Handwritten Bangla Numeral Recognition using Deep Long Short Term Memory

Mahtab Ahmed, M. A. H. Akhand
Dept. of Computer Science and Engineering
Khulna University of Engineering & Technology
Khulna, Bangladesh

M. M. Hafizur Rahman
Dept. of Computer Science, KICT
International Islamic University Malaysia
Selangor, Malaysia

***Abstract*—Recognition of handwritten numerals has gained much interest in recent years due to its various application potentials. Bangla is a major language in Indian subcontinent and is the first language of Bangladesh; but unfortunately, study regarding handwritten Bangla numeral recognition (HBNR) is very few with respect to other major languages such as English, Roman etc. Some noteworthy research works have been conducted for recognition of Bangla handwritten numeral using artificial neural network (ANN) as ANN and its various updated models are found efficient for classification task. The aim of this study is to develop a better Bangla handwritten numeral recognition system and hence investigated deep architecture of Long Short Term Memory (LSTM) method. LSTM is a variant of recurrent neural networks (RNN) and is applied efficiently for image classification with its distinct features. The proposed HBNR-LSTM normalizes the written numeral images first and then employs two layers of LSTM to classify individual numerals. Unlike other methods, it does not employ any feature extraction technique. Benchmark dataset with 22000 hand written numerals with different shapes, sizes and variations are used in this study. The proposed method is shown satisfactory recognition accuracy and outperformed other prominent exiting methods.**

Keywords—Bangla Handwritten Numeral, Recurrent Neural Network, Long Short Term Memory, Deep Neural Networks.

I. INTRODUCTION

Handwritten numeral recognition has gained much attention in present days because of its potential to be applied in various fields like postal system automation, passports and document analysis, automatic bank cheque processing and even for number plate identification [1]. Research on unconstrained handwritten numeral recognition has shown impressive progress in Roman, Chinese and Arabic script [1, 2, 3]. On the other hand, research on recognition of handwritten Bangla numeral hasn't shown that much progress although it has ranked 7th in the list of languages based on native speakers [4] as well as is the first language of Bangladesh.

Some noteworthy research works have been conducted for recognition of Bangla handwritten numeral using artificial neural network (ANN) as ANN and its various updated models are found efficient for classification task. Khan et al. [5] utilized evolutionary approach in training artificial neural network (ANN) with handwritten numeral data in Bengali language. At first, they used boundary extraction for extracting numeral in a single window by horizontal and vertical scanning; and scaled the image into fixed sized matrix. Then Multi-Layer Perceptron (MLP) are developed for recognition. Basu et al. [3] used Dempster-Shafer (DS) technique where they combined the classification decisions of two MLP based classifiers for handwritten Bangla numeral using two different feature sets. They investigated two feature sets called shadow feature and centroid feature.

Bhattacharya and Chaudhuri [6] presented a multistage cascaded recognition scheme using MLP classifiers. In their scheme, first they computed features using wavelet-filtered image at different resolutions and then used a cascade of three MLP classifiers. If a decision about the possible class of an input numeral cannot be made by any of the MLPs of the first stage, then its class conditional probabilities are fed to another MLP of the second stage. Finally, the numeral image is either classified or rejected according to a precision index in the second stage.

Wen et al. [7] proposed a recognition system for handwritten Bangla numeral based on Support Vector Machine (SVM) that can be applied in automatic letter sorting machine. They used Principal Component Analysis (PCA) along with kernel PCA (KPCA) as extensive feature extractor and SVM as the classifier. Most recently, Nasir and Uddin [8] inaugurated a hybrid system of Bangla handwritten numeral recognition for automated postal system, which uses k-means clustering, Baye's theorem and Maximum a Posteriori for feature extraction, later SVM was used for the recognition process.

