ISSN: 2229-6956 (ONLINE)
DOI: 10.21917/ijsc.2022.0364

BANGLA HANDWRITTEN CHARACTER RECOGNITION USING CONVOLUTION NEURAL NETWORK

Shankha De and Arpana Rawal

Department of Computer Science and Engineering, Bhilai Institute of Technology, India

Abstract

Since, last one-decade, numerous deep learning models have been designed to resolve handwritten character recognition task in languages, namely, English, Chinese, Arabic, Japanese and Russian. Recognition of Bengali handwritten character from document image datasets is undoubtedly an open challenging task. Due to the advancement of neural network, many models have been developed and it is improving performance. The LeNet is a pioneering work in the field handwritten document image recognition specially hand written digits from the images by using CNN. This paper focuses on designing a convolution neural network with refinements on layers and its parameter tuning for Bengali character recognition system for classification of 50 different fonts. Our revised CNN model outperforms on some existing approach and shows font-recognition accuracy of 98.46%.

Keywords:

Convolution Neural Network, Handwritten Character, LeNet

1. INTRODUCTION

Most of the OCR research work concerned European, Arabic, Chinese and Japanese scripts [1]-[9]. In recent years, due to the advancement of machine learning and deep learning techniques, researchers are using MLP and CNN based classification techniques for character recognition of various languages. But it needs high configuration system and large number of training dataset. Among Indian Script, Bengali script is remained open active research area. Complexity in graphics structure and similarity of some characters makes Bengali script a more challenging. Success rate of printed text recognition is always better than handwritten character recognition.

Various conventional approaches are already designed made for developing Bangla character recognition system. Syntactic approach, Template matching, Hidden Markov Model, Support vector machine and Multi-layered Perceptron based ANN techniques has been used in both, print (digitized) character and handwritten (scanned) character recognition domains.

It was realized that print character classification models were much easier to implement than hand-written character recognition models owing to the challenges raised by scripting font styles, sizes, shapes and stroking components. As a result, averagely satisfactory performance of these approaches paved way to much more scopes of variant simulation deep-learning modeling of font-classification models. This paper proposed a CNN model which is capable to recognize handwritten Bengali characters with sufficiently high accuracies (98.46%) applied on standard CMATERdb 3.1.2 dataset [10]-[15].

This paper is organized as follows: section 2 describes use of various machine learning approaches of Bengali Script recognition process. Section 3 gives basic concept and components description of convolution neural network. Section 4 describes details of CNN model experimental setup and data set used. Section 5 shows experimental results followed by the conclusion of this experiments and its future scope.

2. LITERATURE REVIEW

Recent implementations in deep learning (DL) based handwritten font-classification approaches have resulted in implementation of highly efficient hand-written character-recognition models. In this section various recent research paper is reviewed where CNN is applied in many other languages for character recognition. Success rate of recognizing English, Chinese, Japanese, Russian and even Arabic script is noticeable in compare with Indic scripts. Among Indic script, Hindi, Tamil, Gujrati, Oriya and Bengali scripts recognition has been started and getting acceptable accuracy rate. But till date it is an open area for exploring new methods and techniques to improve result.

Bengali Character Recognition process accelerated by Choudhary and Pal [16] by their approach for developing a complete Bengali OCR system for printed character [16]. In continuation with Pal [17] uses water reservoir techniques for unconstrained character recognition.

Bhowmik et al. [18] applied MLP based classifier on stroke features for recognizing handwritten character set and achieved around 84% of accuracy. Basu et al. [19] uses MLP based classifier on 76 different features based on longest run, centroid and shadow features which gives around 75% of accuracy. Advancement of machine learning CNN is also playing an important role for Bengali handwritten script recognition. Various CNN based model has been developed in Bengali handwritten character recognition.

Rahaman et al. [20] achieved 85% accuracy using CNN. Rabby et al. [21] [22] have designed BornoNet and Ekushnet to identify handwritten Bengali fonts using convolution neural network and achieves 95.71% to 98% accuracy in various dataset like ISI, Banglalekha, CMATERdb. Some researchers achieved more accuracy rate using CNN by their own dataset.

Rizvi et al. [23] proposed a hybrid feature extraction method by applying zoning along with density, zone centroid and histogram of gradient features to create a feature vector of size 26. A 2-D sobel mask is used for gradient magnitude and orientation calculation. Cell histogram of 9 bins, varies from 0 to 180º, each bin of 20º is calculated by applying weighted votes. The feature set feed to SVM and CNN (ResNet 18) achieved 87% and 98.04% accuracy. Chatterjee et al. [24] achieves accuracy of 96.12% in 47 epochs by Resnet 50.

Khandokar et al. [25] achieves 92.91% accuracy on NIST dataset whereas Hasan et al. [26] proposed a low-cost convolution neural network for Bengali character recognition. This paper proposed a CNN model to improve accuracy and achieved 98.46% on standard dataset CMATERdb 3.1.2.

