Study Of Hopfield Neural Network For Fingerprint Verification Based On Fast Fourier Transform

Ramesh Chandra Sahoo, Sateesh Kumar Pradhan, Somesh Kumar

Abstract: In this paper we are analyzing storing capacity and recalling of Hopfield neural network of memorized fingerprint image patterns by Hebbian rule through Fast Fourier Transform (FFT). In this process we measure the success rate of the network in terms of recalling original input patterns for testing and also noisy input patterns of the fingerprint images in MATLAB using an image database of FVC2002 and the simulated results are presented here to explain the better performance of Hopfield network for recalling of the stored fingerprint patterns.

Index Terms: Hopfield Neural Network, FFT, Fingerprint, FVC2002.

1. INTRODUCTION

Biometric fingerprint authentication is regarded as more reliable than traditional tokens and passwords, because of their uniqueness and consistency over time. Many automatic fingerprint matching approaches are based on extraction of minutiae [5], correlation based matching using Gabor filter[6], and Dass algorithm[7] has been used for pre-processing step for fingerprint [1, 2] and matching them with the stored patterns. In this study a recurrent network is preferred for pattern recognition or data classification with hebbian learning process. The earliest recurrent network had separately begun with Kohonen (1977), Anderson (1977), and Hopfield (1982) presented a detail complete mathematical study of such a subject [4]. As Hopfield neural network is a fully connected and auto-associative recurrent network where node values are updated based on their local computations iteratively [3].

1.1. Hopfield Neural Network

A Hopfield network is a recurrent artificial neural network invented by John Hopfield where nets perform as content-addressable memory systems with binary/bipolar threshold values[8,9]. Converges to a local minima is a guaranteed feature of these recurrent networks, but it does not guaranteed for its convergence to one of the stored patterns. Hopfield nets can have taken input unit values either of 1 or -1, or values of 1 or 0. Then the two possible rules for unit i[th] activation neuron are:

$$\mathbf{a_i} = \begin{cases} 1 & if\ \sum_j w_{ij}s_j > \theta_i \\ -1 & \text{otherwise} \end{cases} \quad \text{--------(1)}$$

Or

$$\mathbf{a_i} = \begin{cases} 1 & if\ \sum_j w_{ij}s_j > \theta_i \\ 0 & \text{otherwise} \end{cases} \quad \text{--------(2)}$$

Where:
w_{ij} ← the strength of connection weight from unit j to unit i
s_j ← is the state of unit j and
θ_i ← The threshold of unit i and the $w_{ij} = w_{ji}$ is symmetric in Hopfield net.

The specification of Hopfield net is the synaptic weights that referred to learning. In our study we used the Hebbian learning for Hopfield neural network to determine the weight matrix and there were a number of learning rules have been suggested since then to improve its performance [10, 11]. This rule has its advantages of being incremental and local minima. The updating principle of a particular connection in the network depends on the information available on each side of the connection and also the pattern that is being immediately added to the network. Recalling of memorized patterns is taken care of effectively with the pattern information is encoded effectively and efficiently. The features of the input pattern should be extracted efficiently to achieve the pattern information. Various methods have already been applied in literature for the feature extraction [12, 13, 14] to perform the job of associative memory in Hopfield neural network architecture.

2. PROPOSED METHOD

This section enlightens the general structure of the fingerprint verification by a recurrent network called Hopfield neural network. The proposed technique principally involves image pre-processing by enhancing the fingerprint image, extracting the features by Fast Fourier Transform, storing these extracted features of patterns into the Hopfield neural network with Hebbian learning and lastly recall these stored images with introducing 10 to 50 percent explicit noisy in the recall patterns process.

