop in the Matlab toolbox. Figure 6 shows the cropped iris:

$$\text{Initial coordinate } (i, j) = (x_c\text{-}r, y_c\text{-}r) \quad (3)$$

$$\text{Size} = (r\times 2)\times(r\times 2) \quad (4)$$

where (x_c, y_c) = iris center coordinate and r = iris radius.

The pupil size was estimated by cropping each image manually. The length of the image is the diameter of the pupil. The maximum diameter was selected to the size of the pupil. Figure 7 shows the iris by eliminating pupil.

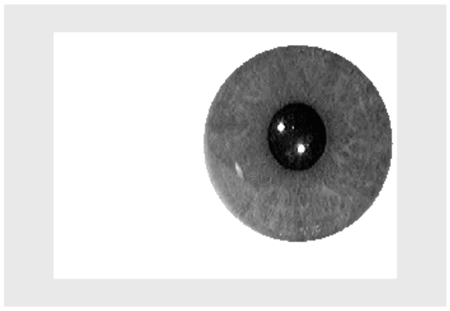


Fig. 5: Reconstructed iris

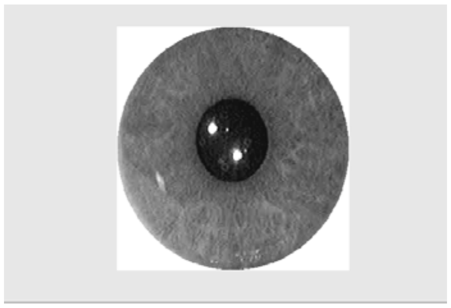


Fig. 6: Cropped image

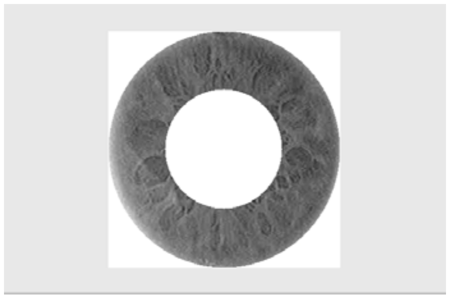


Fig. 7: Pupil eliminated

The iris image is transformed into rectangular form by taking the pixel in 180° as the initial point and turn clock wise until the initial point. Then the image data perform Discrete Fast Fourier Transform. From the original image, the pattern of the iris is growing from the center of the iris. The original transformation of Discrete Fourier Transform is changed to polar coordinate.

The amplitude spectrum of the image after the polar transformation shows that most of the information is assembling in the middle of the image. The middle neighborhood data of the amplitude spectrum have significant value when compared with the maximum amplitude. This neighborhood data extracted from the amplitude spectrum. Then the amplitude was normalized before put into the network for training or recognizing image.

DESIGN OVERVIEW

There are many architectures of how to connect the neurons within a neural network. Normally most of the architecture has the data inputs of the neural network then go into the layer of neurons (the hidden layer). Each input will connect to each of the neurons in the hidden layer which includes the weight matrix, the summers, the bias vector, the transfer function boxes and the output vector. In this project, we have chosen a two layer neural network with the input layer having three neurons, hidden layer having two neurons and the output layer having one neuron.

The input is the vector of the feature extracting of different iris images. The vectors are in pixel form.

As for the output, it is to match the testing input iris with the trained iris which had stored. If both of them are matched, then the output will display the numbered of the particular iris; else it will display 0.

In this project we have chosen multilayer perceptron architecture with backpropagation algorithm since it is the most frequently used architecture and learning algorithm in the field of neural network.

The network is configurable to enable any combination of number of input neurons, hidden neurons and output neurons. The hidden layer and output layer neurons which use sigmoid transfer function and combination of these neurons is coded in a top-level design using structural approach. Random number generator and normalization circuit is added to initialize the weights of the neural network and process the input data to a suitable range.

The developed neural network is divided into 3 mode of operation which are random weight generation mode, training mode and testing mode.

