DOI: 10.21917/ijsc.2010.0012

A NEURAL NETWORK BASED IRIS RECOGNITION SYSTEM FOR PERSONAL IDENTIFICATION

Usham Dias[1], Vinita Frietas[2], Sandeep P.S.[3] and Amanda Fernandes[4]
Department of Electronics and Telecommunication, Padre Conceicao College of Engineering, Goa, India
E-mail: ushamdias@gmail.com[1], vinita_freitas@yahoo.co.in[2], sendmail2sandeep@gmail.com[3], amanda_7616@yahoo.com[4]

Abstract
This paper presents biometric personal identification based on iris recognition using artificial neural networks. Personal identification system consists of localization of the iris region, normalization, enhancement and then iris pattern recognition using neural network. In this paper, through results obtained, we have shown that a person's left and right eye are unique. In this paper, we also show that the network is sensitive to the initial weights and that over-training gives bad results. We also propose a fast algorithm for the localization of the inner and outer boundaries of the iris region. Results of simulations illustrate the effectiveness of the neural system in personal identification. Finally a hardware iris recognition model is proposed and implementation aspects are discussed.

Keywords:
Biometric, Iiris Recognition, Artificial Neural Network

1. INTRODUCTION

In today's world, security has become very important. Iris Recognition Security System is one of the most reliable leading technologies for user identification [1]-[3]. The human iris has random texture and it is stable throughout the life, it can serve as a living passport or a living password that one need not remember but is always present [4]. The iris recognition system model is as shown in Fig 1.

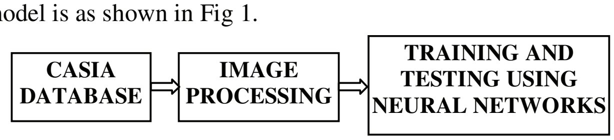

Fig.1. System Model

1.1 WHY THE IRIS? [2], [5]

- *Accuracy*: Iris recognition has highest proven accuracy and has no false matches in over two million cross – comparisons.
- *Uniqueness*: No two irises are alike. There is no detailed correlation between the iris patterns of even identical twins, or the right and left eye of an individual.
- *High information Content*: The amount of information that can be measured in a single iris is much greater than fingerprints.
- *Real time*: It allows high speed processing and the individual needs to just look into a camera for a few seconds.
- *Stability & permanence*: The iris is stable for each individual through his or her life and do not change with age.
- *Low circumvention*: Less susceptible to spoofing

2. IMAGE ACQUISITION

Iris recognition system works by first capturing a digital color image of the eye [6]. Normally a high resolution camera is used with infrared illumination facility. This research paper uses gray scale eye photographs from the CASIA database [7].

3. IMAGE PRE-PROCESSING

3.1. IRIS LOCALIZATION

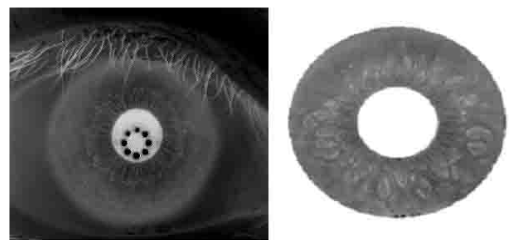

Fig.2. Isolation of iris

As shown in fig 2 the first stage of iris recognition is to isolate the actual iris region in a digital eye image. The iris region is approximated by two circles, one by the iris-sclera boundary and another, interior to the first, by the iris-pupil boundary. In Iris Localization step we calculate the centre co-ordinates and radius of the pupil and the iris.

3.1.1 The Pupil Circle Detection

Here first the edge image is obtained by edge detection algorithms and Circular Hough transform is applied to it to detect the radius and centre of iris [8].

PROPOSED METHOD

The image is first converted into a binary image by applying a suitable threshold. This is followed by an edge detector algorithm such as canny edge [9]. Finally Hough transform is applied to the edge image for the detection of pupil circle. This results in faster and accurate pupil detection.

Edge Detection: Some of the edge detection methods present are: [9] and courtesy matlab help.

