Bangla Handwritten Digit Recognition Using Convolutional Neural Network

Conference Paper · December 2018

DOI: 10.1007/978-981-13-1951-8_11

4 authors:

Akm Shahariar Azad Rabby
University of Alabama at Birmingham

Sheikh Abujar
University of Alabama at Birmingham

Sadeka Haque
Daffodil International University

Syed Akhter Hossain
Daffodil International University

Bangla Handwritten Digit Recognition Using Convolutional Neural Network

AKM Shahariar Azad Rabby, Sheikh Abujar, Sadeka Haque and Syed Akhter Hossain

Abstract Handwritten digit recognition has always a big challenge due to its variation of shape, size, and writing style. Accurate handwritten recognition is becoming more thoughtful to the researchers for its educational and economic values. There had several works been already done on the Bangla Handwritten Recognition, but still there is no robust model developed yet. Therefore, this paper states development and implementation of a lightweight CNN model for classifying Bangla Handwriting Digits. The proposed model outperforms any previous implemented method with fewer epochs and faster execution time. This Model was trained and tested with ISI handwritten character database Bhattacharya and Chaudhuri (IEEE Trans Pattern Anal Mach Intell 31:444–457, 2009, [1], BanglaLekha Isolated Biswas et al. (Data Brief 12, 103–107, 2017, [2]) and CAMTERDB 3.1.1 Sarkar et al. (Int J Doc Anal Recogn (IJDAR) 15(1):71–83, 2012, [3]). As a result, it was successfully achieved validation accuracy of 99.74% on ISI handwritten character database, 98.93% on BanglaLekha Isolated, 99.42% on CAMTERDB 3.1.1 dataset and lastly 99.43% on a mixed (combination of BanglaLekha Isolated, CAMTERDB 3.1.1 and ISI handwritten character dataset) dataset. This model achieved the best performance on different datasets and found very lightweight, it can be used on a low processing device like-mobile phone.The pre-train model and code for all these datasets can be found on this link https://github.com/shahariarrabby/Bangla_Digit_Recognition_CNN.

A. S. A. Rabby · S. Abujar (✉) · S. Haque · S. A. Hossain
Department of Computer Science and Engineering, Daffodil International University, Dhanmondi, Dhaka 1205, Bangladesh
e-mail: sheikh.cse@diu.edu.bd

A. S. A. Rabby
e-mail: azad15-5424@diu.edu.bd

S. Haque
e-mail: sadeka15-5210@diu.edu.bd

S. A. Hossain
e-mail: aktarhossain@daffodilvarsity.edu.bd

A. Abraham et al. (eds.), *Emerging Technologies in Data Mining and Information Security*, Advances in Intelligent Systems and Computing 755,
https://doi.org/10.1007/978-981-13-1951-8_11

Keywords Bangla handwritten recognition · Convolutional neural network
Pattern recognition · Deep learning · Computer vision · Machine learning

1 Introduction

Handwritten digit recognition development research is rapidly evolving and reshaping the must automation fields like—automatic check reading, automatic number plate reading, digital postal service, Optical Image Recognizing (OCR), etc. Due to its various aspects of uses, computer vision researchers verily feel to work on it and improve—quality and performance indeed. But handwritten recognizing is more challenging compared to the typed letter. Because different people write in a different way and which creates a higher degree of variance in written style. Also, there are some similarities between different characters shape. The situation of overwriting makes it more challenging for accurately classifying the handwritten digit.

Nowadays, deep learning, especially the Convolutional neural network is working better in the purpose of classifying these types of recognition work rather other machine learning methods.

The Bangla alphabet incorporates the writing system for the Bangla language together with the Assamese alphabets. Bangla is the fifth most widely writing language in the world. With 250 million speakers, Bangla is the seventh most spoken language in the world by population. Also, it is one of the major languages in the Indian subcontinent and is the first language in Bangladesh. But the research held on Bangla handwriting recognition is very few compared with other languages like English, Arabic, Hindi, Chinese, etc.

