A Comparative Analysis on Bangla Handwritten Digit Recognition with Data Augmentation and Non-Augmentation Process

Md. Abdullah Al Nasim[1], Refat E Ferdous[2], Mahim Anzum Haque Pantho[3] and Atiqul Islam Chowdhury[4]
Dept of Research & Development[1,3,4], Dept of CSE[2]
Pioneer Alpha[1,3,4], Ahsanullah University of Science and Technology[2]
Dhaka, Bangladesh
nasim.abdullah@ieee.org[1], refatcseaust16@gmail.com[2], mahimanzum@gmail.com[3] and achowdhury201036@mscse.uiu.ac.bd[4]

***Abstract*—Determination of Bangla handwritten digit is a momentous image classification task. Though object recognition technology is getting smarter day by day, still Bangla handwritten digit recognition remains inconclusive. Researchers are becoming more concerned about handwritten digit recognition for it's educational and advantageous importance. But it is a matter of trouble that the improvement in Bangla handwritten digit recognition is significantly less as compared to the other languages. To improve the performance of the Bangla handwritten digit recognition system, we have designed a model, in which all basic Bangla digits have been classified. Furthermore, we have also demonstrated Densenet121 architecture in our system. For recognizing Bangla handwriting digits, we proposed CNN (Convolution Neural Network) model. Our system has been experimented on the NumtaDB dataset for recognizing Bangla digit both with augmentation and non-augmentation.**

***Index Terms*—Bangla digit, NumtaDB, handwritten, CNN, augmentation, DenseNet121**

I. Introduction

In the area of computer vision, the handwritten character recognition problem is a classical one as it works on pattern matching [1]. Some important automation fields like – license plate recognition, digital postal service, optical image recognition (OCR), etc. are nowadays rapidly evolved by handwritten digit recognition development. Researchers nowadays highly trying to improve the efficiency and performance of handwritten digit recognition commercially or academically [2]. In some cases, some characters are written isolated (e.g, Japanese Thai, Laos). Also in many cases, there are some enfold characters and sometimes they are combined with each other. In the field of Natural Language Processing (NLP) this challenge is already recognized by many researchers. Recognizing handwritten characters is more complicated compared to the typed or printed form of characters and digits. Because of the huge handwritten style of different writers make a higher dimension of variance in writing style. Also, the similarities in distinct digits make the accurate classification of the digit recognition process a challenge.

Bangla is spoken by more than 200 million people all over the world. Hence, the recognition of Bangla digits and characters has a great significance. In Bangla language, there are 10 unique numbers. By classifying those 10 digits into 10 different classes, we can do the recognition task. But the research held on Bangla handwritten recognition is very few compared with other languages like English, Arabic, Chinese, etc. Researchers mainly focused on English digit recognition and developed the MNIST database.

In Bangla handwritten digit recognition process, researchers faces many difficulties for example lack of publicly available database which should be unbiased. To lighten this problem, we have combined numerals from several sources and united them together to form a dataset, NumtaDB. In our system, the most important and challenging task is about getting high appropriateness for large, unbiased, and highly augmented NumtaDB dataset. Our dataset was gathered from different sources containing Additive Gaussian noise, Gaussian blur, Coarse dropout, Contrast Normalization, Rotation, Salt-pepper noise, Shear, Super-impose, Translation, etc. We have processed those types of images and a Deep Convolutional Neural Network (DCNN) has been used to classified those processed images.

We organized our paper as follows: The related studies will be presented in section II. Section III will describe the dataset description of our research. Our proposed method is described in section IV. In section V, we will discuss the experimental analysis. Finally, the conclusion and future work will be briefed in section VI.

II. Related Study

In this section, we briefly describe some research works dependent on Bangla handwritten digit recognition. In the case of the digit recognition process, Deep Convolutional Neural Network (DCNN) methods have been shown more efficient results for recognizing deep high-level features. So, we have developed our system in such a way that, at first it will take Bangla handwritten digit as an input, next the input digit will be processed and at last instruct the Deep Convolutional Neural Network models to acknowledge the digits.

