Semi-supervised Hopfield-Type Neural Network for change detection in remotely sensed images

Conference Paper · March 2012

DOI: 10.1109/RAIT.2012.6194450

4 authors:

Dr. Moumita Roy
Indian Institute of Information Technology Guwahati

Sukhen Das
Jadavpur University

Susmita Ghosh
Jadavpur University

Ashish Ghosh
Indian Statistical Institute

Semi-supervised Hopfield-Type Neural Network for Change Detection in Remotely Sensed Images

Moumita Roy, Suvadeep Das and Susmita Ghosh
Department of Computer Science and Engineering
Jadavpur University
Kolkata 700032, INDIA
moumita2009.roy@gmail.com,
suvadeep123789das@gmail.com
susmitaghoshju@gmail.com

Ashish Ghosh
Center for Soft Computing Research
Indian Statistical Institute
Kolkata 700108, INDIA
ash@isical.ac.in

Abstract: **In this article, we propose a change detection technique using semi-supervised Hopfield-Type Neural Network (HTNN). The purpose of the work is to show the usefulness of semi-supervision over existing unsupervised/fully supervised methods when we have only a few labeled samples. Here, training of HTNN is performed iteratively using a few labeled patterns along with a number of unlabeled patterns. A method has been suggested to propagate the label information using a kind of K-nearest neighbor approach. To check the effectiveness of the proposed method, experiments are carried out on multi-temporal remotely sensed images. Results are compared with other state of the art techniques and found to be significantly better.**

Keywords: Change detection;multi-temporal images; Hopfield neural network;Semi-supervised learning

I. INTRODUCTION

Change detection [1], [2] is a process of detecting temporal effects of multi-temporal images. This process is used for finding out changes in a land cover over time by analyzing remotely sensed images of a geographical area captured at different time instants. The changes can occur due to natural hazards (e.g., disaster, earthquake), urban growth, deforestation etc. There are various applications of change detection like analysis of damage assessment, land use change analysis, day/night analysis of thermal characteristics, burned area identification etc. Change detection process can be performed by manual analysis of the images. But human participation is costly and time consuming. Therefore, automated techniques for detecting changes are required to reduce human effort.

Change detection can be viewed as an image segmentation problem where two groups of pixels are formed, one for the changed class and the other for the unchanged one. Methodology of change detection can be broadly classified into two categories: supervised [3] and unsupervised [4]-[6]. Supervised technique has certain advantages like it can explicitly recognize the kinds of changes occurred and it is robust to different atmospheric and light conditions at two acquisition dates. Various techniques exist in literature to carry out supervised change detection techniques e.g., post classification method [1], direct multi-date classification method [1], kernel based methods [3] etc. Besides several advantages, the applicability of supervised methods in change detection is poor due to the mandatory requirement of sufficient amount of ground truth information which is expensive, hard and monotonous. On the contrary, in unsupervised approach, there is no need of additional information like ground truth. Due to the depletion of labeled patterns, unsupervised techniques seem to be compulsory for change detection. Generally, three consecutive steps are followed for unsupervised change detection. These are image preprocessing, image comparison and image analysis [1]. Images of the same geographical area, captured at different time instants, constitute the input of the change detection process. In preprocessing step, these images are made compatible by operations like radiometric and geometric corrections, co-registration, noise reduction etc. After preprocessing, image comparison is carried out, pixel by pixel, to generate a difference image (DI) which is subsequently used for change detection. There are various methods for generating the DI like univariate image differencing, change vector analysis (CVA), image ratioing [1]. In the present work, CVA technique [1] is used for creation of DI. Unsupervised change detection process can be divided into two categories: context insensitive [1] and context sensitive [4]-[7]. Histogram thresholding [1] is the simplest unsupervised context insensitive change detection method which has the main disadvantage of not considering spatial correlation between neighborhood pixels in the decision process. To overcome this difficulty, context sensitive methods using Markov random field (MRF) [7] are developed. These techniques also suffer from certain difficulties like requirement of the selection of proper model for statistical distribution of the changed and the unchanged class pixels. Change detection methodologies based on neural networks, both using supervised and unsupervised learning [4], [5], are found in literature which are free from such limitations.