Every language has its own Wscript. Bangla came from Sanskrit script which is completely different from English or other popular scripts. There are 50 characters, 10 numerical digits, more than 200 compound characters and modifier are available in Bangla language. Also, it has many similar characters, some of them are different considering small dot and line. And compound characters can be made by joining two characters. Almost all Bangla consonant can be used to make a new compound character. This makes it more difficult to achieve a good result and better performance with Bangla Handwritten character recognition. There are many other applications which may use this Bangla digit recognition system. Such as Bangla Handwritten character base OCR (Optical Character Recognition), Picture to text to speech, Bangla ID card reading, Number plate reading, vehicle tracking, Post office automation, etc.

2 Related Work

Several studies show that CNN can be successfully applied in the complex image classification work. For English MNIST [4] CNN got a remarkable result.

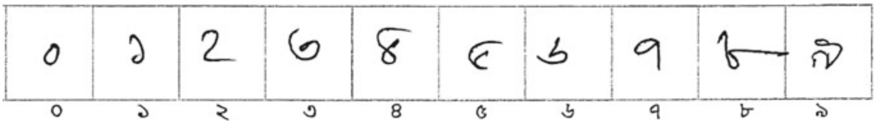

Fig. 1 Example of Bangla digit

There are few works available on Bangla digit recognition who reaches remarkable result in Bangla Handwritten digit recognition. "*Handwritten numeral databases of Indian scripts and multistage recognition*" works on mixed numerals [1]. The main feature of their model includes matra (the upper part of the character), vertical line and double vertical line and for the MLF classifier, the feature is constructed from the stroke feature of the characters (Fig. 1).

"*Automatic Recognition of Unconstrained Off-Line Bangla Handwritten Numerals*" [5] is also concluded with some good research work. The proposed method mentioned, used to extract features from a concept called water reservoir. It was implemented and continued further use in postal automation sector. Another work is "*Handwritten Bangla Digit Recognition Using Deep Learning*" [6]. Where a CNN model was used, that achieves 98.78% accuracy on CMATERDB 3.1.1 datasets.

3 Proposed Methodology

The proposed method uses a Convolutional Neural Network which has many phases, such as Dataset preparation, model training, etc., which is described below.

3.1 Datasets

This method used three datasets. ISI handwritten character database, CMATERDB 3.1.1 and BanglaLekha-Isolated datasets. Those datasets have Bangla characters and digits.

ISI handwritten character database has total 23392 numerical images where 19392 is for train and 4000 for the test. The image has different pixel size and images edge look smooth.

The CMATERDB 3.1.1 has the total 6000-digit image. Each class contains 600 BMP format 3 channel image. Most of the images are noise-free 32×32 pixels and almost correct labeling and no overwriting characters images. But the image edge looks blockier (Fig. 2).

BanglaLekha-Isolated datasets contain 19748 image each class contains average 1974 image each in 1 channel grayscale png format. Each image has different pixel size. Some image has incorrect labeling and some image is overwritten with others.

But the image ages are looking smooth. Then also join these two datasets and make a new dataset with 25748 images.

3.2 Dataset Preparation

For ISI handwritten character database and CMATERDB 3.1.1, we first converted it to one channel grayscale to reduce computational expense and invert the color. Black for background and white for digit. For BanglaLekha-Isolated, first fixed some incorrect labeling and delete some incorrect image. This dataset already containing inverted images. Than resized all images to 28 × 28 pixel.

Then converted both dataset's 28 × 28-pixel image into a 784 + (1 label) D matrix and store all the image pixels into a CSV file to reduce hard disk read and fast computation. Also, perform a Minmax (1) normalization to reduce the effect of illumination differences. Moreover, the CNN converge faster on [0…0.1] data than on [0…0.255].

$$Z_i = \frac{X_i - \text{minmum}(X)}{\text{maximum}(X) - \text{minmum}(X)} \tag{1}$$

Then convert the 10 labels into one hot encoding.