1) Sobel
2) Prewitt
3) Roberts
4) Log
5) Canny

Out of these five edge detection techniques, it was found that Canny edge detection gave the best results. Thus it was implemented. Matlab allows varying two parameters of the canny function namely; *Sigma* (smoothening parameter of the

ISSN: 2229 – 6956(ONLINE)
DOI: 10.21917/ijsc.2010.0012

Gaussian filter) and *Threshold* (the upper value used in threshold hysteresis). In order to find the values which give optimum results, the value of sigma was kept constant and threshold was varied. It was found that threshold value of 0.2 gave best result. Next threshold was kept constant at 0.2 and it was found that sigma values ranging from 2 to 4 gave optimum results.

Circle Detection on Edge image* [10]*:

We first initialize a 3-dimentional Hough accumulator array to store the weights given to each pixel. All the weights are initially assigned a zero value. With each edge pixel as center we draw circles of a particular radius. Weights of all the pixels which form a part of the drawn circle are incremented by one. As the process of drawing circle goes on there will be circle overlaps. The weight of a particular pixel in the Hough array is same as the total number of circle overlaps. The maximum weight corresponds to the centre of the circle. This is carried out for all the probable values of iris radius. The centre co-ordinates of iris/pupil will be that of the edge pixel which has the maximum value in the Hough accumulator array. As shown in Fig 3 the point that has maximum overlap is indeed the center of the edge image.

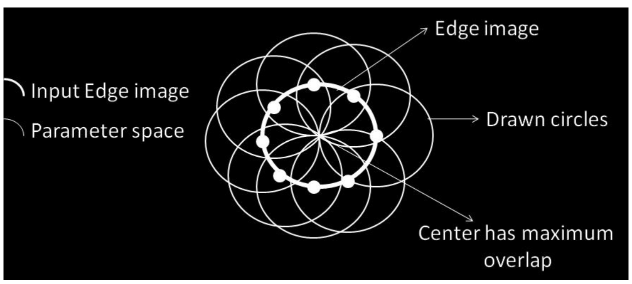

Fig.3. Circular Hough Transform

3.1.2 Iris Circle Detection

Circular Hough transform is the most widely used method for iris circle detection [8]. But this method fails in case the iris edge is not properly detected by the edge detector.

PROPOSED METHOD

The detection of iris circle is done by first scanning groups of pixels towards the sclera. In the process difference in intensities of the groups of pixels are registered. Once the iris-sclera boundary is reached, the difference in intensities becomes maximum, giving an indication of the presence of a boundary. Thus, detecting the iris accurately as shown in Fig.4

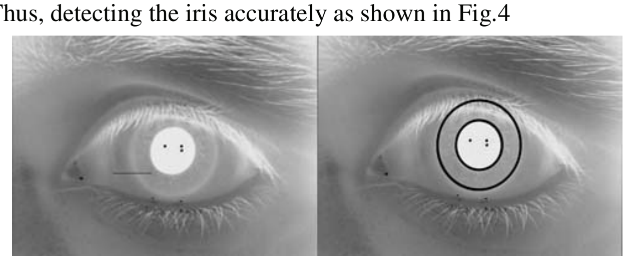

Fig.4. Iris Circle detection

The Fig.5 shows the intensity variations across the sclera boundary. As expected the intensity variations peak at a particular point. The pixel group that has the maximum value is taken. Now the radius is given by the distance of the middle pixel of the pixel group and the detected pupil center.

