Automatic Iris Recognition Using Neural Networks and Wavelet

Mohamed Elgendi[*a], Mohamed El-Adawy[b], Hisham Keshk[c]

[a]Faculty of Engineering, October 6 University, Egypt;
[b,c]Faculty of Engineering, Helwan University, Egypt

ABSTRACT

It's known that any iris recognition system is composed of four steps: iris image acquisition, iris texture isolation, features extraction, and classification. In the acquisition phase, our system begins with identifying the human iris automatically in a given image using neural networks to test whether it contains iris or not .If the iris exists in the acquired image it goes to the second step which is the iris texture isolation to localize the pupil boundaries, localize the iris boundaries, extract the iris texture, convert the iris texture to the polar coordinate system and then equalize the histogram of the rectangular iris texture. In the features extraction phase, our system uses the Haar wavelet to extract 72 features. In the classification phase, our system uses the matching ratio to identify or reject the subject.

Keywords: Iris Recognition, Iris Identification, Iris Biometrics

1. INTRODUCTION

Iris (as shown in Fig. 1) is layered beneath the cornea and has distinctive patterns that are intricate, richly textured, and composed of many furrows and ridges.

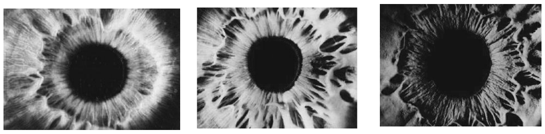

Fig.1: Iris Patterns

The uniqueness of the iris texture is complex enough to be used as a biometric signature [1]. Compared with other biometric features such as face and fingerprint, iris patterns are more stable and reliable. It is unique to people and stable with age. The irises of identical twins are clearly distinguishable. Furthermore, iris recognition systems can be non-invasive to their users [2-3].

2. BACKGROUND

Finding the human iris automatically in a scene is a significant problem. It is the first step in a fully automatic human iris recognition system. Most of well-known iris recognition systems [2-3] assume that the iris image (as shown in Fig. 2) is only presented in the front of the camera. But if the image in the front of the camera is not really iris image (as shown in Fig. 3) our system should be able to detect this difference.

In the process of making iris recognition system, it is essential to know most of all iris recognition systems that already applied and what techniques they are used. Thus, we will simplify our search in this field by a comparative brief look (as shown in Table 1) from the recent remarkable works.

[*] mohu3@ieee.org ; phone +20 10 510 2820

Scientist	***Iris Texture Isolation***		***Features Extraction***	***Classification***
	Segmentation	***Normalization***		
Daugman 1994	Integro-differential operator	Rubber sheet model	2D Gabor Filter	Hamming Distance
Wildes 1997	Hough	Image Registration	Laplacian of Gaussian Filter	Normalized Correlation
Boles 1998	Virtual Circles	N-Level of Circles	Zero-Crossing 1D Wavelet	Dissimilarity Function
Zhu 2000	Virtual Circles	Rubber sheet model	DAUB4 Wavelet	Weighted Euclidean Distance
Lim 2001	Hough	Rubber sheet model	Haar Wavelet	LVQ Neural Networks
Tisse 2002	Integro- differential operator +Hough	Rubber sheet model	2D Hilbert Transform	Hamming Distance
Ma 2002	Hough	Rubber sheet model	2D Gabor Filter	Weighted Euclidean Distance
Noh 2002	Any Traditional Technique		M- ICA	Fisher's Discriminant Ratio

Table 1: Comparison among the most remarkable iris recognition systems

3. PROPOSED SYSTEM

3.1 iris image acquisition

Our technique used to identify human irises automatically by testing the window image (280*320 pixels) whether contains iris (as shown in Fig. 2) or not (as shown in Fig. 3) using neural networks.

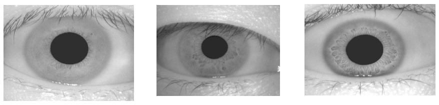

Fig. 2: Iris inside Examples

We used feed-forward backpropagation neural networks with hidden layer which contains 7 neurons, and one output layer contains only one neuron (0 or 1) to test the existence of an iris inside the image or not with the following specifications.

- <u>Transfer function</u>: Log sigmoid transfer function (logsig) for the two layers.
- <u>Backpropagation network training function</u>: Gradient descent with momentum and adaptive learning rate backpropagation (logsig).
- <u>Backpropagation weight/bias learning function</u>: Gradient descent with momentum (learngdm).
- <u>Performance function</u>: Mean squared error performance function (mse).

