Signature Verification on Bank Checks Using Hopfield Neural Network

Vijaypal Singh Dhaka[1] Mukta Rao[2] Manu Pratap Singh[3]

Abstract

The used topology i.e. Associative Hopfield Memory is a very useful form of Artificial Neural Networks. This paper outlines an optimization relaxation approach for signature verification based on the Hopfield neural network (HNN). The standard sample signature of the customer is cross matched with the one supplied on the check. The difference percentage is obtained by calculating the different pixels in both the images. The network topology is built so that each pixel in the difference image is a neuron in the network. Each neuron is categorized by its status, which in turn signifies that if the particular pixel is changed. The network converges to unwavering condition based on the energy function which is derived in experiments. The Hopfield's model allows each node to take on two binary state values (changed/unchanged) for each pixel. The performance of the proposed technique is evaluated by applying it in various binary and gray scale images. This paper contributes in finding an automated scheme for verification of authentic signature on bank check. The derived energy function allows a trade-off between the influence of its neighbourhood and its own criterion. This device is able to recall as well as complete partially specified inputs. The network is trained via a storage prescription that forces stable states to correspond to (local) minima of a network "energy" function.

[1,2&3]Department of Computer Science, ICIS, Dr. B.R. Ambedkar University, Agra. Email : vijaypal.dhaka@gmail.com, manu_p_singh@hotmail.com, mukta.mca@gmail.com

Keywords: Signature Verification, Image Matching, Energy Minimization, Hopfield Neural Network (HNN).

1. Introduction

In 1982, John Hopfield published a very influential paper which drew the attention to the associative properties of a class of neural nets [1]. The analysis was based on a definition of "energy" in the net and a proof that the net operated by minimizing this energy when settling into suitable patterns of operation [2], this helped to reignite the dormant world of Neural Networks once again. This network has attracted considerable interest both as a content address memory (CAM) and, more interestingly, as a method of solving difficult optimization problems [3-5]. The focus for solving optimization problems for researchers have been Hopfield network because of the advantages such as massive parallelism, convenient hardware implementation of the neural network architecture, fully interconnected and a common approach for solving various optimization problems [6]. It has been proven by Hopfield that the network converges when the weights are symmetric with zero diagonal elements (i.e. $w_{ij} = w_{ji}$ & $w_{ii} = 0$) for the hard limiting nonlinear neuron [1] or nonzero diagonal elements (i.e. $w_{ij} = w_{ji}$ & $w_{ii} \# 0$). This stands true when monotonic "sigmoid" neurons with the nonzero diagonal elements are included in the energy function [2]. Some researchers have studied the possessions of the Hopfield neural networks with nonzero diagonal elements in the area of combinatorial optimization problems resolution were also discussed [7]. The experiments conducted were based on a binary Hopfield neural network with a negative self-feedback on same problem domain. An oscillatory neuron unit was also proposed by adding a simple self-feedback, which could also be negative or positive along with an energy value extraction circuit [8-9]. However, each of them attempts to make some local minima unstable by introducing a negative self-feedback or using an oscillatory neuron.

Some researchers apply the Continuous Hopfield Network to three-dimensional object matching problem [10]. However, Continuous Hopfield Network takes much computational time in simulating a differential equation even though it provides good solutions. The Discrete type Hopfield Network has been used for two-dimensional objects matching problems [11]. However, DHN is an approximation method and gives only rough solutions, but it reduces computational time. Unlike the travelling salesman problem implemented by the Hopfield neural network, the matching problem is handled by normalizing features made by a fuzzy function which gives distinguishable values to a connectivity matrix.

The rest of the paper is organized as follows. Section 2 provides a brief description of the Hopfield model as well as some terminologies to be used throughout the paper. Section 3 explains the implementation procedure. Section 4 details the pre-processing, training & test signature samples. Section 5 describes the experimental results. Finally, Section 6 concludes the paper. Next section contains references.

