Human Iris Recognition using Elman Neural Networks

Shaymaa M. Al-mashahadany
Dept of Computer Sciences / College of Computer Sciences & Math.
University of Mosul

Received 28 / 01 / 2009
Accepted 03 / 06 / 2009

الملخص

تعد قزحية عين الإنسان واحدة من أهم المقاييس الحيوية في جسم الإنسان لتمييز الأنماط. تم وصف نظام لتمييز القزحية والذي يتكون من جزئين رئيسين هما: معالجة الصورة ومن ثم تمييزها.

معالجة الصورة تتبع اكتساب الصورة، ثم تقطيع الصورة، وتحديد موقع القزحية وذلك بتحديد موقع الحدود الداخلية والخارجية لمنطقة القزحية، ومن ثم إلغاء تأثير الرموش والأجفان.

لتجاوز اختلاف منطقة القزحية المحددة يتم تحويلها إلى شكل مستطيل بأبعاد ثابتة تمثل مجموعة البيانات بإتباع طريقة المعايرة. يتم استخدام هذه البيانات كإدخال للشبكة العصبية المستخدمة في تصنيف القزحية. يمكن تطبيق الشبكة المصممة على بيانات القزحية لتحديد صورة قزحية أي شخص. تم استخدام برنامج MATLAB®٢٠٠٨a لهذا الغرض.

ABSTRACT

The human iris is one of the best biometric features in the human body for pattern recognition. In this paper, the iris recognition system is described that consist of two main parts: image processing and its recognition.

Image processing follows the capture of image, then followed by image segmentation, then localization of iris region by determining the inner and outer boundaries of the iris region and by removing the eyelashes and eyelids.

To overcome the discrepancies in the iris region, a rectangular shape with constant dimensions is proposed, which is represented by a set of data following the normalization procedure.

Using these data set as input data of the Elman Artificial Neural Networks in order to classify the iris patterns. The adaptive learning strategy could be applied to verify each iris image of a particular person. This is done by using MATLAB®2008a.

1. Introduction

Biometrics technology plays important role in public security and information security domains. Using various physiological characteristics of human, such as face, facial thermograms, fingerprint, iris, retina, hand geometry etc., biometrics accurately identify each individual and distinguish one from another [1]. Iris recognition is one of important biometric recognition approach in a human identification is becoming very active topic in research and practical application. Iris region is the part between the pupil and the white sclera. This field is sometimes called iris texture. The iris texture provides many minute characteristics such as freckles, coronas, stripes, furrows, crypts, etc [2,3].

These visible characteristics are unique for each subject. Such unique feature in the anatomical structure of the iris facilitates the differentiation among individuals. The human iris is not changeable and stable. From one year of age until death, the patterns of the iris are relatively constant over a person's lifetime [1,4]. Because of this uniqueness and stability, iris recognition is a reliable human identification technique.

This paper will be divided into two parts: the first one describes the preprocessing techniques, the second one uses data generated by the first part of the paper for pattern classification using an Elman Neural Network.

Iris recognition consists of the iris capturing, pre-processing and recognition of the iris region in a digital eye image. Iris image pre-processing includes pupil detection, iris localization, normalization, and enhancement. Each of these steps uses different algorithms. In iris localization step, the determination of the inner and outer circles of the iris and the determination of the upper and lower boundaries of the eyelids are performed. The inner circle is located between the iris and pupil boundary, the outer circle is located between the sclera and iris boundary.

Today with the development of Artificial Intelligence (AI) algorithms, a speed is gained in iris recognition systems, hardware simplicity, accuracy and learning ability. In this paper an iris recognition system based on neural networks is proposed.

This paper is organized as follows: section 1 iris image acquisition; section 2 iris preprocessing steps that include iris localization, normalization and enhancement; section 3 neural network used for iris pattern recognition; section 4 conclusions.

1.1 Iris Biometric Features

The use of the human iris as a biometric feature offers many advantages than other human biometric features. The iris is the only internal human body organ that is visible from the outside [5,6], thus well protected from external modifiers. A fingerprint for example may suffer transformations due to harm or aging, voice patterns may be altered due to vocal diseases. Yet, the human iris image is relatively simple to image and may be done so in a non-intrusive way. The human iris starts forming still in the mother uterus, in the third month of gestation [7,8] and the visible structures are formed by the eighth month. [5,6] It is particularly good for automatic recognition because of its complex pattern of many distinctive features such as arching ligaments, furrows, ridges, crypts, rings, corona, freckles, and a zigzag collarette [9]. Some of these patterns may be seen in figure (1).