In change detection, a situation may occur where the categorical information of a few labeled patterns can be collected easily by the experts. If the number of these labeled patterns is less, then this information may not be sufficient

978-1-4577-0697-4/12/$26.00 ©2012 IEEE

for developing supervised methods. In such a scenario, knowledge of labeled patterns, though very poor but expensive, will be completely unutilized if unsupervised approach is considered. Under this circumstance, semi-supervised approach [8], [9] can be opted instead of unsupervised or supervised one. Semi-supervision has been used successfully for improving the performance of clustering and classification [10] when insufficient amount of labeled data are present. Semi-supervised learning using neural networks is explored in various domains [11]. Work along this direction for change detection, is also carried out by effective application of multilayer perceptron (MLP) [12] and modified self-organizing feature map neural network [13].

In the present work, neural network based change detection methodology using domain knowledge is proposed. One of our earlier works on unsupervised change detection technique based on Hopfield-Type neural network (HTNN) [4] is modified to utilize the available labeled patterns (though, a few) under semi-supervised framework. The network architecture considered is similar to the one as used in [4]. Each neuron in HTNN represents a spatial position in DI. Here, re-initialization of the external bias of HTNN is done phase by phase. In each phase, the network is made to converge and attains an output status corresponding to a stable state (a local minimum) in the similar way as it was done in [4] and [14]. Thereafter, the output status of the unlabeled patterns is re-estimated using K-nearest neighbor approach to propagate the label/class information. The output status of the representative neurons of the labeled patterns are assigned to +1 for the changed class pixels and -1 for the unchanged ones and are not updated during the entire learning process. Modified output status is used for re-initializing the external bias for the next phase of HTNN. Learning of the network is continued iteratively until convergence.

To assess the effectiveness of the proposed method, experiments are carried out on multi-temporal and multi-spectral images and the results are compared with other state of art techniques. This study concludes that the proposed semi-supervised change detection technique is better than the existing ones.

II. Background: Hopfield Neural Networks

A Hopfield neural network [15, 16] consists of a set of neurons. The output of each neuron is fed back to each of the other neurons in the network. The input U_i to the i^{th} neuron comes from two sources, namely i) input V_j from other units (to which it is connected) and ii) external input bias I_i, which is a fixed bias applied externally to the unit i. Thus, the total input to a neuron i is given by

$$U_i = \sum_{j=1, j\neq i}^{n'} W_{ij} V_j + I_i, \tag{1}$$

where the weight W_{ij} represents the synaptic interconnection strength from neuron j to neuron i, and n' is the total number of neurons in the network. The connection strengths are assumed to be symmetric, i.e., $W_{ij} = W_{ji}$. The output V_i of neuron i is defined as:

$$V_i = g(U_i), \tag{2}$$

where, $g(\, . \,)$ is an activation function. There are two types of Hopfield models (i.e., discrete and continuous) based output values that a neuron can assume.

A. Discrete Model

In the discrete model, the output V_i of neuron i is either +1 or -1. In this model, the activation function $g(\, . \,)$ is defined as:

$$V_i = g(U_i) = \begin{cases} +1, & U_i \geq \theta_i \\ -1, & U_i < \theta_i, \end{cases} \tag{3}$$

where θ_i is the predefined threshold of neuron i.

The energy function E of the discrete model is given by:

$$E = -\sum_{i=1}^{n'} \sum_{j=1, i\neq j}^{n'} W_{ij} V_i V_j - \sum_{i=1}^{n'} I_i V_i + \sum_{i=1}^{n'} \theta_i V_i. \tag{4}$$

B. Continuous Model

In this model, the output of a neuron is continuous and can assume any real value between [-1, +1]. In the continuous model, the activation function $g(\, . \,)$ must satisfy the following conditions: i) it is a monotonic non-decreasing function and ii) $g^{-1}(\, . \,)$ exists.