[1, 0, 0, 0, 0, 0, 0, 0, 0, 0] = ০

[0, 1, 0, 0, 0, 0, 0, 0, 0, 0] = ১

[0, 0, 1, 0, 0, 0, 0, 0, 0, 0] = ২

[0, 0, 0, 1, 0, 0, 0, 0, 0, 0] = ৩

[0, 0, 0, 0, 1, 0, 0, 0, 0, 0] = ৪

[0, 0, 0, 0, 0, 1, 0, 0, 0, 0] = ৫

[0, 0, 0, 0, 0, 0, 1, 0, 0, 0] = ৬

[0, 0, 0, 0, 0, 0, 0, 1, 0, 0] = ৭

[0, 0, 0, 0, 0, 0, 0, 0, 1, 0] = ৮

[0, 0, 0, 0, 0, 0, 0, 0, 0, 1] = ৯

Then convert the 784 D into 28 × 28 image matrix.

(a) ISI (b) CMATERDB (c) BanglaLekha

Fig. 2 Example image of dataset

3.3 Proposed Model

Algorithm 1:

```
ADAM (Learning Rate)
For 30 iterations in all batch do:
    Convolution 1 (Filter, Kernel Size, Stride, Padding, Activation)
    Convolution 2 (Filter, Kernel Size, Stride, Padding, Activation)
    MaxPool 1 (Pool Size)
    Dropout (Rate)
    Convolution 3 (Filter, Kernel Size, Stride, Padding, Activation)
    Convolution 4 (Filter, Kernel Size, Stride, Padding, Activation)
    MaxPool 2 (Pool Size)
    Dropout (Rate)
    Dense (Units, Activation, Kernel initializer, Bias Initializer)
    Dropout (Rate)
    Dense (Units, Activation, Kernel initializer, Bias Initializer)
end for
```

Proposed Model in this paper use ADAM [7] optimizer with a learning rate of 0.001. The model has 9-layer CNN. For convolution 1 and 2, where filter size is 32, kernel size is (5 × 5), Stride is (1 × 1), "same" padding with ReLU (2) activation. Followed by a 5 × 5 max-pooling layer. Then used 25% dropout [8] to reduce overfitting.

$$ReLU(X) = MAX(0, X) \tag{2}$$

For convolution 3 and 4, the filter is 64, kernel size is (3 × 3), Stride is (1 × 1), "same" padding with ReLU activation. Followed by a 2 × 2 max-pooling layer. Then used 25% dropout [8]. Then flatten the layer and use a Dense layer with 256 units with ReLU activation and 50% dropout. At final output layer, used 10 units with SoftMax (3) activation. Figure 3 is showing the neural network architecture.

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} for\ j = 1, \ldots k \tag{3}$$

3.4 Optimizer and Learning Rate

The choice of optimization algorithm can make a sufficient change for the result in Deep Learning and computer vision work. The Adam paper says, "…many objective functions are composed of a sum of sub-functions evaluated at different subsamples of data; in this case, optimization can be made more efficient by taking gradient steps w.r.t. individual sub-functions…" [8]. The Adam optimization algorithm is an

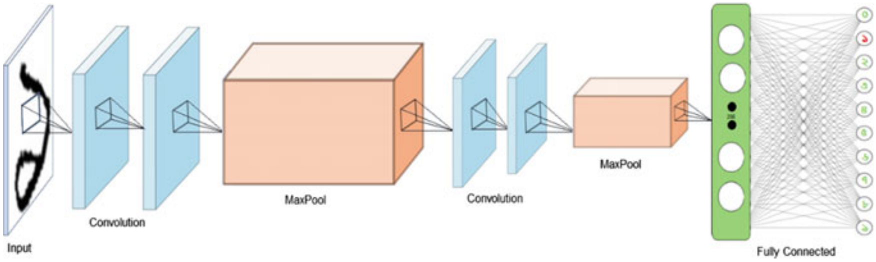

Fig. 3 Proposed CNN model

extension to stochastic gradient descent that recently adopting most of the computer vision and natural language processing application. The method computes individual adaptive learning rates for different parameters from estimates of first and second moments of the gradients.