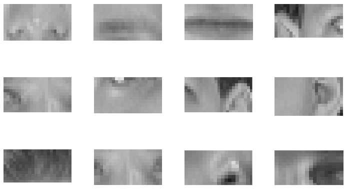

Fig. 3: Not iris inside Examples

The major difficulty we faced is the learning process because the neural network needs a large number of images contain iris/not iris. So, we selected 60 distinguished patterns (30 for iris and 30 for not iris).Therefore, increasing the number of patterns in the learning process improves the classification rate but also increases the database space.

3.2 Iris Texture Extraction

The iris texture extraction phase consists of two main steps: localizing the pupil center, and localizing the iris boundaries.

3.2.1 Localizing the Pupil Center

We will illustrate our system by using the following gray-scale iris image (as shown in Fig. 4).

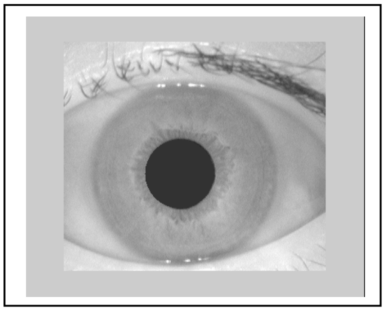

Fig.4: Iris Image

3.2.1.1 Converting the image to BW

Because of the existence of the black pupil in the iris image (as shown in Fig. 4), we found that the conversion of the iris image to black and white image is more effective in our system

3.2.1.2 Enhancing the BW image

Using linear filter 15-by-15 containing equal weights of ones to remove the garbage around the pupil, we get a clear pupil to determine the perfect center for the pupil (as shown in Fig. 5).

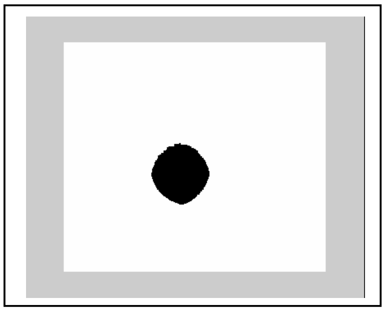

Fig. 5: Filtered BW image

3.2.1.3 Locating the pupil center

We determine the center of the pupil $(x_{o,},y_o)$by counting the number of black pixels (zero value) as following: Count all pixels in each row.

1. Get the maximum number of pixels of all rows.
2. Get the positions of the first and last pixels, (x_f,y_f) and (x_l,y_l) respectively, of this row. Then find the center of this row by, $x_o= (x_f+x_l) /2$.
3. Similarly, apply the previous steps for determining the center of the column of maximum number of pixels by $y_o= (y_f+y_l) /2$.
4. As $x_o \neq y_{o,}$, so we select the center point to be the most frequently crossed point. Consequently, the radius of virtual circle of the pupil can be determined (as shown in Fig.6).

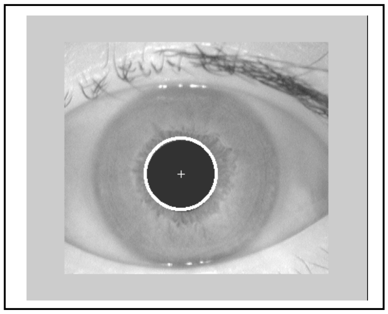

Fig. 6: Image after pupil center localization

3.2.2 Localizing the Iris boundaries

We segment the image of the iris from the eye by applying boundary detection techniques to localize the pupillary boundary. Based on merging the existing edge (the maximum number of point of the edge), this technique is segmented into boundaries by edge linking that described as following:

1. Define the size of neighborhoods 4 x 4.
2. Link similar points which having closed values; the entire image undergoes this process, while keeping a list of linked points.
3. When the process is completed the boundary is determined by the linked list, which can be apparent by using the mid-point algorithms of circle and ellipse. Similar steps can be extended using a coarse scale to locate the outer boundary (as shown in Fig.7).

