Adaptive Metaheuristic Deep Learning for Study towards Human Cognition using Complex Handwriting Pattern

Chandranath Adak[1], *Student Member, IEEE*
Sponsoring Supervisor: Simone Marinai[2]
Supervisor: Michael Blumenstein[3], *Senior Member, IEEE,* and Bidyut B. Chaudhuri[4], *Life Fellow, IEEE*

***Abstract* —This paper deals with study towards complex handwriting patterns and their understanding through deep neural networks. We also handle with hand-crafted features and merged it with auto-extracted deep features. A metaheuristic deep learner is also attempted here. All the techniques are tested on complex Bengali handwritten pattern and the system adaptability is also verified by various writers. We have obtained fairly good results in terms of writer verification task.**

Keywords—CNN, Handwriting; Metaheuristic deep learning; PSO; RNN.

1. INTRODUCTION

"Handwriting" is basically a kind of pattern. However, from the pre-historic era, it bears the connotation of human civilization. The handwriting instrument progressed from finger and wedge (on clay/sand and stone-based medium) to quill, pencil, fountain/ball-point pen, and again finger (on the touch-screen of a smart device). Though the world is going fast towards paperless e-world, "handwriting remains just as vital to the enduring saga of civilization (− Michael R. Sull)" [1].

For computer scientists, *automated analysis of handwriting* is a recognized field-of-study owing to the ever-increasing complexity of extreme variation of writing. In data mining and image processing task, the handwritten digit/character classification [2] has become a benchmark to test any classifier performance. Also, promising results with fair accuracy have been obtained in recognizing free-form running texts in Roman-based western as well as some oriental scripts [3, 4]. However, handwriting analysis is challenging in multi-script (e.g. *Alphabetic*: English, Spanish etc., *Abugida*: Devanagari, Bengali etc., *Logographic*: Chinese, Hieroglyph etc., *Abjad*: Arabic, Hebrew etc.) environment [3-4, 39].

In data/information retrieval task and online archiving of handwritten documents having historical importance [6-8], major contribution is awaited. Though handwritten keyword spotting [5] takes a vital attention, the content and context analysis directly from handwritten document image is not attained by researchers. Such content/context analysis can play a major role in document classification and can assist in e-archiving. Writer identification [9] is also a standard problem in handwriting analysis. This problem can be perceived as a multi-class classification task to classify (identify) multiple writers.

"Adaptability" in the context of handwriting analysis can be defined as the ability to adjust/recognize with the change in writing-strokes/patterns acquired in terms of individual, space, time, writing medium etc. A particular individual can write various forms of a certain character, known as allograph. Handwriting also varies at a different time with cursiveness, free-form, conscious writings. Therefore, an automated system should be nicely coped up with the new dataset and being learned continuously with time. The aspect of adaptiveness takes care such online learning over time.

The "Cognition" refers "the mental action or process of acquiring knowledge and understanding through thought, experience, and the senses" [11]. Human "vision begins with the eyes, but actually takes place in the brain" [12]. In relation to our research work, after seeing the handwritten pattern image, the human brain signals its connotation. Actually, the human brain matches the writing pattern (through human vision system) with previously known/ learned patterns. Sometimes, in handwritten pattern matching or recall from memory becomes very difficult. A human brain grows up (from child to adult) with training. After being taught of vocabulary, writing patterns, multi-script knowledge etc., a brain is linguistically developed [13].

In this era of Artificial Intelligence (AI), one of the main goals of computer scientists is to build the artificial brain. However, before that is possible to be fully functional, it is essential to train the (artificial) brain. Our present work is to teach computers (machines) through handwriting - from simple to complex.

