Extraction of Sequence from Bangla Handwritten Numerals and Recognition Using LSTM

Shadmaan Hye, Md. Rahat-uz-Zaman, M. A. H. Akhand
Dept. of Computer Sceince and Enginnering
Khulna University of Engineering & Technology
Khulna, Bangladesh
Email: praptishadmaan@gmail.com, rahatzamancse@gmail.com, akhand@cse.kuet.ac.bd

Abstract—**In the promising era of Handwritten Numeral Recognition (HNR), despite Bangla being one of the major languages in the Indian subcontinent, fewer explorations have been done on Bangla numerals compared to other languages. Among the existing methods, several convolutional neural network (CNN) based method outperformed other methods. But CNN always gets confused with some specific Bangla numerals due to the similarity of shape and size of different numerals. The main purpose of this study is to expand Bangla HNR by considering a novel methodology with a Long Short-Term Memory (LSTM) network. In the proposed method, images are thinned and a sequence is extracted. These extracted sequences are used to classify using LSTM network. Both single-layer LSTM and Deep LSTM models are trained and performance tested on a benchmark dataset with a large number of samples. On the other hand, traditional CNN is also trained for better understanding. Experimental outcomes revealed that the proposed LSTM based method outperformed CNN with remarkable accuracy for the similar shaped numerals. Finally, the proposed method achieved a test set recognition rate of 98.03% which is better than or competitive to other prominent existing methods.**

Keywords—Handwritten Numeral Recognition, Convolutional Neural Network, Long Short-Term Memory, Sequence Extraction

I. INTRODUCTION

Handwritten Numeral Recognition (HNR) has attracted a great deal of consideration in performing various tasks because of its application in wide areas in recent times. Although Bangla achieved 5th position in the native speakers' language list [1], the groundwork on Bangla HNR (BHNR) is less in comparison to other languages such as roman. A few research works have been done on HNR of Bangla and other Indian scripts [2]–[7]. Most of these research works are based on image processing and convolutional neural networks (CNN).

CNN is proven to perform better for image-based classification [8]. But HNR, CNN always gets confused with some specific Bangla numerals due to the similarity of shape and size of different numerals. Although the shape and size remain the same, the strokes of the numerals are different from each other. The strokes can be fed into the Long Short-Term Memory (LSTM) model for better accuracy in some cases. For example, the classification of Bangla script ১ and ৯ by CNN classifier is very poor due to their similar shape. LSTM classifiers work by observing the sequence of these scripts. The digit ৯ is more cursive than digit ১ which will be recognized by LSTM classifiers.

Santosh and Iwata [9] built a stroke-based character recognition system focusing on Devanagari handwritten characters. This can distinguish the characters of any online stroke and order. But the strokes cannot be extracted if a pre-written image is provided. Ahmed et al. [10] proposed a Deep LSTM method with two LSTM layers to distinguish singular Bengali numerals. But the methodology used has padded to input in LSTM layer which adds unnecessary information. If these unnecessary information by padding are not added, the result may become much better.

The goal of this study is to construct a BHNR system with stroke extraction and LSTM classifier. The methodology is primarily focused on the extraction of sequence of the numerals and feed it into the LSTM model since LSTM works well with sequences of data compared to other methods.

The rest of this paper is organized in three sections. Section II explains the proposed methodology. Section III presents experimental outcomes on a benchmark dataset. Finally, the paper is concluded with future works of this study in Section IV.

II. WRITING SEQUENCE EXTRACTION OF HANDWRITTEN NUMERALS AND CLASSIFICATION WITH LSTM

The first stage of the proposed HNR system is feature extraction by generating a sequence from the numeral image. The second stage is to use these extracted features to classify the image with a good classifier. LSTM recurrent neural network is used as the classifier. The whole system is shown in Fig. 1. For sequence extraction, the preprocessed binary image needs to be thinned to get a skeleton, then a modified algorithm to solve the traveling salesman problem (TSP) is applied. These two parts are discussed following.