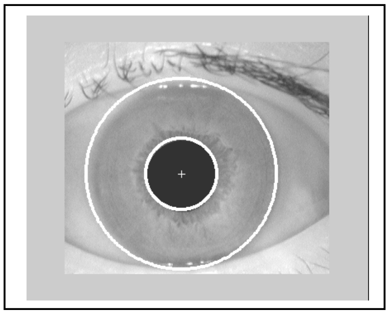

Fig. 7: Image after iris localization

4. Extract 37 iris circles with 360 feature points per circle (as shown in Fig.8).

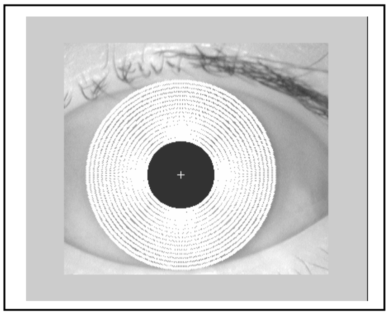

Fig. 8: iris texture

3.2.3 Cartesian to polar reference transform

The circular iris texture, (as shown in Fig. 8) which is in the Cartesian form, should be transformed into polar coordinate systems (37*360 rectangle as shown in Fig. 9). The transformation from Cartesian to polar reference, suggested by Daugman [2], authorizes the equivalent rectangular representation of the zone of interest. In this way, we compensate the stretching of the iris texture as the pupil changes in size, and we unfold the frequency information contained in the circular texture in order to facilitate next features extraction. Moreover, this new representation of the iris breaks the no-eccentricity of the iris and the pupil. The θ ($\theta \in [0; 2\pi]$) parameter and dimensionless ρ ($\rho \in [0; 1]$) parameter describe the polar coordinate system. Thus the following equations implement:

$$I(x(\rho,\theta),y(\rho,\theta)) \rightarrow I(\rho,\theta)$$

$$\begin{cases} x(\rho,\theta) = (1-\rho) * x_{\mathrm{p}}(\theta) + \rho * x_{\mathrm{i}}(\theta) \\ y(\rho,\theta) = (1-\rho) * y_{\mathrm{p}}(\theta) + \rho * y_{\mathrm{i}}(\theta) \end{cases} \quad \text{(Eq.1)}$$

With:
$$\begin{cases} x_p(\rho,\theta) = x_{p0}(\theta) + r_p * \cos(\theta) \\ y_p(\rho,\theta) = y_{p0}(\theta) + r_p * \sin(\theta) \end{cases} \quad \text{(Eq.2)}$$

$$\begin{cases} x_i(\rho,\theta) = x_{i0}(\theta) + r_i * \cos(\theta) \\ y_i(\rho,\theta) = y_{i0}(\theta) + r_i * \sin(\theta) \end{cases} \quad \text{(Eq.3)}$$

Where r_p and r_i are respectively the radius of the pupil and the iris, while $(x_p(\theta), y_p(\theta))$ and $(x_i(\theta), y_i(\theta))$ are the coordinates of the pupillary and limbic boundaries in the direction θ.

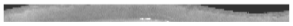

Fig. 9: Iris rectangular representation (37*360)

3.2.4 Enhancing the iris rectangular

The original iris image has low contrast and may have non-uniform illumination caused by the position of the light source. These may impair the results of the texture analysis. We enhance (as suggested in [4]) the iris image and reduce the effect of non-uniform illumination by using the local histogram equalization (as shown in Fig.10).

Fig. 10: Iris texture after enhancement (37*360)

3.3 Features Extraction

Gabor transform and wavelet transform are typically used to extract the iris features. Our system used Wavelet transform to extract this features from the iris image and we chose the Haar wavelet as a basis function (as shown in Fig. 11). After the enhanced 37*360 iris texture rectangular is obtained, we apply the wavelet transform four times in order to get the 3*23 sub-image to get 69 features from the HH sub-image of the high pass filter in the fourth transform.

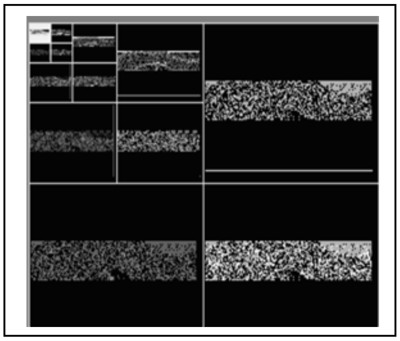

Fig. 11: Appling wavelet transformation 4 times to get 3*23 sub-image

- The dimension of the resulting feature vector is 72 (as shown in Table 2).
  - 69: from the HH sub-image of the high pass filter in the fourth transform.
  - 03: from the average value for the three remaining high pass filter areas in (HH1, HH2, and HH3).
- Each value of 72 dimensions has a real value between –1.0 and 1.0 to reduce the space we convert the positive value to 1 and the negative value to 0.

