Offline Verification of Hand Written Signature using Adaptive Resonance Theory Net (Type-1)

Tirtharaj Dash
Veer Surendra Sai University of Technology, Dept. of Computer Science and Engineering, Burla-768018, India
Email: tirtharajnist446@gmail.com

Tanistha Nayak
National Institute of Science and Technology, Department of Information Technology, Berhampur-761008, India
Email: tanisthanist213@gmail.com

Subhagata Chattopadhyay
Camellia Institute of Engineering, Dept. of Computer Science and Engineering, Kolkata-700129, West Bengal, India
Email: subhagatachatterjee@yahoo.com

***Abstract*— Recognizing hand-written signature is important due to several legal issues. Manual verification is often difficult when very much 'similar-looking' but forged signature is produced. In this work an effort has been made to automate such kind of signature verification process offline, using Adaptive Resonance Theory type-1 (ART-1). It is implemented using both serial and parallel processing, the performance of which are then compared. The said network has been trained with the original signature and tested with two forged signatures. The grade of similarity has been computed by introducing the term 'Similarity Index' (SI). Performance analysis reveals that after a careful tuning of vigilance parameter (ρ), both serial and parallel processing are able to learn the exemplary patterns with 100% accuracy. While testing, it is noted that parallel processing performs better than the serial processing in terms of speed as well as identifying the forged signatures by computing the mismatch.**

***Index Terms*—-hand-written signature, automatic verification, ART-1, similarity index, forged signatures**

I. Introduction

Learning exemplary patterns is an important property of a Neural Network (NN). Based on the learning mechanisms, there are several types of NN, such as Adaptive Linear Net (ADALINE), multiple ADALINE (MADALINE), Perceptrons, Hopfield net and so forth (uses supervised learning methods), and Kohonen's Self Organizing Map, Adaptive Resonance Theory (ART) net, which work by unsupervised methods [1]. This paper focuses on ART type-1.

Adaptive Resonance theory (ART) networks were first developed by Steven Grossberg and Gail Carpenter in 1987[2]. ART is of two types i.e. type-1 and type-2. ART-1 takes binary input vector, whereas, ART-2 takes analog/continuous input vector [3]. In this research, ART-1 network has been considered for automatic verification (offline) of hand-written signature. The reason for considering ART-1 in this study is that ART-1 learns faster, consumes less memory space, and can adapt newer patterns without losing the initially stored patterns [3].

Signature of a person is the most widely used technique for identity verification. However, the issue is that, every new signature of a same person may vary due to re-initialization of scribbling in each time, which might be misused or forged by others. ART is able to store all the patterns including the variations, which might not be possible with other NN-based techniques, such as ADALINE/MADALINE or Perceptron. Hence, it invites a vast scope of research.

Due to increase in computation power, a lot of fast algorithms have been developed for hand written signature verification [4]. Here, we have implemented handwritten signature verification (offline) with ART using both (a) serial and (b) parallel programming techniques. In fact, for parallel programming we have used an API package commonly known as openMP (Open Multiprocessing) [4]. We compared the two algorithms with respect to time of execution and results. Available literature shows that using NNs and based on the parameters, such as (i) height, (ii) slant, (iii) pressure of pen tip and (iv) velocity of signing, signatures could be verified with 97% accuracy for the best case [5]-[7].

Hidden Markov Model (HMM) was used for signature-verification. It was performed by simple analysis of alphabet sequences within the signature. According to this model, a signature was considered as a sequence of vectors related to each point of signature in its trajectory. In HMM matching between model and signature was performed by using the probability of how the original signature was calculated. If the probability is high, then only the signature was accepted. The use of Back propagation NN was also proposed with success for off-line signature verification [5].

Contour technique was proposed by [8], where the contour of the signature was generated by dilating the signature by various levels generating a band like structure. This dilated signature template was then compared with the test signature template. Here they used color bands for various dilation levels and an EX-OR were used for color bands to find variation in signature segments. This technique was fast and simpler to implement and gave an accuracy of 75%.