A. Dataset Preparation

The CVPR, ISI dataset [11] [12] is used in this study which contains a total number of training and test samples are 19392 and 4000 written by 1106 individuals. For the preprocessing of this dataset OTSU thresholding [13] is applied to the image to make it binary image. The image is then cropped to the size of numeric height and width. This is done by cropping the portion of the image where the values of the rows or columns are 0. Since each of the images is variable in size, the image is further

978-1-7281-7366-5/20/$31.00 ©2020 IEEE

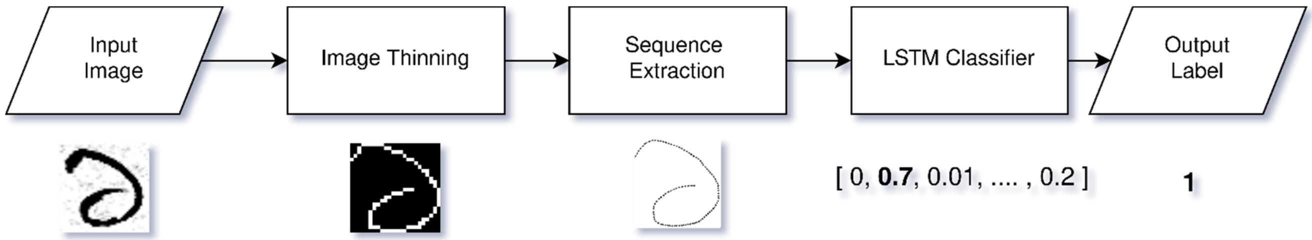

Fig. 1. Complete system diagram of proposed HNR system.

resized into a fixed dimension. 1-pixel padding is added to the image as the selected thinning algorithm only works if the image does not have ones in boundary. All the preprocessing steps are demonstrated in Fig.2.

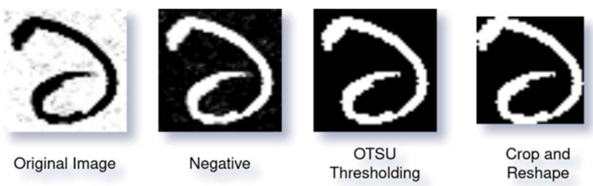

Fig. 2. Preprocessing steps applied on a single sample. image.

B. Image Thinning

Zhang Suen Thinning (ZST) algorithm is used for thinning. At first, only the largest connected non-zero pixels are taken and all other pixels are converted to zeros with morphological operations. ZST algorithm [14] is then applied to the preprocessed image. The algorithm only works with binary images. This is the reason why the preprocessing must return a binary image. ZST algorithm works in such a way that, it turns a pixel into black if it is a pixel of white boundary in an iterative process until it reaches a termination condition. The outcome of ZST algorithm is the thinned image having a stroke of a single-width pixel digit skeleton. The output of the thinning algorithm on a sample image is demonstrated in Fig.3.

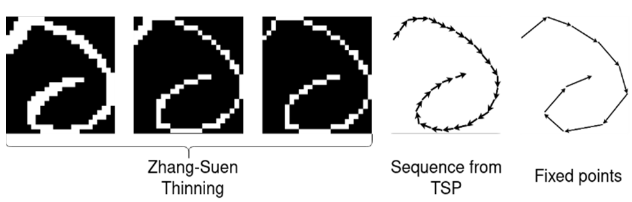

Fig. 3. Output of each step of sequence extraction process.

C. Sequence Extraction

The thinned image having a one-pixel stroke width in all points is processed for sequence extraction. Each pixel with high value in this image is abstracted as a node in a graph. These nodes are connected to their four neighboring nodes first. If it is not possible, then if possible, they are connected to their eight neighboring high pixels. One notable thing is, all nodes will preserve its original pixels location which will be used later in the process. This process will create a connected graph.