3. CNN MODEL FUNCTIONALITY

CNN is one of most used advanced models of neural network. It is made up of huge number of neurons that are interconnected and divided into three layers as input layers which takes user input data passes to next layer, multiple hidden layers used to learn non-linear combination of features to classify or predict object and output layers produces final output. Basic components of any CNN are [9] [27]:

- **Input Layer**: Work as a buffer to takes input data and passes to next layer.
- **Convolution Layer**: Performs feature extraction by moving the kernel over input and found sum of product for each location.
- **Rectified Linear Unit**: An activation function that imposes non linearity as well as convert negative value to 0 that improves learning process.
- **Pooling Layer**: Max pooling and average pooling manipulate spatial size of feature to improve computation and accuracy by extracting more traits. Max pooling works better in presence of noise.

Suppose an input image size $n \times n$. Convolution filter size is ($f \times f$), padding p, stride s, the number of filter is k, Image features are extracted and matched by sub sampling the image size by using the Eq.(1).

$$m = ((n+2p-f)/s)+1 \tag{1}$$

The resulting image size will be k @ $m \times m$ [28].

A simple CNN architecture is represented in Fig.1 that can be used for Bengali handwritten character recognition. Suppose it takes a 32×32-pixel image as an input. The architecture uses two convolution layers, two pooling layer and one flat layer followed by two fully connected layer and a softmax layer as a final classifier.

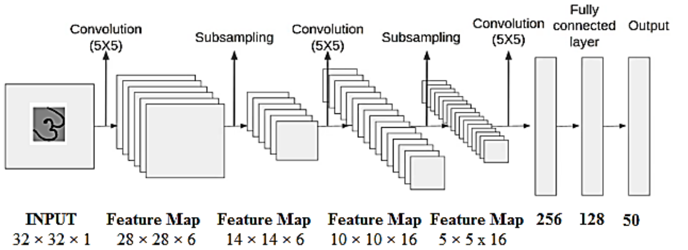

Fig.1. Components of a CNN model

For gray scale image, number of color channel is one, so the input image is a 32×32×1 image for the input layer as shown in the Fig.1. First convolution layer uses a 6 filters of size 5×5 filter and stride as 1 and, the output image size will be (32-5+1)×(32-5+1)×6 = 28×28×6 as number of channels is equal to number of filters. In pooling layers using averaging operation i.e. Average pooling layers reduces the feature map to half as 14×14×6 keeping the number of channels intact. Next convolution layer uses sixteen filter with kernel size 5×5 that again reduces feature map to 10×10×16. Next sub sampling layer will convert it to 5×5× 16. Then final convolution layer having 256 filter of size 5 x 5 makes 1×1×256 feature map that flatten converts to a 256 value. Then a fully connected layer of 128 neurons and at last output layer of 50 neurons for 50 different class of hand written character fonts.

CNN model takes input character image, extract different local features using learnable weights and biases and categorizes to predefined classes. It is performing well in object recognition as well as in character recognition system.

4. DATASET AND FRAMEWORK

The proposed model is using standard dataset which consist of 50 individual Bengali handwritten character set that are labeled properly. It contains two different datasets, one as a training dataset of 12000 handwritten characters for different shape, 240 different sample images in each font class, are considered and another dataset of 3000 images, 60 sample image for each class, for testing purpose. Another important feature is that all images are noise free. All these image set are grouped and labeled to 50 different folders from 0 to 49 for 50 different classes.

The implementation of our revised CNN model is found to perform feature extraction step that is invariant of scaling, rotation as well as robust to other distortions. This makes the revised CNN model to skip inevitable image document preprocessing stages unlike in traditional character recognition processing stages. Our model consisted of two parts, the pre-trained ResNet-34 part and our custom model. Pre-trained part contains sequential 34 layers of sequential and basic blocks. First convolution layer uses kernel size 7 with feature map 64 and 33 padding. It is followed by 3×3 convolutions and feature map dimension vary from [64.128.256.512]. Function Adam at 0x00000186CDCF8CA0 optimizer is used and loss functions are FlattenedLoss of CrossEntropyLoss(). Model unfrozen Callbacks are TrainEvalCallback, Recorder and ProgressCallback. Custom part contains AdaptiveConcatPool2d, Flattern, Relu, 2 BatchNorm1d, 2 Dropout and 2 Linear layers.

5. RESULT AND DISCUSSION

Fastai library and Python version 3.9.6 are used for coding and the model facilitates with different input image sizes as it has AdaptiveConcatPool2d layers which re-adjust the shape/size of the data. The training set consisted of 50 classes, each of which had 240 images each.