A typical choice of the function $g(\, . \,)$ is

$$g(U_i) = \frac{2}{1 + e^{-\phi_i (U_i - \tau)}} - 1, \tag{5}$$

where the parameter τ controls the shifting of the sigmoidal function $g(\, . \,)$ along the abscissa, and Φi determines the steepness (gain) of neuron i. The value of $g(U_i)$ lies in [-1, +1] and is equal to 0 at $U_i = \tau$. The energy function E of the continuous model is given by

$$E = -\sum_{i=1}^{n'} \sum_{j=1, i\neq j}^{n'} W_{ij} V_i V_j - \sum_{i=1}^{n'} I_i V_i + \sum_{i=1}^{n'} \frac{1}{R_i} \int_0^{V_i} g^{-1}(V_i) dV, \tag{6}$$

when $\Phi i = \infty$ for all i, the maxima and the minima of the continuous model become identical to those of the corresponding discrete Hopfield model. In this case, the energy function takes the form:

$$E = -\sum_{i=1}^{n'} \sum_{j=1, i\neq j}^{n'} W_{ij} V_i V_j - \sum_{i=1}^{n'} I_i V_i. \tag{7}$$

III. The Proposed Algorithm Selecting

In the proposed technique, concept of semi-supervision is integrated with the existing unsupervised change detection technique based on HTNN [4]. Semi-supervised learning of HTNN requires a small amount of labeled patterns. These labeled patterns can be collected in many ways. In the present methodology, for experimental purpose, labeled patterns from both the classes with equal percentage are acquired randomly from the ground truth. The ground truth is generated by manually analyzing the two multi-temporal images of the same geographical area at different times. Detailed description of the proposed change detection technique is given below.

A. Generation of Input Pattern

The difference image $D = \{l_{mn}, 1 \leq m \leq p, 1 \leq n \leq q\}$ is produced by CVA technique [1] from two co-registered and radiometrically corrected γ-spectral band images Y_1 and Y_2 of size p x q of the same geographical area at different times T_1 and T_2. Here, gray value of the difference image D at spatial position (m, n), denoted as l_{mn}, is calculated as,

$$l_{mn} = (int)\sqrt{\sum_{\alpha=1}^{\gamma}(l^{\alpha}_{mn}(Y_1) - l^{\alpha}_{mn}(Y_2))^2},$$

where, $l^{\alpha}_{mn}(Y_1)$ and $l^{\alpha}_{mn}(Y_2)$ are the gray values of the pixels at the spatial position (m, n) in the α^{th} band of the images Y_1 and Y_2, respectively.

B. Hopfield-Type Neural Network Architecture

As mentioned earlier, the network architecture used in the present article is the same as it was used in [4], [14]. Each neuron corresponds to the $(m,n)^{th}$ pixel position of D. Each neuron is connected to all the neurons in its neighboring system (N^d). The selection of the d^{th} order neighborhood system (N^d) depends on the problem at hand. Here, for investigation purpose, we consider the first order neighboring system, i.e., $N^1 = (\pm1, 0), (0, \pm1)$. Let $W_{mn, uv}$ be the connection strength between the $(m, n)^{th}$ and $(u, v)^{th}$ neurons. It is assumed that $W_{mn, uv} = 1$ if $(u, v) \in N^d_{mn}$; else 0.

In continuous model, the generalized fuzzy S-function defined over a finite domain is used as the activation function. The form of the S-function is given as follows:

$$V_{mn} = g(U_{mn}) = \begin{cases} -1, & U_{mn} \leq a \\ 2^r\{\frac{(U_{mn}-a)}{(c-a)}\}^r - 1, & a < U_{mn} \leq b \\ 1 - 2^r\{\frac{(c-U_{mn})}{(c-a)}\}^r, & b < U_{mn} \leq c \\ 1 & U_{mn} > c, \end{cases} \quad (8)$$

where $r \geq 2$, and $b = (a+c)/2$. In this case, $g(U_{mn})$ lies in [-1, +1] with $g(U_{mn}) = 0$ at $U_{mn} = b$. The domain of U_{mn} is $[a, c]$. The value of r tunes the sharpness (steepness) of the function. For the first-order neighborhood, the number of neighbors being four, the input value to a neuron lies in [-5, +5], i.e., $a = -5$, and $c = 5$.

