Project Report
on

Bengali Handwritten Digits Recognition using CNN

Author1: Md Ahsanul Haque
Author2: Amir Husen

Instructor: Dr. Olac Fuentes

Submitted to the faculty of the Department of Computer Science

Department of Computer Science
University of Texas at El Paso

Abstract

Bangla handwritten digit recognition is a convenient starting point for building an OCR in the Bengali language. Lack of large and unbiased dataset, Bangla digit recognition was not standardized previously. But in this project, a large and unbiased dataset known as NumtaDB is used for Bangla digit recognition. The challenges of the NumtaDB dataset are highly unprocessed and augmented images. So different kinds of preprocessing techniques are used for processing images and a vgg-style deep convolutional neural network (CNN) is used as the classification model in this project. The deep CNN model has shown an excellent performance with 98.5% validation accuracy. We train the same model again with image augmentation technique for random rotation of -20 to 20 degree and set a callback function with learning rate decays, then we got the validation accuracy of 99.33%.

1 INTRODUCTION

Bangla handwritten digit recognition is a classical problem in the field of computer vision. There are various kinds of practical application of this system such as OCR, postal code recognition, license plate recognition, bank checks recognition etc. Recognizing Bangla digit from documents is becoming more important. The unique number of Bangla digits are total 10. So the recognition task is to classify 10 different classes. The critical task of handwritten digit recognition is recognizing unique handwritten digits. Because every human has his own writing styles. But our contribution is for the more challenging task. The challenging task is about getting robust performance and high accuracy for large, unbiased, unprocessed, and highly augmented NumtaDB dataset [1]. The dataset is a combination of six datasets that were gathered from different sources and at different times containing blurring, noise, rotation, translation, shear, zooming, height/width shift, brightness, contrast, occlusions, and superimposition. We have not processed all kinds of augmentation of this dataset. We have processed blur and noisy images mainly. Then our processed images are classified by a deep convolutional neural network.

2 MOTIVATION

In delving into the realm of Bengali Handwritten Digits Recognition using Convolutional Neural Networks (CNN), our motivation is grounded in the profound impact this technology can have on various applications, from enhancing character recognition systems to facilitating seamless integration of Bengali language processing in digital platforms. The unique challenges posed by Bengali script necessitate a tailored approach, and leveraging the power of CNNs can unlock new possibilities in accurately deciphering handwritten digits. By contributing to this endeavor, we aim to bridge technological divides, empower local language processing, and pave the way for inclusive, efficient solutions in the realm of digit recognition for Bengali script. This venture resonates with my commitment to advancing technology that is culturally relevant and has the potential to improve accessibility and usability for diverse linguistic communities.

3 DATASET AND PROPOSED APPROACH

The approaches we take in this project are explained in the figure 1. First of all, we collected the dataset. Then, we preprocessed the data. After that, we splited into training set and validation set. We train the training data on a CNN model that we built using tensorflow keras. Then, we validate the model using validation set. We tried to improve the accuracy of our model by repeatedly changing the parameters, image augmentation techniques and reducing the learning rate. At the end, we analyze our model with previous models.