Now-a-days the "deep learning architecture" [19] is used as a strong approach and also an up-and-coming promising machine learning tool. Automated handwriting analysis domain [14-16] is also taking some advantages of this architecture. Still, optical handwritten character recognition engine requires more sophistication. In this present research, taking assistance from deep learning architecture, we focus on to know how human cognition takes place to understand the handwritten patterns. It is perceived to help us to train the machine for automated analysis of complex handwriting. On the other hand, it is

[1] C. Adak is from IIIS, School of ICT, Griffith University, Gold Coast Campus, Australia-4222 (e-mail: adak32@gmail.com). Major portion of this work carried out when C. Adak visited DINFO, University of Florence, Italy-50139.
[2] S. Marinai is from DINFO, University of Florence, Italy-50139.
[3] M. Blumenstein is from School of Software, University of Technology Sydney, Australia-2007.
[4] B.B. Chaudhuri is from CVPR Unit, Indian Statistical Institute, India-700108.

also interesting to verify a writer from his/her complex handwritten pattern.

In this paper, we merge handcrafted feature with the auto-extracted feature to experiment the impact of such hybrid learning technique. We have also approach towards a metaheuristic learning technique. All the learning techniques are tested on handwriting pattern towards tackling the problem of writer verification.

The rest of the paper is as follows. Sec. 2 deals with the proposed method and Sec. 3 discusses the experimental results. Sec. 4 finally concludes the paper.

2. PROPOSED METHOD

We first extract two types of handcrafted features: textural and allographic for handwriting analysis. Then these handcrafted features are merged with CNN deep features. The writer verification task through complex handwriting pattern is dealt with CNN merged with MLP and Siamese net separately (refer Sec. 2.1). We also handle writer verification task with auto-derived features and RNN in Sec. 2.2. A PSO-based metaheuristic neural net learner is also attempted to deal with complex handwriting along with writer verification in Sec. 2.3.

2.1. Writer verification with handcrafted features:

Here we deal with writer verification using handcrafted features merged with CNN deep features.

2.1.1. Feature Extraction:

Here we discuss the employed hand-crafted features.

2.1.1.1. Textural feature:

In [36], for the contour hinge feature f_{ch}, two contour fragments, joined to a common end, making angles φ1 and φ2 (φ2 ≥ φ1), spanning to all four quadrants (360°), are considered. A normalized histogram is generated with a joint probability distribution p_f(φ1, φ2). Here, the number of histogram bins (n_b) has been set to 12, leading to $n_b(2n_b +1)$ = 300-dimensional feature vector. However, a total number of combinations of φ1 and φ2 is $4n_b^2$ = 576.

2.1.1.2. Allographic Feature:

Allographic features are extracted from various forms of characters. It is assumed that a writer acts as a stochastic generator of grapheme/ fragment (small broken-off part) shapes. The probability distribution of such grapheme shapes may be used as the feature.

The handwritten patterns are segmented into graphemes/ fragments by segmenting the ink-trace at the minima of the lower contour vertically thorough the ink-stroke-width of the upper contour. The grapheme codebook generation is standardized using Kohonen Self-Organizing Map (KSOM)-2D [29]. We use 25X25 KSOM-2D organization for an ordered map. For this purpose, some alternatives are k-means and KSOM-1D. We choose neither the k-means due to its disordered organization nor the KSOM-1D owing to linear unidirectional order.

Now, we calculate the feature as distribution function of grapheme. A histogram of 625 (25X25) bins is generated considering every grapheme in one bin. A sample grapheme is matched with the nearest codebook prototype and put in a histogram bin, similar to [36]. From the histogram, we obtain a 625-dimensional normalized feature vector.

2.1.2. Writer verification:

Writer verification can be perceived as a binary classification problem, where the task is to classify a handwritten specimen written by a writer or not. Therefore, it can be perceived as binary classification problem having classes “yes” or “no”.

Here the handcrafted features are merged with deep CNN features.

2.1.2.1. CNN-MLP:

Here we use the convolutional neural network (CNN), whose afterward portion is actually multi-layer perceptron (MLP).

a) Textural CNN-MLP: We consider a square textural hinge map of 24X24. Although, one side of the main diagonal of this map is flat owing to φ2 ≥ φ1, it works well for being input to the CNN.