978-1-7281-9352-6/20/$31.00 ©2020 IEEE

There are many research works have been done on Bangla handwritten digit recognition using deep learning. And some biased dataset like CMATER-DB 3.1.1 [3] was used in most of the research works, since NumtaDB dataset was not obtainable at that point. Recently, some researchers have shown an improved exactness of 99.50% by conducting the auto-encoder and significant CNN structure for CMATERDB 3.1.1 dataset [4].

Handwritten digit recognition is a tough process and many research works have been done to solve this problem. LeNet, was the first Convolutional Neural Network (CNN) architecture and many researchers used this architecture for their digit identification research works. The architecture gave better output for the recognition system. So, nowadays researchers mainly use deep CNN methods for digit recognition works. Besides CNN, there are also some other classifiers for handwritten digit recognition, for example, Neural Network (NN), Support Vector Machine (SVM), etc. But comparing with other classifiers, CNN has shown better performance and exactness. Therefore, CNN is the most widely used architecture for digit recognition.

Besides, we studied some significant research works that have been done in English handwritten digit recognition. EMNIST and MNIST are the most prominent dataset for English handwritten character and digit recognition. Though there are not many pieces of research that have been done in Bangla handwritten digit recognition comparing with English digit recognition, there are some remarkable research works in Bangla digit recognition.

A Bangla handwritten character recognition system with the help of CNN, was developed by Rahman et al. [5] in 2015 which achieved 85.96% accuracy on 50 classes. A deep learning approach was applied to Bangla numeric digits using dataset CMATERDB 3.1.1 by Zahangir et al. [6] which achieved 98.8% accuracy on the 10 classes. A research on offline handwritten numeral recognition by Bhattacharya et al. [7] and the approached method used in this case was, an ensemble of MLPs (Multi-Layer Perceptrons) [8] [9] which were combined by Adaboost.

In a research, Chowdhury et al. [10] presented Bangla handwritten word recognition system using fuzzy logic. There was no specific dataset used as the data was collected as a time-ordered sequence of coordinates. Another online Bangla handwriting recognition system was developed by K.Royetal [11]. About 2500 numeric Bangla digits and 12500 Bangla characters were tested on that system. This method got 98.42% accuracy based on 10 numeric data classes and 91.13% accuracy based on 50 classes of character data. An offline or image-based Bengali handwritten character recognition system was implemented using Deep Convolutional Neural Network by Bishwajit et al. [12]. The process achieved 91.23% accuracy on classifying 50 alphabets. The Banglalekha isolated dataset was used for that research.

III. Dataset Description

A. Dataset

Digit recognition using a Deep Convolutional Neural Network is a difficult process. So, we need a large and publicly available handwritten dataset for our model. That's why we used the NumtaDB dataset, which has a large collection of different types of unclassified digit images. This NumtaDB dataset is consists of more than 85,000+ images. There are six different sources (codename: A, B, C, D, E, F) of NumtaDB test datasets and additionally, the test set A and test set C generates two augmented datasets [1]. For our experimental purpose, we divide the dataset into a training set and testing set. The testing set is mainly used to get the final results. Among 85000+ images training set and test set split ratio of the NumtaDB dataset is 85%-15% [13]. In our model, the training set contains 72,048 images and the testing set contains 17,628 images, which gave approximate 80%-20% train and test split ratio. Again, we divided the training sets into 5 folders and testing sets into 7 folders. Among 7 testing set folders, there are 2 folders that contain augmented images, and the rest of the 5 folders contain non-augmented images. Also, the images of the dataset are of different sizes and dimensions. For this reason, we try to bring all images to a fixed square size which is 64*64 pixels and augmented each image to add variety to our dataset.

B. Data Preprocessing

In data preprocessing, we have completed filtering the images and applied various augmentation process. The images in Fig. 1 will show the preprocessing process along with their operation name.