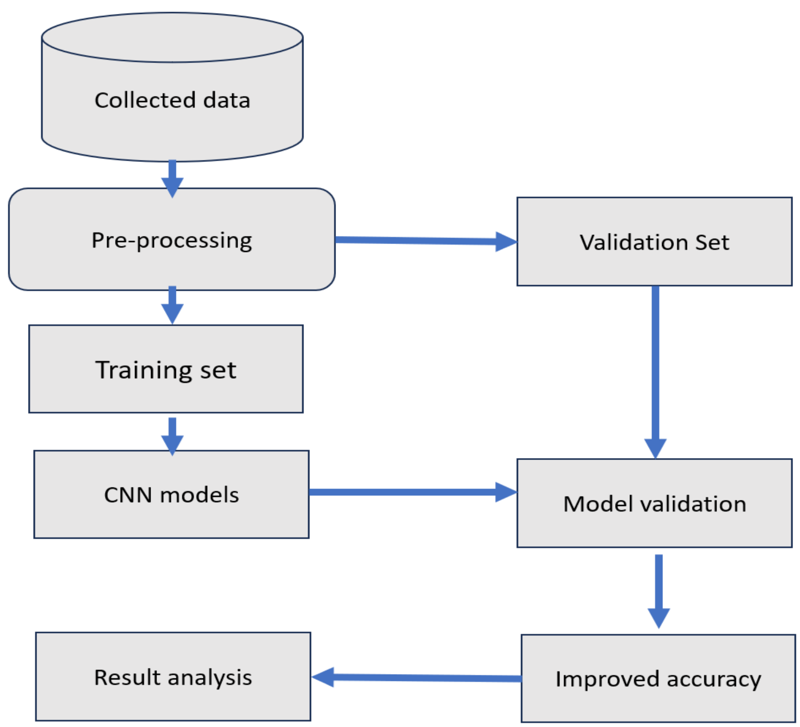

Figure 1: Methodology.

A DATASET

In our proposed approach, we give importance to the NumtaDB dataset which consists of 85,000+ Bangla handwritten digit images with 10 classes from 0 to 9. NumtaDB is a standard, unbiased (in terms of geographic location, age, and gender), large, unprocessed, and reviewed dataset. The NumtaDB dataset can verify our proposed approach performance perfectly and we hope that our proposed approach will get approximately the same accuracy for reallife handwritten digit recognition. The images of the NumtaDB dataset are real-world images without any preprocessing. The NumtaDB dataset is an assembled dataset from six different sources. According to NumtaDB, "The sources are labeled from 'a' to 'f'. The training and testing sets have separate subsets depending on the source of the data (training-a, testing-a, etc.). All the datasets have been partitioned into training and testing sets so that handwriting from the same subject/contributor is not present in both. Dataset-f had no corresponding metadata for contributors for which all of it was added to the testing set (testing-f)". Each image of the dataset is about 180×180 pixels. The sample images of the NumtaDB dataset are shown in Fig. 2.

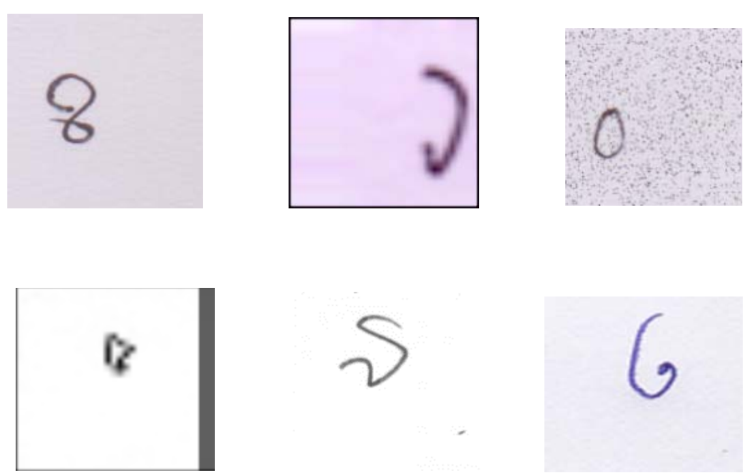

Figure 2: Bangla handwritten digit images from NumtaDB.

According to NumtaDB, "Two augmented datasets (augmented from test images of dataset 'a' and 'c') are appended to the testing set which consists of the following augmentations”

- Spatial Transformations: Rotation, Translation, Shear, Height/Width Shift, Channel Shift, Zoom
- Brightness, Contrast, Saturation, Hue shifts, Noise.
- Occlusions
- Superimposition (to simulate the effect of text being visible from the other side of a page).

So, NumtaDB dataset has made Bangla handwritten digit recognition more challenging by augmented images.