Proposed method used ADAM (4) Optimizer [8] with learning rate = 0.001.

$$\nu_t = (1 - \beta_2) \sum_{i=1}^{t} \beta_2^{t-i} \cdot g_i^2 \tag{4}$$

When using a neural network to perform classification and prediction task. A recent study shows that cross-entropy function performs better than classification error and mean square error [9]. Cross-entropy error, the weight changes do not get smaller and smaller and so training is not s likely to stall out. Proposed method used categorical cross entropy (4) as loss function.

$$\mathrm{L_i} = -\sum_{\mathrm{j}} \mathrm{t_{i,j}} \log\left(\mathrm{p_{i,j}}\right) \tag{5}$$

To make the optimizer converge faster and closer to the global minimum of the loss function, using an automatic Learning Rate reduction method [10]. Learning rate is the step by which walks through the minimum loss. If higher learning rate use it will quickly converge and stuck in a local minimum instead of global minima. To keep the advantage of the fast computation time with a high Learning Rate, after each epoch model dynamically decreases the learning rate by monitoring the validation accuracy.

After some epochs manually checked the accuracy and decrease the learning rate to reach the global minima.

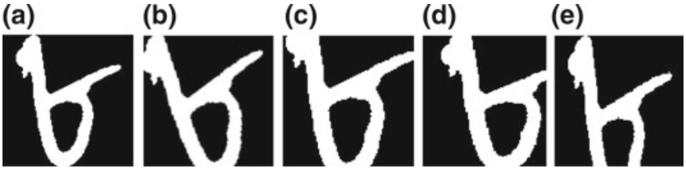

Fig. 4 **a** Original Image, **b** Rotated image, **c** Zoomed image, **d** Width shifted image, **e** Height shifted images

3.5 Data Augmentation

To avoid overfitting, artificially expand the handwritten dataset. This data transformation will create some variance that can occur when someone else writing the digits. For Data augmentation, several methods are chosen:

- Randomly shifting height and width 10% of the images.
- Randomly rotate our training image 10°
- Randomly 10% zoom the training image (Fig. 4).

3.6 Training the Model

Trained the model with different training set and validation set with a batch size of 86. In training time, the Learning Rate reduction formula will monitor the validation accuracy and reduce the learning rate. After 30 epochs, manually monitored the accuracy result and reduce the learning rate and train again 5–10 epochs and again manually set the learning rate couple of times.

The pre-train model and code for all these datasets can be found on this link https://github.com/shahariarrabby/Bangla_Digit_Recognition_CNN.

4 Evaluate the Model

The proposed model is applied to different datasets and get a pretty good result on train, test and validation sets which is shown below.

4.1 Train, Test, Validation Sets

For ISI handwritten character database used 15513 images as training set and 3879 images (20%) in the validation set. And the test set with ISI's 4000 images. For

BanglaLekha-Isolated Datasets used 15766 images as training set and 3942 images (20%) in the validation set. And make the test set with CMATERDB 3.1.1 datasets 6000 images. For CMATERDB 3.1.1 datasets used 4800 images as training set and 1200 image (20%) in the validation set. And make the test set with BanglaLekha-Isolated Dataset's 19708 images. For Mixed dataset used 80% from all of three dataset which is 39279 images on the training set, and 10% (4911) images on validation set and remaining 10% (4910) images in the test set.

4.2 Model Performance

For ISI handwritten character database, after 30 epoch model gets 99.35% accuracy on the training set and 99.74% accuracy on the validation set. Then tested the model with testing set and got 99.58% accuracy. Figure 5 shows the loss value and accuracy of the training set and the validation set and the Confusion Matrix (Table 1).

For BanglaLekha-Isolated dataset, after 50 epoch model gets 99.38% accuracy on the training set and 98.93% accuracy on the validation set. Then tested the model with CMATERdb 3.1.1 dataset and got 98.58% accuracy. Figure 6 shows the loss value and accuracy of the training set and the validation set and the Confusion Matrix.