The generated graph is finally reduced into a directed acyclic graph by applying a solution algorithm of TSP [15]. Any fast TSP solving algorithm can be used to create this acyclic graph. But the last and first node of the travel path must not be connected which is the modification of the solution algorithm. The node with one degree whose stored pixel location is the top-left one will be selected as the starting point.

Depending on the image, the total number of points in the sequence extracted can vary a lot. But a static neural network architecture such as LSTM must have a fixed input size. For this reason, spline curve fitting interpolation [16] method is used to interpolate and create a fixed number of equidistant points. Finally, these points are fed into the LSTM.

D. Classification with LSTM Network

Any sequence classifier can be used for the identification of the above-extracted sequences. Here LSTM network and deep LSTM neural network are used. LSTM is a Recurrent neural Network variant which takes a sequence as input and outputs probabilities of targets. The architecture of the two models are provided in Fig. 4 and Fig. 5. AdaDelta optimizer [17] is used to minimize the categorical cross-entropy loss function.

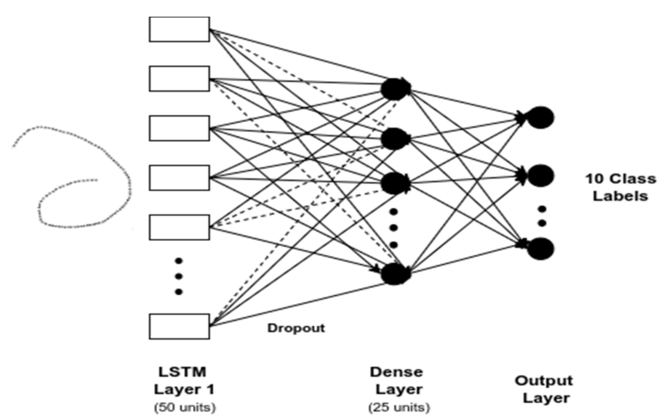

Fig. 4. Single layer LSTM neural network Architecture.

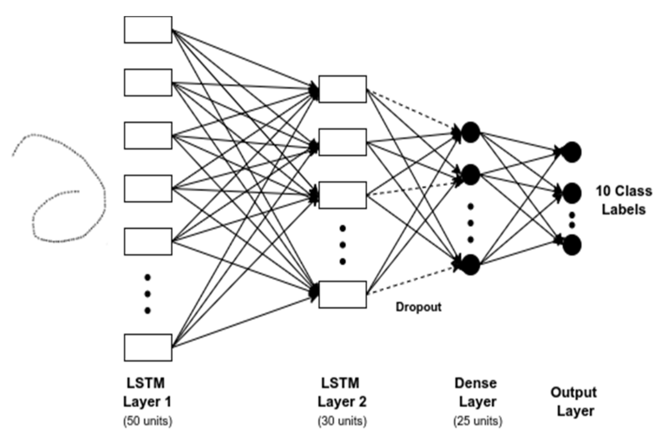

Fig. 5. Deep LSTM neural network Architecture.

III. Experimental Studies

The training of LSTM network performed on Kaggle using GPU support. The test set accuracy per epoch is shown in Fig. 6 and Fig. 7 for single layer LSTM and deep LSTM classifiers, respectively. It can be seen that the accuracy is highest on the input sequence size of 20. Apparently, more input size should outcome better results, but after thinning the pixels of a line might not be aligned to each other in a single row or column. This creates a jagged effect in the extracted sequences. The jagged artifacts are the main reason for the decreased recognition rate in LSTM training. As the early stopping method [18] is used, the most number of epochs is used by the network whose input size of 20 for both models.