However, for quick convergence, we use the domain of U_{mn} in [-1, +1] and an activation function $g(.)$ is defined as follows:

$$V_{mn} = g(U_{mn}) = \begin{cases} -1, & U_{mn} \leq -1 \\ (U_{mn}+1)^r - 1, & -1 < U_{mn} \leq 0 \\ 1 - (1 - U_{mn})^r, & 0 < U_{mn} \leq 1 \\ 1 & U_{mn} > 1. \end{cases} \quad (9)$$

The energy function, E, considered here is the same as it was used in [4]. The expression of the energy value is given by:

$$E = -\sum_{m=1}^{p}\sum_{n=1}^{q}\sum_{(u,v)\epsilon N^d_{mn}} W_{mn,uv}V_{mn}V_{uv} - \sum_{m=1}^{p}\sum_{n=1}^{q} I_{mn}V_{mn}. \quad (10)$$

C. Initialization of HTNN

If the class label of the $(m, n)^{th}$ pixel is known (i.e., if it is a labeled pattern), the output status of the representative neuron is assigned to +1 (if it belongs to the changed class) or -1 (if it belongs to the unchanged class) and the same status is retained for the entire learning process. On the other hand, if the same is not known (i.e., if it is an unlabeled pattern), the input of the neuron is initialized in the following way based on the type of the model used. In discrete model, the input U_{mn} to $(m, n)^{th}$ neuron is taken as +1, if the gray value of the corresponding pixel in D is greater than an initialization threshold value t; otherwise, a value of -1 is assigned. On the other hand, in continuous model, the input U_{mn} to $(m, n)^{th}$ neuron is proportional to $(l_{mn}/t) - 1$ (if $(l_{mn}/t) - 1 > 1$ then the value +1 is used for initializing the corresponding neuron). Here, the threshold values are computed in a similar way as it was done in [4].

D. A Phase of HTNN Learning Until Convergence

For both the discrete and continuous models, total input to a neuron is calculated using (1). The output status of the unlabeled patterns are then updated using (2). For discrete model, (3) is used for status updation where the threshold value $\theta_{mn} = 0$, for all (m,n). The activation function for the continuous model is defined by (9).

For both the models, at each iteration itr, the external input bias $I_{mn}(itr)$ for the $(m,n)^{th}$ neuron updates its value by taking the output value $V_{mn}(itr - 1)$ of the previous iteration. When the network begins to update its state, energy value is gradually reduced until the minimum (stable state) is reached.

After each iteration, the energy value is calculated using (10). Learning procedure of HTNN continues as long as $E(itr) < E(itr\text{-}1)$, where $E(itr)$ and $E(itr\text{-}1)$ are the overall energy values of the network for the current and the previous iterations, respectively. Otherwise, it is checked whether the energy value is monotonically non-decreasing for the next few pre-defined number of iterations. If so, it is assumed that the convergence is reached for that phase.

E. Iterative Semi-supervised Learning of HTNN

On convergence of HTNN, a local minima is reached which corresponds to a stable state of the network. Now, the output status of the unlabeled patterns are modified using the K-nearest neighbor approach to incorporate label information. To search for the K nearest neighbors, instead of considering all the patterns, we consider only those which lie within a window around that unlabeled pattern. Let, M be the set of K nearest neighbors (spectrally closer) of the $(m,n)^{th}$ unlabeled pattern. Now, the modified output status, $V'(m, n)$ of $(m, n)^{th}$ neuron is computed as,

$$V'_{mn} = \left[\frac{\sum_{(i,j)\in M} V_{ij}}{K} \right]. \quad (11)$$

Then for the continuous model, the external input bias I_{mn} of the $(m, n)^{th}$ neuron (for unlabeled pattern) is re-initialized by V'_{mn} for the next phase of learning. While, in case of discrete model the same is re-initialized by +1 if $V'_{mn} \geq 0$; else -1.

This iterative learning is continued as long as the minimum energy value of the present phase of HTNN is less than that of the previous phase. After the minimum energy value is reached, to ensure stability, the monotonic non-decreasing nature of the energy value is checked for the next few phases.

IV. RESULTS AND ANALYSIS

To evaluate the effectiveness of the proposed method, experiments are conducted on three multi-temporal remotely sensed images. Similar findings are obtained for all the data sets. For illustration, results on Mexico data set are presented here. Performance of the proposed technique is compared with those of the existing unsupervised techniques using HTNN models [4], k-means algorithm [17], a supervised method using MLP and constrained k-means algorithm [10]. Performance measuring indices used are the number of missed alarms (changed class pixels identified as unchanged ones-*MA*), the number of false alarms (unchanged class pixels classified as changed ones-*FA*), the number of overall error (*OE*) and Kappa measure (*KM*) [18]. The average (Avg.) and standard deviation (written in brackets in the table) values (over 10 simulations) of overall error and Kappa measure are also considered for comparison. The best results (denoted by `Min') and the worst results (denoted by `Max') for *MA*, *FA* and *OE*, considering all the simulations, are also shown in a table. To find out K nearest neighbors for each unlabeled pattern, window size was taken as 51 X 51 and the value of K was fixed to 9 including the pattern under consideration. After a minimum was reached, three subsequent steps were checked to ensure stability.