Recently, deep neural network (DNN) have been found to be very potential in pattern classification imitating the underlying features of the data through its deep architecture. Convolutional neural network (CNN) has shown better numeral recognition accuracy than tradition NN based methods. Akhand et al. [9] proposes a CNN based Bangla handwritten numeral recognition method in which they trained a two layer CNN with the normalized gray scale value of image pixels as the features and use a fully connected NN at the output layer to predict the appropriate class labels.

A few works of Bangla handwritten numeral recognition are also available whose uses different techniques rather than ANN. Bashar et al. [10] investigated a recognition system for numeral data based on windowing and histogram techniques. Windowing technique extracts uniform features from scanned image files and then these generated features produce a

978-1-5090-4521-1/16 $31.00 © 2016 IEEE
DOI 10.1109/ICT4M.2016.62

histogram. Finally, recognition of the digit is achieved on the basis of that generated histogram. Pal et al. [2] introduced a new method based on the idea of water overflow from the reservoir for extracting features and then used binary tree classifier for Bangla handwritten numeral recognition. Recently, Wen and He [11] proposed a method to recognize handwritten Bangla numeral based on kernel and Bayesian Discriminant.

The aim of this study is to develop a better Bangla handwritten numeral recognition system and hence investigated deep architecture of Long Short Term Memory (LSTM) method. LSTM is a variant of recurrent neural networks (RNN) and is applied efficiently for image classification with its distinct features. It easily learns about sequence alignment and work without the need of any segmented data, but it takes much longer time to converge. On the other hand, DNNs perform better than single layer LSTM for speech recognition [12] which gives a scope to the use of deep LSTM architecture for handwritten numeral recognition.

The deep LSTM architecture used in this work consists of two LSTM layers, followed by a reshape layer, then a dense-layer and finally the output layer. Handwritten numeral images are normalized first and then employ LSTM to classify individual numerals. Unlike other related works it does not employ any feature extraction method. Benchmark dataset with large number of handwritten numeral images is used to identify the effectiveness of the proposed method. In order to highlight the significance of the deep architecture, a single layer LSTM network was also tested on the same dataset. It is observed that deep LSTM gives higher accuracy in very less number of training steps than single layer LSTM. Experimental studies reveal that the proposed deep LSTM based method shows satisfactory classification accuracy and outperformed some other exiting methods.

The rest of the paper is organized as follows. Section II explains proposed HBNR using LSTM which contains dataset Preparation, preprocessing and classification using LSTM. Section III presents experimental results of the proposed method and performance comparison with other related works. Finally, a brief conclusion of the work is given in Section IV.

II. Handwritten Bangla Numeral Recognition using LSTM (HBNR-LSTM)

This section explains proposed HBNR-LSTM in detail which has two major steps: preprocessing of raw images of numerals and classification using LSTM. The following subsection gives brief description of each steps. At first it explains Dataset preparation for better understanding.

A. Dataset Preparation

Here we have constructed a moderately large dataset for Bangla handwritten isolated numerals. For handwritten scripts, we have reviewed around several individuals from different education levels, sex and ages. The dataset accommodates 22000 images with extensive variation of distinct numeral images by different people practicing different writing styles. Fig. 1 shows few sample images of each numeral.