Number format design: Since finite storage devices, also referred to as sequential elements, are being used in the neural network chip, the internal signals must have a numbering format. In this study, number-scaling method is used to reduce the complexity and to facilitate synthesis process. This method multiplies integer type of VHDL with 1024 or 1024^2 depending on the number of multiplication and later divides the number with the same constant in order to get back its initial value.

System level design overview: The VHDL code is divided into 5 entities which 4 of it are link together by 1 entity. The top-level entity is MATCHING which contain the architecture BEHAVIOUR to link the components using VHDL structural approach. The other 4 entities are INPUT, HIDDENLAYER, OUTPUTLAYER and BACKPROPAGATIONLAYER.

Input layer neuron design: The input layer of the neural network design will consist of 3 neurons. The purpose of the input layer is to act as a buffer between the data and the hidden layer neurons. The entity for the input layer neuron is INPUT and its architecture is BEHAVIOUR.

Hidden layer neuron design: The hidden layer of the neural network design consists of two neurons that utilize a sigmoid transfer function. Each of the hidden layer neurons has the three system data of the current exemplar as input. The entity for the hidden layer neuron is HIDDENLAYER and its architecture is BEHAVIOUR. The code is written in a way that each neuron is capable of performing 3 modes of operations.

The first mode of operation is random weights generator which generates random weights for each of the three inputs. The first mode of operation is triggered if the input to the "mode" pin is set to the value "01" of std_logic_vector type.

The second mode of operation is the forward mode. In this mode, the network propagates forward its input to produce output. Each of the hidden layer neurons multiplies each of the three input data attributes by a unique internal weight generated by the random weights generator. These three results then be added together and applied to the internal transfer squashing function that is the sigmoid function. The output of the transfer function is the activation output of the neuron. The output of the each hidden layer neurons is fed to the output layer neuron. The second mode is used for testing the network or when the network is actually as iris recognizer. The reason of the second mode, the network does not perform any weight adjustment as no learning algorithm is performed. The second mode of operation is triggered if the input to the "mode" pin is set to the value "10" of std_logic_vector type.

The third mode of operation is the backward propagate mode. In this mode, the error of the network is backpropagated and the weights and the threshold is adjusted to reduce the error of the network if the same input is applied to the network again. Backpropagation learning algorithm is applied in this process. The error from the hidden layer neuron is transferred to port hidden_error1 and hidden_error2 for each of the two hidden layer neuron respectively for monitoring purposes. When the neural network chip is running in the 3rd mode, the network is running in training mode and output_desired port fed with the desired output data. The third mode of operation is triggered if the input to the "mode" pin is set to the value "11" of std_logic_vector type.

For an efficient operation, another entity, BACKPROPAGATIONLAYER is created for the third mode. In order to adjust for the weights of the hidden layer neuron, output error is needed. Therefore, the weights are adjusted after the output error is obtained. BACKPROPAGATIONLAYER is located after the output layer neuron so that it can obtain the output error and adjust the weights according to the backpropagation learning algorithm. The adjusted weights feedback to the hidden layer neuron.

Random weights generator: Random weights generator generates a random number with uniform distribution in the interval between -1024 and 1024. The weights are set between these ranges since the input data that is used for calculation is between these ranges of value.

The random weights generator uses a random number between 0 to 1024 to generate the desired number derived from the Eq. 5-7:

$$X(n) = A + (B - A)*R(n) \tag{5}$$

Where:

- A and B are the limits of the interval
- X is a random number between A and B
- R is a random number between 0 and 1
- Given interval A = -1024 and B = 1024

Therefore,

$$X = -1024 + (2048*R) \tag{6}$$

Since, we are going to feed random number of range between 0 and 1024, the following formula is applied to the network instead of the formula above:

$$X = -1024 + (2048*R/1024) = -1024 + (2*R) \tag{7}$$

The value of 1024 is chosen because it is a power factor of 2. This is important since only value of power factor of 2 can be used in the division arithmetic operation for synthesis.

