Scientific Research Publishing

Journal of Computer and Communications, 2021, 9, 158-171
https://www.scirp.org/journal/jcc
ISSN Online: 2327-5227
ISSN Print: 2327-5219

Bangla Handwritten Character Recognition Using Extended Convolutional Neural Network

Tandra Rani Das[1], Sharad Hasan[1], Md. Rafsan Jani[1], Fahima Tabassum[2], Md. Imdadul Islam[1]

[1]Department of Computer Science & Engineering, Jahangirnagar University, Savar, Dhaka, Bangladesh
[2]Institute of Information Technology, Jahangirnagar University, Savar, Dhaka, Bangladesh
Email: tanudast95@gmail.com, hasansharad@gmail.com, rafsan@juniv.edu, fahima@juniv.edu, imdad@juniv.edu

How to cite this paper: Das, T.R., Hasan, S., Jani, Md.R., Tabassum, F. and Islam, Md.I. (2021) Bangla Handwritten Character Recognition Using Extended Convolutional Neural Network. *Journal of Computer and Communications*, **9**, 158-171.
https://doi.org/10.4236/jcc.2021.93012

Received: February 14, 2021
Accepted: March 28, 2021
Published: March 31, 2021

Abstract

The necessity of recognizing handwritten characters is increasing day by day because of its various applications. The objective of this paper is to provide a sophisticated, effective and efficient way to recognize and classify Bangla handwritten characters. Here an extended convolutional neural network (CNN) model has been proposed to recognize Bangla handwritten characters. Our CNN model is tested on "BanglalLekha-Isolated" dataset where there are 10 classes for digits, 11 classes for vowels and 39 classes for consonants. Our model shows accuracy of recognition as: 99.50% for Bangla digits, 93.18% for vowels, 90.00% for consonants and 92.25% for combined classes.

Keywords

Loss and Accuracy, Deep Neural Network, Image Classification, Noise Removal, CNN and HCR

1. Introduction

In this present era of digitization, the importance of handwritten character recognition is increasing and its application is prevalent in computer vision. With the improvement of computer technology, governments are trying to computerize their information repository which includes a large amount of handwritten scripts. The traditional method is manually retyping everything which requires huge manpower and a considerable amount of time. Handwritten character recognition has the ability to automate this process and this automation will help us in many areas e.g. postal code identification, passport and document verification, handwritten license plate recognition, automatic processing of bank cheques, converting hard-copy data to softcopy, ID card reading, signature verification etc. But the challenge is recognition of handwritten characters and it is far difficult compared to recognition of printed characters. The main reason behind this is the size and shape of handwritten characters which varies from person to person; moreover, writing style and inclination is also not identical.

Approximately 260 million people speak Bangla worldwide which ranks it as the seventh most spoken language in the world and second in the Indian sub-continent. Bangla handwritten characters have versatility in size, shape, stroke and writing style for different people. Therefore, a sophisticated model like CNN is necessary which is able to extract the features from images automatically without any explicit description.

Researchers have proposed some notable methods for recognizing Bangla handwritten characters. Majority of the works extract features explicitly from the character images using various methods and create feature vectors [1] [2] [3] [4] [5]. Then the feature vectors are fed into classifiers e.g. SVM, KNN etc. These explicit feature extraction methods face difficulties because of the complex shapes and similarity of the Bangla characters. Some similar characters differ from one another by just a single dot mark. This feature extraction task becomes more challenging because of different writing styles with distinct strokes and varied spacing of different individuals. Moreover, some of the works have considered similar characters as same class and reduced the number of classes [3] [4]. These factors affect the classification accuracy negatively. Some researchers have used convolutional neural network in their works [6] [7] [8]. But some of the works do not consider all the character classes. Besides, works that consider all character classes face less overall recognition accuracy in comparison to individual categories (vowels, consonants, numerals).

In this paper we intend to solve the problem of explicit feature extraction and propose a method that will automatically select and extract feature from the character images irrespective of the individual writing style and spacing. We have also considered 50 classes of basic letters (11 vowel classes and 49 consonant classes) and 10 classes of digits and tried to propose a method that will have better overall accuracy than existing methods.

CNN was first used in [9] for character recognition. It is different from traditional machine learning algorithm in the way that, for extracting features it does not require explicit specification. It can automatically extract the necessary features; that's why it is widely used for the classification tasks. Visual patterns are directly recognized from the image pixels in CNN. It is also translational and scale invariant. For these reasons we have used convolutional neural network to recognize Bangla handwritten characters.

The rest of the paper is organized as: Section 2 deals with some previous works in character recognition, Section 3 provides the system model where a modified CNN is used to recognize Bangla character, Section 4 provides results based on analysis of Section 3 and Section 5 concludes the entire work.

2. Literature Review

This section provides state-of-the-art pertinent to character recognition based on machine learning. Shopon *et al.* in [6] recognized Bangla handwritten digits employing unsupervised pre-training. They have used auto encoder with deep CNN. Their proposed architecture contains three convolutional layers and one max pooling layer. Authors got 99.5% accuracy for digit recognition which is the best accuracy so far. Purkayastha *et al.* in [7] proposed a convolutional deep model for recognizing Bangla handwritten characters including digits and other character classes. Specialty of their study was recognizing 20 mostly used compound characters. Their CNN model consists of two convolutional layers and two pooling layers followed by three densely connected layers.

According to Pak and Kim of [10], convolutional neural network is the most prominent deep learning approach for image processing and pattern recognition. They made a comparison among the successful and popular deep learning architectures namely AlexNet, VGG, GoogLeNet and ResNet. Vaidya *et al.* [11] developed a system based on CNN for handwritten English character recognition. Their system has two parts: an Android application for taking image of handwritten text to be recognized and a server in the backend having a trained neural network model.