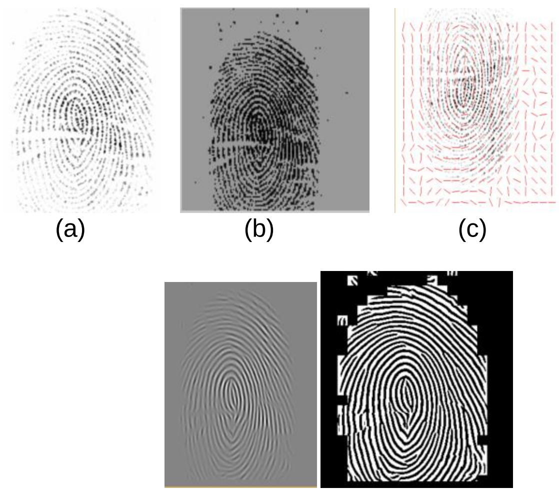
2.1. Fingerprint enhancement

As the patterns are not enough high quality fingerprint images, it is very important to apply a proper method to improve the quality of fingerprint images those are affected by noise like smudgy area, breaks in ridges, low contrast ridges, over inked area, under inked area to make it easier for further operations. Hear the fingerprint images we considered for our study is FVC2002 database. Firstly we apply histogram equalization which refers to boost the pixel value

- *Ramesh Chandra Sahoo, Sateesh Kumar Pradhan, Somesh Kumar*
- *Research Scholar, Utkal Unuversity, Bhubaneswar-751004, India, rsahoo22@gmail.com*
- *Deaprtment of Computer Science, Utkal Unuversity, Bhubaneswar-751004, India, sateesh1960@gmail.com*
- *Department of Computer Science, Moradabad Institute of Technology, Moradabad, INDIA, kumarsomesh507@gmail.com*

distribution of images to increase the perceptional information. Then the histogram equalization images are passed through the following function to get the binary image which will further passed to our second step for feature extraction through fast fourier transform.

Enhancement_Procedure(image)
{
Image1= fftenhancement(image,6) % enhancement with block 6X6
Image1=fftenhancement(image1,12)%enhancement with block 12X12
Iamge1=fftenhancement(image1,24)% enhancement with block 24X24
Normalize=ridgesegment(image,1blocksize, threshold)
Oimage=ridgeorient(normalize)
[normalize mask]=ridgesegment(image1 blocksize threshold)
*binaryImage=ridgefilter(normalize,oimage,frequency.*mask)*
}



(a) (b) (c) (d) (e)

Figure-1: *(a): Original Image, (b) Histogram equalization Image (c) Ridgeorient (d) Ridge Filter (e) Enhanced Binary Image*

2.2. Fast Fourier Transform

Mathematically images are represented in Spatial domain in form of their spatial variable function f(x, y), where these variables x and y is the intensity of a particular location or point of the image in a 2D plane. Alternatively the same image can also be represented by its frequency domain in terms of their frequency, phase or other complex exponentials. Fast Fourier Transform (FFT) is a Fourier transform whose inputs and outputs are discrete samples means the input image of spatial domain is transform to frequency domain and the inverse Fourier transform retransform from frequency domain to spatial domain over a finite region 0 ≤ x ≤ X-1 and 0 ≤ y ≤ Y-1. The FFT and its inverse of a 2D image are given by the following equations:

$$F(x) = \sum_{n=0}^{N-1} f(n) e^{-j2\pi(x\frac{n}{N})} \quad ----(2.1)$$

$$f(n) = \frac{1}{N} \sum_{n=0}^{N-1} F(x) e^{j2\pi(x\frac{n}{N})} \quad ----(2.2)$$

Where:
f(m,n) = the pixel at coordinates (m, n),
F(x,y) = the value of the image in the frequency domain corresponding to the coordinates x and y,
M X N =the dimensions of the image.
Since FFT is separable, the 2D transform can be done as two 1D transforms as shown below and the end result is equivalent to performing the 2D transform in the frequency space.

$$F(x,y) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m,n) e^{-j2\pi\left(x\frac{m}{M} + y\frac{n}{N}\right)} -(2.3)$$

$$f(m,n) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} F(x,y) e^{j2\pi\left(x\frac{m}{M} + y\frac{n}{N}\right)} -(2.4)$$

We applied FFT to the enhanced binarized images which were enhanced through our enhancement procedure to producing extracted feature vectors. The filtered transform feature vectors are then applied to Inverse Transform to produce the refined images. Now we converted the features obtained after FFT to bipolar pattern (i.e an image where pixel is either +1 or -1 value) as we are storing these images in Hopfield Network. Finally the images are resized to dimension 60 X 60 and converted to bipolar pattern vectors.