2. Preliminaries And Terminology

Recurrent networks are networks those have closed loops in the network topology. In this paper, a form of recurrent neural network suitable for auto associative (content) addressable memory and its characteristics, optimization and constraint satisfaction application is considered. This network is often referred to as Hopfield network in the honour of John Hopfield. The topology is very simple: it has 'n' neurons, which are all networked with each other or fully interconnected. The net has symmetric weights with no self connections i.e. all the diagonal elements of the weight matrix of Hopfield net are zero.

The two main differences between Hopfield and iterative auto associative net (recurrent associative net) are that, in the Hopfield net,

- Only one unit updates its activation at a time, and,
- Each unit continues to receive an external signal in addition to the signal from the other units in the net.

2.1. Architecture And Mathematical Foundation Of Hopfield Network

The architecture shown in Figure 1. The Hopfield network consists of a set of neurons and a corresponding set of unit delays, forming a multiple loop feedback system. The number of feedback loops is equal to the number of neurons. Basically, the output of each neuron is fed back, via a unit delay element, to each other neuron in the network.

Hopfield network considered here is made up of neurons which possess additive model (current summing model).

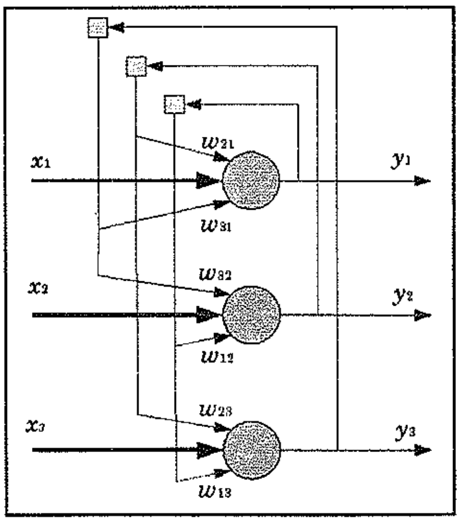


Figure 1: A Hopfield Net With 3 Nodes. Each Node's Output Is Fed-Back To All Other Nodes.

From *Kirchoff's current low*, on the input node of the HNN we get

$$C_j \frac{dv_j(t)}{dt} = -\frac{v_j(t)}{R_j} + \sum_{i=1}^{N} w_{ji} x_i(t) + I_j, \quad j = 1, 2, \ldots, N \tag{2.1}$$

Where

R_j = leakage resistance

$x_i(t)$ = stimuli acting on synaptic weights.

w_{ji} = synaptic weight.

$v_j(t)$ = induced local field at the input

C_j = leakage capacitance

I_j = current source/ externally applied bias

Recognizing that $x_i(t) = \varphi(v_j(t))$

$$C_j = -\frac{d}{dt} v_j(t) = \frac{v_j(t)}{R_j} + \sum_{i=1}^{N} w_{ji} \varphi_i(v_i(t)) + I_j, \quad j = 1, 2, \ldots, N \tag{2.2}$$

Where $\phi(v_j(t))$ is the non-linear activation function.

This dynamic has got following assumptions:-

1. The matrix of synaptic weights is symmetric:

$w_{ji} = w_{ij}$ for all i and j

2. Each neuron has a *non linear* activation of its own – hence $\phi(v_j(t))$ used in equation.

3. The *inverse* of the nonlinear activation function exists, so we may write :-

$$v = \varphi_i^{-1}(x) \tag{2.3}$$

Let the sigmoid function $\phi(v_j(t))$ be defined by the hyperbolic tangent function

$$x = \varphi_i(v) = \tanh\left(\frac{a_i v}{2}\right) = \frac{1 - \exp(-a_i v)}{1 + \exp(-a_i v)} \tag{2.4}$$