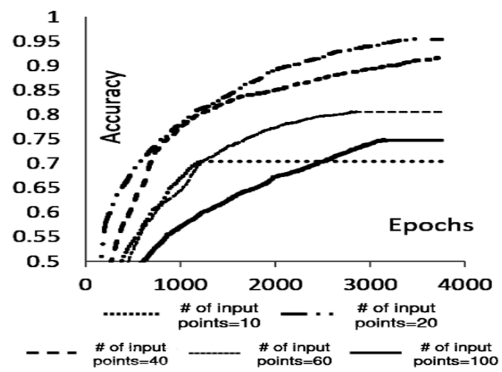

Fig. 6. Accuracy vs Epochs for training Single layer LSTM classifier.

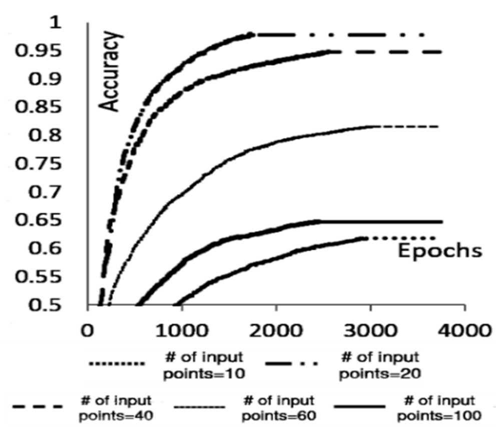

Fig. 7. Accuracy vs Epochs for training Deep LSTM classifier.

Table I compares test set accuracy between single layer LSTM network and Deep LSTM network for different input points. It is observed from the that the deep LSTM performs better than the LSTM network with a single layer. The best recognition rate of this methodology is 98.03% with Deep LSTM network for 20 inputs.

For a better understanding of the proposed method's proficiency, standard CNN is also considered in the investigation. A standard CNN model is trained in the same CVRP, ISI dataset, and its resulting confusion matrix is shown in Table II. On the other hand, the confusion matrix for best performed deep LSTM (with 20 input points) is shown in Table III. It is observed from Table II that a large number of samples from ১ and ৯ are misclassified interchangeably by CNN; 12 handwritten numerals of ১ misclassified as ৯ and on the other hand, eight cases of ৯ misclassified as ১. The numerals ১ and ৯ are shown very similarly in handwritten cases and interchangeably misclassification scenario clearly depicts the shortcomings of image-based operation of CNN. The effectiveness of the proposed sequence-based classification with LSTM is clearly revealed from Table III for the two similar shaped numerals. For the deep LSTM network, only two numerals misclassified for each of ১ and ৯. Since general writing sequences of ১ and ৯ are different then outperformance of sequence-based LSTM is logical with prominent aspects.

The achieved test set recognition accuracy of proposed deep LSTM method is compared to recent works on same CVRP, ISI dataset in Table IV. The proposed method is shown competitive performance with the existing methods. Although the proposed method is not outperformed others, the major contribution of this research is achieved to increase HNR for similar shaped numerals (e.g., ১ and ৯) where CNN is weak in classification.

Table I. Test set accuracy comparison between single layer LSTM network and Deep LSTM network for different input points.

# of input points	Single layer LSTM accuracy (%)	Deep LSTM accuracy (%)
5	10.58	10.58
10	70.48	61.80
20	95.45	98.03
40	91.65	94.84
60	80.63	81.62
80	59.60	64.80
100	74.88	62.13

Table II. Confusion matrix of BHNR using CNN.

Class	Total samples of a particular numeral classified as									
	০ (0)	১ (1)	২ (2)	৩ (3)	৪ (4)	৫ (5)	৬ (6)	৭ (7)	৮ (8)	৯ (9)
০ (0)	391	2	0	2	1	2	1	0	1	0
১ (1)	1	377	4	0	1	2	1	2	0	12
২ (2)	0	3	391	0	1	1	1	1	1	1
৩ (3)	4	0	0	393	0	2	1	0	0	0
৪ (4)	1	3	2	0	391	3	0	0	0	0
৫ (5)	2	0	1	2	2	389	1	1	1	1
৬ (6)	0	0	2	4	0	2	389	0	2	1
৭ (7)	0	2	1	0	5	1	0	391	0	0
৮ (8)	0	0	1	1	1	1	1	1	394	0
৯ (9)	0	8	0	0	3	0	1	1	0	387

Table III. Confusion matrix of BHNR using proposed Deep LSTM Network.