Statistical Approach using correlation was proposed by [9]. But it failed in case of skilled forgeries. As this method usually describes the characteristics of the signature related to the shape of signature.

Structural or Syntactic approach is an pattern recognition technique which represents patterns in the form of symbolic data structures viz. Trees, graph, strings etc. To verify a forge signature, its' symbolic data was compared with number of prototypes which are stored in the database. These structural features used modified direction and transition distance feature (MDF) [10] which extracts the transition locations. MDF are based on relational organization of low-level features into higher-level structures.

Wavelet-based approach was used by [11] where both static and pseudo dynamic features could be extracted from the original signature and processed by wavelet transform, which were then converted to sub-bands. This increased the difference between original and forged signatures. This method gave an average error rate of 12.57% for English signature and 13.96% for Chinese Signature.

Handwritten Signature verification (offline) is also done by using Adaptive Resonance Theory Method [12]. In this work, they have used ART-2 net for recognizing very similar looking but forged signature. In this study, the accuracy rate is almost 100%.

Associative Memory Net (AMN) approach was used for recognizing Handwritten Signature Verification [13], [14]. Here, the accuracy rate 92.3%.

Adaptive Resonance Theory (ART) method is also used for signature verification purpose. Based on 1-bit quantized pressure pattern in time domain, the work was being proposed. The timing information is used for screening for first stage screening of incoming signatures using ART-1 networks with various values of vigilance parameter [15].

Edge Detection can be another method for signature verification, where edge of the signature can be considered for training and testing [16].

II. METHODOLOGY

The study has been performed in the following steps:

Step-1: Collecting hand written signatures

Step-2: Extracting the features of all the signatures by to get all the gray-scale values

Step-3: development of ART-1 net on 'C' language with (i) serial and (ii) parallel processing

Step-4: Training ART net with the 'original' signature, and

Step-5: Testing ART net with two forged signatures and compute the error by noting the % of mismatch.

Step-1: Hand signature collection:

Figures 1, 2a, and 2b show the hand-written signatures – both original and forged, collected for this work. The size of the signatures is 324x210 dpi.

In the next stage, Similarity Index (SI) has been computed between each of the forged signatures and the original signature using equation 1. The arrangements of pixels ('on' and 'off') in the forged signatures are compared with that of the original signature row and column-wise. The disparities/dissimilarities are then noted. In equation 1, D_p is the number of 'dissimilar pixels' and T_p is the total number of pixels. The SIs computed for forged signature 1 and 2 are ~51%.

$$SI = \frac{1 - D_p}{T_p} \cdot 100 \qquad (1)$$

Step-2: Feature extraction:

We extracted the pixel values in RGB (Red-Green-Blue) format and then converted it to gray scale format by using the following standard relation.

Gray value= 0.33 × R + 0.56 × G + 0.11 × B (2)

It is important to note that we have used a text file to extract the pixel value in the binary form using the following conditional statements.

If ***gray value > 0***

Then append a '**1**' to the text file.

Else

Append a '**0**'.

A sample of pixel values are given in Appendix-A.

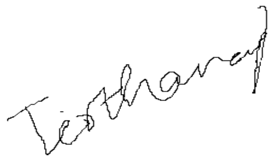

Figure 1: The original signature.

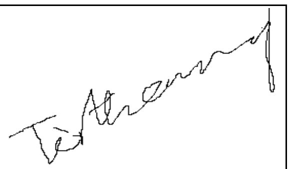

Figure 2(a): Forged signature 1.

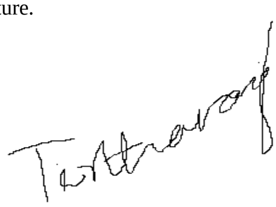

Figure-2(b): Forged signature 2.

Step-3 and 4: Development of ART-1 algorithm and its training:

As already mentioned, it has been developed on 'C' language. Figure 3 shows the schematic diagram of ART-1 structure. The implementation algorithm is as follows:

Step-0: Initialize the parameters:

$$\alpha > 1 \text{ and } 0 < \rho \le 1$$

Initialize the weights:

$$0 < b_{ij}(0) < \alpha/(\alpha-1+n) \text{ and } t_{ji}(0) = 1$$

Step-1: Perform Steps 2-13 when stopping condition is false.

