

# Iris Recognition Using Modified Fuzzy Hypersphere Neural Network with different Distance Measures

S. S. Chowhan

Dept. of Computer Science  
COCSIT  
Latur, India

U. V. Kulkarni

Dept. of Computer Science  
SGGS IET  
Nanded, India

G. N. Shinde

Dept. of Elec. & Computer Science  
Indira Gandhi College  
Nanded, India

**Abstract**—In this paper we describe Iris recognition using Modified Fuzzy Hypersphere Neural Network (MFHSNN) with its learning algorithm, which is an extension of Fuzzy Hypersphere Neural Network (FHSNN) proposed by Kulkarni et al. We have evaluated performance of MFHSNN classifier using different distance measures. It is observed that Bhattacharyya distance is superior in terms of training and recall time as compared to Euclidean and Manhattan distance measures. The feasibility of the MFHSNN has been successfully appraised on CASIA database with 756 images and found superior in terms of generalization and training time with equivalent recall time.

**Keywords**- Bhattacharyya distance; Iris Segmentation; Fuzzy Hypersphere Neural Network.

## I. INTRODUCTION

Iris recognition has become the dynamic theme for security applications, with an emphasis on personal identification based on biometrics. Other biometric features include face, fingerprint, palm-prints, iris, retina, gait, hand geometry etc. All these biometric features are used in security applications [1]. The human iris, the annular part between pupil and sclera, has distinctly unique features such as freckles, furrows, stripes, coronas and so on. It is visible from outside. Personal authentication based on iris can obtain high accuracy due to rich texture of iris patterns. Many researchers work has also affirmed that the iris is essentially stable over a person's life. Since the iris based personal identification systems can be more noninvasive for the users [2].

Iris boundaries can be supposed as two non-concentric circles. We must determine the inner and outer boundaries with their relevant radius and centers. Iris segmentation is to locate the legitimate part of the iris. Iris is often partially occluded by eyelashes, eyelids and shadows. In segmentation, it is desired to discriminate the iris texture from the rest of the image. An iris is normally segmented by detecting its inner areas (pupil) and outer (limbus) boundaries [3] [4]. Well-known methods such as the Integro-differential, Hough transform and active contour models have been successful techniques in detecting the boundaries. In 1993, Daugman proposed an integro-differential operator to find both the iris inner and outer borders Wildes represented the iris texture with a laplacian pyramid constructed with four different resolution levels and used the normalized correlation to determine whether the input image and the model image are from the same class [5]. O. Byeon and

T. Kim decomposed an iris image into four levels using 2DHaar wavelet transform and quantized the fourth-level high frequency information to form an 87-bit code. A modified competitive learning neural network (LVQ) was used for classification [6]. J. Daugman used multiscale quadrature wavelets to extract texture phase structure information of the iris to generate a 2048-bit iris code and he compared the difference between a pair of iris representations by computing their Hamming distance [7]. Tisse. used a combination of the integro-differential operators with a Hough Transform for localization and for feature extraction the concept of instantaneous phase or emergent frequency is used. Iris code is generated by thresholding both the models of emergent frequency and the real and imaginary parts of the instantaneous phase[8]. The comparison between iris signatures is performed, producing a numeric dissimilarity value. If this value is higher than a threshold, the system generates output as a nonmatch, meaning that each iris patterns belongs to different irises [9]. In this paper, we have applied MFHSNN classifier which is a Modification of Fuzzy Hypersphere Neural Network (FHSNN) proposed by Kulkarni et al. [10]. Ruggero Donida Labati, et al. had represented the detection of the iris center and boundaries by using neural networks. The proposed algorithm starts by an initial random point in the input image, and then it processes a set of local image properties in a circular region of interest searching for the peculiar transition patterns of the iris boundaries. A trained neural network processes the parameters associated to the extracted boundaries and it estimates the offsets in the vertical and horizontal axis with respect to the estimated center [12].