The working process of these operations are given shortly below:

- **Additive Gaussian Noise:** It added any noise that might be intrinsic to the information.
- **Gaussian Blur:** In our system Gaussian blur was used. At first, it adds blur and then subtracts blur from the real image. This is a linear filter applied to the image to smooth.
- **Coarse Dropout:** In the dataset, we black out some rectangular from the image. We remove some black rectangular from the dataset.
- **Contrast Normalization:** To control the contrast of the digit side or background.
- **Rotation:** We rotate an image when a digit is rotated in an image. We rotate an image from -45 degrees to +45 degrees.
- **Salt-Pepper Noise:** To sharp the image by reducing sudden disturbance.
- **Shear:** Calculate pixel distance of a digit from left and displace it to the left or right randomly.
- **Super-Impose:** In the dataset, we found some images where 2 digits in one image. We take only a weighted digit and replicate the effect of a weighted digit.

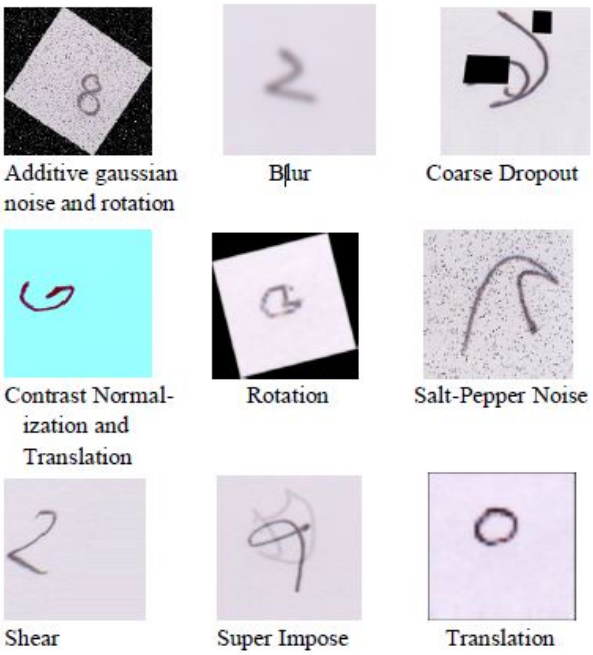

Fig. 1. Data preprocessing operations

- **Translation:** If a digit is a side of the image, we remove the blank area of the image so that it comes in the center. It is called translation.

IV. Our Methodology

In the last few years, deep learning has played an outstanding performance in the field of machine learning. Some most popular architectures, for example, Deep Belief Net(DBN), Stacked Auto Encoder(SAE), and CNN are part of Deep Convolutional Neural Network. The highly varying nonlinear can be represented by DNN models and these models are giving more appropriate results compared to the shallow learning approaches. DNN models are a combination of classification layers and feature extraction. And for this reason, DNN models are easier for learning. There are three types of DNN models and they are low, middle, and high. Among them, the low and middle-level abstract features from target images and the high level classify the extracted features. As a result, an end-to-end framework is formed with the integration of all necessary modules within a single network. Therefore, often we get better performances and accuracy from DNN models comparing to the other machine learning methods.

Among different types of neural networks, CNN has shown the best performances on image data classification. Also, it is providing image classification, object detection, segmentation, face recognition, and much more obviously.

In many user authentication applications, handwritten digit recognition plays a significant role. We use the NumtaDB dataset in our system, where there are 85000+ Bangla numeral pictures which are written manually. In the case of computers, the CNN image classifier takes an input image, processes it, and classifies it under certain categories. In our system, we proposed image preprocessing and DCNN process to classify handwritten NumtaDB digits. For image classification process, first of all, we collect the target images, process it (shown in Fig. 1) and next DCNN is done. After that digit recognition process takes place.

We used a manually bounded NumtaDB dataset in our proposed approach and classify it into two important terms. These terms are:

- Preprocessing of Input Images
- Deep Convolutional Neural Network section

Preprocessing steps were discussed in the previous section. In this, we will discuss our network model.

We used two Convolutional Neural Network with different depth and different image augmentation and compared their performances. Our first model is of depth 5 and we also down-sampled images to 26*26 and used augmentation of Keras library. We feed this input to our ConvNet which had 8, 5*5 filters, and then used maxpool of size 2*2. Then we used 16, 3*3 filters, and then again maxpool of size 2*2 with stride 2. Again we used 32, 3*3 filters followed by maxpool layer of 2*2 with stride 2. Then we flattened the output and connected them with 64 dense neurons and after that with 10 output layers which lead to the accuracy of 95.56%.