Which has the slope of $a_i/2$ at the origin as shown by -

$$\frac{a_i}{2} = \frac{d\varphi_i}{dv}\Big|_{v=0} \tag{2.5}$$

Henceforth we refer to as the gain of neuron i

The inverse output-input relation of eq (2.4) may thus be rewritten in the form

$$v = \varphi_i^{-1}(x) = -\frac{1}{a_i} \log\left(\frac{1-x}{1+x}\right) \tag{2.6}$$

The standard form of the inverse output-input relation for a neuron of unity gain is defined by:-

$$\varphi_i^{-1}(x) = -\log\left(\frac{1-x}{1+x}\right) \tag{2.7}$$

We may rewrite eq (2.6) in terms of this standard relation as:-

$$\varphi_i^{-1}(x) = \frac{1}{a_i} \varphi^{-1}(x) \tag{2.8}$$

The energy (Lyapunov) function E of the Hopfield net work is defined by

$$E = -\frac{1}{2}\sum_{j=1}^{N}\sum_{j=1}^{N} w_{ji} x_i x_j + \sum_{j=1}^{N} \frac{1}{R_j} \int_0^{x_j} \varphi_j^{-1}(x) dx - \sum_{j=1}^{N} I_j x_j \tag{2.9}$$

This energy function may have a complicated *lands cape* with many minima. The dynamics of the network are described by a mechanism that seeks out those minima. Hence differentiating E with respect to time, we get

$$\frac{dE}{dt} = -\sum_{j=1}^{N}\left(\sum_{i=1}^{N} w_{ji} x_i - \frac{v_j}{R_j} + I_j\right)\frac{dx_j}{dt} \tag{2.10}$$

By replacing values on right hand side from eq(2.2) we get

$$\frac{dE}{dt} = -\sum_{j=1}^{N} C_j \left(\frac{dv_j}{dt}\right)\frac{dx_j}{dt} \tag{2.11}$$

We recognize the inverse relation that defined vi in terms of xj. The use of eq (2.6) in eq (2.11) yields

$$\frac{dE}{dt} = -\sum_{j=1}^{N} C_j \left[\frac{d}{dt}\varphi_j^{-1}(x_j)\right]\frac{dx_j}{dt}$$

$$= -\sum_{j=1}^{N} C_j \left(\frac{dx_j}{dt}\right)^2 \left[\frac{d}{dt}\varphi_j^{-1}(x_j)\right] \tag{2.12}$$

If we graphically represent this function then we may find that inverse output-input relation $\varphi_j(x_j)$ is a monotonically increasing function of the output. It follows therefore that

$$\frac{d}{dx_j}\varphi_j^{-1}(x_j) \geq 0 \text{ for all } x_j \qquad (2.13)$$

We also note that

$$\left(\frac{d}{dt}(x_j)\right)^2 \geq 0 \text{ for all } x_j \qquad (2.14)$$

Hence, all the factors that make up the sum on the right hand side of eq 31 are non negative. In other words, for the energy function E defined in Eq (2.12) we have

$$\left(\frac{d}{dt}E\right) \leq 0 \qquad (2.15)$$

From the definition of eq (2.15), we note that the function E is bounded. Accordingly, we may make the following two statements:-

1. The energy function E is a Lyapunov function of the continuous Hopfield model.

2. The model is stable in accordance with Lyapunov's Theorem 1.

2.2. The Model As Content-Addressable Memory

John Hopfield originally designed the network as a device that would retrieve or complete certain patterns if only part of the pattern, or even a slightly distorted part of the pattern, was given to it as the starting state or input pattern. This is described as an associative memory: the network will associate or map the incomplete input with the full pattern it has 'learned" or stored, and it will retrieve this. This mapping of Hopfield network is different from Feed Forward Neural Network in the sense that it maps states into states instead of input to output. The network input is the initial state, and the mapping is through one or more states to form the network. The Hopfield networks have memory and therefore can store a set of unit outputs or system information state for use as a content-addressable memory (CAM). The network provides nearest-neighbour association of the input (initial state) with the set of stored patterns.