This CNN architecture contains two convolution layers (C1 and C2), each followed by a sub-sampling layer (S1 and S2, respectively).

The C1 convolutional layer has 16 feature maps of size 24X24. Each feature map is connected to a 3X3 neighborhood in the input (filter size N_F =3X3).

The S1 subsampling (max-pooling) layer contains 16 feature maps of size 12X12, each connected to a 2X2 (N_K) neighborhood in the corresponding feature map in C1.

The C2 convolution layer consists of 32 feature maps of size 12X12. Here, N_F =3X3.

The S2 sub-sampling layer contains 32 feature maps of size 6X6. Here, N_K =2X2.

Here, ReLU (*Rectified Linear Unit*) is used as activation function to the outcome of C1 and C2.

The afterward portion of CNN is an MLP with 1 hidden layer (H1).

The output of the S2 layer (32@6X6) is fully connected with the following layer F1, which is the input layer of the adjunct MLP. The input of F1 has 1152 nodes (1152@1X1). The following hidden layer H1 contains 512 nodes.

The next layer is the output layer (OP1) and it contains 3 output nodes. One extra node is remained for handling void or null (ϵ).

b) Allographic CNN-MLP: Here, we use the abovementioned allographic feature map of size 25X25 to be fed to this CNN.

The allographic CNN-MLP is quite similar to the above textural CNN-MLP, having 2 convolution layers (C1 and C2) each followed by 2 sub-sampling layers (S1 and S2).

Here also C1 convolutional layer has 16 feature maps of size 24X24 with N_F=3X3. Likewise, the S1 sub-sampling (max-pooling) layer contains 16 feature maps of size 12X12. Here N_K= 2X2.

The C2 and the S2 layer are little-bit different. Here, the C2 layer has 32 feature maps each of size 8X8. Here, N_F=3X3. The S2 subsampling layer is of 32 feature maps each sized 8X4. Here, N_K=1X2.

Here also, we have used ReLU to the outcome of C1 and C2.

The output of S2 is fed into an MLP with 1 hidden layer. The input layer (F1) of MLP (F1) is fully connected with S2 (32@8X4). The F1 has 1024 nodes (1024@1X1). The following hidden layer contains 512 nodes. The output layer (OP1) contains 3 nodes having one extra node as void (ϵ).

2.1.2.2. CNN-Siamese:

Siamese neural network is a well-known architecture for ranking the similarity between two inputs [37, 38]. Here we use Siamese net for writer verification to know whether a handwritten sample is written by a particular writer or not.

We embed the above textural CNN and allographic CNN to the Siamese net.

a) *Textural CNN-Siamese*: Here the previous textural CNN is used without the MLP portion. So, the fully connected layer F1 having 1152 nodes is inputted to the Siamese. Siamese twin CNN comprises the same architecture. The similarity metric involves the contrastive loss [30] function.

b) *Allographic CNN-Siamese*: Here also, we employ the previous allographic CNN without the MLP part. Therefore, the fully connected layer F1 with 1024 nodes is fed to the Siamese neural network. We use this same CNN for the Siamese twin. The contrastive loss function [30] is used as a similarity metric.

2.2. Writer verification with auto-extracted features:

We also perform the writer verification task using auto-extracted features followed by RNN classification.

2.2.1. Auto-derived CNN feature:

For auto-derived feature extraction, we use a patch-based selection [31] to feed into the CNN. The input patch size is chosen as 64X64. This CNN architecture comprises of 4 convolutional layers (C1, C2, C3, and C4), each followed by a subsampling layer (S1, S2, S3, and S4).

The C1, C2, C3, C4 layers contain 8, 16, 32, 32 feature maps of size 64X64, 64X32, 32X16, 16X8 respectively. For each convolution layers, N_F=5X5.

The S1, S2, S3, S4 sub-sampling (max-pooling) layers contain 8, 16, 32, 32 feature maps of size 64X32, 32X16, 16X8, 8X4 respectively. For S1, S2, S3, S4, the N_Ks are 1X2, 2X2, 2X2, 2X2, respectively.