## II. TOPOLOGY OF MODIFIED FUZZY HYPERSPHERE NEURAL NETWORK

The MFHSNN consists of four layers as shown in Fig 1(a). The first, second, third and fourth layer is denoted as  $F_R$ ,  $F_M$ ,  $F_N$ , and  $F_o$  respectively. The  $F_R$  layer accepts an input pattern and consists of  $n$  processing elements, one for each dimension of the pattern. The  $F_M$  layer consists of  $q$  processing nodes that are constructed during training and each node represents hypersphere fuzzy set characterized by hypersphere membership function [11]. The processing performed by each node of  $F_M$  layer is shown in Fig 1(b). The weight between  $F_R$  and  $F_M$  layers represents centre points of the hyperspheres.

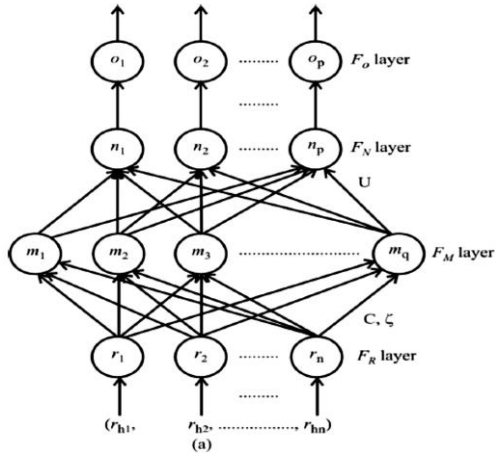


Figure 1. (a) Modified Fuzzy Hypersphere Neural Network

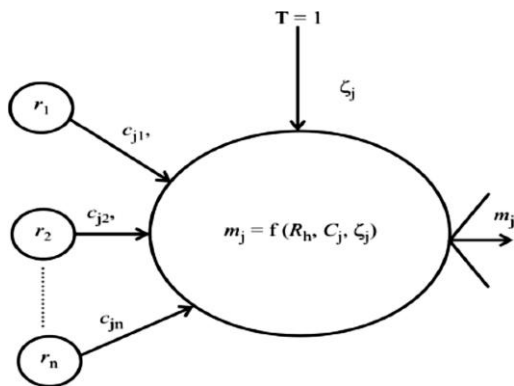


Figure 1. (b) Implementation Modified Fuzzy Hypersphere Neural Network

As shown in above Fig 1(b),  $C_j = (c_{j1}, c_{j2}, \dots, c_{jn})$  represents center point of the hypersphere  $m_j$ . In addition to this each hypersphere takes one more input denoted as threshold,  $T$ , which is set to one and the weight assigned to this link is  $\zeta_j$ .  $\zeta_j$  represents radius of the hypersphere,  $m_j$  which is updated during training. The center points and radii of the hyperspheres are stored in matrix  $C$  and vector  $\zeta$ , respectively. The maximum size of hypersphere is bounded by a user defined value  $\lambda$ , where  $0 \leq \lambda \leq l$  is called as growth parameter that is used for controlling maximum size of the hypersphere and it puts maximum limit on the radius of the hypersphere. Assuming the training set defined as  $R \in \{R_k | h = 1, 2, \dots, P\}$ , where  $R_h = (r_{h1}, r_{h2}, \dots, r_{hn}) \in I^n$  is the  $h$ th pattern, the membership function of hypersphere node  $m_j$  is

$$m_j(R_h, C_j, \zeta_j) = 1 - f(l, \zeta_j, \gamma) \quad (1)$$

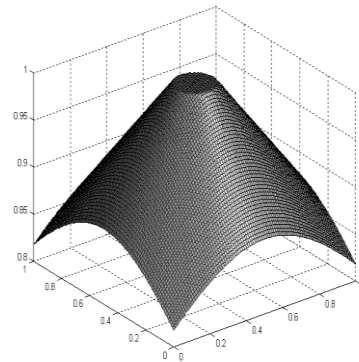
Where  $f()$  is three-parameter ramp threshold function defined as  $f(l, \zeta_j, \gamma) = 0$  if  $0 \leq l \leq \zeta_j$ , else  $f(l, \zeta_j, \gamma) = l\gamma$ , if  $\zeta_j < l \leq l$ ,  $l \leq l$ , and argument  $l$  is defined as:

$$l = \sum_{i=1}^n \sqrt{\sum c_{ji} - \sum r_{hi}} \quad (2)$$

The membership function returns  $m_j = 1$ , if the input pattern  $R_k$  is contained by the  $R_k$  hypersphere. The parameter  $\gamma, 0 \leq \gamma \leq l$ , is a sensitivity parameter, which governs how fast the membership value decreases outside the hypersphere when the distance between  $R_h$  and  $C_j$  increases. The sample plot of membership function with centre point  $[0.5 \ 0.5]$  and radius equal to  $0.3$  is shown in as shown in Fig 2.