The content-addressable memory is such a device that returns a pattern when given a noisy or incomplete version of it. In this sense a content-addressable memory is error-correcting as it can override provided inconsistent information.

The discrete Hopfield network as a memory device operates in two phases: *storage phase* and *retrieval phase*. During the storage phase the network learns the weights after presenting the training examples. The training examples for this case of automated learning are binary vectors, called also fundamental memories. The weights matrix is learned using the Widrow-Hoff rule. According to this rule when an input pattern is passed to the network and the estimated network output does not match the given target, the corresponding weights are modified by a small amount. The difference from the single-layer perceptron is that no error is computed; rather the target is taken directly for weight updating.

The summation block of the i-th neuron is computed as:

$$S_i = \sum_{j=1}^{N}\left[w_{ji}\, x_j(t)\right] \qquad (2.16)$$

and next thing is to modify the weights depending on the result. There are two cases to consider:

if $s_i >= 0$ and $x_i = 0$ the output should be made negative, so each weight has to be decreased by :

$$w_{ji} = w_{ji} - (0.1 + s_i)/n \qquad (2.17)$$

if $s_i < 0$ and $x_i = 1$ the output should be made positive, so each weight has to be increased by:

$$w_{ji} = w_{ij} + (0.1 - s_i)/n \qquad (218)$$

where learning rate coefficient is 0.1

During the retrieval phase a testing vector called probe is presented to the network, which initiates computing the neuron outputs and evolving the state. After sending the training input to the recurrent network its output changes for a number of steps until reaching a stable state. The selection of the next neuron to fire is asynchronous, while the modifications of the state are deterministic. After the state evolves to a stable configuration, that is the state is not more updated, the network produces a solution. This state solution can be envisioned as a fixed point of the dynamical network system. The solution is obtained after adaptive training.

2.3. Learning And Training Algorithm For The Proposed Hopfield Network

Learning: Present the given training (binary) vectors to the Hopfield net, and calculate the weights using the following method:

$$\text{if } s_i >= 0 \text{ and } x_i = 0 : w_{ji} = w_{ji} - (0.1 + s_i)/n \quad (2.17)$$

$$\text{if } s_i < 0 \text{ and } x_i = 1 : w_{ji} = w_{ji} + (0.1 - s_i)/n \quad (2.18)$$

Initialization: Let the testing vector become initial state *x(0)*

Repeat

-*update* asynchronously the components of the state *x(t)*

$$x'_i(t)=1 \text{ if } s_i(t) = >= 0 \text{ or } x'_i(t)=0 \text{ if } s_i(t) <0 \qquad (2.19)$$

-*continue* this updating until the state remains unchanged

Until convergence

Generate output: return the stable state (fixed point) as a result.

2.4. Energy Minimization

The energy is a behavioural characteristic that can be used to examine the network performance. It is well known that independently from the initial conditions the network will stabilize; it cannot oscillate even at the same energy level. The energy of Hopfield nets is defined in equation (2.9).

The energy in the Hopfield model either decreases or remains the same, the energy never increases. If it decreases it can never return to a previous state. The only case when the energy remains the same is when a neuron output changes from 0 to 1. The convergence of the network can be considered as a process of reducing the network energy until reaching energy well, which is the stable network states are energy wells. Which of the possible energy wells will be reached after training depends on the initial system state. The labelled information in the neighbourhood of a pixel is mapped as an energy function taking into account the inter label relationships. Hence under similar labels the energy function described in equation (2.9) decreases. Also the energy decreases if the pixel under evaluation and its neighbourhood have high probability values computed from Gaussian distributions. Hence it's clear that an efficient use of contextual information improves the signature change detection performance.