Step-2: Perform Steps 3-12 for each training input.

Step-3: Set activation of all F_2 units to zero. Set the activation of $F_1(a)$ units to input vectors.

Step-4: Calculate the norm of s:

$$||s|| = \sum_i s_i \quad (3)$$

Step-5: Send input signal form F_1(a) layer to F_1(b) layer:

$$x_i = s_i \quad (4)$$

Step-6: For each F node that is not inhibited, the following rule should hold:

$$\text{If } y_j \neq -1, \text{ then } y_j = \sum_i b_{ij} x_i \quad (5)$$

Step-7: Perform Steps 8-11 when reset is true.

Step-8: Find J for $y_J \geq y_j$ for all nodes j. If $y_J = -1$, then all the nodes are inhibited and note that this pattern cannot be clustered.

Step-9: Recalculate activation of X of F_1(b):

$$x_i = s_i \ t_i \quad (6)$$

Step-10: Calculate the norm of vector x:

$$||x|| = \sum_i x_i \quad (7)$$

Step-11: Test for the reset condition.

If $||x||/||s|| < \rho$, then inhibit node J, $y_J = -1$. Go back to Step 7 again

Else if $||x||/||s|| \geq \rho$, then proceed to the next step (Step 12).

Step-12: Perform weight updation for node J (fast learning):

$$b_{iJ}(\text{new}) = \alpha x_i \ / \ (\alpha - 1 + ||x||) \quad (8)$$

$$t_{Ji}(\text{new}) = x_i \quad (9)$$

Step-13: Test for the stopping condition. The stopping conditions may be:

a. No change in weights.
b. No reset of units.
c. Maximum number of epoch reached.

The symbols used in this algorithm:

n = number of components in input training vector;
m = maximum number of cluster units that can be formed;
ρ = vigilance parameter (set between 0 and 1);
α = learning trials;
b $_{ij}$ = bottom-up weights;
t $_{ji}$ = top-down weights (Weights from Y_j of F_2 layer to X_i unit of F_1(b) layer);
s = binary input vector;
x = activation vector for F_1(b) layer;
||x|| = norm of vector x and is defined as the sum of components of x_i (i=1 to n).

It is important to mention that both the serial and parallel processing using *OpenMP* were executed in Linux environment.

System specification: It may be noted that, both 'serial' and 'parallel' processing has been executed in a PC with Intel dual core processor, 1 GB RAM, and 2 GHz.

Step-5: Testing the performance of ART-1

The forged signature is passed to the trained ART-1 network and the number of updated b_{ij} is counted. Now the b_{ij} updated during the training and the new b_{ij} after testing is compared.

The ratio of equal b_{ij} s to the total b_{ij} of training $\times 100$ gives the percentage of matching of the two signatures from this result the tested signature can be accepted or rejected.

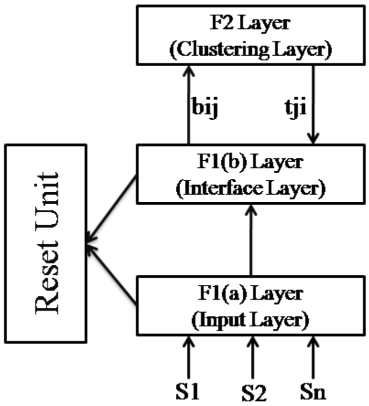

Figure-3: Structure of ART-1.

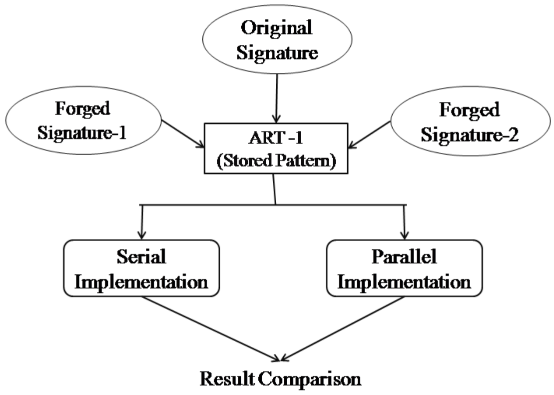

Figure-4: Training and testing of ART-1.