The value of the weights generated is stored in signal class of data object because an object of this class holds not only the current value of a type but also the past value and the set of scheduled future values that are to appear on that signal. A signal is assigned to a value using a signal assignment statement. Furthermore, a signal can be assigned a value only at a future point of time; that is, the current value of a signal can never be changed.

Normalization: Normalization is used to reduce the range of the data set to the value appropriate for input to the activation function. The input data is normalized by using the Eq. 8-9:

$$\text{New value} = \frac{(\text{Input value-Mean})}{(\text{Maximum Range})} \tag{8}$$

Form the vector data from iris image:

- Mean = 68
- Maximum = 350
- Minimum = 10

$$\text{New value} = \frac{(X - 68)}{(340)}, x = \text{Input value} \tag{9}$$

Sigmoid function and threshold value: A sigmoid transfer function acts to squash the output of the neuron into a range of zero and one. Piecewise linear approximation of the sigmoid function is used into the network (Larsson *et al*., 1996). Since the range of the input is between -1024 and 1024, the range of the output must be between the ranges 0 to 1024 after the sigmoid function is applied to the network. Therefore, all the set of the equation are multiplied by 1000 which is applied to the VHDL design.

The bias or the threshold value of the network is set to the mean value of the image which is 68.

Output layer neuron: The output layer of the neural network consists of one neuron that will utilize a sigmoid transfer function. The entity for the output layer neuron is OUTPUTLAYER and its architecture is BEHAVIOUR. The output layer neuron has the activation output of each of the two hidden layer neurons as inputs. The output layer neuron also has the target value, as input. The output layer neurons multiply each of the two input datavalues from the hidden layer neurons by a unique internal weight generated by the random weights generator. The two results then added together and applied to the internal sigmoid transfer function and the output is the activation of the neuron. The activation of the output layer neuron is the activation of the entire neural network chip. This activation is compared to the expected target input from the data exemplar of the system to get the current exemplar error of the neural network. The same modes of operation as the hidden layer neurons are applied to the output layer neurons. The same random weights generator, normalization method and piecewise linear approximation of the sigmoid function as the hidden layer is used in the output layer.

SIMULATION AND DISCUSSION

Simulation is done by feeding testbench into the VHDL code. Training is done by feeding the iris vector into the network iteratively. The data was fed in 3 times only due to the limitation of the Active-HDL software which cannot process mass input data due to low CPU processing speed. The training process is done between 0 to 36000 nsec. This process is in mode “10”.

The first 100 nsec are used to initialize the weights of the hidden layer neurons and the output layer neurons. This process is done in mode “01”.

Testing is done between 36100 and 36800 nsec. This process is done in mode “11”.