For CMATERdb 3.1.1 dataset, after 30 epoch model gets 99.05% accuracy on the training set and 99.42% accuracy on the validation set. After the train tested the model with the BanglaLekha-Isolated dataset and got 92.65% accuracy. The test

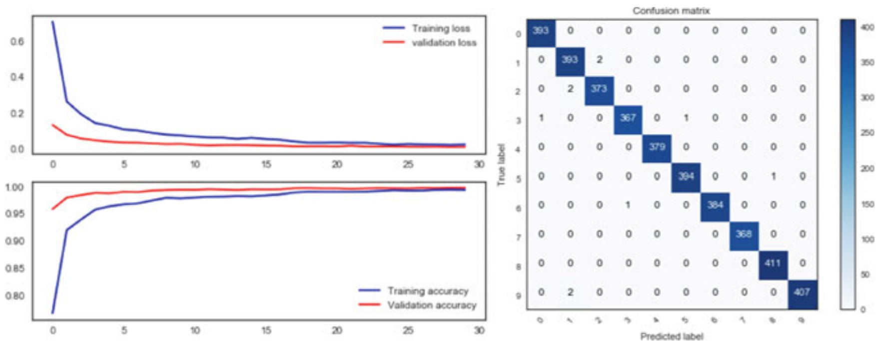

Fig. 5 ISI handwritten character database

Table 1 Result comparison in different dataset

Datasets	Tr. Loss	Val Loss	Tr. Acc.	Val Acc.	Test Acc.
ISI	0.0221	0.0090	99.35	99.74	99.58
BanglaLekha	0.0199	0.0510	99.38	98.93	98.58
CMATERdb	0.0339	0.0207	99.05	99.42	92.65
Mixed dataset	0.0256	0.0181	99.23	99.43	99.51

Table 2 Result comparison in different model

ISI numeral dataset		CMATERDB 3.1.1	
Work	Accuracy (%)	Work	Accuracy (%)
Bhatt and Chaudhuri	98.20	Haider et al. [11]	94
Wen and He [12]	96.91	Hassan et al. [13]	96.7
Nasir and Uddin [14]	96.80	Sarkhel el at. [15]	98.23
Akhnad et al. [16]	97.93	Basu et al. [17]	95.10
HybridHOGPPC8 [18]	99.03	HybridHOGPPC8 [18]	99.17
Proposed model	99.74	Proposed model	99.42

accuracy is too poor because CMATERdb 3.1.1 contains the noise-free image that fails to learn the noisy image of the BanglaLekha-Isolated dataset. Figure 7 shows the loss value and accuracy of the training set and the validation set and the Confusion Matrix.

For combined dataset, after 40 epoch models get 99.23% accuracy on the training set and 99.43% accuracy on validation set and 99.51% accuracy on the test set. Figure 8 shows the loss value and accuracy of the training set and the validation set and the Confusion Matrix.

4.3 Result Comparison

See Table 2.

5 Error Observations

From Fig. 9, for these twelve cases, it is clear that this model is doing great jobs in classifying digits. Some of those errors can also make by humans. Variance in handwriting model is getting confused with some digits, especially 5, 6 and 9, 1. Figs are sorted by error rate in ascending order.

6 Conclusion and Future Work

This paper presented a new CNN model that performs better classification accuracy in the different dataset for both train and validation set for lesser epochs and less computation time compared to the other CNN model. Also, the cross-validation from different distribution's data proposed model achieve a great result that makes it as a