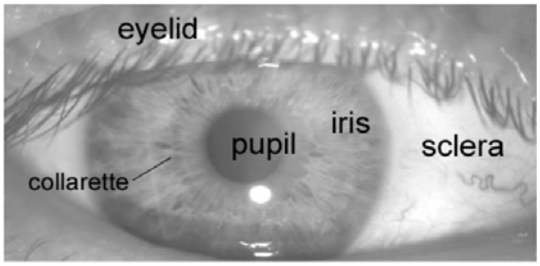


Fig. (1): Human iris with pupil and biometric features

2. Iris recognition system

Figure (2) bellow shows an overview of the iris recognition system steps taken in this paper.

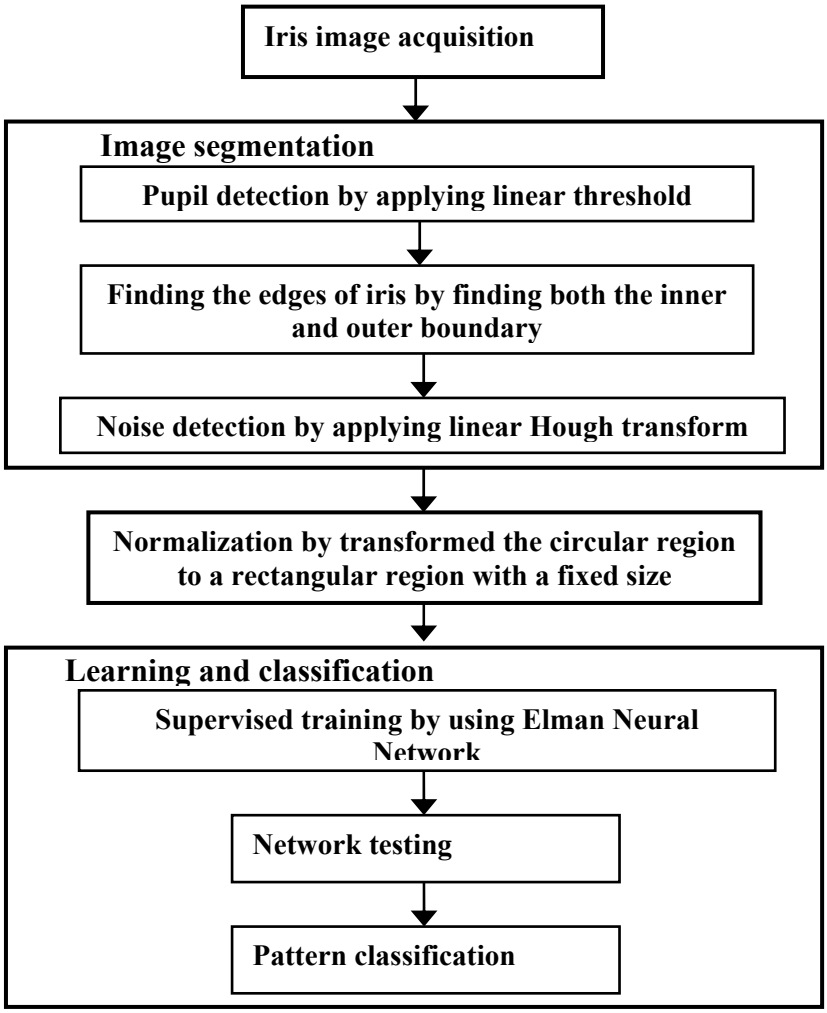


Fig. (2): The structure of the iris recognition system

2.1 Iris Image Acquisition

An important and difficult step of an iris recognition system is the image acquisition. Since the iris is small in size and dark in color (especially for Asian people), it is difficult to acquire good images for analysis using the standard CCD camera and ordinary lighting. The acquired image always contains not only the “useful” parts (iris) but also some “irrelevant” parts (e.g. eyelid, pupil,… etc.). Under some conditions preprocessing is needed on the original captured image. Usually the brightness is not uniformly distributed. In addition, the distance between the eye and the camera may result in different image sizes of the same eye [9,10].