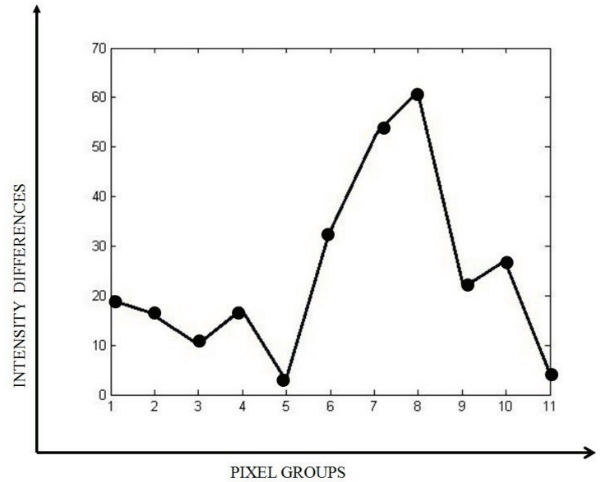

Fig.5. Graph of intensity versus pixel groups

3.2 NORMALIZATION

Once the iris region is successfully segmented from an eye image, the next stage is to transform the iris region so that it has fixed dimensions in order to allow comparisons. The original image which is in Cartesian co-ordinates is transformed into polar [11]. Thus the donut shaped iris is transformed into a rectangular strip as shown in figure 6. The normalization process will produce iris regions, which have the same constant dimensions, so that two photographs of the same iris under different conditions will have characteristic features at the same spatial location.

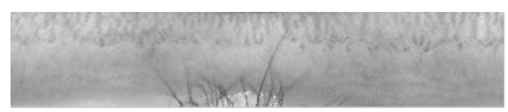

Fig.6. Iris Normalization

3.3 IMAGE ENHANCEMENT

To remove high frequency noises and also improve the contrast of projected iris ribbon, histogram equalization (HE) will be used for the iris zone as shown in fig 7. Also the effects of background illumination are removed [12]. The size of the image obtained is large and is unsuitable for giving to neural network. Thus averaging is performed on this image so as to reduce its dimensions. Now this reduced dimension template thus obtained itself will act as feature vector to the artificial neural network. Thus there is no explicit feature extraction done and performance is tested taking the intensity values itself as the features. This reduces the computations to a lot extent thereby speeding up the image preprocessing stage.

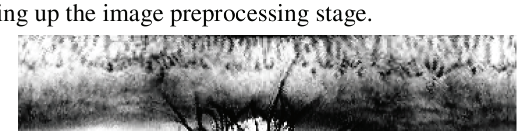

Fig 7: Histogram Equalized Iris

DOI: 10.21917/ijsc.2010.0012

4. EXPERIMENTAL RESULTS USING ANN [12], [13] and *courtesy MATLAB help*

The Artificial Neural network is used which involves training and testing of the extracted feature vectors from the iris images. In the CASIA database each person's eye (left) had 20 images, of which 10 were used for training and 10 for testing. These images were preprocessed as mentioned in the image processing section. The final result was a strip of data for each image which was fed to the neural network. All the neural networks in this paper consist of only one hidden layer. Accuracy is calculated using the formula;

Accuracy = (No. of correct classification/ total number of testing images) *100

The results obtained are as mentioned below. Under the column for transfer function two are mentioned, first for the hidden layer and second for the output layer. The choice of transfer function was random. Log indicates unipolar sigmoid transfer function, tan indicates bipolar and pure indicates linear transfer function.

4.1 CLASSIFICATION USING EBPA ALGORITHM WITH ADAPTIVE LEARNING RATE AND MOMENTUM FACTOR

These networks classify 99 people and were trained using error back propagation algorithm with adaptive learning rate (traingda). A total of 990 images were given for testing. Results are shown in table 1.

No. of inputs: 500

Table.1. Results for EBPA algorithm

Sl. No.	No. of Hidden Nodes	Transfer Function	Epoch	Accuracy
1	2500	Log, Tan	3000	86.26%
2	1500	Tan, Tan	5000	86.36%

4.1.1. Simultaneous Training And Testing

For 99 classes the network took a lot of time to train. The number of iterations was varied to get higher accuracy. This required the neural network to be retrained again from start and is therefore time consuming. Simultaneous training and testing provides some relief. Here a network is trained and tested for certain iterations (say 3000); if results are not satisfactory reload this network and train it for few more iterations, e.g. 2000 iterations. Therefore the effective number of iterations for which the network is trained is 5000. This is continued till a point of maximum accuracy.

At start EBPA algorithm with adaptive learning rate (traingda) was used. This gave problems of local minima; therefore the training algorithm was changed to EBPA with momentum factor and adaptive learning rate (traingdx).