A full connection layer (FC1) is followed by S4 layer and we have obtained a 1024 dimensional auto-extracted CNN feature vector.

2.2.2. RNN classification:

The 1024 dimensional auto-derived feature is fed into a bi-directional Recurrent Neural Network (RNN) [32] for writer verification task. The number of nodes in the RNN input layer is the dimension of the feature vector. The output layer contains 3 nodes. One ϵ node at output layer is extra and remains as null. We use two distinct hidden layers for forward and backward sequences separately. Here, LSTM (Long Short-Term Memory) [33] blocks are used as hidden units. These two hidden layers contain 256 and 128 LSTM memory cells, respectively.

2.3. Writer Verification with Metaheuristic learner:

Here we deal with writer verification by feed-forward metaheuristic neural network.

2.3.1. Feature Extraction:

Here we use the same textural feature “hinge” of size 300 (refer Sec. 2.1.1.1).

2.3.2. Classification using FNN with PSO:

Classification with the feed-forward neural network is quite traditional technique. However, our main intention is to work with a metaheuristic learner [35]. Hence, in this current task, we start with the feed-forward neural network (FNN) to get the essence of metaheuristic optimization. Meta-heuristic algorithms have proved their efficiency in optimizing neural nets alongside neural net optimization techniques such as backpropagation [35].

Optimization of a neural network comprises of many tasks such as node optimization, layer optimization, learning rule optimization, connection weight optimization, and sometimes overall architecture optimization. Here we deal with only the simultaneous optimization of the connection weights and transfer parameters.

Particle Swarm Optimization (PSO) is a popular technique among population-based state-of-the-art meta-heuristic algorithms. It imitates the searching behavior of swarms, and depends on the velocity and position of a swarm. The velocity is updated in order to update the position of the particles in a swarm. So, the whole population moves towards an optimal solution. More details on PSO can be found in [34].

Here, we have employed the non-linear transfer function tan-hyperbolic adaptive.

3. Experimental Result and Discussion

3.1 Employed Database:

For experimental analysis, we have used an in-house dataset of 100 writers. Each writer wrote 3 pages of Bengali writing. We perform the experiment on Bengali script, since it has more about 300 complex character patterns. For writing consistency, the writers were provided pen and paper. Otherwise, the handwriting samples were collected in an uncontrolled fashioned.

3.2. Experimental Results:

We have shown the writer verification performance measure in terms of EER (*Equal Error Rate*). The results are shown in the following TABLE I. We have obtained the best performance on textural CNN-Siamese.

TABLE I. WRITER VERIFICATION PERFORMANCE

Method	EER (%)
Textural CNN-MLP	3.6
Allographic CNN-MLP	4.1
Textural CNN-Siamese	2.4
Allographic CNN-Siamese	3.3
CNN-RNN	2.9
PSO-FNN	4.7

The state-of-the-art method [36] using hinge feature with χ^2 distance produced EER of 5.2 while testing on our database. In comparison with this state-of-the-art method [36], our PSO-FNN method performed well. Overall the textural feature worked better. The performance of CNN and RNN hybridization is also impactful.

4. CONCLUSION

In order to access the effectiveness of the conceived learning mechanism, we have considered the handwriting trait to assess its applicability in digital document analysis domain. The handcrafted feature embedding in CNN is new here to the best of our knowledge. The hybridization of CNN and RNN is relatively new in literature for such task. The metaheuristic neural net learner is also relatively very new and required to be investigated more. Although we have obtained fairly well outcome using PSO on FNN, in future we would like to investigate on different neural architecture.

This research has high potential to produce adaptive metaheuristic deep learning model (with aid from human cognition), which can be applied in automated handwriting analysis. We will also explore the promising adaptive techniques in handwriting research and build a reliable everlasting learning artificial cognitive model. On the other hand, automated handwriting analysis has positive impacts on the fields of Forensics, Biometrics, Library and Data Science. So, this research will discover encouraging practical contribution to these fields.