B PREPROCESSING OF IMAGES

Resizing and Grayscaling

The original size of NumtaDB images are 180×180 pixels which are too large for preprocessing efficiently. So we reduce the size of images to 32×32 pixels. We also convert all RGB images to GRAY scale images. The color channel is converted to 1 channel from 3 channel.

Interpolation

Images can lose much important information due to resizing. Inter-area interpolation is preferred method for image decimation. This method is resampling using pixel area relation. We use inter-area interpolation after resizing images.

Removing Blur from Images

We use Gaussian blur to add blur at first and then subtract the blurred image from the original image. Then we add a weighted portion of the mask to get de-blurred image.

Removing Noise from Images

We remove salt and pepper noise from NumtaDB images. We use the median filter to remove salt and pepper noise. After preprocessing, the images become clear, sharp and salt and pepper noiseless.

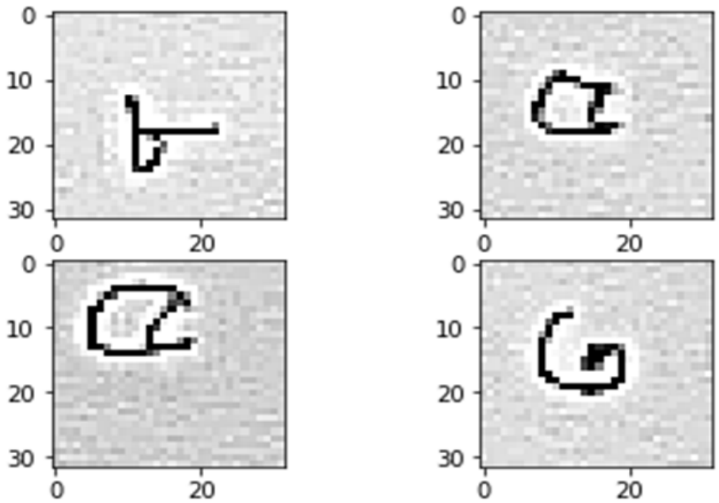

Figure 3: Preprocessed Images.

C DEEP CONVOLUTIONAL NEURAL NETWORK

Deep learning has been providing the outstanding performance in the field of handwritten digit recognition since the last few years. Deep learning is more efficient learning technique than others because deep learning is the combination of feature extraction and deep classification layers. That's why we have chosen the deep learning for Bangla handwritten digit recognition. We have built a custom architecture for deep learning. The architecture is illustrated in the following.

CNN model Architecture

Our proposed architecture consists of 6 convolutional layers and 2 fully connected dense layer. The first two layers have 32 filters and each filter size is 5×5. The middle two layers have 128 filters and each filter size is 3×3. The last two layers have 256 filters and each filter size is 3×3. Rectified Linear unit (ReLu) is used as an activation function for all layers. Maxpooling layers and Batch normalization are used after every two layers. The pool size of maxpooling layer is 2×2. Batch normalization is used for speed up learning. Dropout (20%) is added after first dense layer to reduce overfitting. Among the two fully connected layers, the first one has 64 filters and the last one has 10 filters for the 10 digits. The last activation function is a softmax function for the classification. We use the Adam optimizer to update weights. Fig. 4 shows the design of our proposed deep CNN architecture. From input to output every configuration is marked properly.