The CASIA iris image database 2 is used, which is collected by Institute of Automation, Chinese Academy of Sciences. Images are 320x280 pixels gray scale taken by a digital optical sensor designed by NLPR (National Laboratory of Pattern Recognition) [11].

2.2 Image Segmentation

Several researches were made in the subject of iris finding and segmentation. The main objective here is to remove non useful information, namely the pupil segment and the part outside the iris (sclera, eyelids, skin). Wildes [12] uses Hough transforms to detect automatically the iris contour. Daugman [13] proposes an integral differential operator to find both the pupil and the iris contour.

The segmentation of the iris region is to localize the actual iris region in a digital eye image. Two circles can approximate the iris region, one for the iris/sclera boundary and another, interior to the first, for the iris/pupil boundary. The eyelids and eyelashes normally occlude the upper and lower parts of the iris region. Also, specular reflections can occur within the iris region corrupting the iris pattern [14]. A segmentation technique is required to isolate and exclude these artifacts as well as locate the circular iris regions.

2.2.1 Pupil Detection

To find the pupil, a linear threshold is applied on the eye image i.e. pixels with intensity less than a specified empirical value are converted to 0 (black) and pixels greater than or equal to the threshold are assigned 1 (white). Freeman's chain code algorithm [15] is used to find regions of 8-connected pixels having the value 0. It is also possible that eyelashes may satisfy the threshold condition, but they have a much smaller area than the pupil. Using this knowledge, we can cycle through all regions and apply the Freeman's chain code algorithm to retrieve the black pupil in the image. From this region, the central moment is obtained [4,16]. The edges of the pupil are found by creating two imaginary orthogonal lines passing through the centroid of the region. Starting from the center to both the extremities, boundaries of the binarized pupil are defined by the first pixel with intensity 1.

2.2.2 The Iris Detection

Having acquired images of iris, the next step is to isolate the iris from the rest of the image, namely, finding both the inner (pupillary) boundary and the outer (sclera) boundary [17].

The algorithm for finding the edges of the iris from eye image I(x, y) is as follows:

1. Center of pupil (C_{px}, C_{py}) and radius r_p are known using the pupil detection algorithm.
2. Apply Linear Contrast Filter on image I(x,y) to get the linear contrasted image P(x, y).
3. Create vector A = {a1, a2,…, aw} that holds pixel intensities of the imaginary row passing through the center of the pupil, with w being the width of the image P(x, y).
4. Create vector R from the vector A which contains elements of A starting at the right fringe of the pupil and ending at the right most element of vector A. Similarly, another vector L is created which contains elements of A starting at the left fringe of the pupil and ending at the left most element of vector A.
5. For each side of the pupil (vector R for the right side and vector L for the left side):
   a. Calculate the average window vector Avg = {b1,…,bn}

      where n = |L| or n = |R|. Vector Avg is subdivided into I windows of size z. The value of every element in the window is replaced by the mean value of that window.
   b. Locate the edge point for both the vectors L and R as the first increment in value of Avg that exceeds a threshold t.

Thus, the pupil, the iris center and the radius are calculated and a circle is drawn using these values to locate the pupil and iris edges as shown in figure (3).

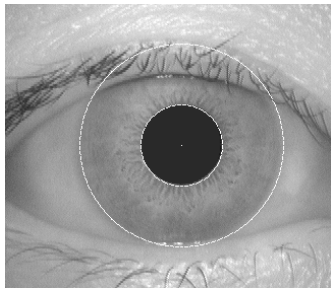


Fig. (3): Pupil and Iris edges Detected

2.2.3 Eyelash and Eyelid Detection

The eyelashes and eyelids are the upper and lower regions of the iris. The search space depends on the pupil radius. The upper eyelid is considered as the area over the pupil from the upper of image to a pre-defined point. The pre-defined point is calculated as the difference between the center of pupil in X direction and the pupil radius. The lower eyelid is considered as the area under the pupil that is from the lower of image to the pre-defined point. Here the pre-defined point is the result of

the addition of the center of pupil in X direction and the pupil radius. This gives the eyelids region as shown in figure (4).