The results for 500input and 125input are shown below.

Classification with 500 input

The network information and parameters are mentioned in Table 2. Accuracy at different epochs is shown in Fig 8.

Table.2. Network information

No. of Hidden Nodes	Transfer Function	Epoch	Accuracy
3000	Tan, Tan	--	--

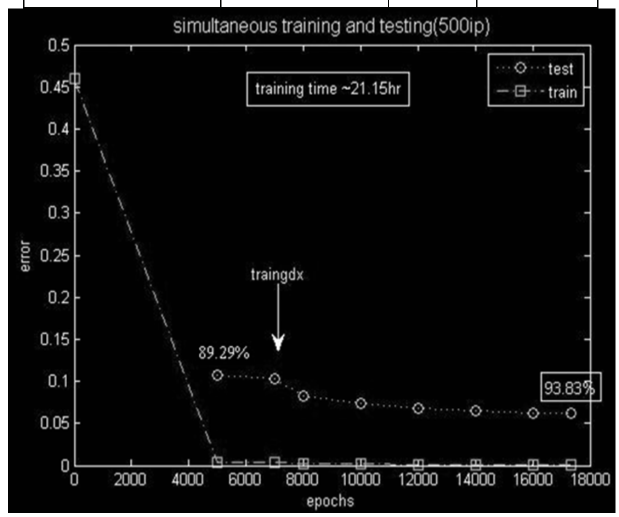

Fig.8. Simultaneous training and testing for 500 input

This gave satisfactory result with 93.83% accuracy. Total number of epochs (iterations) and approximate time taken to achieve this result is 17326 and 21:15 hours respectively.

Classification with 125 input

Here same parameters as mentioned in table 2 were used but the number of inputs was 125. Accuracy at different epochs (iterations) is shown in Fig 9.

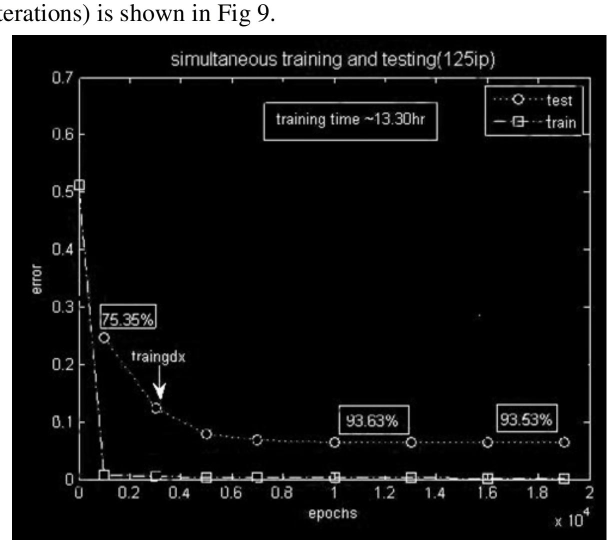

Fig.9. Simultaneous training and testing for 125 inputs

This gave satisfactory result with 93.53% accuracy. Total number of epochs (iterations) and approximate time taken to achieve this result is 19000 and 13:30 hours respectively.

4.2 TRAINING USING CONJUGATE GRADIENT ALGORITHMS

Simultaneous training and testing using traingda and traingdx algorithms helped improve the accuracy. But the training time was large. Therefore conjugate gradient algorithms were tried courtesy matlab help, which gave good results; as shown below.

Table.3. Results with Fletcher-Reeves update (traincgf)

Sl. No.	No. of hidden nodes	Transfer Function	Epoch	Apprx. time (hr.)	Accuracy
1	2000	Tan , pure	723	1:00	53.9%
2	3000	Tan, pure	1000	2:20	66.46%
3	3500	Tan , Tan	1000	5:00	91.01%
4	3000	Tan , Tan	1000	2:00	**92.83%**

Table.4. Results with Polak-Ribiere update (traincgp)

Sl. No.	No. of hidden nodes	Transfer Function	Epoch	Apprx. time (hr.)	Accuracy
1	3000	Tan , Pure	810	1:55	60.5%
2	3000	Tan, Tan	1000	2:00	**93.93%**