The Mexico data set [4], [5] consists of two multi-spectral images captured by the Landsat Enhanced Thematic Mapper Plus (ETM+) sensor over an area of Mexico taken on 18th April, 2000 and 20th May, 2002. From the entire available Landsat scene, a section of 512 X 512 pixels has been selected as test site. A fire destroyed a large portion of the vegetation in the considered region between two acquisition dates. Initially, we performed some trials in order to determine the most effective spectral bands for detecting the burnt area in the considered data set. On the basis of the results of these trials, band 4 is observed to be more effective to locate the burnt area. Figs. 1(a) and 1(b) show the band 4 images corresponding to April, 2000 and May, 2002, respectively. The difference image (Fig. 1(c)) created by spectral band 4 using CVA technique is only used for further analysis. For evaluation of the proposed approach, a reference map (Fig. 1(d)) was manually constructed according to a detailed visual analysis of both the available multi-temporal images and the said difference image. Different color composites of the reference images were used to highlight all the portions of the changed area in the best possible way. The reference map, thus obtained, contains 25599 changed and 236545 unchanged pixels.

For visual illustration, the change detection maps, corresponding to minimum overall error (obtained over 10 simulations), using the unsupervised method based on HTNN, the proposed semi-supervised method and the supervised method are depicted in Fig. 2. Fig. 2(a) shows the map obtained using the unsupervised technique based on 1st order continuous HTNN model while Fig. 2(b) shows the corresponding map using the proposed technique (with 0.5% training pattern from both the classes). The change detection map using the supervised strategy (with 0.5% training pattern from both the classes) is displayed in Fig. 2(c). It has been observed that the map obtained using the proposed method is a more accurate resemblance of the reference map. From the change detection maps, it is clearly visible that the erroneous classification of the unchanged areas as the changed ones (i.e. false alarms) has been significantly reduced by the proposed method. But, it failed to detect some of the small and scattered changed areas (i.e. missed alarms) where the pixels are on the boundary or the area is surrounded by a vast amount of unchanged regions. This is obviously due to the neighboring effect. It is noticed that this effect is visibly intense, if we consider 2nd order neighboring system for our investigation.

Table I presents the results obtained using the unsupervised method based on HTNN model [4], the supervised method based on MLP and the proposed semi-supervised method. The entries put within the brackets in column titled `Model' represent the threshold values used. From the table it is noticed that the proposed method (considering all the cases of percentage of training patterns) outperforms the corresponding unsupervised version using 1st order discrete and continuous models and the supervised method based on MLP in almost all the cases. It is also observed that in case of the proposed strategy, the average values of both the measuring indices are significantly better than those of the corresponding unsupervised method, but the standard deviation are little more. This might be due to the fact that different training patterns are present over different simulations in case of semi-supervised methodology. On the contrary, selecting training patterns is not at all needed for unsupervised technique. By comparing the standard deviation values of Table I, it has been found that for all the performance measures MLP based supervised technique showed much higher values than those obtained using unsupervised and semi-supervised approaches. This may be due to unavailability of sufficient number of training samples to carry out any supervised method and it might be a typical example of any real life scenario.

The results of the proposed strategy are also compared with the unsupervised k-means algorithm [17] and the semi-supervised constrained k-means algorithm [10] and it has been found that the proposed strategy gives noticeably better results than both the algorithms. The average values of overall error and Kappa measure (over 10 simulations), obtained using k-means algorithm, are 3772 and 0.914725, respectively. The corresponding values, obtained using constrained k-means algorithm with 0.1% training patterns, are 3767.5 and 0.914831, respectively. In short, the results show that the proposed semi-supervised version has an edge over the unsupervised as well as the supervised techniques, when a few labeled patterns are available.