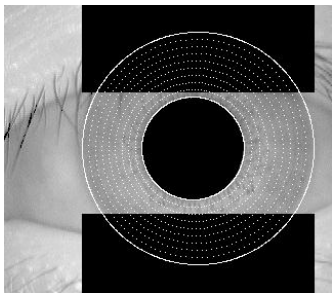


Fig. (4): Eyelashes and eyelid in an eye image

Because the isolated eyelashes appear as lines, it is decided to apply a filter coefficient for directional edge detection [9,18]. The linear Hough transform is applied to find the line coordinates. The multiple eyelashes and eyelids are detected using a threshold. Each value which is lower than a threshold is considered as noise (eyelash or eyelid).

2.3 Iris normalization

The irises captured from the different people have different sizes. The size of the irises from the same eye may change due to illumination variations, distance from the camera, or other factors. At the same time, the iris and the pupil are non concentric. These factors may affect the result of iris matching. In order to avoid these factors and to achieve more accurate recognition, the normalization of iris images is implemented. In normalization, the iris circular region is transformed to a rectangular region with a fixed size. With the boundaries detected, the iris region is normalized from Cartesian coordinates (x,y) to polar representation (r,θ). This operation is done using the following equations defining the important points marked in figure (5.b). Points P_a through P_d are interpolated along line segments P_1-P_3 and P_2-P_4 [9, 19].

$$P_1 = C_p + R_p (\cos(\theta) - \sin(\theta)) \quad \ldots \quad (8\text{ a})$$

$$P_2 = C_p + R_p (\cos(\theta + d\theta) - \sin(\theta + d\theta)) \quad \ldots \quad (8\text{ b})$$

$$P_3 = C_i + R_i (\cos(\theta) - \sin(\theta)) \quad \ldots \quad (8\text{ c})$$

$$P_4 = C_i + R_i (\cos(\theta + d\theta) - \sin(\theta + d\theta)) \quad \ldots \quad (8\text{ d})$$

$$P_a = P_1\left(1 - \frac{k}{N}\right) + \left(\frac{P_3 k}{N}\right) \quad \ldots \quad (9\text{ a})$$

$$P_b = P_1\left(1 - \frac{k+1}{N}\right) + \left(\frac{P_3 (k+1)}{N}\right) \quad \ldots \quad (9\text{ b})$$

$$P_c = P_2\left(1-\frac{k}{N}\right)+\left(\frac{P_4 k}{N}\right) \quad \ldots \quad (9\text{ c})$$

$$P_d = P_2\left(1-\frac{k+1}{N}\right)+\left(\frac{P_4 (k+1)}{N}\right) \quad \ldots \quad (9\text{ d})$$

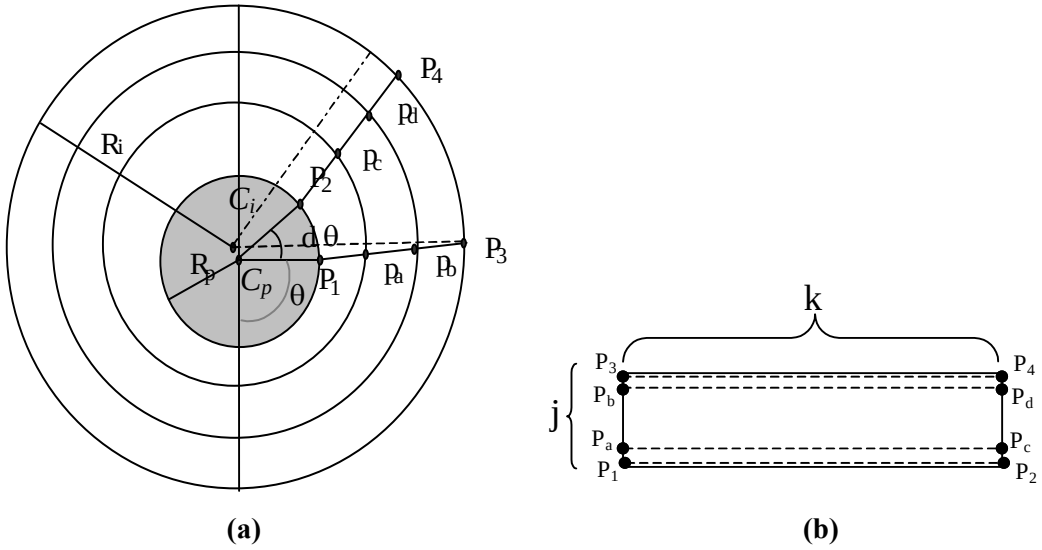


Fig. (5): The normalization technique used to equalize the iris a): Cartesian. b): Polar.