Table.5. Result with powell-beale restarts (traincgb)

Sl. No.	No. of hidden nodes	Transfer Function	Epoch	Apprx. time (hr.)	Accuracy
1	3500	Tan , Tan	1400	2:30	92.62%
2	3000	Tan, Tan	1000	2:30	**94.24%**

Table.6. Result with scaled conjugate gradient (trainscg)

Sl. No.	No. of hidden nodes	Transfer Function	Epoch	Apprx. time (hr.)	Accuracy
1	3500	Tan , Tan	1000	1:40	92.52%

4.3 RESULTS USING ONE STEP SECANT (QUASI NEWTON ALGORITHM) courtesy matlab help

Results with One Step Secant Algorithm (trainoss) are shown in Table.7.

Table.7. Results for One-step secant method

Sl. No.	No. of hidden nodes	Transfer Function	Epoch	Apprx. time (hr.)	Accuracy
1	2000	Tan , Tan	200	0:23	72.32%
2	3000	Tan, Tan	200	0:36	81.01%
3	3000	Tan , Tan	800	2:05	**93.13%**

4.4 YOUR LEFT EYE AND RIGHT EYE ARE UNIQUE!!!

Here left and right eye of 10 people were used to fire different nodes. The results are as in Table.8.

Table 8: Results proving uniqueness of left and right eye

Sl. No.	No. of hidden nodes	Transfer Function	Epoch	MSE (10^-3)	Accuracy
1	500	Tan, Tan	200	1.3325	95%
2	250	Tan, Tan	200	2.2559	92.5%
3	125	Tan ,Tan	200	5.6952	**95.5%**

4.5 TRAINING DEPENDS ON INITIAL WEIGHTS

Training is sensitive to the initial weights. The same program as in table 8, sr. no.3 was run. But because the weights are initialized to random values the results we got are as in table 9.

Table.9. Results for random weight initialization

Sl. No.	No. of hidden nodes	Transfer Function	Epoch	MSE (10^-3)	Accuracy
1	125	Tan , Tan	200	5.1445	91%
2	125	Tan, Tan	200	7.3774	92.5%
3	125	Tan , Tan	200	10.328	91.5%
4	125	Tan , Tan	200	5.0571	89%
5	125	Tan , Tan	200	7.2295	94.5%

4.6 THE MORE YOU TRAIN THE BETTER THE RESULTS…WELL THINK AGAIN!!!

Here the same network is loaded as in table no. 8, sr. no.3. The initial weights are kept constant throughout, only epochs are varied. The results are as in table10.As the number of epochs increase beyond 500, accuracy reduces.

Table.10. Results for over-training

Sl. No.	No. of hidden nodes	Transfer Function	Epoch	MSE (10^-3)	Accuracy
1	125	Tan , Tan	250	4.185	95.5%
2	125	Tan, Tan	300	3.3118	96.5%
3	125	Tan , Tan	400	1.96	96.5%
4	125	Tan , Tan	500	1.3557	96.5%
5	125	Tan , Tan	525	1.2386	96%
6	125	Tan , Tan	550	1.1112	96%
7	125	Tan , Tan	600	0.9017	95.5%
8	125	Tan , Tan	800	0.5119	95%

4.7 SUMMARY ON BACK PROPAGATION ALGORITHMS

There are several different backpropagation training algorithms. They have a variety of different computation and

DOI: 10.21917/ijsc.2010.0012

storage requirements, and no one algorithm is best suited for all application courtesy matlab help. Table 11 summarizes the training algorithms.