Figure 2. Plot of Modified Fuzzy Hypersphere membership function for  $\gamma = l$

Each node of  $F_N$  and  $F_o$  layer represents a class. The  $F_N$



layer gives fuzzy decision and output of  $k$ th  $F_N$  node represents the degree to which the input pattern belongs to the class  $n_k$ . The weights assigned to the connections between  $F_M$  and  $F_N$  layers are binary values that are stored in matrix  $U$  and updated during learning as,  $u_{jk} = 1$  if  $m_j$  is a HS of class  $n_k$ , otherwise  $u_{jk} = 0$ , for  $k = 1, 2, \dots, p$  and  $j = 1, 2, \dots, q$ , where  $m_j$  is the  $j$ th  $F_M$  node and  $n_k$  is the  $k$ th  $F_N$  node. Each  $F_N$  node performs the union of fuzzy values returned by HSs described as

$$n_k = \max_{j=1}^q m_j u_{jk} \text{ for } k = 1, 2, \dots, p \quad (3)$$

Each  $F_o$  node delivers non-fuzzy output described as,  $O_k = 0$ , if  $n_k < T$ , otherwise,  $O_k = 1$ , if  $n_k = T$ , for  $k = 1, 2, \dots, p$  where  $T = \max(n_k)$ , for  $k = 1, 2, \dots, p$ .

### III. MFHSNN LEARNING ALGORITHM

The supervised MFHSNN learning algorithm for creating fuzzy hyperspheres in hyperspace consists of three steps.

#### A. Creating of HSs

Given the  $h$ th training pair  $(R_h, d_h)$  find all the hyperspheres belonging to the class  $d_h$ . These hyperspheres are arranged in ascending order according to the distances between the input pattern and the center point of the

hyperspheres. After this following steps are carried sequentially for possible inclusion of input pattern  $R_h$ .

Step 1: Determine whether the pattern  $R_h$  is contained by any one of the hyperspheres. This can be verified by using fuzzy hypersphere membership function defined in equation (4). If  $R_h$  is contained by any of the hypersphere then it is included, therefore in the training process all the remaining steps are skipped and training is continued with the next training pair.

Step 2: If the pattern  $R_k$  falls outside the hypersphere, then the hypersphere is expanded to include the pattern if the expansion criterion is satisfied. For the hypersphere  $m_j$  to include  $R_h$  the following constraint must be met.

$$\left[ \sum_{i=1}^n \sqrt{\sum c_{ji} - \sum r_{hi}} \right] \leq \lambda \quad (4)$$

Here we have proposed a new approach for testing expansion of new hyperspheres based on Bhattacharyya distance which is sum of the absolute differences yields superior results as compared to Euclidian distance and Manhattan distance.

If the expansion criterion is met then pattern  $R_k$  is included as

$$\zeta_j^{new} = \left[ \sum_{i=1}^n \sqrt{\sum c_{ji} - \sum r_{hi}} \right] \quad (5)$$

Step 3: If the pattern  $R_k$  is not included by any of the above steps then new hypersphere is created for that class, which is described as

$$C_{new} = R_h \text{ and } \zeta_{new} = 0 \quad (6)$$

### B. Overlap Test

The learning algorithm allows overlap of hyperspheres from the same class and eliminates the overlap between hyperspheres from different classes. Therefore, it is necessary to eliminate overlap between the hyperspheres that represent different classes. Overlap test is performed as soon as the hypersphere is expanded by step 2 or created in step 3.