The k^{th} pattern vector is represented of form :

$$p_k = [p_{k1}, p_{k2}, p_{k3}, \dots \dots \dots \dots\ p_{kN}],$$

where N= 1 to 3600.

Then all pattern vectors of images are combined together in a matrix of size N X K, where K is the number of patterns of size 3600 X 1 each, is submitted for pattern storage to the Hopfield network and the total pattern matrix is given as below:

$$P = \begin{pmatrix} p_{11} & p_{21} & \cdots & p_{K1} \\ \vdots & & \ddots & \vdots \\ p_{1N} & p_{2N} & \cdots & p_{KN} \end{pmatrix} \quad \text{--------------(2.5)}$$

2.3. Pattern Storage in Hopfield Neural Network

In proposed method, the Hopfield Neural network is used to store the extracted feature vectors of images after FFT as in equation 2.5, where each pattern is order of (N * 1) consisting of N processing units and N * N connection strengths and each neuron can either +1 or -1 as state. To store K such patterns the summation of correlation matrices for each pattern is given as:

$$W_{ij} = \frac{1}{N} \sum_{k=1}^{K} p_{ki} * p_{kj} \quad for\ i \neq j \ ....(2.6)$$
$$= 0, \quad for\ i = j,\ 1 \le i \le N$$

Where N= number of units/neurons in the network.

For storing all K patterns the activation equation must be satisfied to achieve the storage capacity, for that every unit receive input from every other unit. The net input of a unit i at any time t is computed as:

$$S_i(t) = \sum_{j \neq i} W_{ij} S_j(t) \dots\dots\dots\dots\dots\dots (2.7)$$

Where:
W_{ij} =weight of connection between unit i and j
S_j =state of unit j at time t.
Now the change in weight to store the first pattern is given by:

$$W_{ij}^{New} = W_{ij}^{Old} + \sum_{i,j} p_{1i} * p_{1j} \dots\dots\dots\dots (2.8)$$

And

$$W_{ij}^{Old} = W_{ij}^{New} \text{ ------------------------}(2.9)$$

Similarly the Kth pattern we have

$$W_{ij}^{K} = W_{ij}^{K-1} + \sum_{i,j} p_{Ki} * p_{Kj} \dots\dots\dots (2.10)$$

This can be generalized by:

$$W_{ij}^{K} = \sum_{k=1}^{K} \sum_{i,j} p_{Ki} * p_{Kj} \dots\dots\dots (2.11)$$

So the normalized weight matrix over all N is given by:

$$W_{ij}^{K} = \frac{1}{N} \sum_{k=1}^{K} \sum_{i,j} p_{Ki} * p_{Kj} \dots\dots\dots (2.12)$$

And the same weight matrix is also being calculated from pattern vectors is:

$$W^{K} = \frac{1}{N} \sum_{k=1}^{K} p_k * (p_k)' \dots\dots\dots\dots\dots\dots (2.13)$$

Then the final learning for all patterns in the Hopfield neural netwoks, the final weight matrix is given as:

$$W^K = \begin{bmatrix} 0 & p_1p_2 & p_1p_3 \dots. & p_1p_N \\ p_2p_1 & 0 & p_2p_3 \dots. & p_2p_N \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ p_Np_1 & p_Np_2 & \dots\dots\dots & 0 \end{bmatrix} \dots.(2.14)$$

The above matrix in equation (2.14) is the stored patterns in the Hopfield neural network. To measure the performance of the network we tested with all memorized patterns and also with some noisy patterns through our recalling procedure.