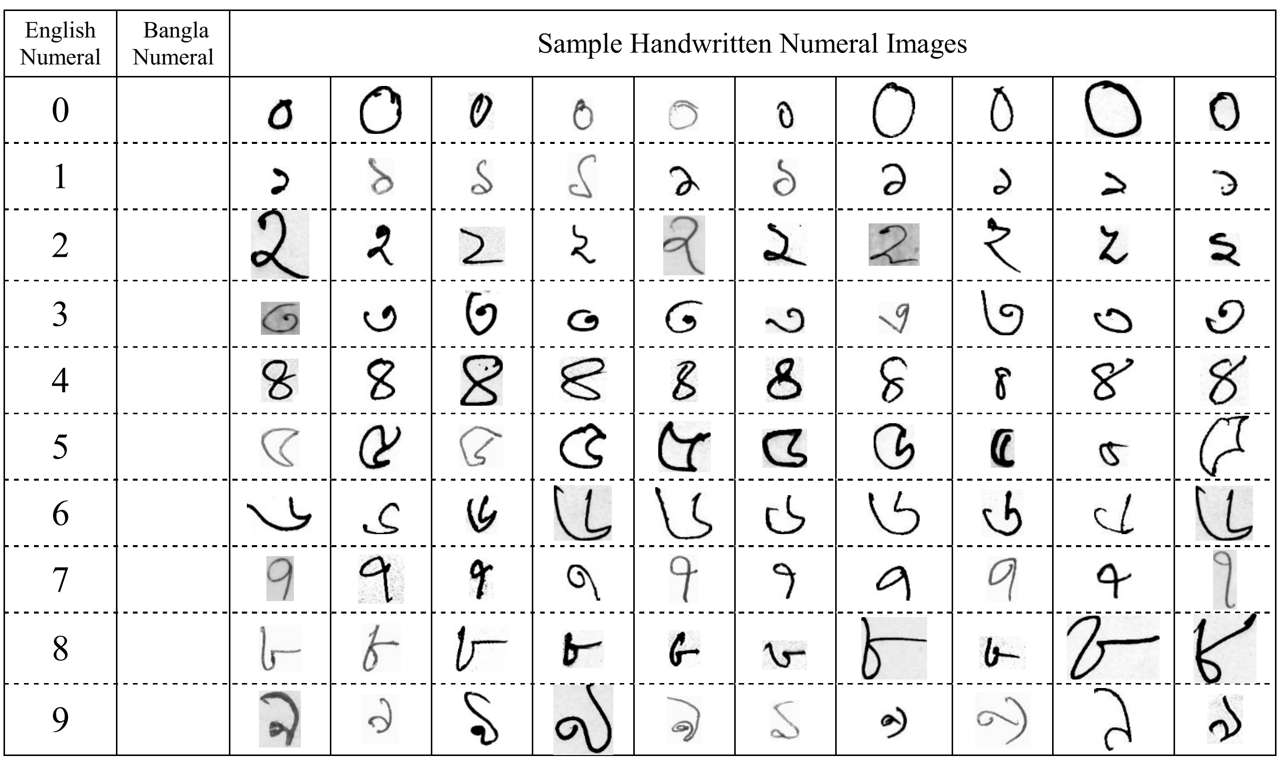


Fig.1. Samples of handwritten Bangla numerals from the dataset.

B. Preprocessing of Raw Numeral Images

In order to make the raw images appropriate to feed into classifiers, some preprocessing techniques are applied. In the first step, handwritten numerals are scanned and converted to gray scale images. In a grayscale image, each pixel value is a single integer number that varies from 0 to 255. This integer value represents the brightness of the pixel. Typically integer value 255 represents white pixel where value 0 represents black pixel. The image files reviewed here even for a numeral are often found in different sizes for different persons because of their different writing style. Finally, to maintain appropriate and equal inputs for all the numerals, the arbitrary images are resized into 28 x 28 dimension.

The grayscale image files contains more white pixels than black for writing, since black color is considered for writing on white paper (background) and this increases computational overhead. In order to compensate this, images are converted through background numeral white to black and foreground changed to white.