Overlap test for step 2: Let the hypersphere is expanded to include the input pattern  $R_k$  and expansion has created overlap with the hypersphere,  $m_v$  which belongs to other class, which belongs to other class. Suppose,  $C_u = [x_1, x_1, \dots, x_n]$  and  $\zeta_u$  represents center point and radius of the expanded hypersphere and  $C_v = [x'_1, x'_2, \dots, x'_n]$  and  $\zeta_v$ , are centre point and radius of the hypersphere of other class as depicted in Fig 3(a). Then if

$$\left[ \sum_{i=1}^n \sqrt{\sum c_{ji} - \sum r_{hi}} \right] \leq \zeta_u + \zeta_v \quad (7) \quad \square$$

means those hyperspheres from separate classes are overlapping.

#### a) Overlap test for step 3

If the created hypersphere falls inside the hypersphere of other class means there is an overlap. Suppose  $m_p$  represents created hypersphere to include the input  $R_h$  and  $m_q$  represents the hypersphere of other class as shown in Fig 4b The presence of overlap in this case can be verified using the membership function defined in equation (1). If  $m_q(R_h, C_p, \zeta_p) = m_q(R_h, C_q, \zeta_q) = 1$  means two hyperspheres from different classes are overlapping.

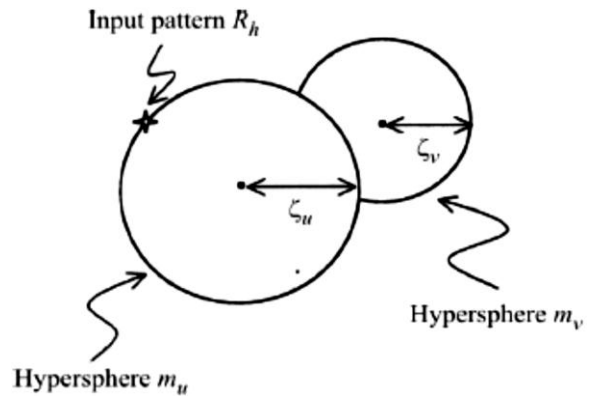


Figure 3. (a) Status of the hypersphere before removing an overlap in step 2

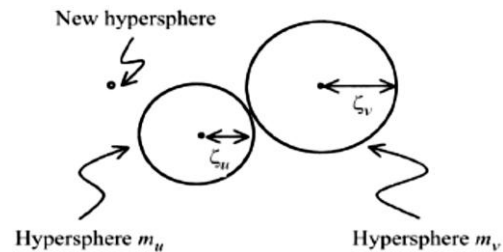


Figure 3. (b) Status of the hyperspheres after removing an overlap in step 2

#### b) Removing Overlap

If step 2 has created overlap of hyperspheres from separate classes then overlap is removed by restoring the radius of just expanded hypersphere. Let,  $m_u$  be the expanded hypersphere then it is contracted as

$$\zeta_u^{new} = \zeta_u \quad (8)$$

and new hypersphere is created for the input pattern as described by equation (11). This situation is shown in Fig 3(b). If the step 3 creates overlap then it is removed by modifying of other class. Let  $C_p = [x_1, x_1, \dots, x_n]$  and  $\zeta_p$ , represents centre point and radius of the  $C_q = [x'_1, x'_2, x'_3, \dots, x'_n]$  and  $\zeta_q$ , are center point and radius of the

$$\zeta_j^{new} = \left[ \sum_{i=1}^n \sqrt{\sum c_{ji} - \sum r_{hi}} \right] - \delta \quad (9)$$

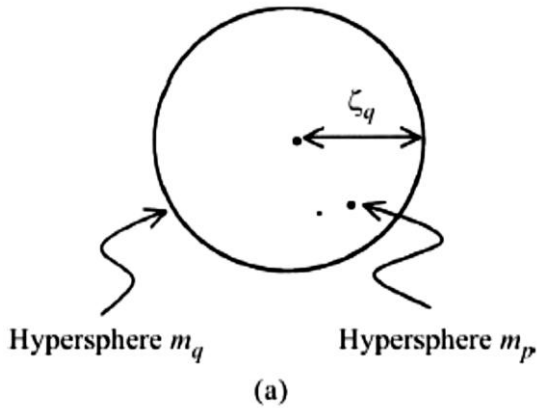


Figure 4. (a) Status of hypersphere before removing an overlap in step 3.

