International Journal of
Innovative Research

International Journal of Innovative Research, **4(1)**:7–12, 2019
ISSN 2520-5919 (online)
www.irsbd.org

RESEARCH PAPER

Bangla Handwritten Digit Recognition Using Convolution Neural Network

Afroza Islam[1], Surma Jahan Api[1], Md. Kamal Hossain Chowdhury[1*]

[1]Department of Computer Science and Engineering, Comilla University, Bangladesh

ARTICLE HISTORY

Received: February18, 2019
Revised : March 5, 2019
Accepted: March 28, 2019
Published: April 30, 2019

*Corresponding author:
kamal.cou@gmail.com

ABSTRACT

Bangla handwritten digit recognition is always a big challenge due to its variation of shape, size, and writing style. Due to the economic and educational values of accurate handwritten recognition, researchers are becoming more thoughtful about it. Several works have been already done on the Bangla Handwritten Digit Recognition. Therefore, in this paper, we investigated the dynamic search process to recognize a digit. The unbiased dataset, NumtaDB is used for Bangla digit recognition. This paper states the development and implementation of a lightweight Convolution Neural Network (CNN) model for classifying Bangla Handwriting Digits. We have systematically evaluated the performance of our method on this image database NumtaDB. And finally, from experiments, we have achieved a 99% accuracy using some of the proposed methods.

Key words: NumtaDB, CNN, Bangla Handwritten Digit Recognition

Introduction

Handwritten Bangla Character Recognition (HBCR) is of academic and commercial interests. Current algorithms are already excelled in learning to recognize handwritten characters. The enormous varieties of handwriting styles by different writers make the handwritten character recognition more challenging than the printed forms because written character style differ from one to another with different aspects such as size and shape (Alom et al. 2017). Bangla, ranked fifth in the world, is one of the most spoken languages. Bangla is also a prominent language with a rich heritage; February 21st is announced as the International Mother Language Day by UNESCO to respect the language martyrs for the language in Bangladesh in 1952. Additionally, Bangla is the first language of Bangladesh and the second most popular language in India. About 220 million people use Bangla as their speaking and writing purpose daily (Chaudhuri & Pal, 1998). Bangla has 10 digits and 50 characters in vowel and consonant in Bangla language. Moreover, Bangla has many similar shaped characters, in some cases, a single dot or mark causes the difference of a character from its similar one. Furthermore, the Bangla language also contains some special characters in some special cases. That makes difficult to achieve better performance with simple technique as well as hinders the development of the HBCR system. In this work, we have investigated HBCR on Bangla digits. At first, we have processed all kinds of augmentation of this dataset. Then our processed images are classified by a deep convolutional neural network.

Background study and related works

Deep convolutional neural network models show a higher performance rate for recognizing and computing high-level deep features in the case of the digit recognition problem. So the main objective of our research is to take handwritten Bangla digits as input, process the digits, train the convolutional neural network model to recognize the digits. Many research and work have been done for solving the digit recognition problem (Wang et al, 2012). The first Convolutional Neural Network (CNN) architecture known as Le-net was also used by some researchers for Bangla digit recognition (Bhattacharya & Chaudhuri, 2009). The Convolutional neural network was introduced for better-supervised learning and accuracy (Akhand and Ahmed, 2016). A notable early attempt in the area of character recognition research was made by Grimsdale in 1958. In this method, the input character pattern obtained by a flying spot scanner is described in terms of length and slope of straight-line segments and length and curvature of curved segments. Pal and Chaudhury (2000) have conducted some exploring works for the issue of recognizing handwritten Bangla numerals. Their proposed schemes are mainly based on the extracted features from a concept called the water reservoir. The reservoir is obtained by considering the accumulation of water poured from the top or the bottom of numerals. They deployed a system towards Indian postal automation. The achieved accuracies of the handwritten Bangla and English numeral classifier are 94% and 93%, respectively.

Data set

In our proposed approach, the unbiased and highly augmented NumtaDB dataset has been used which consists of more than 85,000 Bangla handwritten digit images (Alam et al. 2018). The dataset is a compilation of six datasets that were gathered from different sources and at different times. After rigorous checking, all digits were at least legible to one human being without any prior knowledge. In the NumtaDB dataset, the sources are labeled from 'a' to 'f'. The subsets are separated depending on the source of the data (training-1, testing-a, etc.). All the datasets have been partitioned into training and testing sets so that handwriting from the same subject not present in both. The dimension of the images is about 180×180 pixels.