C. Classification using Long Short Term Memory (LSTM)

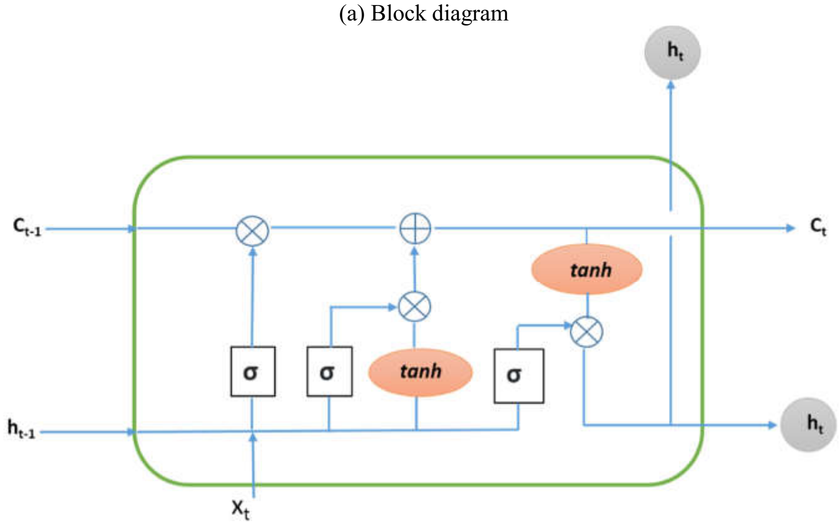
Handwritten numeral classification is a high-dimensional complex task and traditional MLP require much computation to work with grayscale image. Therefore, a number of traditional methods [5] first extract features from the input image and then use MLP based methods for classification task.

Recurrent neural networks (RNN) form a memory by mapping all previous inputs to each output and thus gain the ability to model contextual information from the data sequence by using their recurrent connections. As a result, if the number of hidden layers in RNNs are increased, the gradient/error signal decreases exponentially with the number of hidden layers, making the training of the front layers very slowly [13], [14]. On the other hand, some post-processing scheme would be required to compute the likelihood of consecutive labels as RNN produces localized classification. That's why, a lot of limitation is faced in handling sequential labeling problems like handwriting recognition, speech recognition and several others using RNNs. Moreover, like other artificial neural networks which use gradient based learning methods, RNNs also suffers from vanishing gradient problem.

LSTM [15] architecture with RNNs overcomes the vanishing gradient problem, as an LSTM block acting as a memory unit remembers a value for an arbitrary length of time. Moreover, it also has the ability to access long range context, learn sequence alignment and work without the need of any segmented data. Therefore, LSTM, a variant of RNN, is considered for Bangla handwritten numeral in this study. Each LSTM block contains three multiplicative gates: input gate, forget gate and output gate. These three multiplicative gates operate on the same LSTM block and determine whether the input is important enough to remember or forget the value by activating the forget gate. Thus the memory cells remember the error without perishing till the forget gate is activated resulting in the prevention of error decay.

Figure 2 shows network architecture of this study for classification of Bangla handwritten numeral. Figure 2(a) shows the block diagram of the proposed network which

(a) Block diagram



(b) Enlarged view of an LSTM cell

Fig. 2. Structure of the network for proposed HBNR- LSTM

consists of an input layer, two LSTM layers, a reshape layer, a dense layer and finally the output layer. The first LSTM layer contains 50 hidden LSTM cells and the second one contains 100 hidden LSTM cells respectively. They are separated by feed-forward layers and *tanh* is used as the activation function. The following reshape layer reshapes the second LSTM layer units to the appropriate size and the dense layer creates a fully connected structure with the output layer. Finally the reshape layer reshapes the output layer units according to the 10 class labels.

Figure 2(b) shows the enlarged version of an LSTM cell. An LSTM cell takes previous cell state and the hidden unit values from the previous layer as input and outputs a current cell state and the next layer hidden unit values based on the forget gates. The network works in two passes- Forward pass and backward pass. Based on the data and the feature form the data, forward pass initializes the network parameter values and activates the hidden layer from the external input as well as hidden LSTM layer activations from the previous time-step. During forward pass, an LSTM layer decides whether to keep an information completely or forget it. A sigmoid layer called the forget gate layer makes this decision. It takes $\;_t$ and x_t as input and provides an output between 0 and 1 for each number in the cell state C_{t-1}. If its' output is 1, it will keep the data completely, while for 0 it will forget the data.

$$f_t = \sigma(W_f[\;_{t-1}, x_t] + b_f), \tag{1}$$

In the next step, it decides what new information it would store in the cell state. First a sigmoid layer called input layer followed by a *tanh* layer decides which values it'll update and then the following *tanh* layer generates a vector of new candidate values C_t'.