Acknowledgement

This research has been partially supported by the IEEE Computational Intelligence Society Graduate Student Research Grants 2017. Mr. C. Adak is also thankful to the *AI Team*, DINFO, University of Florence, Italy regarding various academic discussions and support.

REFERENCES

[1] C. Adak, B.B. Chaudhuri, M. Blumenstein, “Writer Identification and Verification from Intra-variable Individual Handwriting”, CoRR abs/1708.03361 (2017).

[2] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, “Gradient-Based Learning Applied to Document Recognition”, Proceedings of the IEEE, vol. 86, no. 11, pp. 2278-2324, November 1998.

[3] R. Plamondon, S. Srihari, “On-line and off-line handwriting recognition: a comprehensive survey”, IEEE Trans. on PAMI, vol.22, no.1, pp. 63-84, 2000.

[4] U. Pal, R. Jayadevan, N. Sharma, “Handwriting recognition in Indian regional scripts: a survey of offline techniques”, ACM Trans. on Asian Language Information Proc., vol.11, no.1, pp. 1-35, 2012.

[5] I. Pratikakis, K. Zagoris, B. Gatos, G. Louloudis, N. Stamatopoulos, “ICFHR 2014 Competition on Handwritten Keyword Spotting (H-KWS 2014)”, Proc. Int. Conf. on Frontiers in Handwriting Recognition (ICFHR), pp. 814-819, 2014.

[6] Bibliotheque de Rouen (Rouen Library), Rouen Cedex-76043, France. Web: http://www.bovary.fr/ .

[7] The Morgan Library & Museum, New York, USA-10016. Web: http://www.themorgan.org/ .

[8] Queensland State Archive, Australia-4113. Online available at: http://www.archivessearch.qld.gov.au .

[9] L. Schomaker, “Advances in Writer Identification and Verification”, Proc. Int. Conf. on Document Analysis and Recognition (ICDAR), vol.2, pp. 1268-1273, 2007.

[10] M. P. Caligiuri, L. A. Mohammed, “The Neuroscience of Handwriting: Applications for Forensic Document Examination”, ISBN: 978-1-4398-7140-9, CRC Press, 2012.

[11] Oxford Dictionary, Oxford University Press, Online available at: http://www.oxforddictionaries.com/definition/english/cognition .

[12] F.-F. Li, “How we're teaching computers to understand pictures”, TED2015, Online: https://www.ted.com .

[13] G. Bounds, “How Handwriting Trains the Brain”, The Wall Street Journal, Oct.5, 2010.

[14] A. Graves, M. Liwicki, S. Fernandez, R. Bertolami, H. Bunke, J. Schmidhuber, “A Novel Connectionist System for Unconstrained Handwriting Recognition”, IEEE Trans. on PAMI, vol.31, no. 5, pp.855-868, 2009.

[15] G. R. Ball, S. N. Srihari, “Semi-supervised Learning for Handwriting Recognition”, Proc. Int. Conf. on Document Analysis and Recognition, pp. 26-30, 2009.

[16] H. Lee, B. Verma, M. Li, A. Rahman, “Machine Learning Techniques in Handwriting Recognition: Problems and Solutions”, in Machine Learning Algo. for Prob. Solv. in Comp. Applications: Intell. Tech., IGI Global, Chap.2, pp.12-29, 2012.

[17] S. Marinai, M. Gori, G. Soda, "Artificial Neural Networks for Document Analysis and Recognition", IEEE Trans. on PAMI, vol.27, no.1, pp. 23-35, 2005.

[18] A.L. Samuel, "Some Studies in Machine Learning Using the Game of Checkers", IBM Journal of Research and Development, vol.3, no.3, pp. 210-229, 1959.

[19] J. Schmidhuber, "Deep Learning in Neural Networks: An Overview", Neural Networks, vol.61, pp.85-117, 2015.