Table.11. Summary on back propagation algorithms

Function	Description	Discression
Traingd	Basic gradient descent.	Slow response, no satisfactory result
Traingdm	Gradient descent with momentum	Slow response, no satisfactory result
Traingdx	Adaptive learning rate. Faster training than traingd, but can only be used in batch mode training.	It gave satisfactory response. But took approx. 21 hours of training
Trainrp	Resilient backpropagation. Simple batch mode training algorithm with fast convergence and minimal storage requirements.	No appreciable results
Traincgf	Fletcher-Reeves conjugate gradient algorithm. Has a smallest storage requirement out of the conjugate gradient algorithms.	Fast training (~2 hr), highest accuracy =92.83%
Traincgp	Polak-Ribiére conjugate gradient algorithm. Slightly larger storage requirements than traincgf. Faster convergence on some problems.	Fast training (~2 hr), highest accuracy =93.93%
Traincgb	Powell-Beale conjugate gradient algorithm. Slightly larger storage requirements than traincgp. Generally faster convergence.	Fast training (~2:30hr), highest accuracy =94.24%
Trainscg	Scaled conjugate gradient algorithm. The only conjugate gradient algorithm that requires no line search. A very good general purpose training algorithm.	Fast training (~1:40 hr), highest accuracy=92.52%
Trainbfg	BFGS quasi-Newton method. Requires storage of approximate Hessian matrix and has more computation in each iteration than conjugate gradient algorithms, but usually converges in fewer iterations BFGS quasi-Newton method	No results. System ran out of memory.
Trainoss	One step secant method. Compromise between conjugate gradient methods and quasi-Newton methods	Fast training (~2:05 hr), highest accuracy =93.13%
Trainlm	Levenberg-Marquardt algorithm. Fastest training algorithm for networks of moderate size. Has memory reduction feature for use when the training set is large.	Our system ran out of memory. No results dispite the use of reduction in memory feature
Trainbr	Bayesian regularization. Modification of the Levenberg-Marquardt training algorithm to produce networks that generalize well. Reduces the difficulty of determining the optimum network architecture	No results System ran out of memory.

5. PROPOSED HARDWARE IMPLEMENTATION

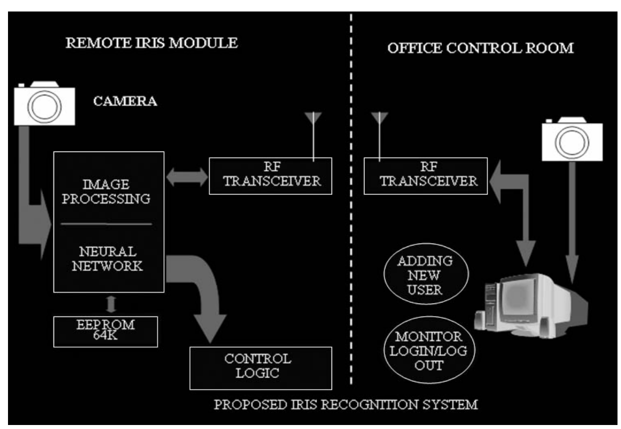

Fig.10. Proposed System Model

The proposed system model is as shown fig 10. The remote iris module does the testing while training is primarily done in the office control room. If a user is identified by the iris module a control logic is activated. The user identity and time of login is sent to the server through a wireless network. A new user can be enrolled only by the administrator at the office control room. This requires updating the network in remote iris module via the RF link.

REMOTE IRIS MODULE (RIM):

It consists of a core capable of image processing, testing and training the neural network. High speed DSP processors may be used for image processing. The network is stored in the EEPROM interfaced with processor. The feature vectors of all the people in the database are also stored in the EEPROM.

Neural Network chips are used for training and testing. Testing is done by the Module independently using the stored network. But training is done under the complete control of the office control room server.

OFFICE CONTROL ROOM:

Continuously monitors login and logout. It keeps track of the number and identity of people entered. A new user in enrolled by taking the iris photographs and processing the images using the same algorithm used in remote iris module DSP processor. Now the feature vectors extracted are stored in the EEPROM of RIM.

Now during training, the RIM is shown offline and the neural network processor in RIM is trained all over again including the feature vectors of the newly enrolled user. Once the training is done the RIM is again brought online. The RIM is now ready with new user in the database.

Iris Recognition system can be implemented in hardware in two ways. a) Microcontroller based. b) Single chip (ASIC). A suitable implementation is using Microcontroller.