2.4. Recall procedure from Hopfield Network

Let one of the memorized pattern P and its noisy pattern form be P+ε where ε is the percentage (number of bits) of error, for example 10% means 360 bits, 20% means 720 bits, 30% means 1080bits, 40% means 1440bits and 50% means 1800bits since our stored pattern is of form 3600 bits. Let the stored i^{th} pattern be:

$$N(p^i) = \{p_1^i, p_2^i \dots\dots, p_N^i\} \dots\dots\dots\dots\dots\dots\dots\dots (2.15)$$

To recall the stored fingerprint pattern, the sample pattern P and its distorted pattern form P+ε are offered to the network. The output state of activation function of the network for P and P+ε patterns respectively as:

$$(p_i^i) = \sum_{j=1}^{N} W_{ij}^k \, (p_j^i)(t+1) \dots\dots\dots\dots\dots (2.16)$$

And

$$(p_i^i + \varepsilon) = \sum_{j=1}^{N} W_{ij}^k \, (p_j^i + \varepsilon)(t+1) \dots (2.17)$$

Then the activation dynamics for the memorized pattern in equation (2.16) and the distorted pattern in equation (2.17) is executed for testing by using the weight matrix (2.14) and are given as:

$$(p_i^i)(t+1) = (p_i^i)(t), \; for\ all\ i = 1\ to\ N. (2.18)$$

And

$$(p_i^i + \varepsilon)(t+1) = (p_i^i)(t), for\ all\ i = 1\ to\ N (2.19)$$

That shows the network settle the same stable state that already stored pattern even if for noisy pattern. In our study we shows the percentage of recall for different percentages of noisy patterns in our result section.

3. SIMULATION DESIGN AND RESULTS

In this simulation process with Hopfield Net, which consider with 3600 processing units is trained with a feedback network storage with Hebbian Learning Rule after patterns have been pre-processed through pattern enhancement procedure and FFT. Here we are analyzing the efficiency of the memorized patterns through the recalling procedure and the algorithmic form is given as:

Hopfield_Storage()
{
Set the initial Weight matrix(N X N) to zero;
do
{
Perform matrix multiplication of first pattern p_1 with its transpose;
And then the new weight matrix will be the old weight matrix with product matrix;
}
While until $p_1 \neq K$
Now Assign the diagonal elements to zero;
Then Normalize weight matrix (divide each element by N);
}

After storing all patterns to the network, we initiate recalling process for a random input pattern which is memorized and the efficiency measured, shows a sign of the storage capacity of Hopfield Network. Thereafter we analyzed for distorted patterns for different percentages of errors those are introduced explicitly through our recall procedures. The network finally produces a memorized pattern that best matches with the input pattern in a stable state. The network's efficiency is also measured with new pattern that is not memorized to check that whether it is associated with one of the stored pattern, leads to a false minima or not recognised at all. To recall the input pattern as well as noisy pattern, the recall algorithm is described as follows:

Hopfield_Recall ()
{
load weight matrix;
submit the sample input pattern;
initialize the network with input pattern ;
do
{
do (for randomly selected unit pattern p)
{
calculate net input to p;
add the net input to the existing value of p;
Compare p with θ (threshold);
set activation of p to +1 if p > θ else set it to - 1;
transmit the value of p to all other units;
}
while (p ≠ N)
}
while (convergence not achieved or 500 iterations)
}