Class	Total samples of a particular numeral classified as									
	০ (0)	১ (1)	২ (2)	৩ (3)	৪ (4)	৫ (5)	৬ (6)	৭ (7)	৮ (8)	৯ (9)
০ (0)	391	2	0	2	1	2	1	0	1	0
১ (1)	1	391	4	0	0	0	0	2	0	2
২ (2)	0	3	393	0	1	0	1	1	1	0
৩ (3)	4	0	0	393	0	2	1	0	0	0
৪ (4)	1	3	2	0	391	3	0	0	0	0
৫ (5)	2	0	1	2	2	390	0	1	1	1
৬ (6)	0	0	2	4	0	0	391	0	2	1
৭ (7)	0	2	1	0	5	1	0	391	0	0
৮ (8)	0	0	1	1	1	1	1	0	395	0
৯ (9)	0	2	0	0	3	0	0	0	0	395

TABLE IV. PERFORMANCE COMPARISON OF PROPOSED METHOD WITH PROMINENT METHODS ON CVPR, ISI BANGLA DATASET.

Work Reference, Year	Feature Selection	Classification	Test Set Rec. Accuracy
Bhattacharya and Chaudhuri [12], 2009	Wavelet filter	A cascade of MLPs	98.20%
Akhand et al. [19], 2016	No	CNN	98.45%
Shopon et al. [20], 2016	No	Autoencoder with CNN	98.29%
Akhand et al. [21], 2018	No	Rotation based CNN	98.98%
Ahmed et al. [10], 2019	No	DLSTM	98.25%
Proposed Method (Sequence Extraction and Classification with LSTM)	Sequence	Single Layer LSTM	95.45%
		Deep LSTM	98.03%

IV. CONCLUSIONS

This study investigated the handwritten numeral recognition model through the extraction of sequence and identification using an LSTM classifier. The method has been examined for Bangla numerals and accomplished satisfactory results. The proposed method is shown competitive performance to the recent works. An ensemble of proposed LSTM methods might give a better result and remain as future studies.

ACKNOWLEDGMENT

The authors would like to thank the Dr. Ujjwal Bhattacharya (Computer & Communication Sciences Division, Indian Statistical Institute, Kolkata, India) for the benchmark datasets used in this study.

REFERENCES

[1] Wikipedia, "List of languages by number of native speakers," 2010. [Online]. Available: https://en.wikipedia.org/wiki/List_of_languages_by_number_of_native_speakers.

[2] M. Shopon, N. Mohammed, and M. A. Abedin, "Bangla handwritten digit recognition using autoencoder and deep convolutional neural network," *IWCI 2016 - 2016 Int. Work. Comput. Intell.*, no. December, pp. 64–68, 2017, doi: 10.1109/IWCI.2016.7860340.

[3] Y. Wen, Y. Lu, and P. Shi, "Handwritten Bangla numeral recognition system and its application to postal automation," *Pattern Recognit.*, vol. 40, no. 1, pp. 99–107, Jan. 2007, doi: 10.1016/j.patcog.2006.07.001.

[4] M. Ahmed, A. K. Paul, and M. A. H. Akhand, "Stacked auto encoder training incorporating printed text data for handwritten bangla numeral recognition," in *2016 19th International Conference on Computer and Information Technology (ICCIT)*, 2016, pp. 437–442, doi: 10.1109/ICCITECHN.2016.7860238.

[5] S. S. Roy, M. Ahmed, and M. A. H. Akhand, "NOISY IMAGE CLASSIFICATION USING HYBRID DEEP LEARNING METHOS," *J. Inf. Commun. Technol.*, vol. 17, no. 2, pp. 233–269, 2018.