$$i_t = \sigma(W_i \quad _{t-1} + W_i \quad x_t + b_i) \tag{2}$$

$$C'_t = tan\ (W_c \quad _{t-1} + W_c \quad x_t + b_c) \tag{3}$$

In the next step, LSTM combines these two and updates the cell state from C_{t-1} to C_t. At first, the previous cell state value is multiplied by the forgetting factor f_t, that makes the LSTM forgetting the data and then this value is added with $i_t * C_t'$ to produce the new candidate values

$$C_t = f_t \quad C_{t-1} + i_t \quad C'_t \tag{4}$$

In the output state, the output would be the filtered version of the cell state. The output is generated in two steps. In the first step, a sigmoid layer decides/selects a part of the cell state that would be presented as the output. In the second step, the selected cell states are passed through a *tanh* and multiplied by the output of the sigmoid gate. The selected cell states are passed through the *tanh* to push the values to be between -1 to 1.

$$o_t = \sigma(W_o \quad _{t-1} + W_o \quad x_t + b_o) \tag{5}$$

$$_t = o_t \quad \tanh(C_t) \tag{6}$$

In the backward pass, BPTT algorithm [16] calculates the derivative of the loss function (i.e. gradient) with respect to output layer activation for each point. Finally this gradient is applied to update the network parameters converging the network towards generalization.

III. Results and Discussions

Experimental results of the proposed recognition scheme have been collected based on the samples of the prepared dataset discussed earlier. 22000 samples are divided into 18000 and 4000 samples, for training and testing, respectively. Training samples are evenly distributed over the underlying 10 classes. The recognition performance reported in this paper are based on the test set accuracies. In the test set, equal number of samples (i.e., 400) for each numeral were considered.

We applied this deep LSTM architecture on the resized and normalized grayscale image files without any feature extraction technique. The method is implemented in python using Theano and Lasagne. The experiment has been conducted on MacBook Pro laptop machine (CPU: Intel Core i5 @ 2.70 GHz and RAM: 8.00 GB) in OS X Yosemite environment.

We have observed classification accuracy of the system for various fixed number of iterations and it is observed that the network started to give more than 95% test set accuracy with only 50 iterations. Fig. 3 shows the training set and test set classification (i.e., recognition) accuracy at different iterations for batch size 50. It is observed that recognition accuracy improved with iteration for both training and test sets rapidly at lower iteration values (e.g., up to 100). For the higher iteration values, recognition accuracy was also improved coinciding minimization of E, since LSTM cells are trained with bit values of the patterns from training set images. On the other hand, testing set accuracy did not coincide with training set accuracy because its patterns were unseen by LSTM during training. It is notable that the accuracy on test is more desirable that indicates the generalization ability of a system.

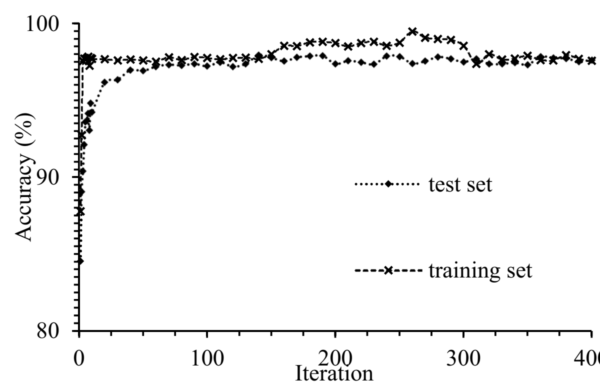


Fig. 3. Training and Test Sets Recognition Accuracy for different iterations with batch size 50.