3. Implementation Procedure

Under the analogy HNN paradigm, the problem of signature verification is to label each pixel of the incoming images as changed or unchanged and the degree of change. With such purpose, we consider the output image as a network of nodes where each node is associated to a pixel location in the difference image, i.e., the number of nodes is exactly the number of pixels of the incoming images. Also, each node is characterized by its state value v_i, ranging in [-1, +1]. The network state is characterized by the states of the nodes. After the optimization process, when the network stability is reached, the nodes have achieved its final state value.

This final value will indicate unchanged (-1) or maximum change (+1); intermediate values give the strength of the change. The optimization process minimizes the energy in (equation no) starting from an initial network state. The interconnection weights in (equation no) are computed by applying data and contextual consistencies and the external inputs through the self-data information. The procedure can be depicted using following diagram:-

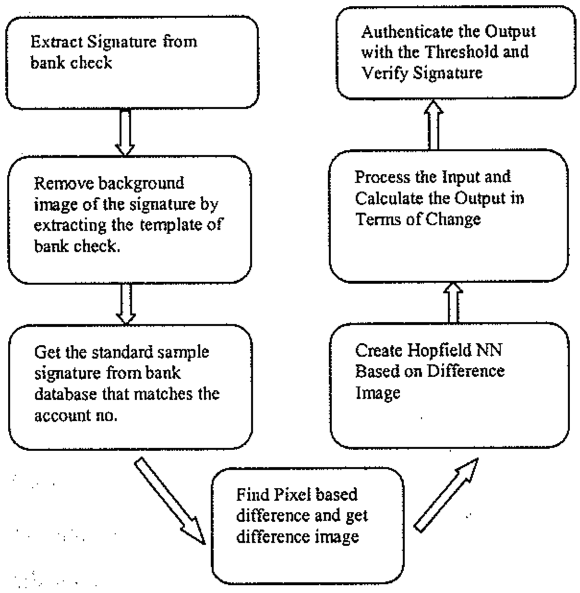


Figure 2: Overall Processing Of Proposed Signature Verification System Based On HNN

3.1 Pre-Processing And Network Initialization

The signature on the bank check is always containing the back-ground, because of the reason that the check leaves cannot be forged. Although this is a security feature for the bank but it leads to specific module in our system. We have to pre-process the check's image before it can be presented to the system.

The back-ground image of a signed bank check's leaf can be removed in two ways:-

1). It is evident from the real-life samples that the back-ground image's intensity is always very less as compared to the signature's ink density. Hence the image of signature can be binarized in way that the intensity threshold removes the back-ground.

2). In this method we can subtract the standard template back-ground image from the one that is signed signature, this subtraction should be done as a Homomorphic image subtraction (gray scale based).

For the proposed system the input image is changed based on above described method 2. The network initialization is carried out by exploiting the characteristics of the difference image. We use the initialization strategy, described in [12], as follows. From the histogram of the difference image, we compute thresholds. Now, given a pixel location in the difference image, its associated node in the network is initialized based on this threshold. In the work of Wang [14], a list of major grouping principles is exhaustively studied. In our signature verification approach, we apply the following two principles: proximity, changed/unchanged pixels that lie close in space tend to group. The State of each pixel is confirmed as following:-

Difference Image is computed as:-

$$D(x,y) = I_1(x,y) - I_2(x,y) \qquad (2.20)$$

Initial State of HNN =

$$O(x,y) = 1, \text{ if } D(x,y) > \text{Threshold (changed)}$$
$$= 0, \text{ Otherwise} \quad \text{(no change)} \qquad (2.21)$$

4. Training & Test Signature Samples and Experiments

The success of any technique is closely dependent on the samples on which the experiments have been conducted. The samples of signatures were gathered from academic staff of Dr. B.R. Ambedkar University. The members gave signature samples on blank sheet whereas cross marked but signed check leaves were provided for verification purpose. 83 such samples were gathered and processed. 40 samples were consumed during learning process of HNN where as rest of the 40 samples were used for testing of the system. 3 samples were discarded because of being distorted. During the experiments one intermediate image was obtained. That is a truth image; which is the simple difference image that state if the pixel is changed or not, this has been derived as a binary image. As shown in the following figures:-