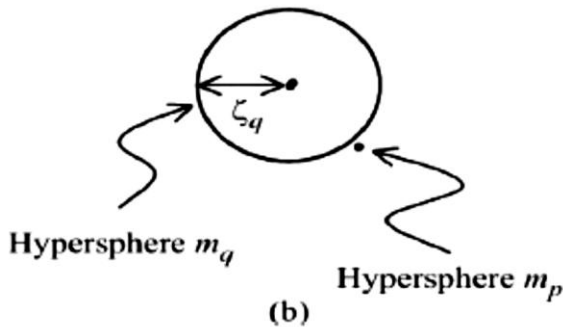


Figure 4. (b) Status of hypersphere after removing an overlap in step 3.

#### IV. IRIS SEGMENTATION AND FEATURE EXTRACTION

Iris Segmentation plays very important role for detecting the iris patterns, Segmentation is to locate valid part of the iris for iris biometric [3]. Finding the inner and outer boundaries (pupillary and limbic) are as shown in the Fig 5(a). Localizing it's upper and lower eyelids if they occlude, and detecting and excluding any overlaid occlusion of eyelashes and their reflection. The best known algorithm for iris segmentation is Daugman's intergro-differential operator to find boundaries of iris as defined.

$$\max(r, x_0, y_0) \left| G_\sigma * \frac{\partial}{\partial r} \int_{(r, x_0, y_0)} \frac{I(x, y)}{2\pi r} ds \right| \quad (10)$$



Figure 5. (a) Inner and Outer boundaries are detected with different radius



Figure 5. (b) ROI is extracted

Iris has a particularly interesting structure and provides rich texture information. Here we have implemented principal component analysis method for feature extraction which captures local underlying information from an isolated image. The result of this method yields high dimension feature vector. To improve training and recalling efficiency of the network, here we have used Singular value decomposition (SVD) method to reduce the dimensionality of the feature vector, and MFHSNN is used for Classification. SVD is a method for identifying and ordering the dimensions along which the feature exhibit the most variation [13]. The most variation are identified the best approximation of the original data points using less dimensions. This can be described as:

$$X = UTV^T \quad (11)$$

the covariance matrix is be defined as:

$$C = \frac{1}{n} XX^T = \frac{1}{n} UT^2U^T \quad (12)$$

$U$  is a  $n \times m$  matrix. SVD performs repetitively order the singular values in descending order, if  $n < m$ , the first  $n$  columns in  $U$  corresponds to the sorted eigenvalues of  $C$  and if the first  $m$  corresponds to the sorted non-zero eigenvalues of  $C$ . The transformed data can thus be written as:

$$Y = U^T X = U^T UTV^T \quad (13)$$

Where  $U^T U$  is a simple  $n \times m$  matrix which is one on the diagonal and zero. Hence Equation 13 is decomposition of equation 11.

#### V. EXPERIMENTAL RESULTS

CASIA Iris Image Database Version 1.0 (CASIA-IrisV1) includes 756 iris images from 108 eyes. For each eye, 7 images are captured in two sessions, where three samples are collected in the first session and four samples in second. For each iris class, we choose two samples from each session for training and remaining as testing samples. Therefore, there are 540 images for training and 216 images for testing. The timing analysis of training and recall, recognition rates in terms of number of hyperspheres, radius and recognition rate are depicted in Table 1 and Table 2.



Figure 6. Specimen Feature Vectors of 20 Iris Pattnrs

TABLE I. PERFORMANCE EVALUATION OF DIFFERENT DISTANCE MEASURES ON MFHSNN

Distance Measures	Training	Recall	Hypersphere	Radius
Euclidean	0.830694	31.780012	347	0.008
Manhattan	0.867152	31.660341	364	0.022
Bhattacharya	0.750051	23.801397	377	0.0000001

TABLE II. RECOGNITION RATE OF MFHSNN ON CASIA IMAGES

Methodology	Recognition rate	
Daugman	HD,SVM	99.25
Wildes	Normalized Correlation	97.43
Y.Wang	WED	99.57
Ali	HD, WED	95.20
	MFHSNN	94.00

## VI. CONCLUSION AND FUTURE WORK

In this paper, we described an iris recognition algorithm using MFHSNN which has ability to learn patterns faster by creating /expanding HSs. It has been verified on CASIA database the result is as shown in Table 1 and Table II. MFHSNN can also be adapted for applications in some other pattern recognition problems. Our future work will be to improve the iris recognition rate using fuzzy neural networks.