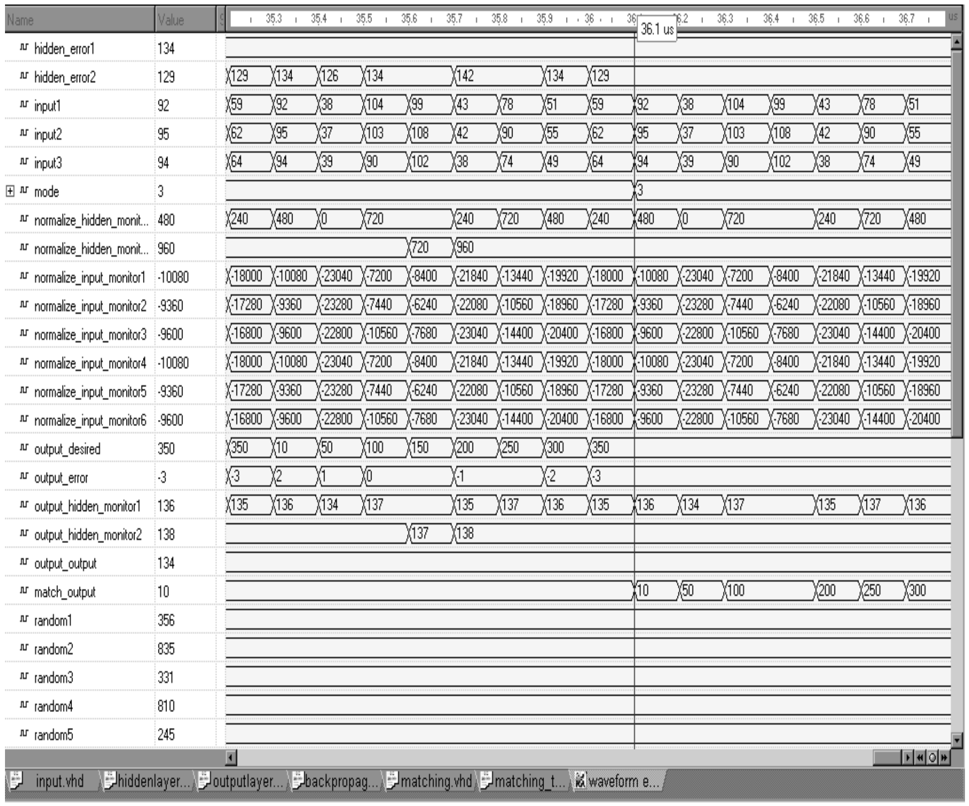
Only the first 100 data from the iris image vector is used to show the functionality of the VHDL code. The subsequent 100 data is used for testing purposes. The normalization function is adjusted accordingly so that it can perform normalization with respect to the specified data only.

Training simulation: At 0 nsec, the neural network chip operates at weight initializing mode. The random number is generated and then converted to the corresponding weight by utilizing the random weights generator.

At 100 nsec, the neural network chip is in training mode. The 3 input to the input layer neuron is accompanied by a desired output. Where desired output is the signature which is chosen to recognize different iris images. Weights are adjusted to minimize the output error.

Several sets of weights generated by the random weight generator are tested randomly before getting the simulation result shown in this study. It has been observed that the value of output error is reduced as more input is fed into the network. The weights are adjusted so that the probability of the next input to get its desired output is higher. In the training mode, the input to the network is first normalized before the calculation is performed.

Testing simulation: At 36100 nsec, the neural network chip is in testing mode. A neural network must pass through weights initialization process and training before testing is performed.



Name	Value
hidden_error1	134
hidden_error2	129
input1	92
input2	95
input3	94
mode	3
normalize_hidden_monit...	480
normalize_hidden_monit...	960
normalize_input_monitor1	-10080
normalize_input_monitor2	-9360
normalize_input_monitor3	-9600
normalize_input_monitor4	-10080
normalize_input_monitor5	-9360
normalize_input_monitor6	-9600
output_desired	350
output_error	-3
output_hidden_monitor1	136
output_hidden_monitor2	138
output_output	134
match_output	10
random1	356
random2	835
random3	331
random4	810
random5	245

Fig. 8: Sample timing diagram of testing simulation at 36100 nsec

During testing, no desired output data is required, only input data is required. The weights of the neural network are considered to have reached its local or global minimum at this period. There wouldn't be any adjustment of weights during training and this also means there won't be any output error. Therefore, it can be observed that the variables stop changing at 36100 nsec.

It can also be observed that changes in weights are less frequent as the network approaches 36100 nsec. The reason behind that the network has almost reached its global or local minimum after performing the backpropagation learning algorithm. Figure 8 is a sample-timing diagram of testing simulation at 36100 nsec through the VHDL programming.

The simulation stopped at 36800 nsec after the neural network chip is fed with 3 iterative inputs for training and subsequent 100 input for testing. The reduction in the number of input is done in order to reduce the complexity of the system and is meant to enhance the understanding on the operating characteristics of the system.