Table I shows the confusion matrix of test set samples after fixed 460 iterations. From the table it is observed that the proposed method worst performed for the numeral “ ” and 383 cases it classified truly out of 400 test cases. In eight cases this numeral classified as “ ”. Both “ ” and “ ” have some kind of ambiguity in handwritten form. In the Bangla handwritten numeral script, “ ” and “ ” looks very similar; therefore in 11 cases “ ” classified as “ ”. Similarly, the numeral “ ” classified as “ ” and “ ” in three and four cases, respectively; it is clearly observed that these confusions in several handwritten images are due to diverse writing styles of individuals. But the proposed method is shown best performance for “ ” and correctly classifies all 400 test samples.

Table II shows some handwritten numeral images those are misclassified. Due to large variation in writing styles, such numeral images are difficult to classify even by human. Finally, the proposed HBNR-LSTM misclassified 70 test samples out of 4000 test cases and achieved accuracy of 98.25%. On other hand, the method misclassified only 90 cases out of 18000 training samples showing accuracy rate of 99.50%. This indicates that there is a chance to improve deep LSTM training and get better performance with the proposed method.

Table III compares the outcome of the proposed method with other prominent works of Bangla handwritten numeral recognition. It also presents distinct features of individual methods. It is notable that proposed method did not employ any feature selection technique whereas an existing method uses one or two feature selection methods. Without feature selection, proposed HBNR-LSTM method is shown to outperform the exiting methods. According to the table, HBNR-LSTM achieved test set accuracy of 98.25%, on the other hand, the test set accuracy are 92.80%, 95.05% and 97.93% for the works of [2], [7] and [9], respectively. Test set accuracy of work [6] is very close to the proposed one, but it used a scaled up version of the original dataset which is 10 times larger than the one used in this study. Although performance compared here are for different datasets, the efficacy of the proposed HBNR-LSTM is quite interesting and identified the ability of deep LSTM based classifier for Bangla handwritten numeral recognition.

TABLE II. SAMPLE HANDWRITTEN NUMERALS THAT ARE MISCLASSIFIED BY HBNR-LSTM.

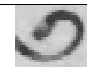
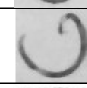
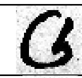
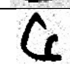
Handwritten Numeral Image	**True Numeral**	**Image Classified as**
		
		
		
		
		
		
		

TABLE I. CONFUSION MATRIX PRODUCED FOR TEST SAMPLES OF BANGLA HANDWRITTEN NUMERALS. TOTAL SAMPLES ARE 4000 HAVING 400 FOR EACH NUMERAL.

English Numeral	Bangla Numeral	Total samples of a particular numeral classified as									
0		398	0	1	0	0	0	1	0	0	0
1		0	382	1	0	3	0	3	0	0	11
2		0	0	400	0	0	0	0	0	0	0
3		1	0	0	393	0	1	3	0	2	0
4		0	0	0	0	399	0	0	0	1	0
5		0	0	0	1	3	392	4	0	0	0
6		1	0	0	11	0	2	386	0	0	0
7		0	0	1	0	0	1	0	398	0	0
8		0	0	0	0	0	1	0	0	399	0
9		1	8	3	4	1	0	0	0	0	383

TABLE III. A COMPARATIVE DESCRIPTION OF PROPOSED HBNR-LSTM WITH SOME CONTEMPORARY METHODS.

The work reference	Feature Selection	Classification	Recognition Accuracy
Pal et al. [2]	Water overflow from the reservoir based feature selection	Binary decision tree	92.80%
Wen et al. [7]	Principal component analysis (PCA) and Kernel PCA	SVM	95.05 %
Basu et al. [3]	Shadow feature and Centroid feature	MLPs with Dempster-Shafer technique	95.10%
Bhattacharya and Chaudhuri [6]	Wavelet filter at different resolutions	Four MLPs in two stages (three + one)	98.20%
Akhnad et al. [9]	No	CNN	97.93%
Proposed HBNR-LSTM	No	LSTM	**98.25 %**

IV. CONCLUSIONS

This paper proposes a Deep LSTM architecture for handwritten Bangla numeral recognition. LSTM has the ability to recognize visual patterns directly from pixel images with minimal preprocessing. Therefore, an LSTM structure is investigated without any feature selection for handwritten Bangla numeral recognition in this study achieving promising result. The study also reveals that the deep networks need lesser data than shallow networks. The method has been tested on a large hand written numeral dataset and outcome compared with existing prominent methods for Bangla. The proposed method is shown to outperform the existing methods on the basis of test set accuracy without scaling the data size. Moreover, the proposed scheme seems efficient in size and computation.