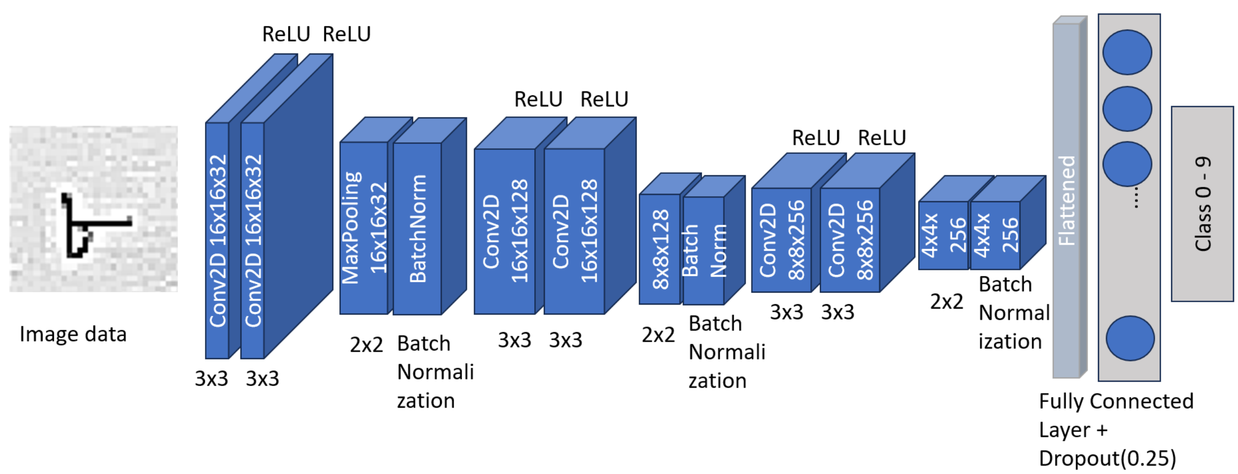

Figure 4: CNN model architecture.

Table I Shows the whole model summary of our proposed deep CNN architecture.

Training Model Parameters

The training model of our architecture is dependent on some parameters. The parameters of the training model are given in Table II.

Table 1: Summary of CNN model 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 2: Parameters of training model

Parameters Name	**Value**
Learning rate	10^-3
Batch size	200
Epochs	30
Shuffle	True

4 EXPERIMENT AND RESULT ANALYSIS

A Experimental Environment

We trained our model on ”Google Colab” using run-time type as T4 GPU.

Training, Validation, and Testing

Among 85000+ images, train and test split ratio of NumtaDB dataset is 85%-15%. In the experiment, we split the training data into training and validation keeping the split ratio about 80%-20%. We use the validation data to evaluate our model performance. At last, our final result is measured by testing dataset.

Result analysis and improved accuracy

We observe mainly three results from our experiment. These are training accuracy, validation accuracy, and testing accuracy. After 30 epochs we get this result. Table III shows our training, validation result as well as the testing result for un-weighted average accuracy.