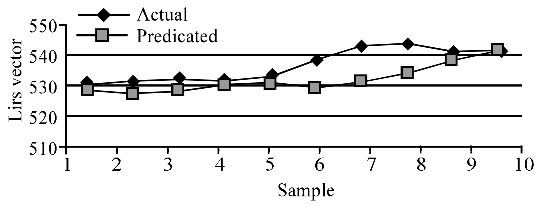


Fig. 9: Actual iris vector Vs predicted iris vector

A comparison study between the actual iris vector and predicted iris vector is shown in Fig. 9. It is concluded from the comparison study that the model identification is 88.6% accurate:

$$\text{\% of accuracy} = 100 - \left[\left(\sum_{i=1}^{N}\frac{X - Y}{X}\times 100\right)\div N\right] = 88.6\%$$

Where:
N = No. of samples
X = Actual iris vector
Y = Predicted iris vector

CONCLUSION

By simulating with iris data the proposed approach of VHDL modeling of neural network based iris recognition system is successfully designed, implemented and tested. The use of IEEE floating point-number format would enhance the accuracy of recognition. Currently, we are conducting further research that consider the floating point number format and further reductions in the hardware complexity in terms of synthesis and finally to download the code into Altera FLEX10K: EPF10K10LC84 FPGA chip on LC84 package for hardware realization.

REFERENCES

Akter, M., M.B.I. Reaz, F. Mohd-Yasin and F. Choong, 2008. Hardware implementations of image compressor for mobile communications. J. Communicat. Technol. Electronics, 53: 899-910.

Bhattacharyya, D., P. Das, S.K. Bandyopadhyay and T.H. Kim, 2009. Feature extraction for IRIS recognition. Communicat. Comput. Infor. Sci., 29: 31-39.

Brunelli, R. and T. Poggio, 1993. Face recognition: Features versus templates. IEEE Trans. Patt. Anal. Mach. Intell., 15: 1042-1052.

Chenhong, L. and L. Zhaoyang, 2005. Efficient iris recognition by computing discriminable textons. Int. Conf. Neural Networks Brain, 2: 1164-1167.

Choong, F., M.B.I. Reaz and F. Mohd-Yasin, 2005. Power quality disturbance detection using artificial intelligence: A hardware approach. Proceedings of 19th IEEE International Parallel and Distributed Processing Symposium, (IPDPS'05), Workshop. Denver, USA, IEEE, 146-146.

Choong, F., M.B.I. Reaz, T.C. Chin and F. Mohd-Yasin, 2006. Design and implementation of a data compression scheme: A partial matching approach. Proceedings of Computer Graphics, Imaging and Visualisation: Techniques and Applications, (CGIV'06), Sydney, IEEE, Australia, 150-155.

Coussy, P., D.D. Gajski, M. Meredith and A. Takach, 2009. An Introduction to high-level synthesis. IEEE Design Test Comput., 26: 8-17.

Daugman, J., 2005. Recognizing persons by their iris patterns. Adv. Biometric Person Authenticat., 3338: 783-814.

Haddadnia, J., M. Ahmadi and W.C. Miller, 2002. Multi-features and multi-stages RBF neural network classifier with fuzzy integral in human face recognition. Proceedings of International Conference on Artificial Intelligence, (IC-AI '02), Las Vegas, Nevada, USA, CSREA Press, 117-123.

Ibrahimy, M.I., M.B.I. Reaz, M.A.M. Ali, T.H. Khoon and A.F. Ismail, 2006. Hardware realization of an efficient fetal QRS complex detection algorithm. WSEAS Transac. Circuits Syst., 5: 575-581.

Jarjes, A.A., K. Wang and G.J. Mohammed, 2010. GVF snake-based method for accurate pupil contour detection. Inform. Technol. J., 9: 1653-1658.