But our 2nd model is much deeper with 8 hidden layers and we used our own implemented 10 random augmentations to each image of size 64*64. We feed this input to our convNet which had 64, 5*5 filters 2 times, and then used maxpool of size 2*2. Then we used 128, 3*3 filters 2 times, and then again maxpool of size 2*2 with stride 2. Again we used 256, 3*3 filters 2 times followed by maxpool layer of 2*2 with

stride 2. Then we flattened the output that had shape 4096 and connected them with 64 dense neurons and after that with 10 output layers which lead to the accuracy of 99.486%. The model of our work is shown in Fig. 2.

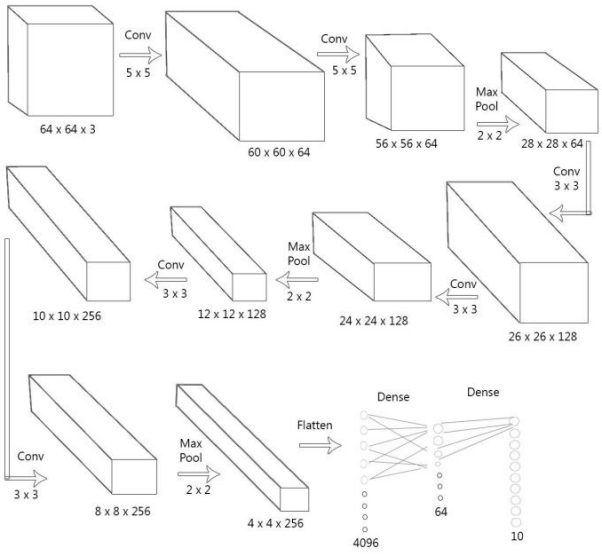

Fig. 2. Proposed model of our work

To develop a powerful deep learning method, data augmentation is a must need process. Data augmentation is a core process for improving the generalization ability of any system. Additionally, it helps to increase the probability of getting relevant data from the training dataset. Besides that, it is widely used to reduce overfitting. Some popular data augmentation techniques are flipping, cropping, rotation, translation, noise injection, etc.

Below we mentioned the data augmentation techniques that we have used for our system:

- Gaussian Blur
- Additive Gaussian Noise
- Affine (scale (2.0), translate_percent, rotate, shear)- rotate= (-45, 45), shear= (-8, 8)
- Coarse Dropout
- Contrast Normalization
- Pepper

To enhance the training sets, each image was introduced to a random augmentation with a 10% probability each image was gaussian blurred. We also used additive Gaussian noise, coarse dropout, salt and pepper noise, shear, rotation of 45 degrees, and multiply each pixel with a random number between 0.8 and 1.2. These augmentations have increased the size of the training sets and reduced the risk of overfitting. Therefore, we try to develop a data augmentation process, from which we get a large amount of suitable data for our training sets.

V. Experimental Analysis

In this section, we describe the experimental and performance analysis of our proposed method.

In our experiment, we assembled all of the datasets including augmented dataset 'A' and augmented dataset 'C'. We notice that these two augmented datasets comprise some uncategorized images that were extremely augmented and difficult to recognize. As we preprocess all kinds of images, so our model can recognize averagely every kind of image. The whole process took around 30 epochs.

To get a clear perception, we designed the confusion matrix (with and without augmented datasets). By analyzing the result and confusion matrix we found that we got the best accuracy for our system using the NumtaDB dataset. We have applied 5-fold cross-validation and for getting better accuracy. Different accuracy has been occupied depending on the learning rate. The learning rate is a hyper-parameter and during training, it helps to update our model's weight concerning loss gradient. It controls the model's learning speed. The value of the learning rate is between 0.0 and 1.0.