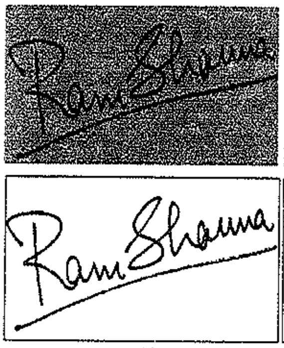


Figure 3(a)

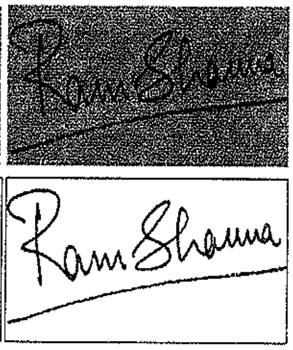


Figure 3(b)

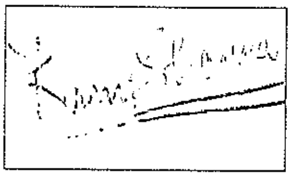


Figure 3(c)

Figure 3(A): Sample Signature Of The Customer Stored In The Bank's Database (Original And Binarized). Figure 3(B): Signature Of The Customer Captured From The Check (Original And Binarized). Figure 3(C): Difference Image Calculated From The Difference Of Both Binary Images Of The Signature.

To deduce the network performance the percentage of correctly labelled pixels was computed based on following formula:-

$$\text{Perc} = ((NT + PT)/(NT+PT+NF+PF))*100 \quad (2.22)$$

Where

NT = Negative True; i.e. number of no change pixels correctly detected by HNN

PT = Positive True; i.e. number of changed pixels correctly detected by HNN

NF = Negative False; i.e. number of no change pixels incorrectly detected by HNN

PF = Positive False; i.e. number of changed change pixels not detected by HNN

Detecting by the HNN simply means as labelling of the state value. The judgement of the signature verification was based on the higher value of the Perc. The network was made o continue the iterations until it reached to the 95% of percentage (network goal). When done so, the number of actual changed pixels were calculated and compared to the threshold. If the threshold is lower that this number, signature is said to be forged.

5. Experimental Results

The performance of the system is analysed in terms of the correct verifications, computation time and number of epochs (network iterations). Table 1 shows the number of correctly verified signatures which were actually correct and incorrect. Figure 4 shows the number of epochs for verification of correct training and testing data sets.

Table 1: Hopfield Neural Network Energy Valuesr The Training And Test Data Set

Epochs	Training Data Set Energy (x 10^6)	Testing Data Set Energy (x 10^6)
1	-1.1167	-0.7167
2	-1.893	-1.193
3	-2.4	-1.26
4	-2.7	-2.1
5	-3.3	-2.43
6	-4.1	-2.91
7	-4.5	-3.25
8	-4.7	-3.3
9	-4.8	-3.6
10	-4.84	-3.65

The graphical representation of data from table 4 is shown below:-

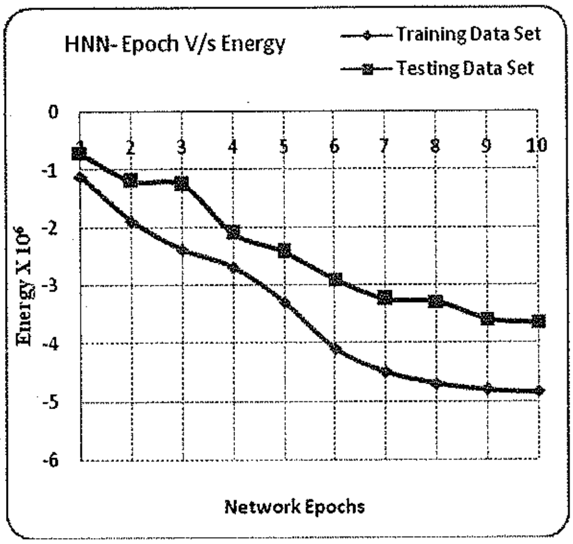


Figure 4: Comparison Of Energy Value Convergence Of The System With Training And Test Samples Of The Signature