III. RESULTS

The average similarity index (SI) between the original and forged signatures near 60%, which may have high chance of matching, instead of rejecting the forged signatures. It is desired that even with slightest difference, the network must be able to differentiate those from the original signature based on its learning and assigned vigilance. It is certainly a real-world challenge to curb this issue. The paper suggests that vigilance parameter (ρ) needs to be optimally set, which is the first challenge. In this work, optimum ρ has been set based on the percentage of mismatch (which is the squared error) through a detail parametric study. Table-1 and 2 show the parametric study of setting ρ. The second challenge is to assure that the network learns the exemplary patterns through several observations. The third challenge is that the learning and its execution must be accomplished at minimum time. When original signature is checked with itself, it is noted that the net learns 100% exemplary patterns at average time of 13.70 sec in case of serial processing (see table 1). In case of parallel processing, 100% exemplary pattern has been learnt with much less time, i.e., 6.20 sec (see table 2).

While detecting the forged signatures, it is noted that up to 50% mismatch could be detected by serial processing with average time of 18.85 sec with ρ=0.99. In case of parallel processing such amount of mismatch could be detected with average of 7.01 sec, which is much less, than the time consumed during serial processing. Also, during the parallel processing the values of ρ widely varies yielding more flexibility to the network. In this study, mismatches are identified with ρ values 0.1, 0.49, 0.68 and 0.99 respectively.

TABLE I. RESULTS OF SERIAL PROCESSING [A = 100]

ρ ↓	Original vs. Original		Original vs. Forged1		Original vs. Forged2	
	Mismatch (%)	Time (s)	Mismatch (%)	Time (s)	Mismatch (%)	Time (s)
0.10	0.01	13.20	48.91	14.40	47.8	13.90
0.20	0.00147	14.00	48.975	13.20	49.171	13.90
0.30	0.001	13.80	49.21	13.90	49.170	13.80
0.49	0.01	*13.70*	49.21	13.90	49.17	13.60
0.68	0.0014	13.7	49.21	13.90	49.07	13.40
0.80	*0.001*	14.00	48.97	14.10	48.76	11.70
0.99	0.01	13.90	*50.007*	*18.70*	*50.07*	*19.00*

TABLE II. RESULTS OF PARALLEL PROCESSING [α = 100]

ρ ↓	Original vs. Original		Original vs. Forged1		Original vs. Forged2	
	Mismatch (%)	Time (s)	Mismatch (%)	Time (s)	Mismatch (%)	Time (s)
0.10	0.01	8.22	48.01	*6.07*	*50.9*	*4.37*
0.20	0.02	6.059	48.07	5.95	49.171	5.051
0.30	*0.001*	*6.20*	*50.01*	*4.71*	49.172	5.93
0.49	0.02	6.9	*50.9*	8.20	*50.7*	7.8
0.68	*0.02*	*6.98*	*50.09*	*8.25*	*50.70*	*7.80*
0.80	0.80	0.01	7.09	48.97	8.99	49.00
0.99	*0.01*	*6.40*	*50.007*	*6.89*	*50.07*	*8.09*

Figure-5 is a plot drawn to compare the results of serial and parallel processing. It shows that, at ρ=0.3 the parallel implementation gives better result by consuming only 5.61 sec during execution. Figure 6a and 6b shows the parametric study on different vigilance parameters (ρ) vs. mismatch (i.e. the squared errors) in case of first forged signature for the serial and parallel processing, while figures 7a and 7b plots the same for the second forged signature. These are discussed below.