Ryo, Karungaru and Terada [12] have proposed a smartphone based system for recognition and interpretation of road navigation signs. They provided techniques for character candidate domain extraction, one-character extraction and removal of noise from the image captured by the smartphone. The CNN was used for training the model and recognizing characters. Ashiquzzaman and Tushar in [13], proposed a deep neural network based algorithm for recognizing handwritten Arabic numerals. They showed that, their neural network model performed significant amount of improved accuracy in comparison with the existing methods for recognizing Arabic numerals.

Selmi *et al.* in [14] presented a deep learning based system, which can detect and recognize license plates. The system detects segments and recognizes characters. The authors have claimed their system to be successful in recognizing dynamic license plates in various complex conditions like low quality and distorted images and intense daylight and dark environment of night. The authors argued that their model requires fewer steps for image preprocessing. Joshi and Risodkar in [15] proposed a system, which can recognize Gujarati handwritten characters into the machine editable format. They have used deep neural network for recognizing the characters. Authors have used K-nearest neighbor, NNC classifier which are popular methodology in the field of OCR.

Tajane *et al.* in [16] analyzed three ways that are being used for coin recognition namely electromagnetic, mechanical and image processing. They proposed a new approach for recognizing and detecting Indian coins, which is based on deep learning model. Authors picked features like texture, color and shape for training the popular CNN architecture. Li *et al.* in [17] described recognition accuracy and inference performance as the key challenging factors in classifying images for any real time application. The authors proposed a solution to accelerate promising residual network (ResNet) framework in the inference application on FPGA (Field Programmable Gate Array) using OpenCL programming language. Authors have provided a convertor to transform any ResNet in CAFFE framework into FPGA platform.

3. System Model

This section provides basic construction of CNN, dataset used for analysis and extended CNN used for Bangla handwritten character recognition.

3.1. Convolutional Neural Network

The CNN is a deep learning approach which is able to extract different features of objects by applying learnable weights and biases which differentiate with objects. This methodology was proved successfully in the field of image classification of as explained in [18].

CNN model takes an image as input, processes the image and categorizes it in one of the predefined classes. A CNN model is first trained with a large amount of images of different categories. In this phase a general model of each category is built. Then, in the testing phase images are tested against the general models of different categories and determined in which category an image belongs to.

For this training and testing, each input image is passed through a series of convolutional layers with different kernels. Convolutional layer also includes batch normalization, ReLU layer and max pooling layer. After convolutional layers there is fully connected layer and softmax layer for probability distribution. In convolution layer, features are extracted from the images by convoluting the input image with different kernels. During this operation a kernel scans the whole image from left to right and top to bottom and performs dot product between the pixel values of input image and the kernel. Different kernels are used for extracting different features. The result of this convolutional layer is one feature map for each of the features as extracted.

The ReLU layer is used for introducing non-linearity to the feature maps then the pooling layer is used to down sample the feature maps. This is used for dimensionality reduction which reduces the spatial size of the feature maps while keeping the important information and spatial relationship among pixels intact. Finally, the fully connected layer where the feature maps are flattened and converted into a vector to feed into a traditional feed forward network which uses backpropagation algorithm. Next, the softmax layer performs the probability distribution based on which an image is categorized.

CNN can extract high level features as well as low level features very effectively compared with the other techniques like support vector machine or Kth nearest neighbor. It is challenging for any visual recognition task to extract effective features from the images and become insensitive to the variance of local features.

The CNN becomes insensitive to the variance of local features by applying replicative feature detector in case of extracting high level features. It has achieved outstanding result in recognizing handwritten characters in different languages hence CNN is used for recognizing Bangla handwritten characters in this paper.

3.2. Dataset

For our study of Bangla handwritten character recognition we have used BanglaLekha-Isolated dataset which is the largest isolated Bangla character dataset. This dataset contains 50 classes of basic letters (11 vowel classes and 49 consonant classes), 10 classes of digits and 24 classes are for compound letters. From the dataset, digit, vowel and consonant classes are taken for the study which sums up to total 60 classes of characters. The character classes that we have picked for recognition from the BangLalekha-Isolated are given in **Tables 1-3**.

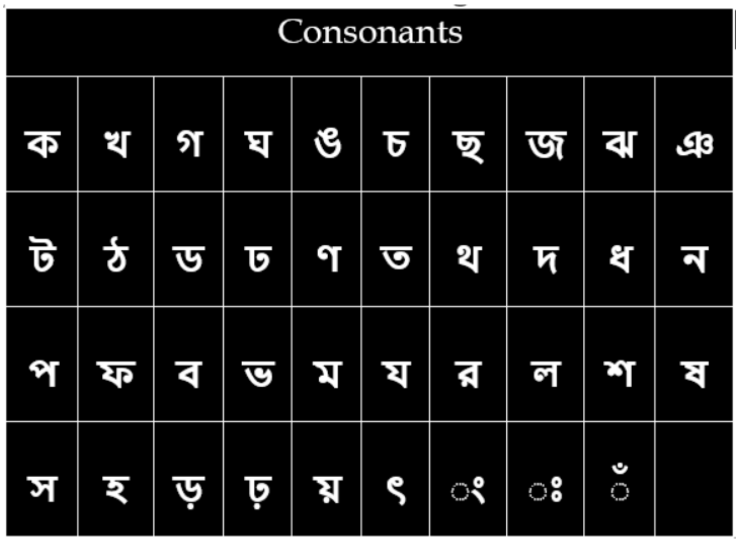
Table 1. Character classes of Bangla numerals.