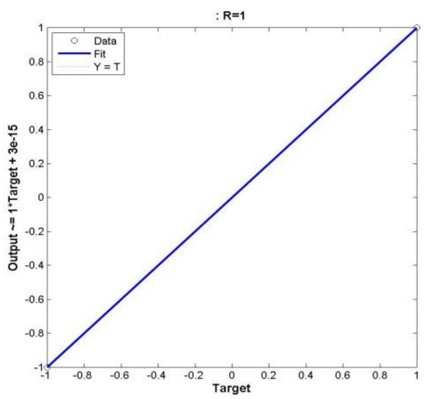


Figure 2.1: *Regression plot for 20% error of 80 Patterns*

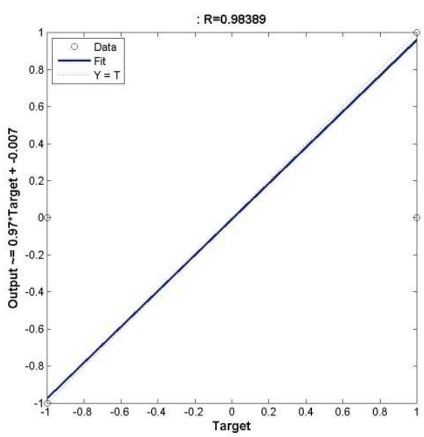


Figure 2.2: *Regression plot for 30% error of 80 Patterns*

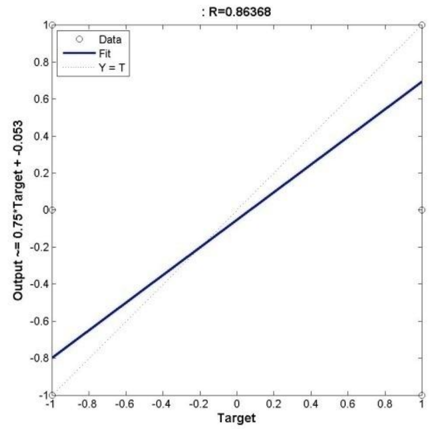


Figure 2.2: *Regression plot for 40% error of 80 Patterns*

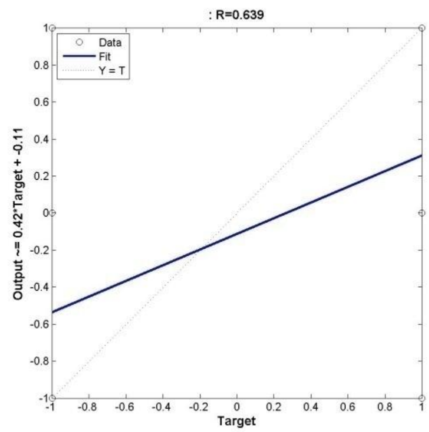


Figure 2.4: *Regression plot for50% error of 80 Patterns*

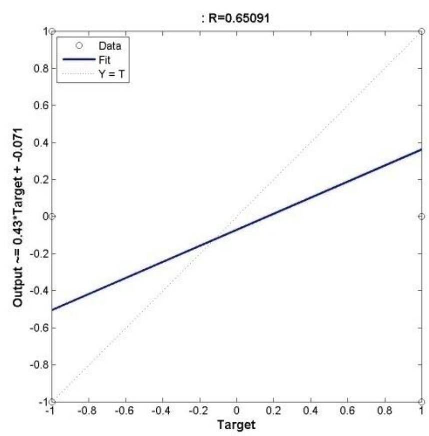


Figure 2.5: *Regression plot for 40% error of 160 Patterns*

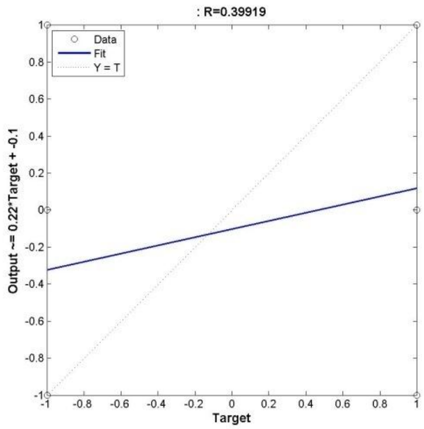


Figure 2.6: *Regression plot for 50% error of 160 Patterns*

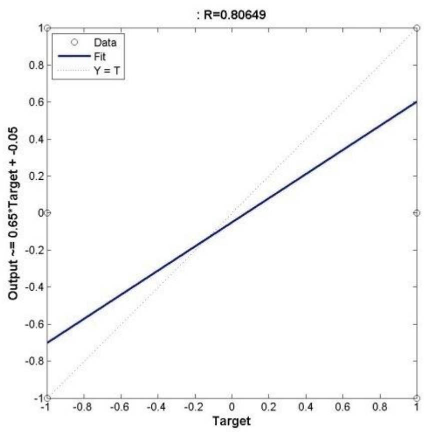


Figure 2.7: *Regression plot for 30% error of 240 Patterns*

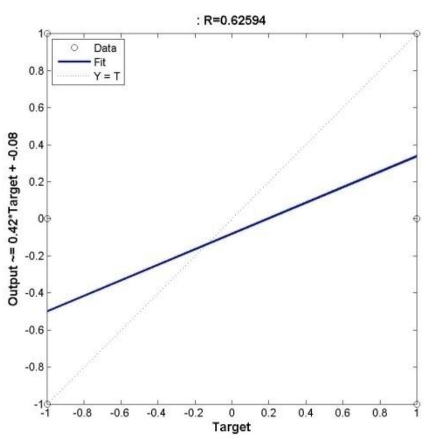


Figure 2.8: *Regression plot for 40% error of 240 Patterns*

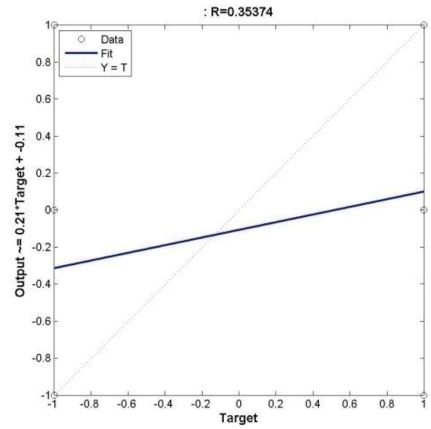


Figure 2.9: *Regression plot for 50% error of 240 Patterns*

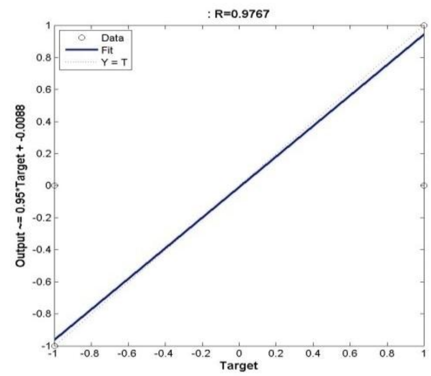


Figure 2.10: *Regression plot for 10% error of 320 Patterns*

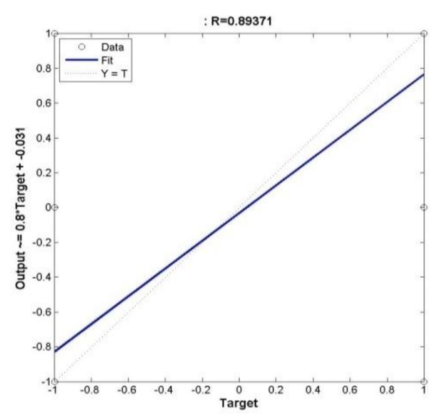


Figure 2.11: *Regression plot for 20% error of 320 Patterns*

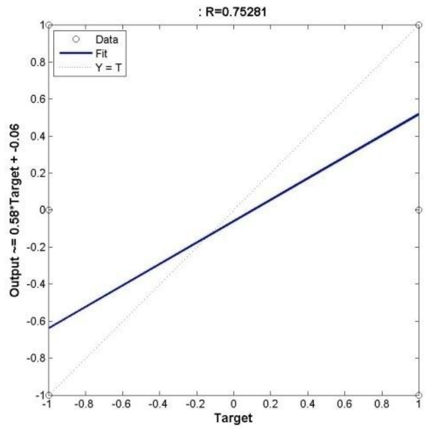


Figure 2.12: *Regression plot for 30% error of 320 Patterns*

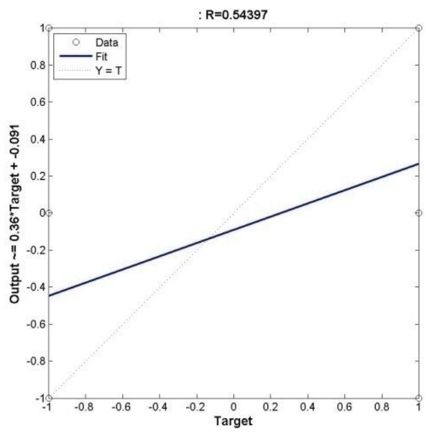


Figure 2.13: *Regression plot for 40% error of 320 Patterns*

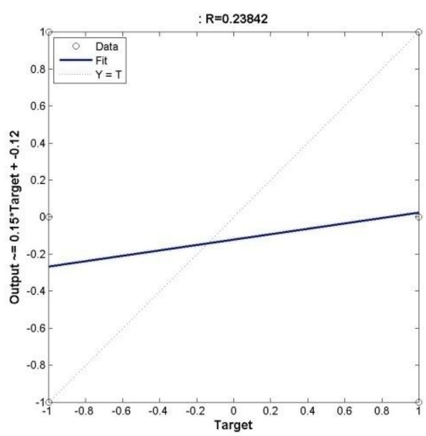


Figure 2.14: *Regression plot for 50% error of 320 Patterns*

Table-1:
Pattern Recall Ability of Hopfield Neural Network through FFT

No. Of Patterns in the Network from FVC2002	% of Noise introduced in actual Pattern	Correct Recalling through FFT feature extraction	False Minima
80 Patterns	10%	80	0
	20%	80	0
	30%	80	1
	40%	70	3
	50%	52	6
160 Patterns	10%	160	0
	20%	150	2
	30%	132	2
	40%	104	4
	50%	64	8
240Patterns	10%	234	0
	20%	220	0
	30%	192	0
	40%	148	0
	50%	84	0
320 Patterns	10%	313	0
	20%	284	0
	30%	240	0
	40%	172	0
	50%	76	0

4. CONCLUSION

The Hopfield recurrent pattern storage network has been used to store and recall the patterns. Here the learning method used is the Hebb. Rule which has an efficiency issue limited to storage capacity and recalling of the distorted patterns which can be improved by changing the learning rules and also by adopting better feature extraction methods. It has been observed that the efficiency of the network in terms of storage and recall can be improved by training these patterns with enhancement procedure followed by Fast Fourier Transform in comparison with training raw image patterns without enhancement. Here the Hopfield network performs better for less number of patterns with less percentage of error that improves the storage capacity and also recalls efficiency. In this study, it has been observed that if the percentage of error increase then the recall rate decreases when more number of patterns were trained and stored as compare to less number of patterns. As in our experiment for 80 