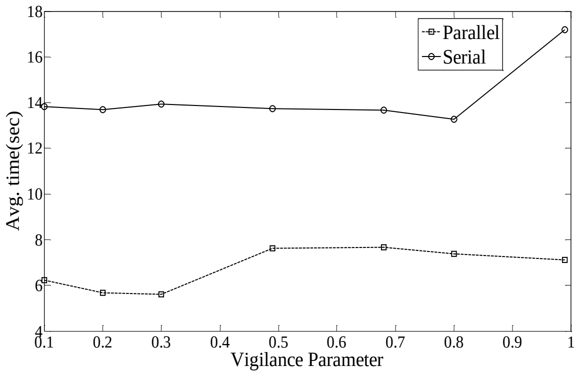

Figure-5. Plot showing the Average execution time vs. Vigilance parameter (ρ) in 'Serial' and 'Parallel' processing.

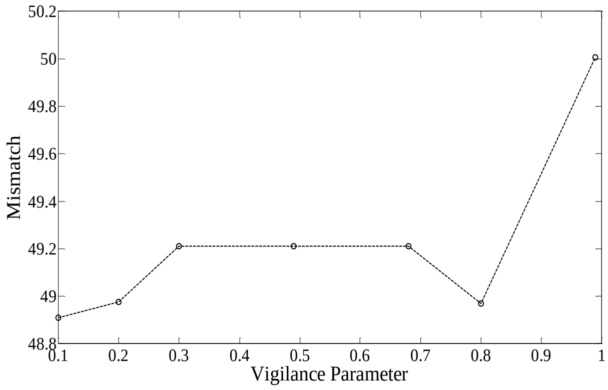

Figure-6a. Plot showing the 'Squared error' produced with several 'Vigilance parameter (ρ)' in 'Serial' processing for first forged signature.

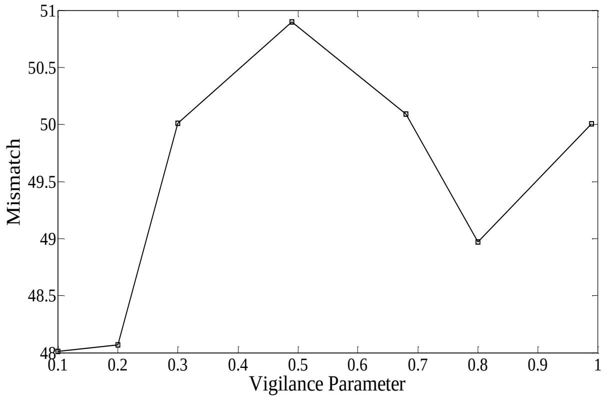

Figure-6b. Plot showing the 'Squared error' produced with several 'Vigilance parameter (ρ)' in 'Parallel' processing for first forged signature.

From figures 6a and 6b, it may be noted that for the first forged signature, in case of serial and parallel processing the maximum mismatches are found with ρ = 0.99 and 0.5, respectively. On the other hand, from figures 7a and 7b, it may be noted that for the second forged signature, in case of serial and parallel processing the maximum mismatches are found with ρ = 0.99 and 0.45 & 0.7, respectively. Therefore, ρ values between 0.45 – 0.99 might be able to catch the forged signatures mostly.

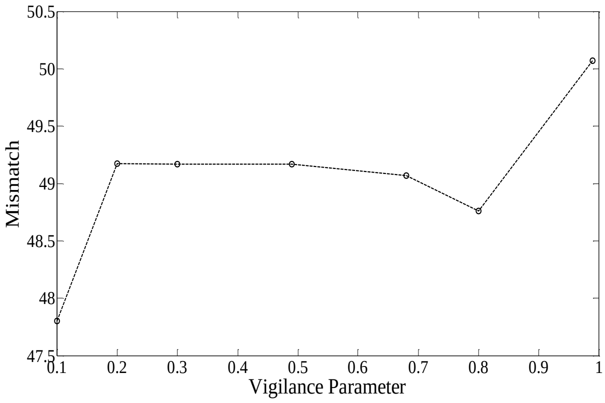

Figure-7a: Plot showing the 'Squared error' produced with several 'Vigilance parameter (ρ)' in 'Parallel' processing for second forged signature.