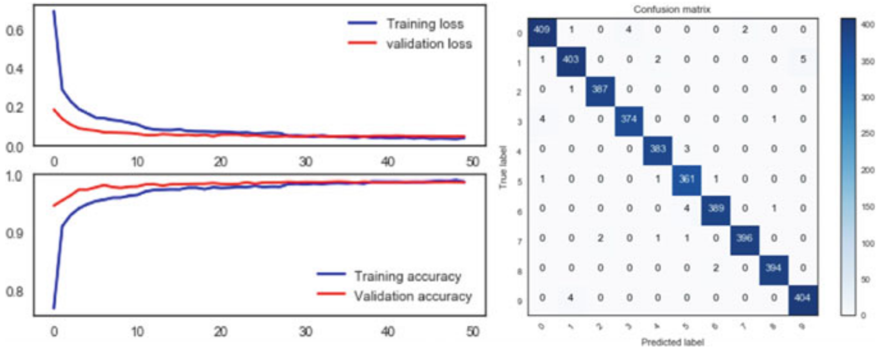

Fig. 6 BanglaLekha-Isolated evaluation

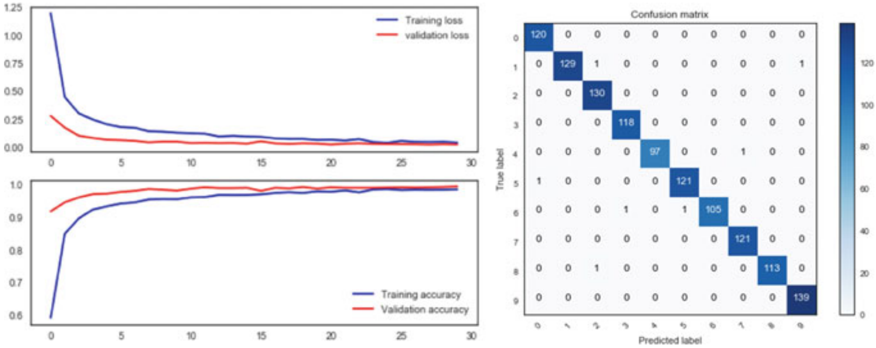

Fig. 7 CMATERdb evaluation

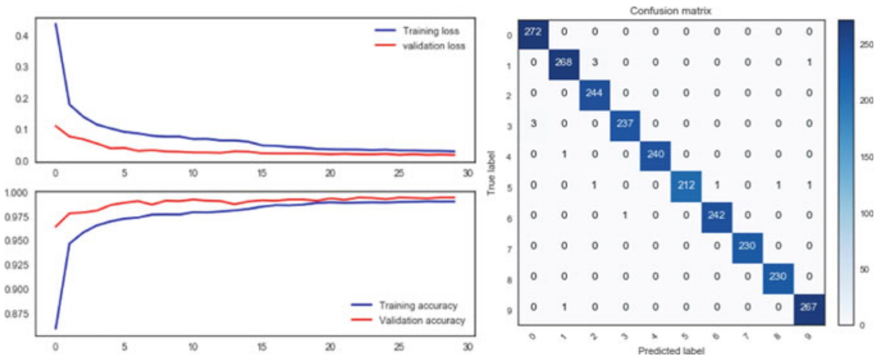

Fig. 8 Mixed datasets evaluation

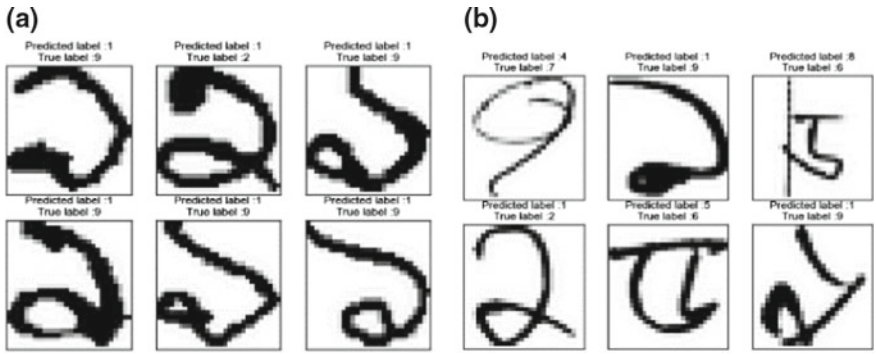

Fig. 9 **a** Validation set error, **b** Test set error

robust model that improve any other previous model. Also, this model got 99.55% accuracy on English MNIST handwritten dataset.