Methodology

The following two major steps were proposed to classify NumtaDB handwritten digits. These steps were shown in the following figure 1 :

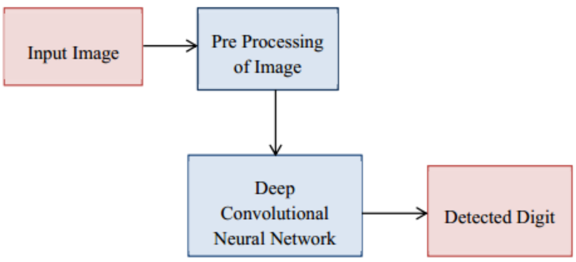

Fig. 1: Block diagram showing the steps of the proposed Bangla digit recognition system

At first stage, input image was collected, and then preprocessing was done and after that deep convolutional of neural network was done and at last digit recognition was taken place. The details of our two major steps were described in Section V and VI.

Image preprocessing

A. Image Resizing and Grayscaling

In the NumtaDB dataset, each image was 180× 180 pixels. So it was important to properly reduce the size of images to 32×32 pixels. To get a binary image, this RGB format image had to be converted to the grayscale format. So we converted all RGB images to grayscale images.

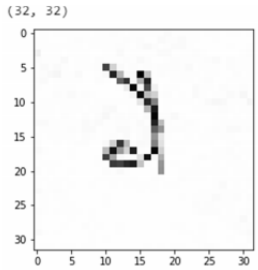

Fig. 2: The image from NumtaDB dataset is resized to 32x32 image

B. Image Interpolation

Interpolation was needed for image enlargement because the image can lose some information due to resizing. So we used inter-area interpolation after resizing images.

C. Removing Blur From Images

We used Gaussian blur to add blur at first and then subtract the blurred image from the original image. Then we added a weighted portion of the mask to get the de-blurred image (Gonzalez & Woods, 2007).

D. Image sharpening

Sharpness was a combination of two factors: resolution and acutance. Resolution is just the size, in pixels, of the image file. Acutance was a little more complicated. It was a subjective measure of the contrast at an edge. After that, by sharpening the apparent sharpness of an image is increased. There were many filters for sharpening images. In this paper, we used the Laplacian filter.

E. Removing noise from images

We removed salt and pepper noise from NumtaDB images by using salt and pepper noise.

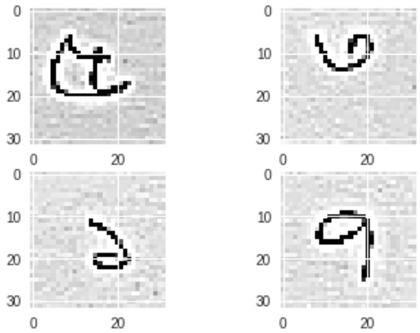

Fig 3: Image after Preprocessing

Then Otsu's Method had been applied to find the threshold value. The method chooses the threshold to minimize the intraclass variance of the thresholded black and white pixels. Figure 3 showed the image after preprocessing.

Deep learning and deep convolutional neural network

Machine learning offered a variety of techniques and models based on application and the size of data to process and solve the problems. Deep learning, a subset of Machine Learning, is generally more complex. So, thousands of images are needed to get reliable results.

High-performance Graphical Processing Units (GPU) needs to analyze all those images to ensure less time for analysis. Deep learning methods use a neural network, which is why deep earning models are often referred to as deep neural networks. Traditional neural networks only contain 2-3 hidden layers, while deep networks can have as many as possible. Convolutional Neural Networks (CNN or ConvNet) is one of the most popular types of deep neural networks. CNN convolves by learning features with input data, and it uses 2D convolutional layers, making this architecture well suited to processing 2D data, such as images. By assigning learnable weights and biases to various aspects/objects in the image and be able to differentiate one from the other. The less preprocessing is needed in a ConvNet compared to other classification algorithms. The architecture is illustrated in subsection A.