In this case, the training will slower and will take significantly more time as we have used low learning rate, which is really helpful for improving any model's performance. In our model, we work in such a way that the learning rate will change automatically after a specific epoch. It checks the accuracy again and again and if you get higher accuracy than before then it will update the accuracy otherwise the accuracy will remain the same as previously. Below we show different accuracy of augmented and non-augmented datasets along with their learning rate during training.

Accuracies of augmented datasets with their learning rates:

- Accuracy: 98.93%; Learning rate: 0.001
- Accuracy: 98.68%; Learning rate: 0.005
- Accuracy: 98.87%: Learning rate: 0.0005
- Accuracy: 99.04%; Learning rate: 0.0005
- Accuracy: 99.46%; Learning rate: 0.0001

Accuracies of non-augmented datasets with their learning rates:

- Accuracy: 94.98%; Learning rate: 0.001
- Accuracy: 94.89%; Learning rate: 0.005
- Accuracy: 95.19%: Learning rate: 0.0005
- Accuracy: 95.21%; Learning rate: 0.0005
- Accuracy: 96.72%; Learning rate: 0.0001

In augmented datasets, we achieve 99.46% of classification accuracy and for non-augmented datasets, we achieve 96.72% accuracy which is quite good for our recognition process. Fig. 3 shows some sample prediction of our digits.

We analyze every dataset and found that different dataset has different accuracy. All of these images are highly augmented and difficult to recognize. According to the prediction level, we can say that our model cannot predict the true value of all datasets in every attempts. But some datasets got 100% accuracy, which is a sign that our model is working perfectly. As machines are not 100% able to recognize digits or characters in every attempt and none of us are above mistakes,

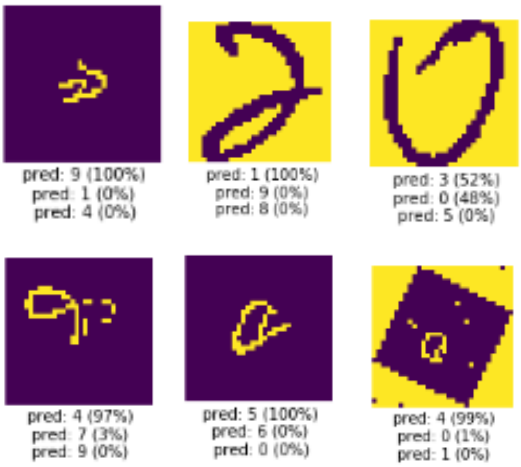

Fig. 3. Sample images of digit which were used in our model

so it can be said that the performance of our model in the digit recognition process is excellently effective. Below we show the confusion matrix of with and without augmented datasets in Fig. 4 and 5.

	0	1	2	3	4	5	6	7	8	9
0	[[1292	35	0	93	2	4	1	0	1	47]
1	[0	1384	3	3	5	0	1	0	0	76]
2	[0	1	1363	1	1	0	0	0	1	98]
3	[2	1	1	1472	2	0	5	0	1	6]
4	[0	0	0	0	1461	1	1	0	3	11]
5	[1	0	0	5	5	1395	10	0	0	8]
6	[0	0	0	6	1	4	1410	0	0	10]
7	[1	0	5	18	9	2	4	1354	4	33]
8	[0	0	1	4	0	0	4	0	1407	6]
9	[98	3	1	1	1	0	0	0	0	1299]]

Fig. 4. Confusion matrix (without augmentation)

	0	1	2	3	4	5	6	7	8	9
0	[[1392	26	0	5	0	3	0	0	0	0]
1	[0	1421	0	0	0	0	0	0	0	6]
2	[0	4	1422	0	0	0	1	0	1	7]
3	[2	2	0	1536	2	0	2	0	1	2]
4	[0	3	0	0	1508	0	1	0	0	2]
5	[0	0	1	2	0	1420	11	0	0	0]
6	[0	0	0	1	0	1	1440	0	0	0]
7	[0	0	2	0	2	5	2	1424	0	10]
8	[0	0	0	0	0	0	0	0	1416	0]
9	[0	6	0	0	0	0	0	0	0	1417]]

Fig. 5. Confusion matrix (with augmentation)

VI. Conclusion and Future Work

In Deep Learning approaches, CNN models are extremely significant. In our research, we have used DenseNet121 architecture for handwritten Bangla digit recognition. DenseNet architecture requires fewer parameters and it is free from training problems that occurred for the flow of information and gradients. We have investigated that our model got an excellent classification accuracy which is 99.46% in augmented datasets, and 96.72% accuracy for non-augmented datasets which was basically large and unbiased dataset. As we have used DenseNet architecture, so we did not mark out the features that were used to classify images. Not only the Deep Convolutional models but also the preprocessing operation of images plays a very significant role in the image recognition process and increase the performances of our model.