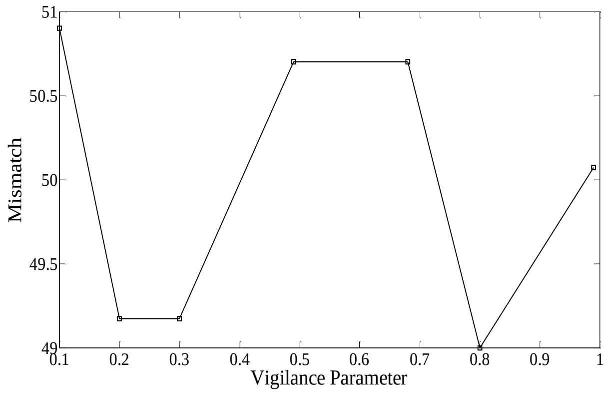

Figure-7b: Plot showing the 'Squared error' produced with several 'Vigilance parameter (ρ)' in 'Parallel' processing for second forged signature.

Finally, the performance (based on detection accuracy) has been compared with other techniques used in handwritten signature verifications (refer to Table-3). It shows that our method gives more accurate result when compared with other techniques. However, in this study we have tested only two forged signatures. This is a limitation. The net needs to be tested with many different types of forged signatures. We are at present working on this issue.

TABLE III. COMPARISON TABLE FOR DIFFERNT METHODS

Model used	Accuracy (%)
HMM[5]	88.9
BPNN [5]	83.9
Modular Neural Net(MNN) [6]	96.6
ANN [6,7]	94.27
Template Matching Technique [8]	75
Wavelet-based approach [11]	87.43; 86.04
ART-2 net [12]	*99.98*
AMN[13]	*92.3*
ART-1(in this work)	*99.99*

IV. CONCLUSIONS AND FUTURE WORK

An ART-1 type NN has been developed in this work for automating the forged signature verification task offline. It has been implemented with both serial and parallel processing. While training, the accuracy of the said net in both the processing techniques is closed to 100%. The accuracy is tested on two closely resembled but forged signatures with similarity index (SI) >60%. The study observes that with parallel processing the mismatches of both the forged signatures are detected completely with ρ equals to 0.49, 0.68, and 0.99. While comparing with other available techniques, the developed net has outperformed other techniques in terms of accuracy in predicting forged signatures. Here mismatch acts as threshold and threshold setting must be situation specific.

APPENDIX A. SIGNATURE PIXELS

Sample pixel grids of *Original* Signature:

1111111111111111111111111111111111111111111111111111111110
0111111111111111111111111111111111111111111111111111111111
1111111111111111111111111111111111111111111111111111111111
1111111111111111111111111111111111111111111111111111111111
11111111111111111111
. . .
. . .
1111111111111111111111111111111111111111111111111111111111
1111111111111111111111111111111111111111111111111111111111
1111111111111111111111111111111111111111111111111111111111
1111111111111111111111111111111111111111111111111111111111
11111111111111111111

Sample Pixel grid of *Forged* Signature 1:

11111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111.
. .
. . .
11111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111
11111111111111111111111111111110111111111111111111111
11111111111111111111111111111111111111111111111111

Sample Pixel grid of *Forged* Signature 2:

11111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111110
11111111111111111111111111111111111111111111111111111
11111111111111111111111111111110011111111111111111
. . .
. . .

111111111111111111111111111111111111111111111111
111111111111111101111111111111111111111111111111
111111111111111111111111111111111111111111111111
11111111111111111111111111

REFERENCES

[1] K. Krishna, A. Goyal, and S. Chattopadhyay, "Non-correlated character recognition using Hopfield network: a study", in *Proc. International Conference on Computer and Computational Intelligence (ICCCI 2011)* Bangkok, Thailand, 2011, pp. 385-389.

[2] G. A. Carpenter and S. Grossberg, "A self organizing neural network for supervised learning, recognition, and prediction," *IEEE Communication Magazine*, vol. 30, issue 9, pp. 38-49, 1992.

[3] S. N. Sivanandan and S. N. Deepa, *Principle of Soft Computing*, 2nd ed., WILEY-INDIA, 2011. ISBN-978-81-265-2741-0, 2011.