After normalization of the iris from Cartesian coordinates to polar representation, the dimension of 8 columns and 128 rows are obtained as shown in figure (6). This corresponds to N=128 wedges, each of angle $2\pi/128$, with each wedge divided radically into 8 sections.

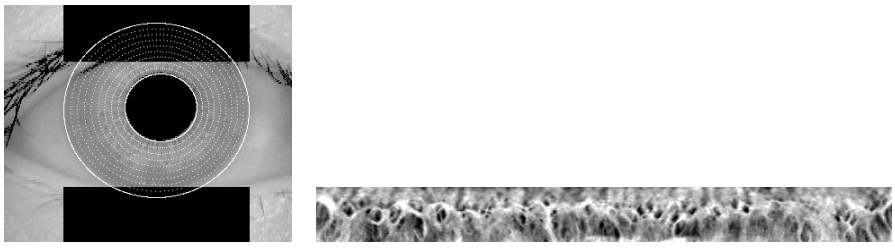


Fig. (6): The Cartesian and polar coordinates of an iris

2.3.1 Normalization Implementation

For the normalization of the iris regions, the center of the pupil is considered as the reference point, and radial vectors pass through the iris region. A number of data points are selected along each radial line and is defined as the radial resolution. The number of radial lines going around the iris region is defined as the angular resolution. Since the pupil could

be non-concentric to the iris, a remapping formula is needed to rescale points depending on the angle around the circle. The normalized pattern is created by backtracking to find the Cartesian coordinates of data points from the radial and angular position in the normalized pattern. From the "doughnut" iris region, normalization produces a 2D array with horizontal dimension as angular resolution and vertical dimension as radial resolution as shown in figure (7.a). Another 2D array is created to mark reflections, eyelashes, and eyelids detected in the segmentation stage as shown in figure (7.b). In order to prevent non-iris region data that corrupting the normalized representation, data points occurring along the pupil border or the iris border are discarded [14].

(a) **(b)**

Fig. (7): Normalized of an iris.
a) Iris template b) Corresponding noise mask

5. Neural Networks

Artificial Neural Networks (ANN's) are programs designed to simulate the way of a simple biological nervous system that is believed to operate. They are based on simulated nerve cells or neurons, which are joined together in a variety of ways to form networks. These networks have the capacity to learn, memorize and create relationships amongst data. ANN is an information-processing paradigm, implemented in hardware or software that is modeled after the biological processes of the brain. An ANN is made up of a collection of highly interconnected nodes called neurons or processing elements. A node receives weighted inputs from other nodes, sums these inputs, and propagates this sum through a function to other nodes. This process is analogous to the actions of a biological neuron. An ANN is learned by an example. In a biological brain, learning is accomplished as the strengths of the connections between nodes. This is true for ANN's also as these strengths are captured by the weights between the nodes. ANN's are most important advantage that they can be used to solve problems of considerable complexity; problems that do not have an algorithmic solution or for which such a solution is too complex to be found. Because of their abstraction from the brain, ANN's are good in solving problems of those humans but computers are not. Pattern recognition and classification are examples of problems that are well suitable for ANN application.

Neural Networks are very diverse fields; and many researchers been invested a great deal of time trying to figure out new ideas and new networks [20].

For the problem in hand, supervised network training and classification, several network models are available, and we've chosen "Elman Neural Networks" model because it is very well suitable for supervised problems.

5.1 Network Architecture

The Elman network is commonly a two-layer network with feedback from the first-layer output to the first layer input. This recurrent connection allows the Elman network to both detect and generate time-varying patterns. A two-layer Elman network is shown in figure (8).