Table 2: Signature Verification Data Result Statistics

Type	Verified	Discarded
Genuine Signatures	82%	18%
Forge Signatures	11%	89%

Table 3: HNN Correct State Percentage Values For The Training And Test Data Set

Epochs	Training Data Set Percentage	Testing Data Set Percentage
1	0.46	0.54
2	0.51	0.57
3	0.57	0.58
4	0.6	0.63
5	0.67	0.64
6	0.76	0.67
7	0.82	0.7
8	0.86	0.72
9	0.88	0.745
10	0.9	0.75

The graphical representation of data from table 3 is shown below:-

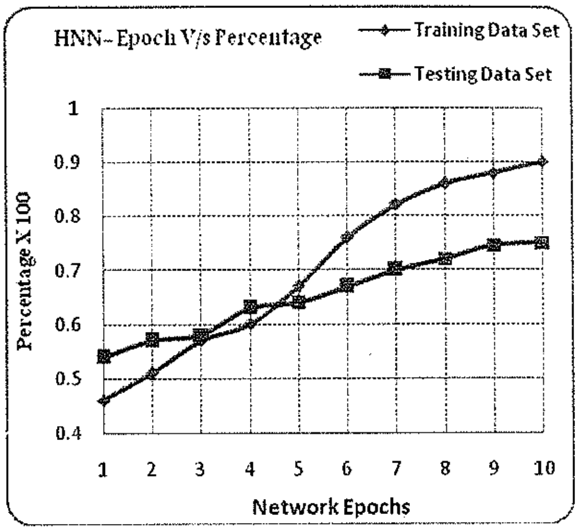


Figure 5: The Perc Values Computed From Equation (2.22) Against The Network Iterations For Training And Test Data Sets.

6. Discussion And Conclusion

In this paper, we have developed a new automatic strategy for verification of authentic signatures in bank checks using Hopfield Neural Networks. Some researchers have developed strategies on pattern recognition using HNN; we have extended their work for the specific area of signature verification. We have also used this kind of information for integrating the data and contextual information in form of an energy function that converges rapidly. The mapping of data information is improved in relation to classical implementations of this information. This improvement is achieved through the inter-data relations. As it is visible from figure 4, the convergence is fast and the results of the HNN are better in the sequence without significant illumination changes. The results in Table 2 display that the performance of network is remarkable. The best performance in terms of accuracy is obtained with iterative methods. Part of the improvement of HNN is due to the mapping of the self-information applied by HNN. This takes special relevance near the borders of changed areas. The initialization process can be important but it is not decisive. The noise removal process of the check's image is still a big challenge for this strategy. The main drawback of Hopfield network is the high evaluation time, so for real life situations and faster responses, this strategy should be implemented on parallel architectures. The process should be further improved to get more accurate results.

References

[1]. J. J. Hopfield, *"Neural network and physical systems with emergent collective computational abilities"*, Proc. Nat. Acad. Sci. USA, Vol. 79, PP.2554–2558, 1982.

[2]. J. J. Hopfield and D. W. Tank, *"Neural computation of decisions in optimization problems"*, Biol. Cybern, Vol. 52, PP. 141–152, 1985.

[3]. J. J. Hopfield *"Neurons with graded response have collective computational properties like those of two-state neurons"*, Proc. Nat. Acad. Sci. USA, Vol. 81, PP. 3088–3092, 1984.

[4]. J. J. Hopfield and D. W. Tank, *"Computing with neural circuits: A model Science"*, No. 233, PP.625–633, 1986.