F. Model Architecture

The main aim of the Convolution Operation was to extract the high-level features such as edges, from the input image. ConvNets need not be limited to only one Convolutional Layer. Conventionally, the first ConvLayer was responsible for capturing the Low-Level features such as edges, color, gradient orientation, etc. With added layers, the architecture adapts to the High-Level features as well, giving us a network that had the wholesome understanding of images in the dataset similar to how human do. Our proposed architecture consists of 6 convolutional layers and 2 fully connected dense layers. The first two layers had 32 filters and each filter size is 5×5. The middle two layers had 128 filters and each filter size is 3×3. The last two layers had 256 filters and each filter size was 3×3. Rectified Linear unit (ReLu) (V.Nair, and Hinton, G. E 2010) was used as an activation function for all layers. Max Pooling and Average Pooling were available. The maximum value from the portion of the image covered by the Kernel was returned by max pooling, while the average of all the values from the portion of the image covered by the Kernel was returned by average pooling. Later one discarded the noisy activations altogether and also performed de-noising along with dimensionality reduction. So, architecture Max pooling layers and Batch normalization were used after every two layers. The pool size of max pooling layer was 2× 2. Among the two fully connected layers, the first one had 64 filters and the last one has 10 filters for the 10 digits. The last activation function was a softmax function for the classification. We used Adam optimizer to update weights. Figure 4 showed the design of our proposed deep CNN architecture. From input to output every configuration was marked properly.

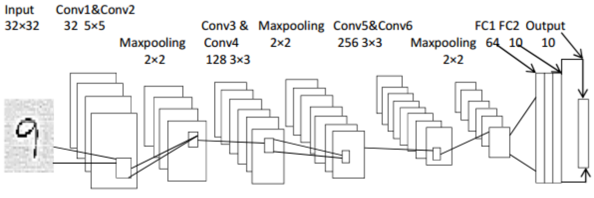

Fig 4: Deep CNN Architecture of the model

Table 1. Model summary of deep CNN architecture

Layer	Output Shape
Conv2D_1	(None, 32, 32, 32)
Conv2D_2	(None, 32, 32, 32)
Batch Normalization_1	(None, 32, 32, 32)
MaxPooling2D_1	(None, 16, 16, 32)
Conv2D_3	(None, 16, 16, 128)
Conv2D_4	(None, 16, 16, 128)
Batch Normalization_2	(None, 16, 16, 128)
MaxPooling2D_2	(None, 8, 8, 128)
Conv2D_5	(None, 8, 8, 256)
Conv2D_6	(None, 8, 8, 256)
Batch Normalization_3	(None, 8, 8, 256)
MaxPooling2D_3	(None, 4, 4, 256)
Flatten_1	(None, 4096)
Dense_1	(None, 64)
Activation_1	(None, 64)
Dropout_1	(None, 64)
Dense_2	(None, 10)
Activation_2	(None, 10)

Table 1 showed the whole model summary of our proposed deep CNN architecture.

BanglaLekha-Isolated Numerals was another independent dataset that was a combination of CMATERDB Numerals and ISI Numerals dataset for Bangla handwritten digits. But this dataset had only 19748 digit images and there were no highly augmented images like NumtaDB. It was mentioned in NumtaDB paper that the dataset 'e' of NumtaDB is BanglaLekha-Isolated Numerals. Table II showed the model summary of Deep CNN architecture for BanglaLekha Isolated Numerals.

Experiment

G. Experimental Environment

Our experimental environment was configured within Google Colaboratory. Entirely cloud based Colaboratory was a free Jupyter notebook environment which required no setup. Each colab session was well-equipped with a virtual machine which had 13GB of RAM and either a CPU, GPU or TPU processor. We had completed our experiment in GPU processor.

H. Training, Validation and Testing

Among 85000+ images, train and test split ratio of NumtaDB dataset was 85%-15% (Alam, Reasat and

Doha, 2018). In the experiment, we split the training data into training and validation keeping the split ratio of about 80%-20%.

Table 2. Model summary of CNN architecture

Layer	Output Shape
Conv2D_1	(None, 128, 128, 32)
Activation_1	(None, 128, 128, 32)
Conv2D_2	(None, 126, 126, 32)
Activation_2	(None, 126, 126, 32)
MaxPooling2D_1	(None, 63, 63, 32)
Dropout_1	(None, 63, 63, 32)
Conv2D_3	(None, 61, 61, 64)
Activation_3	(None, 61, 61, 64)
MaxPooling2D_2	(None, 30, 30, 64)
Dropout_2	(None, 30, 30, 64)
Flatten_1	(None, 57600)
Dense_1	(None, 64)
Activation_4	(None, 64)
Dropout_3	(None, 64)
Dense_2	(None, 10)
Activation_5	(None, 10)

Results and Discussion

We observed our experiment combining all of the datasets that contained augmented dataset 'a' and augmented dataset 'c'. These augmented datasets had such misclassified images that were highly augmented as it was also difficult for the human brain to recognize correctly. From our experiments, it was cleared that our approach cannot detect all kinds of augmented images, because we did not preprocess the rotated, shifted, zoomed, superimposition and occlusion images. We observed training accuracy from our experiments. After 30 epochs we get this result.