Sometimes proposed model confused to understand overwritten character and dataset contained some incorrect labeling images. Also, the model performed poorly if the train on noise-free data. In future work fixing dataset and overcoming the limitation of overwriting digit should fix.

References

1. Bhattacharya, U., Chaudhuri, B.: Handwritten numeral databases of indian scripts and multistage recognition of mixed numerals. IEEE Trans. Pattern Anal. Mach. Intell. **31**, 444–457 (2009). https://doi.org/10.1109/TPAMI.2008.88
2. Biswas, M., Islam, R., Gautam, K.S., Shopon, Md., Mohammed, N., Momen, S., Abedin, A.: BanglaLekha-Isolated: a multi-purpose comprehensive dataset of Handwritten Bangla Isolated characters. Data Brief. **12**, 103–107 (2017). https://doi.org/10.1016/j.dib.2017.03.035
3. Sarkar, R., Das, N., Basu, S., Kundu, M., Nasipuri, M., Basu, D.K.: Cmaterdb1: a database of unconstrained handwritten Bangla and Bangla-English mixed script document image. Int. J. Doc. Anal. Recogn. (IJDAR) **15**(1), 71–83 (2012)
4. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. In: Proceedings of the IEEE, vol. 86(11), pp. 2278–2324, Nov 1998
5. Pal, U., Chaudhuri, B.: Automatic Recognition of Unconstrained Off-Line Bangla Handwritten Numerals **1948**, 371–378 (2000). https://doi.org/10.1007/3-540-40063-x_49
6. Alom, Md.Z., Sidike, P., Tarek, M.T., Asari, V.: Handwritten Bangla Digit Recognition using Deep Learning (2017)
7. Kingma, D.P., Ba, J.: Adam: A Method for Stochastic Optimization, Dec 2014. arXiv:1412.6980
8. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: A simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. **15** 1929–1958 (2014)
9. Janocha, K., Czarnecki, W.M.: On Loss Functions for Deep Neural Networks in Classification (2017). arXiv:1702.05659
10. Schaul, T., Zhang, S., and LeCun, Y.: No More Pesky Learning Rates (2012). arXiv:1206.1106

11. Khan, H.A., Helal, A.A., Ahmed, K.I.: Handwritten Bangla digit recognition using sparse representation classifier. In: 2014 International Conference on Informatics, Electronics & Vision (ICIEV), pp. 1–6. IEEE (2014)
12. Wen, Y., He, L.: A classifier for Bangla handwritten numeral recognition. Expert Syst. Appl. **39**(1), 948–953 (2012)
13. Hassan, T., Khan, H.A.: Handwritten Bangla numeral recognition using local binary pattern. In: 2015 International Conference on Electrical Engineering and Information Communication Technology (ICEEICT), pp. 1–4. IEEE (2015)
14. Nasir, M.K., Uddin, M.S.: Handwritten Bangla numerals recognition for automated postal system. IOSR J. Comput. Eng. **8**(6), 43–48 (2013)
15. Sarkhel, R., Das, N., Saha, A.K., Nasipuri, M.: A multi-objective approach towards cost-effective isolated handwritten Bangla character and digit recognition. Pattern Recogn. **58**, 172–189 (2016)
16. Mahbubar Rahman, S.I.P.S. Md., Akhand, M.A.H., Rahman, M.M.H.: Bangla handwritten character recognition using convolutional neural network. I. J. Image Graph. Signal Process. (IJIGSP) **7**(3), 42–49 (2015)
17. Basu, S., Sarkar, R., Das, N., Kundu, M., Nasipuri, M., Basu, D.K.: Handwritten Bangla digit recognition using classifier combination through ds technique. In: Pattern Recognition and Machine Intelligence, pp. 236–241. Springer (2005)
18. Sharif, S.A.M., Nabeel, M., Mansoor, N., Momen, S.: A hybrid deep model with HOG features for Bangla handwritten numeral classification. In: 2016 9th International Conference on Electrical and Computer Engineering (ICECE), pp. 463–466 (2016)