[6] U. Pal, N. Sharma, T. Wakabayashi, and F. Kimura, "Handwritten numeral recognition of six popular Indian scripts," in *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, 2007, vol. 2, pp. 749–753.

[7] S. Basu, N. Das, and R. Sarkar, "An MLP based Approach for Recognition of Handwritten ' Bangla ' Numerals," in *2nd Indian International Conference on Artificial Intelligence*, 2005.

[8] M. A. H. Akhand, M. Ahmed, and M. M. H. Rahman, "Convolutional neural network training with artificial pattern for bangla handwritten numeral recognition," in *2016 5th International Conference on Informatics, Electronics and Vision (ICIEV)*, 2016, pp. 625–630.

[9] K. C. Santosh and E. Iwata, "Stroke-Based Cursive Character Recognition," *Adv. Character Recognit.*, p. 175, 2012.

[10] M. Ahmed, M. A. H. Akhand, and M. M. Hafizur Rahman, "Recognizing Bangla Handwritten Numeral Utilizing Deep Long Short Term Memory," *Int. J. Image, Graph. Signal Process.*, vol. 11, no. 1, pp. 23–32, 2019, doi: 10.5815/ijigsp.2019.01.03.

[11] U. Bhattacharya and B. B. Chaudhuri, "Databases for research on recognition of handwritten characters of Indian scripts," in *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, 2005, doi: 10.1109/ICDAR.2005.84.

[12] U. Bhattacharya and B. B. Chaudhuri, "Handwritten numeral databases of Indian scripts and multistage recognition of mixed numerals," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 3, pp. 444–457, 2008.

[13] C. Yu, C. Dian-ren, L. Yang, and C. Lei, "Otsu's thresholding method based on gray level-gradient two-dimensional histogram," in *2010 2nd International Asia Conference on Informatics in Control, Automation and Robotics (CAR 2010)*, 2010, vol. 3, pp. 282–285.

[14] T. Y. Zhang and C. Y. Suen, "A fast parallel algorithm for thinning digital patterns," *Commun. ACM*, vol. 27, no. 3, pp. 236–239, 1984.

[15] D. L. Applegate, R. E. Bixby, V. Chvatal, and W. J. Cook, *The traveling salesman problem: a computational study*. Princeton university press, 2006.

[16] I. J. Schoenberg, *Cardinal spline interpolation*, vol. 12. Siam, 1973.

[17] M. D. Zeiler, "Adadelta: an adaptive learning rate method," *arXiv Prepr. arXiv1212.5701*, 2012.

[18] T. Zhang, B. Yu, and others, "Boosting with early stopping: Convergence and consistency," *Ann. Stat.*, vol. 33, no. 4, pp. 1538–1579, 2005.

[19] M. A. H. Akhand, M. Ahmed, and M. M. H. Rahman, "Convolutional Neural Network based Handwritten Bengali and Bengali-English Mixed Numeral Recognition," *Int. J. Image, Graph. Signal Process.*, vol. 8, no. 9, pp. 40–50, 2015, doi: 10.5815/ijigsp.2015.09.06.

[20] M. Shopon, N. Mohammed, and M. A. Abedin, "Bangla handwritten digit recognition using autoencoder and deep convolutional neural network," in *2016 International Workshop on Computational Intelligence (IWCI)*, 2016, pp. 64–68, doi: 10.1109/IWCI.2016.7860340.

[21] M. A. H. Akhand, M. Ahmed, M. M. H. Rahman, and M. M. Islam, "Convolutional Neural Network Training incorporating Rotation-Based Generated Patterns and Handwritten Numeral Recognition of Major Indian Scripts," *IETE J. Res.*, vol. 64, no. 02, p. 176, 2018, doi: 10.1080/03772063.2017.1351322.