Table 3. Un-weighted average accuracy

Training Accuracy	Validation Accuracy
99.70%	98.44%

Table III showed training accuracy result for un-weighted average accuracy.

Here we also plot a graph for an image. From the training dataset, we could plot a graph for every image. The following Figure 5 showed a graph for an image from the training set.

After applying model among 1,360,810 programs, our trainable programs were 1,359,978 and non-trainable programs were 832.

Then we had got two different results for two different datasets. Table IV showed the result comparison between the NumtaDB, dataset and BanglaLekhaIsolated.

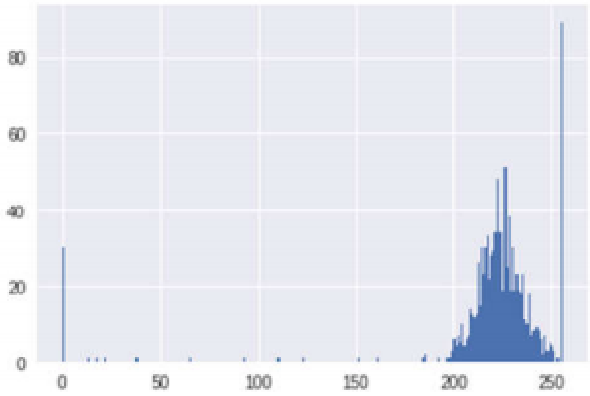

Fig 5: Graph For an Image

Table 4. Result comparison between numtaDB and Banglalekha-isolated numerals

Dataset	Number of Images	Accuracy
NumtaDB	85000+	99%
BanglaLekha Isolated Numerals	19748	90%

NumtaDB was a challenging dataset because it had more complex images which were hard to recognize. We had also completed our experiments for every dataset a,b,c,d,e. So, we observed training accuracy and testing accuracy for those datasets. After 100 epochs, we got these results. We observed these accuracies by applying scikit learn and keras. But we observed that, for keras model we achieved the best accuracy for every dataset and for scikit learn we did not get any best accuracy. Applying keras model our accuracy for every dataset was above 90%.

The following table showed the accuracy for every dataset.

Table. 5. Training accuracy for every separate dataset

Dataset	Accuracy
Dataset-A	92%
Dataset-B	79%
Dataset-C	94%
Dataset-D	91%
Dataset-E	95%

We observed from this table that, accuracy was different for every dataset. But for dataset E accuracy was 95% which was highest. That means that the number of misclassified digits in this dataset was low. The reason for low accuracy was misclassified digits from these datasets.

Some misclassified digit images were shown in the following Figure 6:

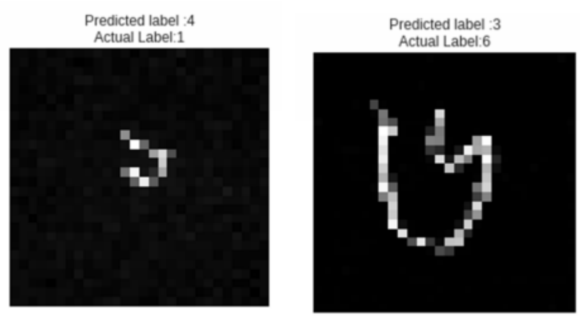

Fig 6: Misclassified Digits.

Some correct predicted digit images were shown in the Figure 7:

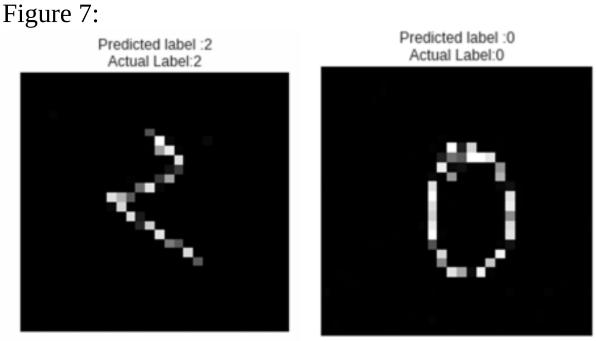

Fig 7: Correct Predicted Digits.

We had plotted a graph for the training process for every dataset. These graphs showed perfect results for training loss, training accuracy and also showed the result for validation loss and validation accuracy.