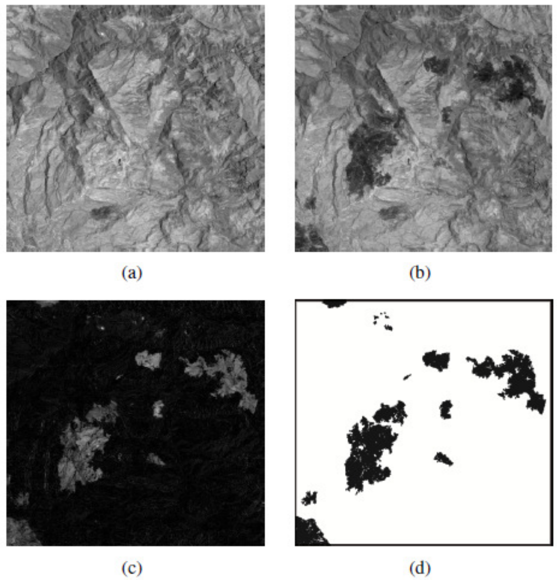

Fig. 1. Images of Mexico area. (a) Band 4 image acquired in April 2000, (b) band 4 image acquired in May 2002, (c) corresponding difference image, and (d) a reference map of the changed area.

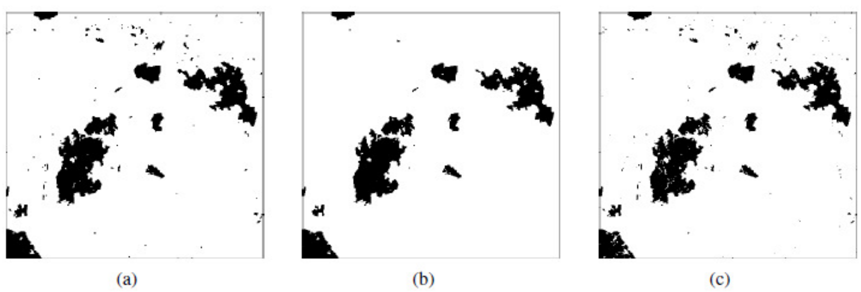

Fig. 2. Change detection maps obtained for Mexico data set: (a) using the unsupervised technique based on 1^{st} order continuous HTNN model, (b) using the proposed semi-supervised technique based on 1^{st} order continuous HTNN model (with 0.5% training pattern), and (c) using the supervised technique based on MLP (with 0.5% training pattern).

TABLE I
RESULTS ON MEXICO DATA SET

Techniques used	Model (t)	Training patterns	Min/ Max	*MA*	*FA*	*OE*	Avg. *OE*	Avg. *KM*
Unsupervised (HTNN)	1st Ord. Dis. (33)	-	-	1102	1858	2960	2960 (0)	0.936761 (0)
	1st Ord. Cont. (31)	-	-	687	2066	2753	2753 (0)	0.941807 (0)
Supervised (MLP)	-	0.1%	Min	1345	1429	2774	3086.4 (186.277857)	0.927558 (0.004676)
			Max	2268	1139	3407		
		0.5%	Min	1269	1406	2675	2834.3 (95.736148)	0.915087 (0.001731)
			Max	875	2192	3067		

Proposed semi-supervised (HTNN)	1st Ord. Dis. (33)	0.1%	Min	1425	1018	2443	2449.6 (3.104835)	0.946602 (0.000067)
			Max	1426	1029	2455		
		0.5%	Min	1403	986	2389	2423.7 (13.833982)	0.947175 (0.000292)
			Max	1419	1023	2442		
	1st Ord. Cont. (31)	0.1%	Min	1434	910	2344	2346.8 (2.227106)	0.948722 (0.000053)
			Max	1445	906	2351		
		0.5%	Min	1415	893	2308	2328.5 (8.476438)	0.949142 (0.000184)
			Max	1408	931	2339		

V. CONCLUSIONS

In this paper, a semi-supervised and context-sensitive technique is proposed for change detection in multi-temporal remotely sensed images based on HTNN, under the scarcity of labeled patterns. Iterative learning of HTNN is performed to incorporate the label information of training patterns using K-nearest neighbor approach. Experiments are carried out on a set of multi-temporal images to confirm the effectiveness of the proposed method. The proposed algorithm significantly outperforms the existing unsupervised technique using the discrete and continuous HTNN models, the k-means algorithm, a supervised method based on MLP and a semi-supervised constrained k-means algorithm. Like most of the existing semi-supervised techniques, the proposed strategy has a drawback of requirement of more computational time.