Numerals				
০	১	২	৩	৪
৫	৬	৭	৮	৯

Table 2. Character classes of Bangla vowels.

Vowels					
অ	আ	ই	ঈ	উ	ঊ
ঋ	এ	ঐ	ও	ঔ	

Table 3. Character classes of Bangla consonants.



Consonants									
ক	খ	গ	ঘ	ঙ	চ	ছ	জ	ঝ	ঞ
ট	ঠ	ড	ঢ	ণ	ত	থ	দ	ধ	ন
প	ফ	ব	ভ	ম	য	র	ল	শ	ষ
স	হ	ড়	ঢ়	য়	ৎ	ং	ঃ	ঁ	

3.3. Proposed System Architecture

In this paper, we have proposed a deep convolutional neural network framework for Bangla handwritten character recognition. The model used for this purpose is shown in Figure 1. It has one image input layer, three convolutional layers and three fully connected layers. The image input layer takes Bangla hand written character images of dimension 32 × 32 × 1.

For every model, preprocessing the inputs is necessary to give the images a common form before feeding to any classifier. The images of BanglaLekha-Isolated are in png format. We have converted the images into tif format. The images found in the BanglaLekha-Isolated were of different size therefore, the most important task of preprocessing was making the images of equal size. All of the images are converted into 32 × 32 pixels and to reduce the computational time we preferred white letters on black background. Then the inputs are fed to the model.

In the 1st convolutional layer, input images are padded with zero padding of size 1. Then 8 kernels each having the dimension 3 × 3 × 1 have been applied to extract eight different features. Each kernel performs convolution operation on the entire image, which results in one activation map. If the kernel has m rows and n columns, then the formula of convolution operation is:

$$Z(x,y)=\sum_{j=1}^{m}\sum_{i=1}^{n}W(i,j)I(x-i)(y-j) \tag{1}$$

where, I = input matrix,

W = kernel matrix,

Z = output matrix.

Stacking the activation maps, we get a 32 × 32 × 8 feature map. The dimension of feature map is:

$$D=\frac{N+2P-F}{S}+1 \tag{2}$$

where, N = Dimension of input image,

P = Padding,

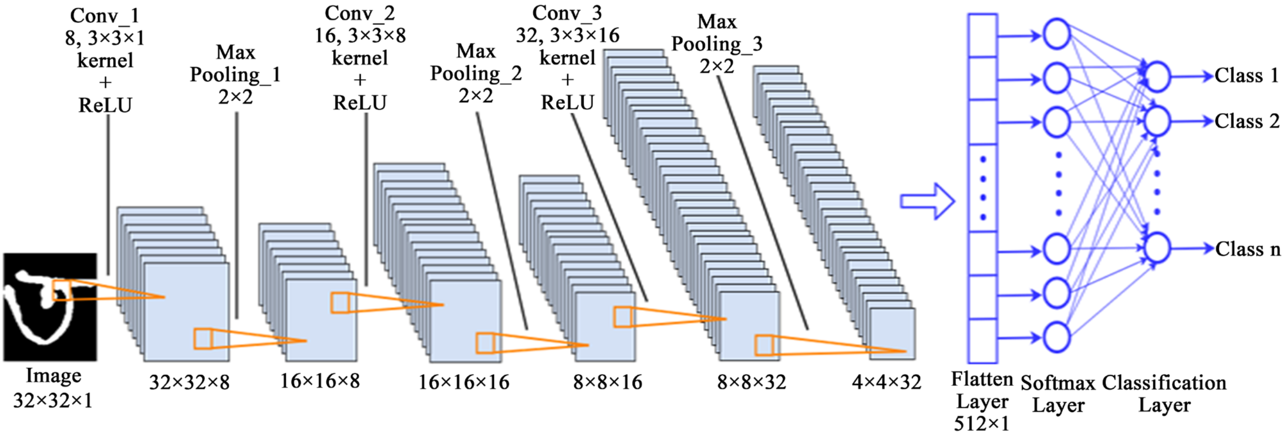


Figure 1. Architecture of the proposed CNN.

F = Dimension of filter,

S = Stride.

This feature map is then passed through a batch normalization layer where it is normalized across mini batches. The normalization layer makes the training process faster and insensitive to the network initialization.

Next, our feature map is passed through the ReLU layer. The purpose of this layer is to include non-linearity. The convolution operation may create negative values in the feature map. We ensured positive values using ReLU activation function as,

$$f(x)=\begin{cases} x, & x \ge 0 \\ 0, & x < 0 \end{cases} \tag{3}$$

where, x denotes value of a pixel.

After that, our model employs a max-pooling layer with kernel size of 2 × 2. This kernel scans across the whole image with stride size 2 and returns the maximum pixel value from its covered regions. If the kernel size is $k \times k$ then it will cover a region M with dimension $k \times k$ of the feature map. Then max-polling will be done using the following formula:

$$V = \max_{i=1,\cdots,k;\, j=1,\cdots,k} M(i, j) \tag{4}$$

where, V = maximum value of the $k \times k$ region,

M = $k \times k$ region of the feature map.

The dimension of the max-pooling layer output is:

$$D_x = \frac{N-F}{S}+1 \tag{5}$$

where, N = Dimension of input to pooling layer,

F = Dimension of filter,

S = Stride.

The Pooling layer reduces height and width of the activation map. The convolutional layer, batch normalization layer, ReLU layer and max-pooling layer forms the 1st layer of our model. There are two more such layers in our proposed model.