The following Figure 8 and Figure 9 showed the training loss and accuracy for dataset-c

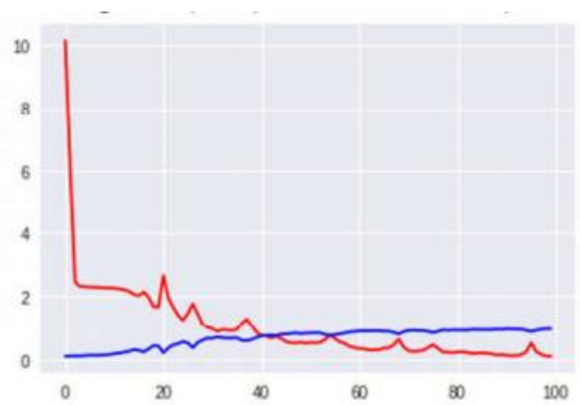

Fig 8: Graph of Training loss and accuracy

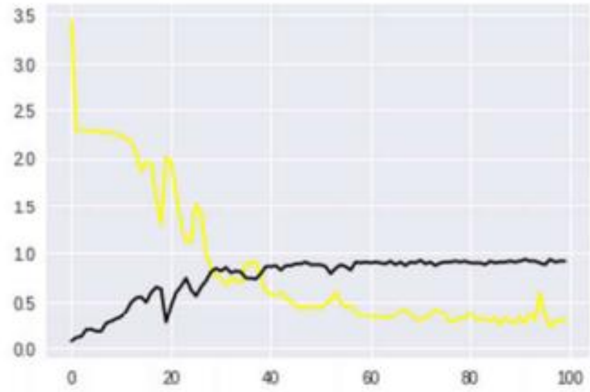

Fig 9: Graph of Validation loss and accuracy

Conclusion

In this paper, we proposed to use Convolution Neural Network (CNN) for Bangla handwritten digit recognition which shows a good performance to recognize most of the input digits. We had chosen CNN because CNN eliminates the need for manual feature extraction, so we did not need to identify features used to classify images. We had achieved 99% accuracy which was a good result for large and unbiased NumtaDB dataset comparing to BanglaLekha Isolated Numerals. In this article, keras (which uses tensorflow backend) library was used with CNN which was a recent development in the pattern recognition field and in Bangla handwritten digit recognition. In future, researchers can explore CNN for the bangla handwritten character recognition.

References

Akhand MAH, Ahmed M and Rahman MMH (2016) Convolutional neural network training with artificial pattern for bangla handwritten numeral recognition. 5th ICIEV. IEEE, PP. 625-630.

Alam S, Reasat T, Doha RM and Humayun AI (2018) NumtaDBAssenbled Bengali, arXiv:1806.02452 [cs.CV].

Alom MZ, Sidike P, Taha TM, Asari VK (2017) Handwritten Bangla Digit Recognition Using Deep Learning , arXiv:1705.02680.

Bhattacharya U and Chaudhuri BB (2009) “Handwritten Numeral databases of Indian scripts and Multistage recognition of mixed numerals” In IEEE transactions on pattern analysis and machine intelligence. IEEE, 31(3): 1-18.

Biswas M, Islam R, Shom GK et al. (2017) Banglalekha-isolated: A multi- purpose comprehensive dataset of handwritten bangla isolated characters,” Data in brief, vol. 12, pp. 103–107.

Chaudhuri BB, Pal U (1998) A complete printed Bangla OCR system, Pattern Recognition. IEEE, vol.31, pp. 531–549.

Gonzalez R and Woods RE (2007) Digital Image Processing, Third Edition, pp. 976.

Grimsdale L (1959) A system for the automatic recognition of patterns. Proceedings of the IEE-Part B: Electronic and Communication Engineering, 106(29):210- 221.

Kingma D and Ba J (2015) Adam: A method of stochastic optimization, arXiv:1412.6980 [cs.LG].

Nair V and Hinton GE (2010) Rectified linear units improve restricted boltzmann machines. Proceedings of the 27th International Conference on Machine Learning (ICML-10).

Otsu N (1979) A Threshold Selection Method from Gray-Level Histograms, IEEE Transactions on Systems, Man, and Cybernetics. IEEE, 9(1):62-66.

Pal U and Chaudhuri BB (2000) Automatic recognition of unconstrained offline Bangla and-written numerals. IEEE, PP. 790-794.

Tan T, Shi Y, Gao W (2000) Advances in Multimodal Interfaces-ICMI 2000, Lecture Notes in Computer Science, vol. 1948, Springer, Berlin, pp. 371–378.

Wang T, Wu DJ, Coates A and Andrew YN (2012) End-to end text recognition with convolutional neural networks. Proceedings of the 21st International Conference on Pattern Recognition (ICPR), IEEE pp. 3304- 3308.