	Daugman	**Lim[9]**	**Our System**
Feature Vector	256 dimensions	87 dimensions	72 dimensions
No of bits per dimension	1 byte/dimension	1 bit /dimension	1 bit /dimension
No of bits per feature vector	2,048 bits	87 bits	72 bits

Table 2: Performance Evaluation according to the size of feature vectors

3.4 Classification

The difference between the stored iris code and the presented iris code is the XOR bit by bit Matching .So, Let A_i and B_i be two iris codes to be compared, the XOR Matching can be calculated as follow:

The XOR Matching Algorithm:

```
For  i = 1  to  72
          Comparing bit by bit code $A_i$ with the first code $B_i$ in the database.
          IF  the result of XOR is ( 0 )    {this mean the 2 bits are the same}
              count the number of zero's
          Else
              don't count, continue
End
```

Thus, we have to calculate The Matching Ratio by the following formula:

$$MR= \frac{N_Z * 100}{T_Z}$$

Where N_Z and T_Z are the number of zero's and total number of bits in each code respectively.

SPEED PERFORMANCE SUMMARY

On an Intel 1.2 GHZ processor with 128 MB RAM with using MATLAB 6.5 as a development tool, the execution times for the critical steps in iris recognition are as follows (as shown in Table 3).

Operation	Execution Time
Iris Existence Test	199 ms
Iris Boundaries Localization	528 ms
Iris Texture Transformation	301 ms
Feature Extraction	112 ms
XOR Comparison of two Iris Codes	10 μs

Table 3: Speeds of various stages in iris recognition process

CONCLUSIONS

In our implementation, we used 20 people's images with 3 iris images per person; two for training and one for testing. The performance of biometric system is usually described by two error rates: The False Acceptance Rate (FAR measures the probability of an enrolled individual being wrongly identified as another individual) and The False Rejection Rate (FRR measures the probability of an enrolled individual not being identified by the system). So, we have to determine the threshold separating FAR and FRR .When we use the threshold of 61.2 % for the matching ratio, we can get the performance about 95% to 96.7%. Our system scored 1.7 % for FRR and 3.1 % for FAR. Table (4) shows the classification rate compared with well known two methods; Daugman's and Lim's that are a little better than ours.
In fact, our system is beginning with testing whether the acquired image contains iris or not but other systems are not consider this test. Also, our features vector consists of 72 features, while Daugman's features vector is 2048 features and Lim features vector is 87 features. So, both methods much higher than ours because they extract features in much smaller local regions. Thus, we expect to improve the performance of our current system.

	Daugman	**Lim[9]**	**Our System**
Recognition Performance	100 %	98.4 %	96.7 %

Table 4: Overall performance comparison

ACKNOWLEDGMENT

We wish to thank Dr. Zhenan Sun, Chinese Academy of Sciences for the database of gray-scale iris images that provided us.

REFERENCES

1. A. Jain, R. Bolle and S. Pankanti, *BIOMETRICS: Personal Identification in Networked Society*, Kluwer Academic Publishers, Chap5, (1998) 103-122.
2. J.G. Daugman, "High Confidence Visual Recognition of Persons by a Test of Statistical Independence", *IEEE Tans. Pattern Analysis and Machine Intelligence*, vol.15, pp.1148-1161, Nov. 1993.
3. R.P. Wildes, "Iris Recognition: An Emerging Biometric Technology", *Proceedings of the IEEE*, vol.85, pp.1348-1363, Sept. 1997.
4. Y. Zhu, T. Tan, and Y. Wang, "Biometric Personal Identification System Based on Iris Pattern", *Chinese Patent Application*, No. 9911025.6, 1999.
5. W. W. Boles. " A security system based on human iris identification using wavelet transform ", *Engineering Applications of Artificial Intelligence*, Vol.11, No, I, pp, 77-85, Elsevier Science Ltd., 1998
6. H. EI-Bakry, "Fast Iris Detection for Personal Verification Using Modular Neural Nets", In *Lecture Notes in Computer Science*, Vol. 2206, pp. 269-283, Springer-Verlag, 2001.
7. Michael Negin, Thomas A. Chmielewski, Marcos Salganicoff, Theodore A. Camus, Ulf M. Cahn von Seelen, Péter L. Venetianer, Guanghua G. Zhang ,"An Iris Biometric System for Public and Personal Use", *IEEE Computer Magazine*, Vol. 33, No.2, pp. 70-75, 2000.
8. D. Ziou, S. Tabbone, "Edge Detection Techniques -An Overview", *Technical Report* No.195, Department de math et infonnatique, Universite de Sherbrooke, 1997.
9. S. Lim, K. Lee,O. Byeon,and T. Kim "Efficient Iris Recognition through Improvement of Feature Vector and Classifier", *ETRI Journal*,Vol.23, No. 2,Korea,2001.