patterns with 30, 40 and 50 percentage of errors, our recall rate are 100%, 87% and 65% in comparison with for 320 patterns for the same percentage of errors are 75%, 54% and 24% respectively. It also has been observed that the false minima is quite more when less number of patterns were trained i.e even on 30 – 40% of distorted pattern, it associated with one of the pre-stored pattern in the network. The study observed in experiment is quite encouraging and it still indicate that there are other scopes and dimensions to be undertake for further research work to improve the storage capacity and recall rate.

REFERENCES

[1] H.C. Lee and R.E. Gaensslen. Advances in Fingerprint Technology. Elsevier, 1991.

[2] A.K. Jain, A. Prabhakar, and L. Hong. A multichannel approach to fingerprint classification. IEEE transactions on pattern analysis and machine intelligence, 21(4):348–359, 1999.

[3] Kevin Takasaki, “Critical Capacity of Hopfield Networks”, MIT Department of Physics, 2007, URL: http://web.mit.edu.physics/.

[4] Krose B., Van der Smagt P., An Introduction to Neural Networks, Chapter 6, Eight Edition, University of Amsterdam, 1996.

[5] Jiang, X., Yau, W.-Y. (2000). Fingerprint minutiae matching based on the local and global structures. Proc. 15th Internet.Conf. Pattern Recognition (ICPR, 2000) 2. pp.1042–1045.

[6] L. Hong. Automatic Personal Identification Using Fingerprints. PhD thesis, Michigan State University, East Lansing, U. S. A., June 1998.