The resulting features of 1st layer of our model are then fed to the 2nd convolutional layer with padding 1. In this convolutional layer, we employed 16 kernels each of dimension 3 × 3 × 8, which gives 16 feature maps each having dimension of 16 × 16 × 1. This set of feature maps are stacked up and treated as input of dimension 16 × 16 × 16 to the next layer. After that, our model has batch normalization layer and ReLU layer followed by 2 × 2 max-pooling layer with stride size 2. This 2nd max-pooling layer outputs a feature map of dimension of 8 × 8 × 16. This feature map is then fed into 3rd convolutional layer with padding 1 like before. In this layer, 32 kernels each of dimension 3 × 3 × 16 are used for convolution, which results in 32 feature maps each of 8 × 8 × 1. Next, the 8 × 8 × 32 feature map is fed through the batch normalization layer, ReLU layer and max-pooling layer. As a result, we get a feature map with dimension of 4 × 4 ×

32. This 3rd max-pooling layer extracts those features, which will be used for recognizing the true classes of the characters.

Now, the feature map is flattened as a column vector of 512 × 1 and fed to the fully connected (FC) layer. Next layer of our model is softmax layer, where the probability of each of the predefined classes of the characters is calculated. The equation of the softmax layer is:

$$p(x_i) = \frac{e^{x_i}}{\sum_j e^{x_i}} \quad (6)$$

where, x_i refers to each element of logits vector,
$p(x_i)$ refers to the probability of x_i,
j is the number of elements in the logits vector.

Finally, a classification layer follows the softmax layer. The task of classification layer is to specify the class of a character based on the results obtained from the softmax layer. This completes a single epoch of the training process. After each epoch the loss is calculated which is used to update the parameters of each of the layers using back propagation. After several epochs, the model is trained enough to distinguish and recognize the classes of the hand written characters. The flowchart of the aforementioned model is shown in Figure 2 and the parameter chart is depicted in Table 4.

4. Result and Discussion

Because of the versatility of Bangla handwritten characters in shape and size, it is quite a complex task to recognize them in comparison to other languages. Our purpose was to classify and recognize Bangla digits, vowels, consonants separately and finally recognizing the combined classes using the same classifier. All the Bangla characters are taken from "Bangla-Lekha Isolated" dataset. Some of Bangla handwritten digits, vowels and consonants are taken randomly before

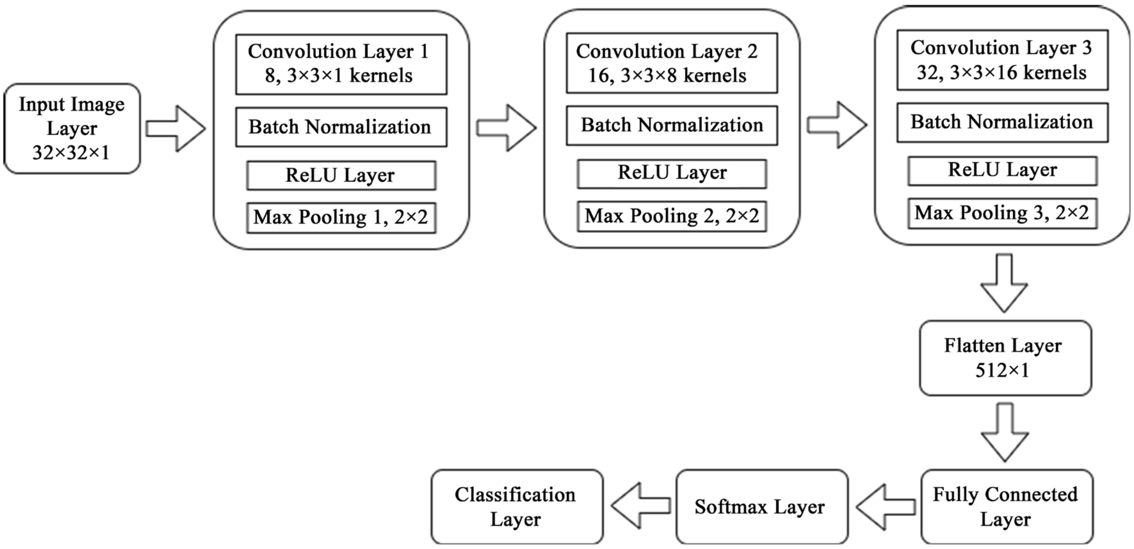
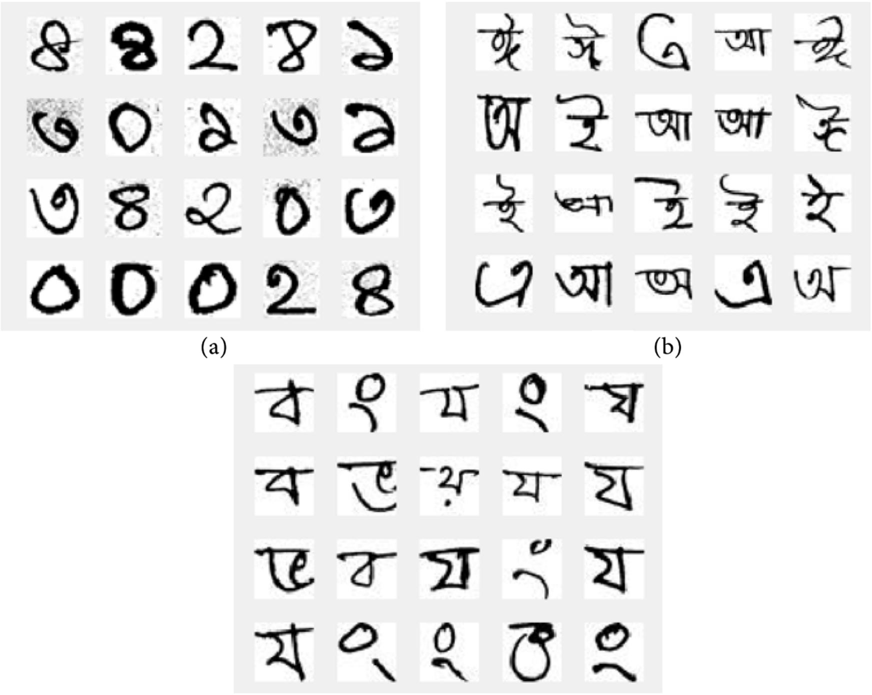


Figure 2. Flowchart of Bangla handwritten characters recognition model.