[20] Y. Bengio, A. Courville, P. Vincent, "Representation Learning: A Review and New Perspectives", IEEE Trans. on PAMI, vol.35, no.8, pp. 1798-1828, 2013.

[21] C. Adak, B. B. Chaudhuri, "An Approach of Strike-through Text Identification from Handwritten Documents", Proc. Int. Conf. on Frontiers in Handwriting Recognition (ICFHR), pp. 643-648, 2014.

[22] L.M.R. Rere, M.I. Fanany, A.M. Arymurthy, "Simulated Annealing Algorithm for Deep Learning", Procedia Computer Science, vol.72, pp. 137-144, 2015.

[23] J. Kennedy, R. Eberhart, "Particle Swarm Optimization", Proc. Int. Conf. on Neural Networks, vol.4, pp. 1942-1948, 1995.

[24] M. Liang, X. Hu, "Recurrent convolutional neural network for object recognition", IEEE Conf. on CVPR, pp. 3367-3375, 2015.

[25] R. Salakhutdinov, A. Mnih, G. Hinton, "Restricted Boltzmann Machines for Collaborative Filtering". Proc. 24th Int. Conf. on Machine Learning (ICML), pp. 791-798, 2007.

[26] G. Alain, Y. Bengio, "What Regularized Auto-Encoders Learn from the Data-Generating Distribution", The Journal of Machine Learning Research, vol. 15, no. 1, pp. 3563-3593, 2014.

[27] R. Salakhutdinov, G. Hinton, "Deep Boltzmann Machines", Proc. Int. Conf. on Artificial Intelligence and Statistics, JMLR Workshop and Conference Proc., vol. 5, pp. 448-455, 2009.

[28] G.E. Hinton, S. Osindero, Y.-W. Teh, "A Fast Learning Algorithm for Deep Belief Nets", Neural Computation, vol. 18, no. 7, pp. 1527-1554, 2006.

[29] T. Kohonen, "Self-Organization and Associative Memory", second ed. Springer Verlag, 1988.

[30] S. Chopra, R. Hadsell, Y. LeCun, "Learning a similarity metric discriminatively with application to face verification", CVPR, 2005, pp. 539–546.

[31] C. Adak, B. B. Chaudhuri, M. Blumenstein, "Legibility and Aesthetic Analysis of Handwriting", 14th Int. Conf. on Document Analysis and Recognition (ICDAR), pp. 175-182, 2017.

[32] M. Schuster, K.K. Paliwal, "Bidirectional Recurrent Neural Networks", IEEE Transactions on Signal Processing, vol.45, no.11, pp.2673-2681, 1997.

[33] S. Hochreiter, J. Schmidhuber, "Long Short-Term Memory", Neural Computation, vol.9, no.8, pp.1735-1780, 1997.

[34] J. Kennedy, R. Eberhart, "Particle swarm optimization," Proc. IEEE Int. Conf. on Neural Networks, 1995, pp. 1942-1948, vol.4.

[35] X. Yao, "Evolving artificial neural networks," Proceedings of the IEEE, vol. 87, no. 9, pp. 1423–1447, 1999.

[36] M. Bulacu, L. Schomaker, "Text-Independent Writer Identification and Verification Using Textural and Allographic Features," IEEE TPAMI, vol. 29, no. 4, pp. 701-717, 2007.

[37] J. Bromley et al., "Signature Verification using A Siamese Time Delay Neural Network", Int. J. Patt. Recognition and Artificial Intelligence. 07, 669 (1993).

[38] G. Koch, "Siamese Neural Networks for One-Shot Image Recognition", M.S. Thesis, University of Toronto, 2015.

[39] D. Ghosh, T. Dube, A. Shivaprasad, "Script Recognition - A Review", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 32, no. 12, pp. 2142-2161, Dec. 2010.

[40] V.K. Ojha, A. Abraham, V. Snášel, "Metaheuristic design of feedforward neural networks: A review of two decades of research", In Engineering Applications of Artificial Intelligence, vol. 60, pp. 97-116, 2017.