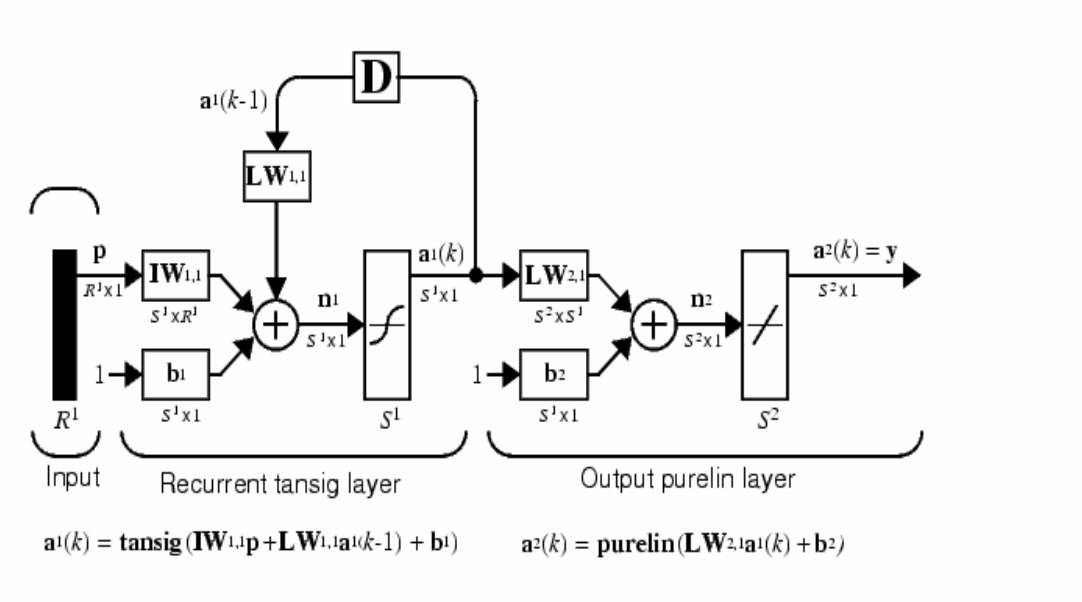


Fig. (8): Elman Neural Network Architecture

The Elman network has tansig (see appendix A) neurons in its hidden (recurrent) layer, and purelin (see appendix B) neurons in its output layer. This combination is special in that two-layer networks with these transfer functions and can approximate any function (with a finite number of discontinuities) with arbitrary accuracy. The only requirement is that the hidden layer must have enough neurons. More hidden neurons are needed as the function being fit increasing in complexity. Note that the Elman network differs from conventional two-layer networks in that the first layer has a recurrent connection. The delay in this connection stores values from the previous time step, which can be used in the current time step. Thus, even if two Elman networks, with the same weights and biases, are giving identical inputs at a given time step, their outputs can be different due to different feedback states.

Because the network can store information for future reference, it is able to learn temporal patterns as well as spatial patterns. The Elman network can be trained to respond to, and to generate, both kinds of patterns [21].

Newelm function from MatlabR2008a is used to create Elman network as follows:

Net = newelm (P,T,[S1S2...S(N-l)],{TF1TF2...TFNl},
BTF,BLF,PF,IPF,OPF,DDF)

takes these arguments,

P R x Q1 matrix of Q1 sample R-element input vectors
T SN x Q2 matrix of Q2 sample SN-element input vectors
Si Size of ith layer, for N-1 layers, default = [].
(Output layer size SN is determined from T).
TFi Transfer function of ith layer. (Default = 'tansig' for hidden layers and 'purelin' for output layer).
BTF Backpropagation network training function (default = 'trainlm')
BLF Backpropagation weight/bias learning function (default = 'learngdm')
PF Performance function (default = 'mse')
IPF Row cell array of input processing functions. (Default = {'fixunknowns','removeconstantrows','mapminmax'})
OPF Row cell array of output processing functions. (Default = {'removeconstantrows','mapminmax'})
DDF Data divison function (default = 'dividerand')

and returns an Elman network.

Algorithm

Elman networks consist of Nl layers using the dotprod weight function, netsum net input function, and the specified transfer function.

The first layer has weights coming from the input. Each subsequent layer has a weight coming from the previous layer. All layers except the last one have a recurrent weight. All layers have biases. The last layer is the network output. Each layer's weights and biases are initialized with Initnw.

Adaption is done with trains, which updates weights with the specified learning function. Training is done with the specified training function. Performance is measured according to the specified performance function[21].