In the future, we will work with more augmented data and build a more advanced model that will able to recognize more augmented images. Also, we will observe the existing problems and try to develop a new architecture for getting a better solution for handwritten Bangla digit recognition.

References

[1] Ashadullah Shawon, Md Jamil-Ur Rahman, Firoz Mahmud, and MM Arefin Zaman. Bangla handwritten digit recognition using deep cnn for large and unbiased dataset. In *2018 International Conference on Bangla Speech and Language Processing (ICBSLP)*, pages 1–6. IEEE, 2018.

[2] Md Zahangir Alom, Paheding Sidike, Mahmudul Hasan, Tarek M Taha, and Vijayan K Asari. Handwritten bangla character recognition using the state-of-the-art deep convolutional neural networks. *Computational intelligence and neuroscience*, 2018, 2018.

[3] Tasnuva Hassan and Haider Adnan Khan. Handwritten bangla numeral recognition using local binary pattern. In *2015 International Conference on Electrical Engineering and Information Communication Technology (ICEEICT)*, pages 1–4. IEEE, 2015.

[4] MAH Akhand, Mahtab Ahmed, and MM Hafizur Rahman. Convolutional neural network training with artificial pattern for bangla handwritten numeral recognition. In *2016 5th International Conference on Informatics, Electronics and Vision (ICIEV)*, pages 625–630. IEEE, 2016.

[5] Md Mahbubar Rahman, MAH Akhand, Shahidul Islam, Pintu Chandra Shill, MM Hafizur Rahman, et al. Bangla handwritten character recognition using convolutional neural network. *International Journal of Image, Graphics and Signal Processing (IJIGSP)*, 7(8):42–49, 2015.

[6] Md Zahangir Alom, Paheding Sidike, Tarek M Taha, and Vijayan K Asari. Handwritten bangla digit recognition using deep learning. *arXiv preprint arXiv:1705.02680*, 2017.

[7] Tarun Jindal and Ujjwal Bhattacharya. Recognition of offline handwritten numerals using an ensemble of mlps combined by adaboost. In *Proceedings of the 4th International Workshop on Multilingual OCR*, pages 1–5, 2013.

[8] Ujjwal Bhattacharya, Malayappan Shridhar, Swapan K Parui, PK Sen, and BB Chaudhuri. Offline recognition of handwritten bangla characters: an efficient two-stage approach. *Pattern Analysis and Applications*, 15(4):445–458, 2012.

[9] DE Rumellhart. Learning internal representations by error propagation. *Parallel distributed processing*, 1:318–362, 1986.

[10] Kanchan Chowdhury, Lamia Alam, Shyla Sarmin, Safayet Arefin, and Mohammed Moshiul Hoque. A fuzzy features based online handwritten bangla word recognition framework. In *2015 18th International Conference on Computer and Information Technology (ICCIT)*, pages 484–489. IEEE, 2015.

[11] K Roy, N Sharma, T Pal, and U Pal. Online bangla handwriting recognition system. In *Advances In Pattern Recognition*, pages 117–122. World Scientific, 2007.

[12] Bishwajit Purkaystha, Tapos Datta, and Md Saiful Islam. Bengali handwritten character recognition using deep convolutional neural network. In *2017 20th International Conference of Computer and Information Technology (ICCIT)*, pages 1–5. IEEE, 2017.

[13] Samiul Alam, Tahsin Reasat, Rashed Mohammad Doha, and Ahmed Imtiaz Humayun. Numtadb-assembled bengali handwritten digits. *arXiv preprint arXiv:1806.02452*, 2018.