[5]. D.W. Tank and J. J. Hopfield, *"Simple neural optimization network: An A/D converter, signal decision circuit, and linear programming circuit"*, IEEE Trans. Circuits Syst., Vol. CAS-33, PP. 533–541, May 1986.

[6]. X. C. Zeng and T. Martinez, *"A new relaxation procedure in the Hopfield neural networks for solving optimization problems"*, Neuron Processing Lett., Vol. 10, PP. 211–222, 1999.

[7]. M. Ohta, A. Ogihara, and K. Fukunaga, *"Binary neural networks with negative self-feedback and its aPPlication to N-Queens problem"*, IEICE Trans. Inform. Syst., Vol. E77-D, no. 4, PP. 459–465, 1994.

[8]. K. Yoshino, Y. Watanabe, and T. Kakeshita, *"Hopfield neural network using oscillatory units with sigmoidal input-average out characteristics", IEICE Trans.*, Vol. J77-DII, PP. 219–227, 1994.

[9]. Y. Watanabe, K. Yoshino and T. Kakeshita, *"Solving combinatorial optimization problem using oscillatory neural network"*, IEICE Trans., Vol. E80-D, No. 1, PP. 72–77, 1997.

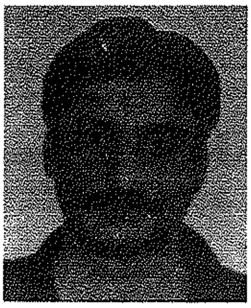
[10].W. Lin, F. Liao. C. Taso and T. Lingutle, *"A hierarchical Multiple-view approach to three-dimensional object recognition"*, IEEE Tram. on Neural Net. Vol. 2 PP. 84-92, 1991

[11].W. Li and M. Nasirabadi *"Object recognition based on graph matching implemented by a hopfield-style neural network"*, Int. J. Cor$ Neural Networks. 11, PP. 287-290, June 18-22, 1989.

[12].L. Bruzzone and D. Fernández Prieto, *"Automatic analysis of the difference image for unsupervised change detection"*, IEEE Trans. Geosci. Remote Sens., Vol. 38, No. 3, PP. 1171–1182, Mar. 2000.

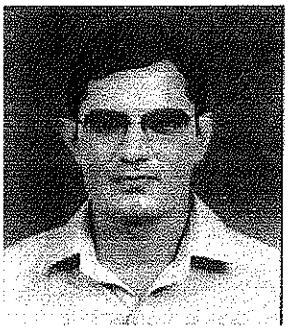
[13].D.Wang, *"The time dimension for scene analysis"*, IEEE Trans. Neural Network., Vol. 16, No. 6, PP. 1401–1426, Nov. 2005.

Author's Biography



Manu Pratap Singh Received his Ph.D. in computational physics from Kumaun university, Nainital,UA, India in 2001. He is currently a Reader in department of computer science, ICIS, Agra University, Agra, UP, India. His research interests are focused on neuro-computing, neuro-science, neuro - informatics, soft computing etc.

Dr. Singh is a member of technical committee of IASTED, Canada since 2004. He has been referee for various international journals. He received Young Scientist Award from International Academy of Physical Science, Allahabad, UP, India. He has more than 40 publications in various international and national journals.



Vijaypal Singh Dhaka is currently pursuing his Ph.D. in computer science from Dr. BR Ambedkar University, Agra, UP, India. He is currently a Sr.Lecturer in department of computer science and engineering ,Maharaja Surajmal Institute of Technology, New Delhi, India. He completed his M. Phil. in computer science in 2007. His research interests are neuro-computing, character recognition, soft-computing and internet technologies.

Mukta Rao is currently working with Interglobe Technologies Ltd, Gurgaon, Haryana, India as a senior software engineer. She has done her MCA from University of Rajasthan, India. She has been associated with Hughes Software Systems and Infosys. Her research areas include Handwriting recognition, Neuro-computing, Web-Acceleration and Network security.