Table 4. Parameter list of the model.

Layers	Formula	Activation Shape	Activation Size	Number of Parameters
INPUT		(32, 32, 1)	1024	0
CONV1 (f = 3, s = 1)	$Z_l = I_{l-1} * W_l$	(32, 32, 8)	8192	80
POOL1	$V = \max_{i=1,\cdots,k;j=1,\cdots,k} M(i,j)$	(16, 16, 8)	2048	0
CONV2 (f = 3, s = 1)	$Z_l = I_{l-1} * W_l$	(16, 16, 16)	4096	1168
POOL2	$V = \max_{i=1,\cdots,k;j=1,\cdots,k} M(i,j)$	(8, 8, 16)	1024	0
CONV3 (f = 3, s = 1)	$Z_l = I_{l-1} * W_l$	(8, 8, 32)	2048	4640
POOL3	$V = \max_{i=1,\cdots,k;j=1,\cdots,k} M(i,j)$	(4, 4, 32)	512	0
FC4		(120, 1)	120	61,560
SOFTMAX	$p(x_i) = \frac{e^{x_i}}{\sum_j e^{x_i}}$	(60, 1)	60	7260

Here, Z_l: pre activation output of layer l; I_l: activation of layer l; W_l: convolution kernel of layer l; *discrete convolution operation.



(a)

(b)

(c)

Figure 3. Few randomly selected Bangla handwritten characters. (a) Bangla digit; (b) Bangla vowels; (c) Bangla consonants.

preprocessing are shown in Figures 3(a)-(c) respectively, provided images are greyscale with size of 32 × 32. After preprocessing, we have taken 80% of the samples from each class to construct training set and 20% samples from each class are taken for testing purpose. We have performed our experiment on Intel Core i3 processor, 8 GB of RAM and Windows 10 environment. 10 epochs have been applied to optimize our result. On each epoch, our model makes a cycle through the entire training dataset. This is performed to have better generalization as the data is presented to the model in different patterns in each epoch. All the experimental results are collected based on the custom "Bangla-Lekha Isolated" dataset about which we have described earlier. The size of our dataset is large. For this reason, we have performed batch wise training in our network. We have kept the batch size as a variable parameter and user have the liberty to change the batch size. At first we have trained and tested our model separately for digits, vowels and consonants to determine the category wise recognition performance. Then the model is trained with all the classes at once for combined recognition performance.

We have used "Accuracy" as our performance evaluation metric. Accuracy is the number of characters our model has classified correctly. Accuracy can be calculated by the following formula:

$$\text{Accuracy} = \frac{\text{Tp} + \text{Tn}}{\text{Tp} + \text{Fp} + \text{Tn} + \text{Fn}} \tag{7}$$

where, Tp (True positive): model classifies a positive sample as positive,
Tn (True negative): model classifies a negative sample as negative,
Fp (False positive): model misclassifies a negative sample as positive,
Fn (False negative): model misclassifies a positive sample as negative.

Running the proposed CNN model on Bangla handwritten digits, we got training loss of 0.0108 and validation loss of 0.0153. Our obtained training accuracy is 99.87% and the validation accuracy is 99.50% after 10 epochs. The variation of loss and accuracy of both training and validation are shown in Figure 4(a) and Figure 4(b). In case of Bangla handwritten vowels, we got training loss of 0.0968 and validation loss of 0.2948. Our obtained training accuracy is 97.39% and the validation accuracy is 93.18% after 10 epochs. The profile of loss and

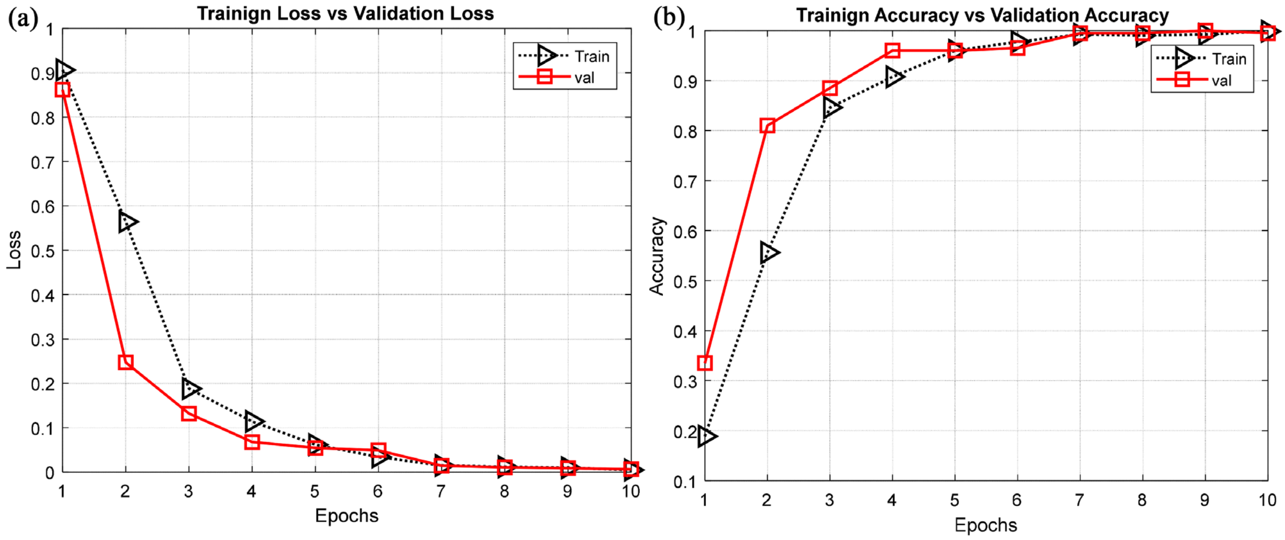


Figure 4. Profile of training and validation of loss and accuracy for Bangla handwritten digits. (a) Variation of loss with epochs; (b) Variation of accuraacy with epochs.