5.2 Network Design

Followings are some parameters set for network training:

Number of layers: 3 layers
Number of nodes in input layer: 128X8 nodes (1024 nodes)
Number of nodes in hidden layer: 5 nodes

Number of nodes in output layer: 40 nodes for 40 people
Training function: trainlm
Initial learning rate: 0.3
Learning rate increment: 1.2
Epochs: 10,000
Error goal: 0.0000005

The iris images needed for this study were captured in the form of 320 X 280 pixel size JPEG format. Later, after segmentation and normalization of these images were transformed into 128 X 8 pixel, that perform 1024 pixel implement the number of nodes in input layer and as mentioned earlier, the number of neurons in the output layer corresponds to the number of classes to recognize which is 40 nodes implements 40 person. When the network is trained in supervised mode, a target vector is also presented to the network. This target vector T has every element set to zero, except on the position of the target class that will be set to 1.

In the system we train 100 iris image and 40 images for testing. The idea behind this design decision is that for each input pattern X presented to the network, an output vector Y is produced. This vector has the same number of elements of output neurons. Each output neuron implements a squashing function that produces a Real number in the range [0,1]. To determine which class is being indicated by the network, the maximum number in Y is selected and set it to 1, while setting all other elements to zero. The element set to one indicates the classification of that input pattern.

6. Conclusions

Iris recognition, as a biometric technology, has great advantages, such as variability, stability and security, thus it is the most promising for high security environments and will have a variety of applications. In this paper, the image was captured in the form 320X280 Jpeg, and applied many algorithms to localize the iris boundaries, after segmentation and normalization obtaining a 128X8 iris image. The located iris after pre-processing is represented by a feature vector. Using this data as a vector of input signal, the Elman neural network is used to recognize the iris patterns by using MatlabR2008a. Elman Neural Network is able to train the iris data to verify each iris image belong to which particular person. The system is able to recognize the sample of irises in training with 100 iris images of 99.5 training recognition rate and testing with 40 images of 87.5 testing recognition rate.

References

1. C. Tisse, L. Martin, L. Torres, and M. Robert. "Person Identification Technique Using Human Iris Recognition". Proc. Vision Interface, pp.294-299, 2002.

2. J. Daugman and C. Downing, "Recognizing iris texture by phase demodulation". IEEE Colloquium on Image Processing for Biometric Measurement, vol.2, pp.1-8, 1994.

3. V. Dorairaj, N. Schmid, and G. Fahmy. "Performance Evaluation of Iris Based Recognition System Implementing PCA and ICA Techniques". Proc. SPIE 2005 Symp. On , Orlando, 2005.

4. Roger F. Larico Chavez, Yuzo Iano and Vicente B. Sablon. "Process of Recognition of Human Iris: Fast Segmentation of Iris". www.decom.fee.unicamp.br/~rlarico/iris/ localizationiris.pdf.

5. J. G. Daugman, "High Confidence Visual Recognition of Persons by Test of Statistical Independence", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.15, No. 11, pp. 1148–1161, 1993.

6. J. G. Daugman, "How Iris Recognition Works", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 14, Number 1, January 2004.

7. W. W. Boles and B. Boashash, "A Human Identification Technique Using Images of the Iris and Wavelet Transform", IEEE Transactions on Signal Processing, Vol. 46, No. 4, 1998, pp. 1185-1188.

8. P. Kronfeld, "Gross anatomy and embryology of the eye In The Eye", H. Davson, Ed. London, U.K.: Academic, 1962.

9. Maha A. H. Al-Gurair, "Biometric Identification Based on Improved Iris Recognition Techniques", Ph.D. Thesis, College of computer Sciences & Math., University of Mosul, Iraq, 2006.

10. Sharp Robin, "User Authentication", Computer Security ©, Informatics and Mathematical Modeling, Technical University of Denmark, 2005.

11. CASIA iris image database, Institute of Automation, Chinese Academy of Sciences,[http://www.sinobiometrics.com]

12. R. Wildes. "Iris recognition: an emerging biometric technology". Proceedings of the IEEE, Vol. 85, No. 9, 1997.

13. K. Jaemin, Cho Seongwon, Choi Jinsu and Robert J. (2004), "Iris Recognition Using Wavelet Features"**,** Journal of VLSI Signal Processing, Vol. 38, pp. 147-156.

14. H. Freeman, "Computer Processing of Line - Drawing Images", Computer Surveys, Vol. 6, No. 1, 1974, pp. 57-97.

15. Gonzalez and Woods, "Digital Image Processing", Second Edition, Pearson Education,2001.

16. S. Taha and A. Nurdan, "Biometric Recognition Systems", IJCI Proceedings of International Conference on Signal Processing, Vol. 1, No. 2, pp. 136-139, 2003.