REFERENCES

[1] R. Plamondon and S. N. Srihari, "On-line and off-line handwritten recognition: A comprehensive survey," *IEEE Trans. on PAMI,* vol. 22, pp. 62-84, 2000.

[2] U. Pal, C. B. B. Chaudhuri and A. Belaid, "A System for Bangla Handwritten Numeral Recognition," *IETE Journal of Research, Institution of Electronics and Telecommunication Engineers*, vol. 52, no. 1, pp. 27-34, 2006.

[3] S. Basu, R. Sarkar, N. Das, M. Kundu, M. Nasipuri and D. K. Basu, "Handwritten BanglaDigit Recognition Using Classifier Combination Through DS Technique," *LNCS*, vol. 3776, pp. 236–241, 2005

[4] List of languages by number of native users. Available: https://en.wikipedia.org/wiki/List_of_languages_by_number_of_native_users

[5] M. M. R. Khan, S. M. A. Rahman and M. M. Alam, "Bangla Handwritten Digits Recognition using Evolutionary Artificial Neural Networks*" in Proc. of the 7th International Conference on Computer and Information Technology (ICCIT 2004),* 26-28 December, 2004, Dhaka, Bangladesh.

[6] U. Bhattacharya and B. B. Chaudhuri, "Handwritten numeral databases of Indian scripts and multistage recognition of mixed numerals", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 31, no. 3, pp. 444-457, 2009.

[7] Y. Wen, Y. Lu and P. Shi, "Handwritten Bangla numeral recognition system and its application to postal automation," *Pattern Recognition*, vol. 40, pp. 99-107, 2007.

[8] M. K. Nasir and M. S. Uddin, "Hand Written Bangla Numerals Recognition for Automated Postal System," *IOSR Journal of Computer Engineering (IOSR-JCE)*, vol. 8, no. 6, pp. 43-48, 2013.

[9] M. A. H. Akhand, Md. Mahbubar Rahman, P. C. Shill, ShahidulIslam and M. M. Hafizur Rahman, "Bangla handwritten numeral recognition using convolutional neural network," in *Proc. of International Conference on Electrical Engineering and Information & Communication Technology (iCEEiCT2015)*, pp. 1-5, May 2015.

[10] M. R. Bashar, M. A. F. M. R. Hasan, M. A. Hossain and D. Das, "Handwritten Bangla Numerical Digit Recognition using Histogram Technique," *Asian Journal of Information Technology*, vol. 3, pp. 611-615, 2004.

[11] Y. Wen and L. He, "A classifier for Bangla handwritten numeral recognition," *Expert Systems with Applications*, vol. 39, pp. 948-953, 2012.

[12] M. Ahmed et al., "Acoustic Modeling of Bangla Words using Deep Belief Network", *International Journal of Image, Graphics and Signal Processing (IJIGSP),* vol. 7, no. 10, pp. 19-27, Sep. 2015.

[13] Vanishing gradient problem. Available: https://en.wikipedia.org/wiki/Vanishing_gradient_problem

[14] S. Hochreiter, Y. Bengio, P. Frasconi, and J. Schmidhuber. "Gradient flow in recurrent nets: the difficulty of learning long-term dependencies." *In A Field Guide to Dynamical Recurrent Neural Networks*. IEEE Press, 2001.

[15] Sepp Hochreiter and J¨urgen Schmidhuber. "Long short-term memory". *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[16] A. Roy, S. Rajeswar and S. Chaudhury, "Text reognition using deep BLSTM networks", *International conference on* adv*ances in pattern recognition (ICAPR)*, pp. 1- 6, Jan. 2015.