Image processing can also be done on various DSP processors available. Some of the DSP processors optimized for image processing are:

I) Texas Instruments TMS32067XX

II) Analog devices Blackfin processor

Neural Network processors available are

I) NeuroMatrix® NM6403 RISC/DSP Microprocessor

II) Accurate Automation Corporation (NNP)

III) The CM1K neural network chip from Recognetics

6. CONCLUSION

A biometric system based on Iris using Neural Network was presented. The Iris localization method proposed uses circular hough transform only for pupil circle detection. Iris circle detection is done by calculating the intensity values of groups of pixels, which reduces the amount of computation that would have been needed if circular hough transform was used. Before applying circular hough transform for pupil circle detection, the image is converted to black and white, and then edge detection is done which reduces the unwanted edges drastically. It was found that out of the many edge detection techniques canny edge detection gave the best results for threshold 0.2 and sigma between 2 and 4.

For recognition neural network is used. The network is trained for 99 people using 990 train images. Testing was carried out using 990 test images and it was found that conjugate gradient algorithm gave best results. Highest accuracy of 94.24% was achieved with Powell-Beale conjugate gradient algorithm. In this paper, we have also shown that a persons left and right eye are unique. In addition, this paper also shows that the training is sensitive to the initial weights and that over-training gives bad results.

ACKNOWLEDGEMENT

We would like to thank Dr. K. R. Pai and Ms. Supriya Patil from Padre Conceicao College of Engineering, Goa for their sincere support and guidance.

REFERENCES

[1] Daugman, J., 2004, “How Iris Recognition Works”, IEEE Transactions on Circuits and Systems for Video Technology, Vol. 14, No. 1, pp.21 – 30.

[2] Chen, Wen-Shiung and Yuan, Shang-Yuan, “An Automatic iris recognition system based on fractal dimension”, VIP-CC Lab., Department of Electrical Engineering, National Chi Nan University.

[3] C.H. Seal, M.M Gifford and D. J. McCartney, 1993, “Iris Recognition for user validation”, IEEE transactions on pattern analysis and Machine Intelligence.

[4] O. Sirvan, B. Karlik, and A. Ugur, 2005, “An Efficient Iris Recognition for security purposes”, GVIP 05 Conference, pp.19-21.

[5] Srinivasa Kumar Devireddy, G.Ramaswami, D.Ravikiran, PSirisha Rani, 2009, “A novel appoarch for an accurate human identification through iris recognition using bitplane slicing and normalisation”, Journal of Theoretical and Applied Information Technology, JATIT

[6] Kefeng FAN, Qingqi Pei, Wei MO, Xinhua Zhao and Qifeng Sun, 2006, “An Efficient Automatic Iris Image Acquisition and Preprocessing System”, Proceedings of the 2006 IEEE International Conference on Mechatronics and Automation, pp.1779-1784, Luoyang, China.

[7] Chinese Academy of Sciences – Institute of Automation, Database of 756 Greyscale Eye Images. http://www.sinobiometrics.com Version 1.0, 2003.

[8] Libor Masek, 2003, “Recognition of Human Iris Patterns for Biometric Identification”, School of Computer Science and Soft Engineering, The University of Western Australia.

[9] Bill, G., “Canny Edge Detection Tutorial”, 2002, http://www.pages.drexel.edu/~weg22/can_tut.html.

[10] Mohamed Rizon et al, “Object Detection using Circular Hough Transform”, American Journal of Applied Sciences, Vol.2, No.12, pp.1600-1605

[11] Lye Wil Liam, Ali Chekima, Liau Chung Fan and Jamal Ahmad Dargam, 2002 “Iris recognition using self organizing neural network”, Student conference on Research and development proceedings, Shah Alam, Malaysia.

[12] Rahib H. Abiyev, Koray Altunkaya, 2007, “Neural Network Based Biometric Personal Identification”,

DOI: 10.21917/ijsc.2010.0012

Frontiers in the Convergence of Bioscience and Information Technologies.

[13] Watts, M. and Woodsford, B., "INFO 331 2003 Laboratory Workbook", Chapter.6, Lab.6: MLP and Backpropagation.