accuracy with 10 epochs are shown in Figure 5(a) and Figure 4(b) respectively. Next, for Bangla handwritten consonants, we got training and validation loss of 0.0039 and 0.3738 respectively. The training accuracy is found as 99.97% and validation accuracy is 90.00% after 10 epochs. The variation of loss and accuracy for consonants are shown in Figure 6(a) and Figure 6(b).

Finally, we run the proposed model on combined Bangla hand written character set (digits, vowels and consonants are mixed together) and we got training loss of 0.0344 and validation loss of 0.3204. Our obtained training accuracy is 99.08% and the validation accuracy is 92.25% after 10 epochs as shown in Figure 7(a) and Figure 7(b).

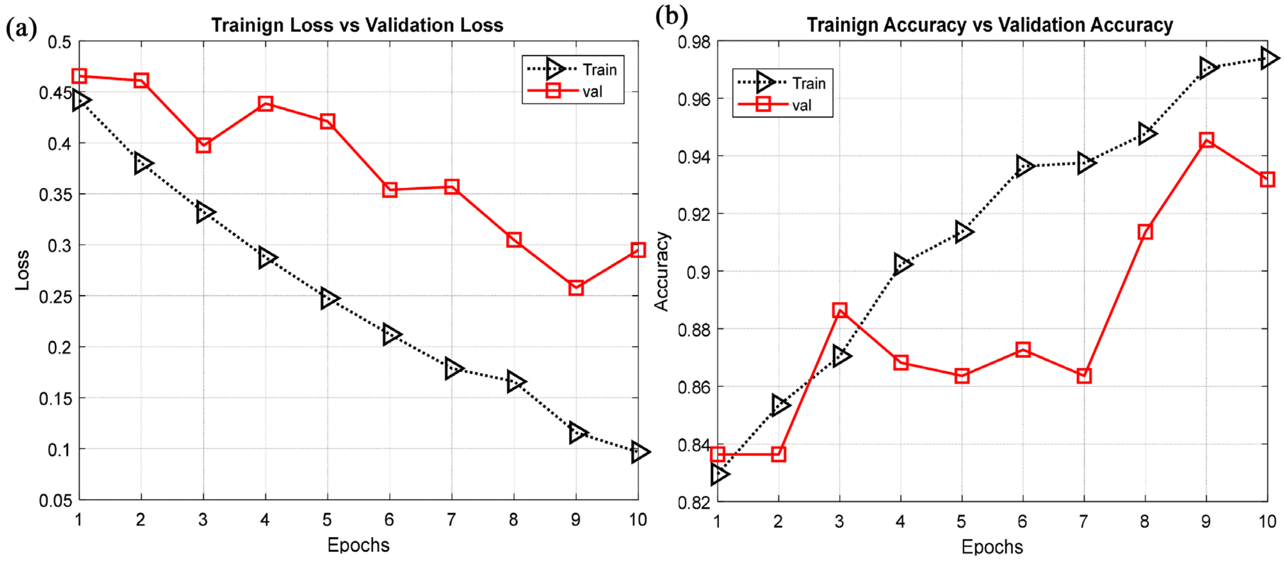


Figure 5. Profile of training and validation of loss and accuracy for Bangla handwritten vowels. (a) Variation of loss with epochs; (b) Variation of accuracy with epochs.