## VII. REFERENCES

[1] Li Ma, T Tan and D.Zhang and Y.Wang. "Personal Identification Based on Iris Texture Analysis," IEEE Trans. Pattern Anal. Machine Intell, 25(12):1519-1533, 2003.

[2] Dey, Somnath, Samanta Debasis. "Improved Feature Processing for Iris Biometric Authentication System," International Journal of Computer Science and Engineering, 4(2):127-134.

[3] Daugman. J.G, "How Iris Recognition Works", Proceedings of 2002 International Conference on Image Processing, Vol. 1, 2002.

[4] Li Ma, Tieniu Tan, Yunhong Wang, and Dexin Zhang. "Efficient Iris Recognition by Characterizing Key Local Variations", IEEE Transactions on Image Processing, 13(6):739-750, 2004.

[5] Wildes R P. "Iris Recognition: An Emerging Biometric Technology". In Proceeding of the IEEE, Piscataway, NJ, USA, 1997, 85:1348-1363.

[6] S.Lim, K.Lee, O.Byeon and T.Kim, "Efficient iris recognition through improvement of feature vector and classifier," ETRI J., 23(2): 1-70, 2001.

[7] Daugman J.G, "Demodulation by complex-valued wavelets for stochastic pattern recognition," International Journal of Wavelets, Multiresolution, and Information Processing, 1(1):1-17, 2003.

[8] Tisse C.-L. and Torres L. Michel, "Robert, Person Identification Technique Using Human Iris Recognition", Proceedings of the 15th International Conference on Vision Interface, 2002, pp. 294-299.

[9] Poursaberi A. and Arrabi B.N, "Iris Recognition for Partially occluded images Methodology and Sensitive Analysis", Hindawi Publishing corporation journal on Advances in Signal Processing, 2007.

[10] U.V. Kulkarni and T.R. Sontakke, "Fuzzy Hypersphere Neural Network Classifier", Proceedings of 10th International IEEE Conference on Fuzzy Systems, University of Melbourne, Australia, December 2001.

[11] Krishna Kanth B. B. M, Kulkarni U. V. and Giridhar B. G. V., "Gene Expression Based Acute Leukemia Cancer Classification: a Neuro-Fuzzy Approach" International Journal of Biometrics and Bioinformatics, (IJBB), Volume (4): Issue (4) pp. 136-146, 2010.

[12] Ruggero Donida Labati et.al, "Neural-based Iterative Approach for Iris Detection in Iris recognition systems", Proceedings of the 2009 IEEE Symposium on Computational Intelligence in Security and Defense Applications (CISDA 2009).

[13] Madsen, R., Hansen, L., &Winther, O. "Singular value decomposition and principal component analysis," isp technical report, 2004.

## AUTHORS PROFILE

**Santosh S. Chowhan** received the M.Sc.(CS) degree from Dr. BAM University, Aurangabad, Maharashtra, India in the year 2000. He is currently working as lecturer in the College of Computer Science and Information Technology, Latur, Maharashtra. His current research interests include various aspects of Neural Networks and Fuzzy Logic, Pattern Recognition and Biometrics

**Uday V. Kulkarni** received Ph.D degree in Electronics and Computer Science Engineering from S. R. T. M. University, Nanded in the year 2003. He is currently working as a professor in Dept of Computer Science and Engineering in SGGGS Institute of Engineering and Technology, Nanded, Maharashtra, India.

**Ganesh N. Shinde** received M. Sc. & Ph.D. degree from Dr. B.A. M. University, Aurangabad. He is currently working as Principal in Indira Gandhi College, Nanded and Maharashtra, INDIA.. He has published 27 papers in the International Journals and presented 15 papers in International Conferences. In his account one book is published, which is reference book for different courses. at Honolulu, U.S.A. He is member of Management Council & Senate of S.R.T.M. University, Nanded, INDIA. His research interest includes Filters, Image processing, Pattern recognition, Fuzzy Systems, Neural Network and Multimedia analysis and retrieval system.