Table 3: UN-WEIGHTED AVERAGE ACCURACY

Training accuracy	**Validation accuracy**
99.95%	98.57%

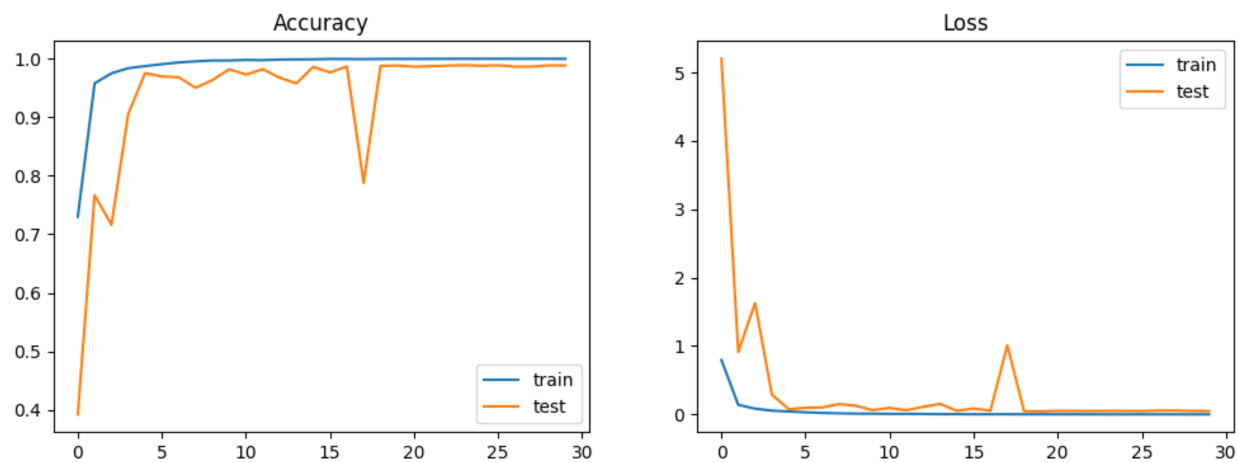

Figure 5: Model accuracy and loss curve.

Then, we augmented images using keras ImageDataGenerator function setting the parameter rotation_range=20 and we used callback function for decaying the learning rate using keras ReduceLROnPlateau function setting the parameters monitor='val_loss',factor=0.5, patience=2, verbose=1. Then, we retrain the same model and improved the accuracy to 99.33%.

Table 4: Improved accuracy on reduced learning rate

Training accuracy	**Validation accuracy**
99.95%	99.33%

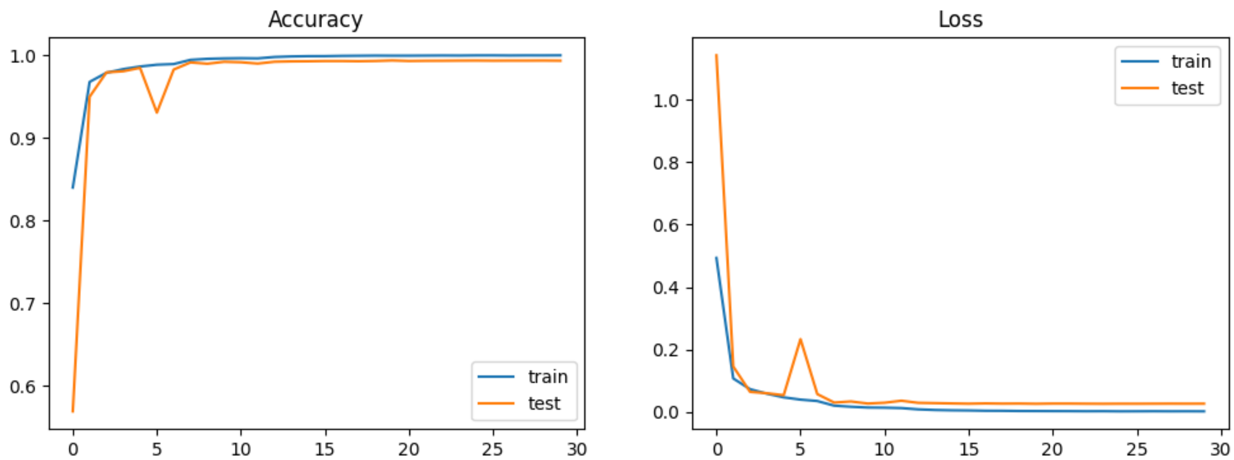

Figure 6: Improved accuracy and reduced loss curve.

5 Conclusion

In this project, we have presented a deep CNN based Bangla digit recognition system for a standard and challenging dataset. We have achieved 99.33% validation accuracy which is a good result for large and unbiased NumtaDB dataset comparing to other biased datasets. We observed that only deep classification model cannot increase performance. All kinds of preprocessing of images are also very important before training. We use some preprocessing technique for blur and noisy images but these are not enough for high performance. As a future work, we think the advanced data augmentation technique and more advanced CNN model can overcome the problems of detecting augmented images. Our aim is to make a reliable model that will achieve 100% accuracy.

Bibliography

[1] S. Alam, T. Reasat, R. M. Doha, and A. I. Humayun, "Numtadb-assembled bengali handwritten digits," *arXiv preprint arXiv:1806.02452*, 2018.