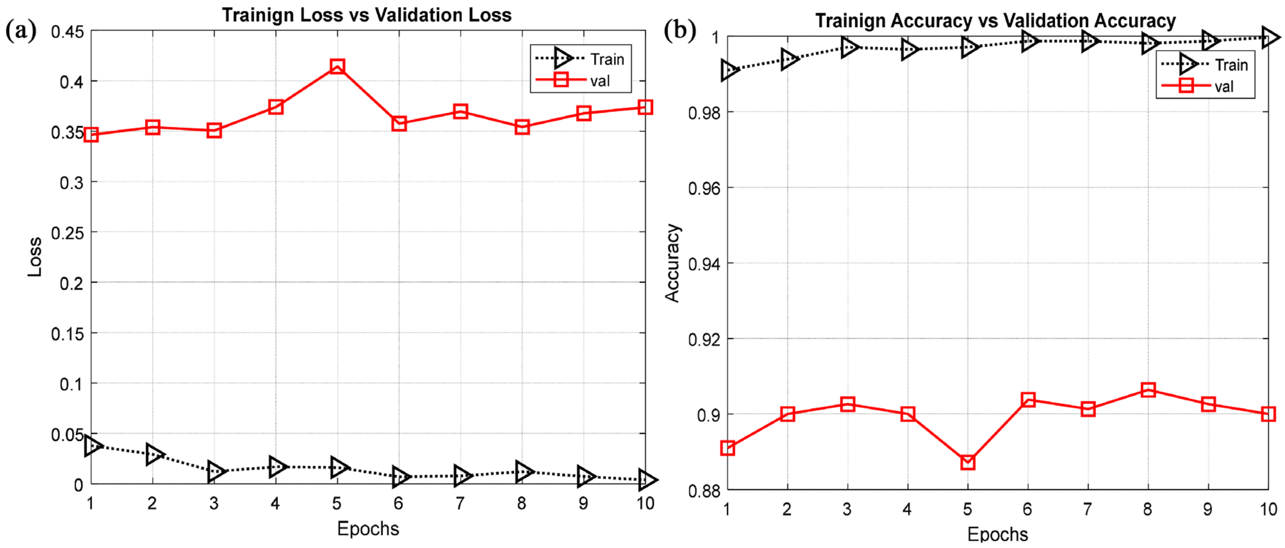


Figure 6. Profile of training and validation of loss and accuracy for Bangla handwritten consonants. (a) Variation of loss with epochs; (b) Variation of accuracy with epochs.

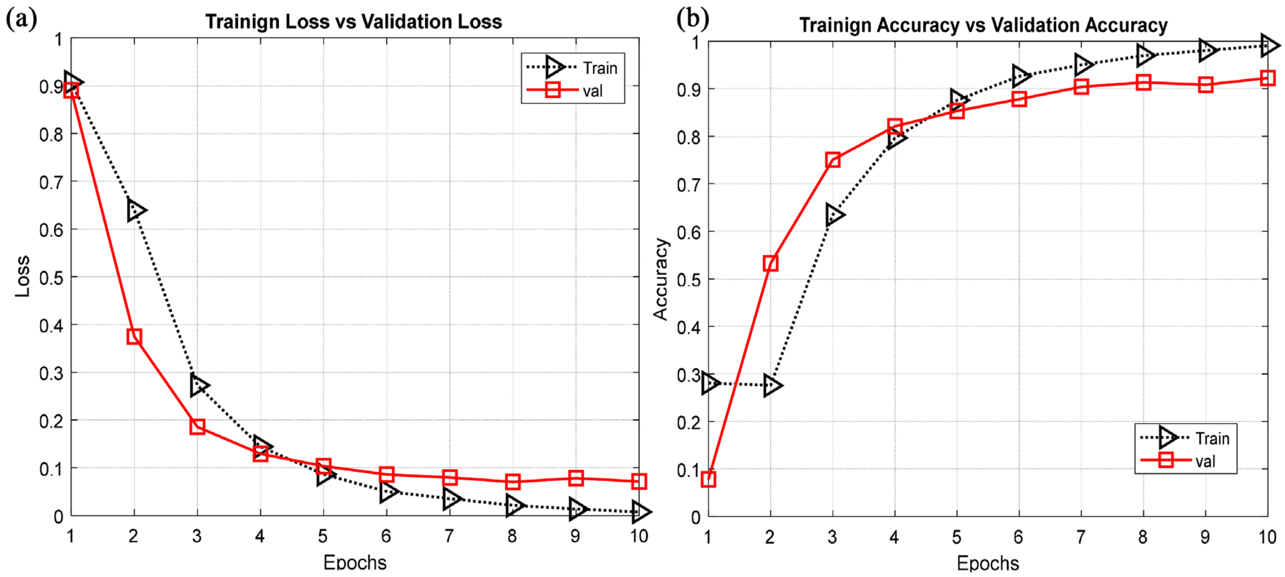


Figure 7. Profile of training and validation of loss and accuracy combined class. (a) Variation of loss with epochs; (b) Variation of accuracy with epochs.

Table 5. Summary of obtained result.

Category	No. of classes	Training Loss	Validation Loss	Training Accuracy	Validation Accuracy
Digits	10	0.0108	0.0153	99.87%	99.50%
Vowels	11	0.0968	0.2948	97.39%	93.18%
Consonants	39	0.0039	0.3738	99.97%	90.00%
Combined	60	0.0344	0.3204	99.08%	92.25%

The entire result of this section is accumulated in Table 5 for visualization at a glance.

Validation accuracy provides an idea about the performance of the model—how the model will perform in predicting the class of a character which is unseen to the model. Training accuracy tells us about how accurately the model has been trained with the training set. From Table 5 we can see, validation accuracy of our model for Bangla digit is 99.5%, that means there is 99.5% probability that our model will correctly classify an unseen digit. The probability of correctly predicting an unseen vowel and consonant is 93.18% and 90.00% respectively. For combined class, this probability is 92.25% which is higher than some state of the art methods. The percentages of training accuracy for different category shows us that, our model has been trained very accurately with the training set. Another noticeable point of Table 5 is that, the training accuracy and the validation accuracy are very close which proved that our model is not overfitted. It fits the training data very well and also generalizes the training which makes it able to make accurate prediction for unseen data.

5. Conclusion

The paper shows an effective utilization of proposed version of CNN model to classify or recognize Bangla handwritten characters and provides better result compared to some previous methods like MLP and SVM. In this paper, we ignored the compound handwritten symbols, which will be included in future with some modification of the proposed CNN. Still we have scopes to extend our work in future in the fields like: face detection, facial expression identification, fingerprint and other biometric item recognition, iris recognition, vehicle detection from both still images and videos.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Bashar, M.R., Hasan, M.A.F.M.R., Hossain, M.A. and Das, D. (2004) Handwritten Bangla Numerical Digit Recognition Using Histogram Technique. *Asian Journal of Information Technology*, **3**, 611-615.