[7] S. C. Dass. Markov Random Field Models for Directional Field and Singularity Extraction in Fingerprint Images. IEEE Transactions on Image Processing , 2004, pp225-238.

[8] B. Yegnanarayana, Artificial Neural Networks, Prentice Hall of India, 2006.

[9] W.Tarkowski, M.Lewenstein, A.Nowak, “Optimal Architectures for Storage of Spatially Correlated Data in Neural Network Memories”, ACTA Physica Polonica B, 1997, Vol. 28, No.7, pp 1695 – 1705.

[10] Neil Davey, S.P Hunt, Rod Adams, “High Capacity Recurrent Associative Memories”, Neurocomputing - IJON, 2004, vol. 62, pp. 459 - 491, DOI: 10.1016/j.neucom.2004.02.007.

[11] Amos Storkey, “Increasing the Capacity of a Hopfield Network Without Sacrificing Functionality”, Artificial Neural Networks – ICANN’97, 1997, 451 – 456.

[12] Nicholas Sandirasegaram, Ryan English, “Comparative Analysis of Feature Extraction (2D FFT & Wavelet) and Classification (Lp metric distances, MLP NN & HNet) algorithms for SAR imagery”, Procedings Of SPIE, 2005, Vol. 5808, pp 314 - 325.

[13] Zafer Iscan, Mehmet Nadir Kurnaz, Zumray Dokur, Tamer Plemz, “Ultrasound Image Segmentation by Using Wavelet Transform and Self Organizing Neural Network”, Neural Information Processing - Letters and Reviews, 2006, Vol. 10, Nos. 8-9, pp.183 - 191.

[14] D.K.Sharma, Loveleen Gaur, Daniel Okunbor, “Image Compression and Feature Extraction with Neural Network”, Proceedings of Academy the of Information and Management Sciences, 2007, Volume 11, No. 1, pp 33 - 37

[15] Nicholas Sandirasegaram, Ryan English, “Comparative Analysis of Feature Extraction (2D FFT & Wavelet) and Classification (Lp metric distances, MLP NN & HNet) algorithms for SAR imagery”, Procedings Of SPIE, 2005, Vol. 5808, pp 314 - 325

[16] Neil Davey, S.P Hunt, Rod Adams, “High Capacity Recurrent Associative Memories”, Neurocomputing - IJON, 2004, vol. 62, pp. 459 - 491, DOI: 10.1016/j.neucom.2004.02.007.

[17] Somesh Kumar, Manu P. Singh, “Study of Hopfield neural network with sub-optimal and random GA for pattern recalling of English characters”, Elsevier, Applied Soft Computing, Volume 12, Issue 8, 2012, 2593 - 2600.

[18] Agnes Meyder, Constantin Kiderlen, “Fundamental Properties of Hopfield Networks and Boltzmann Machines for Associative Memories”, Machine Learning, vt 2008.

[19] S.Y.Kung, “Digital Neural Networks”, New Jersey: Prentice-Hall, 1993.

[20] M. W. Hirsch, Convergent activation dynamics in continuous time networks, Neural Networks, 1989, v.2 n.5, p.331-349.

[21] L. Lam, S. W. Lee, and C. Y. Suen, “Thinning Methodologies-A Comprehensive Survey”, IEEE Transactions on Pattern analysis and machine intelligence, vol. 14, no. 9, 1992.

[22] L. Hong, "Automatic Personal Identification Using Fingerprints", Ph.D. Thesis, 1998.

[23] Jain, A.; Lin Hong; Bolle, R., " On-line fingerprint verification", Pattern Analysis and Machine Intelligence, IEEE Transactions on, Volume 19, Issue 4, Apr 1997 pp. 302 – 314

[24] Wang, C. and Gavrilova, M., (2005) A Novel Topology-based Matching Algorithm for Fingerprint Recognition in the Presence of Elastic Distortions, CGA’05, LNCS, Springer-Verlag, vol.1, pp.748-757.