Kane, R. and M. Milgram, 1994. Financial forecasting and rules extraction from trained networks. Proceedings of IEEE World Congress on Computational Intelligence, 27 June-02 Orlando, FL, IEEE, USA. 3190-3195.

Larsson, L., S. Krol and K. Lagemann, 1996. NeNEB-An application adjustable single chip neural network processor for mobile real time image processing. Proceedings of IEEE International Workshop on Neural Networks for Identification, (ASCNPMRIP'96), Venice, Italy, Venice, IEEE, Italy-154-162.

Marufuzzaman, M., M.B.I. Reaz, M.S. Rahman and M.A.M. Ali, 2010. Hardware prototyping of an intelligent current dq PI controller for FOC PMSM drive. Proceedings of 6th International Conference on Electrical and Computer Engineering, (ICECE'10), Dhaka, Bangladesh, IEEE, 86-88.

Mohammed, G.J., H.B. Rong and A. Al-Kazzaz, 2009. A new localization algorithm for iris recognition. Inform. Technol. J., 8: 226-230.

Pang, W.L., M.B.I. Reaz, M.I. Ibrahimy, L.C. Low, F. Mohd-Yasin and R.A. Rahim, 2006. Handwritten character recognition using fuzzy wavelet: A VHDL approach. WSEAS Transac. Syst., 5: 1641-1647.

Reaz, M.B.I., M.T. Islam, M.S. Sulaiman, M.A.M. Ali, H. Sarwar and S. Rafique, 2003. FPGA realization of multipurpose FIR filter. Parallel and Distributed Computing, Applications and Technologies, (PDCAT'03), Chengdu. 912-915.

Reaz, M.B.I., F. Mohd-Yasin, M.S. Sulaiman, K.T. Tho and K.H. Yeow, 2004. Hardware prototyping of boolean function classification schemes for loss-less data compression. Proceedings of 2nd IEEE International Conference on Computational Cybernetics, (ICCC'04), Vienna, Austria, IEEE, 47-51.

Reaz, M.B.I., F. Mohd-Yasin, S.L. Tan, H.Y. Tan and M.I. Ibrahimy, 2005a. Partial encryption of compressed images employing FPGA. Proceedings of IEEE International Symposium on Circuits and Systems, (ISCAS'05), Kobe, Japan, IEEE. 2385-2388.

Reaz, M.B.I., P.W. Leong, F. Mohd-Yasin and T.C. Chin, 2005b. Modeling of data compression using partial matching: A VHDL approach. Proceedings of 6th World Wireless Congress, (WWC'05), Palo Alto, USA., 411-416.

Reaz, M.B.I., F. Choong and F. Mohd-Yasin, 2006. VHDL Modeling for classification of power quality disturbance employing wavelet transform, artificial neural network and fuzzy logic. Simulation Transac. Soc. Modeling Simulation Inter., 82: 867-888.

Reaz, M.B.I., M.I. Ibrahimy, F. Mohd-Yasin, C.S. Wei and M. Kamada, 2007. Single core hardware module to implement encryption in TECB mode. Informacije MIDEM J. Microelect. Elect. Compon. Materials, 37: 165-171.

Schneiderman, H. and T. Kanade, 2000. A statistical method for 3D object detection applied to faces and cars. Proc. Int. Conf. Comput. Vision Pattern Recog., 1: 746-751.

Wang, F., X. Yao and J. Han, 2007. Minimax probability machine multialgorithmic fusion for iris recognition. Inform. Technol. J., 6: 1043-1049.

Wildes, R., 1997. Iris recognition: An emerging biometric technology. Proc. IEEE, 85: 1348-1363.

Yahya, A.E. and M.J. Nordin, 2008. A new technique for iris localization in iris recognition systems. Inform. Technol. J., 7: 924-929.

Yan, X., S. Chen and X. Niu, 2007. An improved algorithm for iris location. Proceedings of International Conference on Computational Intelligence and Security, Dec. 15-19, Harbin, China, pp: 964-967.