[2] Pal, U., Chaudhuri, C.B.B. and Belaid, A. (2006) A System for BangIa Handwritten Numeral Recognition. *TETE Journal of Research*, *Institution of Electronics and Telecommunication Engineers*, **52**, 27-34. https://doi.org/10.1080/03772063.2006.11416437

[3] Bhowmik, T.K., Ghanty, P., Roy, A. and Parui, S.K. (2009) SVM-Based Hierarchical Architectures for Handwritten Bangla Character Recognition. *International Journal on Document Analysis and Recognition*, **12**, 97-108. https://doi.org/10.1007/s10032-009-0084-x

[4] Basu, S., Das, N., Sarkar, R., Kundu, M., Nasipuri, M. and Basu, D.K. (2009) A Hierarchical Approach to Recognition of Handwritten Bangla Characters. *Pattern Recognition*, **42**, 1467-1484. https://doi.org/10.1016/j.patcog.2009.01.008

[5] Bhattacharya, U., Shridhar, M., Parui, S.K., Sen, P.K. and Chaudhuri, B.B. (2012) Offline Recognition of Handwritten Bangla Characters: An Efficient Two-Stage Approach. *Pattern Analysis and Applications*, **15**, 445-458. https://doi.org/10.1007/s10044-012-0278-6

[6] Shopon, M., Mohammed, N. and Abedin, M.A. (2016) Bangla Handwritten Digit Recognition Using Autoencoder and Deep Convolutional Neural Network. *International Workshop on Computational Intelligence* (*IWCI*), Dhaka, 12-13 December 2016, 64-68. https://doi.org/10.1109/IWCI.2016.7860340

[7] Purkaystha, B., Datta, T. and Islam, M.S. (2017) Bengali Handwritten Character Recognition Using Deep Convolutional Neural Network. 20*th International Conference of Computer and Information Technology* (*ICCIT*), Dhaka, 22-24 December 2017, 1-5. https://doi.org/10.1109/ICCITECHN.2017.8281853

[8] Chowdhury, R.R., Hossain, M.S., ul Islam, R., Andersson, K. and Hossain, S. (2019) Bangla Handwritten Character Recognition Using Convolutional Neural Network with Data Augmentation. 2019 *Joint* 8*th International Conference on Informatics*, *Electronics & Vision* (*ICIEV*) *and* 2019 3*rd International Conference on Imaging*, *Vision & Pattern Recognition* (*icIVPR*), Spokane, 30 May-2 June 2019, 318-323. https://doi.org/10.1109/ICIEV.2019.8858545

[9] LeCun, Y., Boser, B.E., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W.E. and Jackel, L.D. (1990) Handwritten Digit Recognition with a Backpropagation

Network. In: *Advances in Neural Information Processing Systems*, Morgan Kaufmann, San Mateo, Vol. 2, 96-404.

[10] Pak, M. and Kim, S. (2017) A Review of Deep Learning in Image Recognition. 4*th International Conference on Computer Applications and Information Processing Technology* (*CAIPT*), Kuta Bali, 8-10 August 2017, 1-3. https://doi.org/10.1109/CAIPT.2017.8320684

[11] Vaidya, R., Trivedi, D., Satra, S. and Pimpale, P.M. (2018) Handwritten Character Recognition Using Deep-Learning. *Second International Conference on Inventive Communication and Computational Technologies* (*ICICCT*), Coimbatore, 20-21 April 2018, 772-775. https://doi.org/10.1109/ICICCT.2018.8473291

[12] Ryo, M., Karungaru, S. and Terada, K. (2017) Character Recognition in Road Signs Using a Smartphone. 6*th IIAI International Congress on Advanced Applied Informatics* (*IIAI-AAI*), Hamamatsu, 9-13 July 2017, 1039-1044. https://doi.org/10.1109/IIAI-AAI.2017.200

[13] Ashiquzzaman, A. and Tushar, A.K. (2017) Handwritten Arabic Numeral Recognition Using Deep Learning Neural Networks. *IEEE International Conference on Imaging, Vision & Pattern Recognition* (*icIVPR*), Dhaka, 13-14 February 2017, 1-4. https://doi.org/10.1109/ICIVPR.2017.7890866

[14] Selmi, Z., Ben Halima, M. and Alimi, A.M. (2017) Deep Learning System for Automatic License Plate Detection and Recognition. 14*th IAPR International Conference on Document Analysis and Recognition* (*ICDAR*), Kyoto, 9-15 November 2017, 1132-1138. https://doi.org/10.1109/ICDAR.2017.187

[15] Joshi, D.S. and Risodkar, Y.R. (2018) Deep Learning Based Gujarati Handwritten Character Recognition. *International Conference on Advances in Communication and Computing Technology* (*ICACCT*), Sangamner, 8-9 February 2018, 563-566. https://doi.org/10.1109/ICACCT.2018.8529410

[16] Tajane, A.U., Patil, J.M., Shahane, A.S., Dhulekar, P.A., Gandhe, S.T. and Phade, G.M. (2018) Deep Learning Based Indian Currency Coin Recognition. *International Conference on Advances in Communication and Computing Technology* (*ICACCT*), Sangamner, 8-9 February 2018, 130-134. https://doi.org/10.1109/ICACCT.2018.8529467

[17] Li, X., Ding, L., Wang, L. and Cao, F. (2017) FPGA Accelerates Deep Residual Learning for Image Recognition. *IEEE* 2*nd Information Technology, Networking, Electronic and Automation Control Conference* (*ITNEC*), Chengdu, 5-17 December 2017, 837-840. https://doi.org/10.1109/ITNEC.2017.8284852

[18] Tabassum, F., Imdadul Islam, M., Tasin Khan, R. and Amin, M.R. (2020) Human Face Recognition with Combination of DWT and Machine Learning. *Journal of King Saud University—Computer and Information Sciences*. https://doi.org/10.1016/j.jksuci.2020.02.002