17. M. Stephane, "A Wavelet Tour of Processing", Academic Press London, 2nd edition, 1999.

18. Lee Tai Sing, "Image Representation Using 2D Gabor Wavelets", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 18, No. 10, pp. 959-971,1996.

19. Y. Ozbay and B. Karlik. "A Fast Training Back- Propagation Algorithm on Windows", Proceedings of the Third International Symposium on Mathematical & Computational Applications, pp. 204-210, 4-6 September, 2002, Konya, Turkey.

20. Vogl, T. P., J. K. Mangis, A. K. Rigler, W. T. Zink, and D. L. Alkon, "Accelerating the convergence of the backpropagation method Biological Cybernetics", Vol. 59, pp. 257-263, 1988.

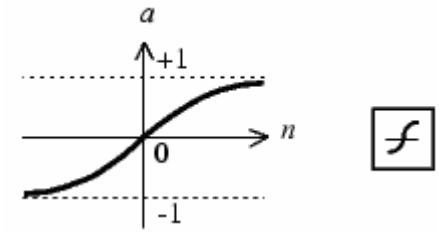
21. J. L Elman, "Finding structure in time", cognitive science, vol.14, 0, pp 179-211,1990.

Appendix A

Tansig

Hyperbolic tangent sigmoid transfer function

Graph and Symbol



a = tansig(n)

Tan-sigmoid transfer function

Syntax

A = tansig(N,FP)
dA_dN = tansig('dn',N,A,FP)
info = tansig(code)

Description

tansig is a neural transfer function. Calculate a layer's output from its net input.

tansig(N,FP) takes N and optional function parameters,
N SxQ matrix of net input (column) vectors
FP Struct of function parameters (ignored)
and returns A, the SxQ matrix of N's elements squashed into [-1 1].

tansig('dn',N,A,FP) returns the derivative of A with respect to N. If A or FP is not supplied or is set to [], FP reverts to the default parameters, and A is calculated from N.

tansig('name') returns the name of this function.
tansig('output',FP) returns the [min max] output range.
tansig('active',FP) returns the [min max] active input range.
tansig('fullderiv') returns 1 or 0, depending on whether dA_dN is SxSxQ or SxQ.
tansig('fpnames') returns the names of the function parameters.
tansig('fpdefaults') returns the default function parameters.

Algorithm

a = tansig(n) = 2/(1+exp(-2*n))-1

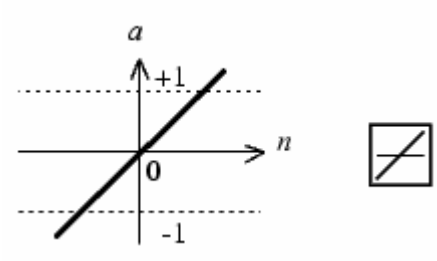
This is mathematically equivalent to tanh(N). It differs in that it runs faster than the MATLAB® implementation of tanh, but the results can have very small numerical differences. This function is a good tradeoff for neural networks, where speed is important and the exact shape of the transfer function is not.

Appendix B

Purelin

Linear transfer function

Graph and Symbol



a = purelin(n) = n
Linear transfer function

Syntax

```
 A = purelin(N,FP)
dA_dN = purelin('dn',N,A,FP)
info = purelin(code)
```

Description

purelin is a neural transfer function. Transfer functions calculate a layer's output from its net input.
purelin(N,FP) takes N and optional function parameters,

N SxQ matrix of net input (column) vectors
FP Struct of function parameters (ignored)
and returns A, an SxQ matrix equal to N.

purelin('dn',N,A,FP) returns the SxQ derivative of A with respect to N. If A or FP is not supplied or is set to [], FP reverts to the default parameters, and A is calculated from N.
purelin('name') returns the name of this function.
purelin('output',FP) returns the [min max] output range.
purelin('active',FP) returns the [min max] active input range.
purelin('fullderiv') returns 1 or 0, depending on whether dA_dN is SxSxQ or SxQ.
purelin('fpnames') returns the names of the function parameters.
purelin('fpdefaults') returns the default function parameters.

Algorithm

a = purelin(n) = n