[4] R. Chandra, L. Dagum, D. Kohr, Maydan, J. McDonald, and R. Menon, *Parallel Programming in OpenMP*, Morgan Kaufmann Publisher. ISBN 1-55860-671-8, 2001.

[5] A. K. M. A. Hossain, Md. G. Rahman, and R. C. Debnath, "Comparative study of signature verification sustem using Supervised (BPNN) and Unsupervised (HMM) Models," in *proc: 6th International Conference on Computer and Information Technology(ICCIT)* Dhaka, Bangladesh, 2003.

[6] O. Mirzaei, H. Irani, and H. R. Pourreza, "Offline signature recognition using modular neural networks with fuzzy response integration" 2011 in *Proc. International Conference on Network and Electronics Engineering IPCSIT* vol. 11, pp. 53-59, Singapore, 2011.

[7] A. McCabe, J. Trevathan, and W. Read, "Neural network-based handwritten signature verification", *Journal of Computers*, vol.3, no. 8, pp. 9-22, 2008.

[8] S. Inglis and I. H. Witten, "Compression-based template matching", in *Proc. IEEE Data Compression Conference*, pp. 106-115, Los Alamitos, CA, 1994.

[9] I. W. McKeague, "A statistical Model of signature verification," pp. 1-25, 2004 (online: http://www.columbia.edu/~im2131/ps/sig-rev1.pdf).

[10] M. Blumenstein, X.Y. Liu, and B. Verma, “A modified direction feature for cursive character recognition,” in *Proc. International Joint Conference on Neural Networks*, pp. 2983-2987, 2004.

[11] W. Tian, Y. Qiao, and Z. Ma, “A new scheme for off-line signature verification using DWT and Fuzzy Net”, in *Proc. 8th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing*, vol. 3., pp. 30-35, 2007.

[12] T. Dash, S. Chattopadhyay, and T. Nayak, “Handwritten signature verification using Adaptive Resonance Theory Type-2 (ART-2) Net”. *Journal of Global Research in Computer Science*, vol. 3 issue 8, pp. 21-25, 2012.

[13] T. Dash, T .Nayak, and S. Chattopadhyay, “Offline handwritten signature verification using Associative Memory Net”. *International Journal Advanced Research in Computer Engineering & Technology*, vol.1 issue 4, pp. 370-374, 2012.

[14] T. Dash, T. Nayak, and S. Chattopadhyay, “Handwritten signature verification (offline) using neural network approaches: A comparative study”. *International Journal of Computer Applications*, vol.57 no.7, pp.33-41, 2012.

[15] L. Y. Tseng and T. H. Huang, “An online Chinese signature verification scheme based on the ART1 neural network,” in *Proc. Int. Conf. Neural Networks,* vol.3, pp.624-630, 1992.

[16] T. Dash and T. Nayak, “A java based tool for basic image processing and edge detection,” *Journal of Global Research in Computer Science*, vol.3 issue 5, pp.57-60, 2012.

Tirtharaj Dash is pursuing his Master of Technology in Department of Computer Sc. & Engg at Veer Surendra Sai University of Technology, Burla. He has published more than 20 research articles in journals of international repute and conferences. His research area includes Parallel Computing, Algorithms, Soft Computing and AI.

Tanistha Nayak has done her Bachelor of Technology in Information Technology at National Institute of Science and Technology, Berhampur. She has contributed 18 papers Soft Computing, Quantum Computing and Image processing researches.

Dr. Subhagata Chattopadhyay received his PhD in Information Technology from Indian Institute of Technology (IIT), Kharagpur and postdoctoral fellowship in e-Health information system at Australian School of Business, the University of New South Wales (UNSW) Sydney Australia. He has published 98 research papers in the field of Soft Computing, Pattern Recognition, AI, Image Processing, Knowledge Engineering and Management. Recently he has published one edited book named "ADVANCES IN THERAPEUTIC ENGINEERING", by CRC Press USA. Dr. Chattopadhyay is presently working as a full professor in the Dept. of Computer Science and Engineering at Camellia Institute of Engineering, India. He is also the Director of the said institute.