ACKNOWLEDGMENTS

The authors like to thank the Department of Science and Technology (DST), Government of India and University of Trento, Italy, the sponsors of the ITPAR program and Prof. L. Bruzzone for providing the data. Moumita Roy is also grateful to Council of Scientific & Industrial Research (CSIR), India for providing her a Senior Research Fellowship [No. 09/096(0684)2k11-EMR-I].

REFERENCES

[1] [1] A. Singh, "Digital change detection techniques using remotely-sensed data," *International Journal of Remote Sensing*, vol. 10, no. 6, pp. 989–1003, 1989.

[2] M. J. Canty, *Image Analysis, Classification and Change Detection in Remote Sensing*. Taylor & Francis: CRC Press, 2006.

[3] G. Camps-Valls, L. Gomez-Chova, J. Munoz-Mari, J. L. Rojo-Alvarez, and M. Martinez-Ramon, "Kernel-based framework for multitemporal and multisource remote sensing data classification and change detection," *IEEE Transactions on Geoscience & Remote Sensing*, vol. 46, no. 6, pp. 1822–1835, 2008.

[4] S. Ghosh, L. Bruzzone, S. Patra, F. Bovolo, and A. Ghosh, "A context-sensitive technique for unsupervised change detection based on Hopfield-type neural networks," *IEEE Transactions on Geoscience & Remote Sensing*, vol. 45, no. 3, pp. 778–789, 2007.

[5] S. Ghosh, S. Patra, and A. Ghosh, "An unsupervised context-sensitive change detection technique based on modified self-organizing feature map neural network," *International Journal of Approximate Reasoning*, vol. 50, no. 1, pp. 37–50, 2009.

[6] A. Ghosh, N. S. Mishra, and S. Ghosh, "Fuzzy clustering algorithms for unsupervised change detection in remote sensing images," *Information Sciences*, vol. 181, no. 4, pp. 699–715, 2011.

[7] T. Kasetkasem and P. K. Varshney, "An image change detection algorithm based on Markov random field models," *IEEE Transactions on Geoscience & Remote Sensing*, vol. 40, no. 8, pp. 1815–1823, 2002.

[8] X. Zhu, *Semi-supervised Learning Literature Survey*, Computer Sciences TR1530, University of Wisconsin, Madison, 2008.

[9] O. Chapelle, B. Sch□olkopf, and A. Zien, *Semi-supervised Learning*. Cambridge: MIT Press, 2006.

[10] S. Basu, A. Banerjee, and R. Mooney, "Semi-supervised clustering by seeding," in *Proceedings 19th International Conference on Machine Learning (ICML-2002)*, Sydney, Australia, 2002, pp. 19–26.

[11] A. Verikas, A. Gelzinis, and K. Malmqvist, "Using unlabelled data to train a multilayer perceptron," *Neural Processing Letters*, vol. 14, no. 3, pp. 179–201, 2001.

[12] S. Patra, S. Ghosh, and A. Ghosh, "Change detection of remote sensing images with semi-supervised multilayer perceptron," *Fundamenta Informaticae*, vol. 84, pp. 429–442, 2008.

[13] S. Ghosh and M. Roy, "Modified self-organizing feature map neural network with semi-supervision for change detection in remotely sensed images," in *Proceedings 4th International Conference on Pattern Recognition and Machine Intelligence (PReMI'11)*, vol. LNCS 6744. Moscow, Russia: Springer, 2011, pp. 98–103.

[14] A. Ghosh, N. R. Pal, and S. K. Pal, "Object background classification using Hopfield type neural network," *International Journal of Pattern Recongnition and Artificial Intelligence*, vol. 6, no. 5, pp. 989–1008, 1992.

[15] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," in *Proceedings the National Academy of Sciences of the United States of America*, vol. 79, no. 8, United States of America, 1982, pp. 2554–2558.

[16] ——, "Neurons with graded response have collective computational properties like those of two state neurons," in *Proceedings the National Academy of Sciences of the United States of America*, vol. 81, United States of America, 1984, pp. 3088–3092.

[17] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review," *ACM Computing Surveys,*, vol. 31, no. 3, pp. 264–323, 1999.

[18] R. G. Congalton and K. Green, Assessing the Accuracy of Remotely Sensed Data: Principles and Practices, 2nd ed. Boca Raton, London, New York: CRC Press, Taylor & Francis Group, 2009.