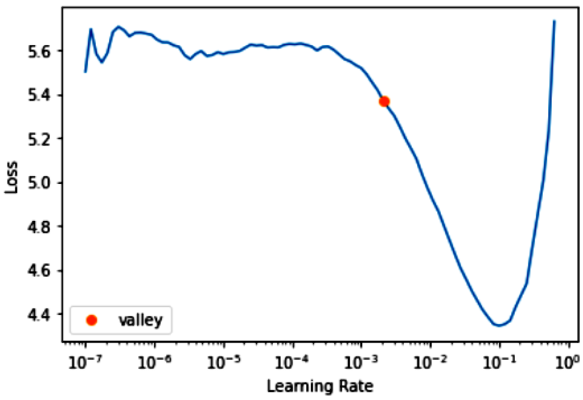

Fig.2. Learning Rate of the Model

Then while importing the dataset into the code, it is splitted into train set and validation set in the ratio 4:1 (80% train set {9600 images}, 20% val set {2400 images}). Model uses the concept of Transfer learning and downloaded a pre-trained ResNet-34 model. Then it looks for the learning rate for training. After finding the proper learning rate as shown in fig.3 to train the model, training performed only to custom model part while the layers of pre-trained ResNet model were freezed for training.

Custom model part is trained on top of pre-trained ResNet model for 20 epochs. It results more than 97% accuracy on the validation set after this step.

Table.1. Custom Model Part Training Details

Epoch	**Training Loss**	**Validation Loss**	**Accuracy**	**Error Rate**	**Time**
0	1.548345	0.982588	0.720833	0.279167	01:14
1	1.160024	0.769645	0.788750	0.211250	00:40
2	1.645433	1.634427	0.763750	0.236250	00:41
3	4.276985	9.914360	0.479583	0.520417	00:41
4	5.259116	3.377505	0.837500	0.162500	00:41
5	5.553296	3.478723	0.872500	0.127500	00:41
6	4.480882	5.890141	0.842083	0.157917	00:41
7	4.436390	2.156400	0.930000	0.070000	00:41
8	4.010653	3.583086	0.906667	0.093333	00:41
9	3.389378	2.591322	0.928333	0.071667	00:41
10	2.695751	2.515882	0.918750	0.081250	00:41
11	2.001156	1.907865	0.942917	0.057083	00:42
12	1.286886	1.750469	0.951250	0.048750	00:41
13	1.096363	1.348161	0.956250	0.043750	00:42
14	0.698854	1.058288	0.962917	0.037083	00:41
15	0.442667	0.925769	0.965833	0.034167	00:42
16	0.288842	0.780559	0.972083	0.027917	00:42
17	0.195554	0.771035	0.972917	0.027083	00:42
18	0.111543	0.739849	0.972500	0.027500	00:42
19	0.156328	0.703851	0.974583	0.025417	00:42

Then unfreeze the previously freezed layers of pre-trained ResNet mode, and train whole network altogether for 12 epochs. Accuracy reaches around 98.91%.

Table.2. Full Model Training Details

Epoch	**Training Loss**	**Validation Loss**	**Accuracy**	**Error Rate**	**Time**
0	0.281411	0.074470	0.991667	0.008333	00:50
1	0.727259	0.485151	0.971250	0.028750	00:51
2	0.907035	0.857859	0.951667	0.048333	00:50
3	0.602399	0.910061	0.950417	0.049583	00:51
4	0.570222	0.418590	0.970000	0.030000	00:51
5	0.475016	0.396487	0.975417	0.024583	00:51
6	0.262390	0.302233	0.981667	0.018333	00:51
7	0.177938	0.239381	0.983750	0.016250	00:51
8	0.130220	0.200896	0.986667	0.013333	00:51
9	0.095973	0.156732	0.988333	0.011667	00:50
10	0.052336	0.152095	0.988750	0.011250	00:51
11	0.036836	0.153460	0.989167	0.010833	00:52

Now trained model is applied to predict on previously unseen data test data set. The trained model correctly classified 2954 out of 3000 testing images, thereby giving an accuracy of 98.466%.

The Fig.3 shows the confusion matrix for all 50-font class of Bengali script. The highlighted cell value depicts the row font class misclassified more than two times. Maximum misclassification value for a particular class is 5 that occur only one time for Bengali font 'naa'.

	অ	আ	ই	ঈ	উ	ঊ	ঋ	এ	ঐ	ও	ঔ	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	য়	ড়	ঢ়	ৎ	ং	ঃ	ঁ
অ	60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
আ	0	60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ই	0	0	56	1	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
ঈ	0	0	0	60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
উ	0	0	0	0	58	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ঊ	0	0	0	0	1	59	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ঋ	0	0	0	0	0	0	57	0	0	0	0	0	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
এ	0	0	0	0	0	0	0	57	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ঐ	0	0	0	0	0	0	0	0	59	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ও	0	0	0	0	0	0	0	0	0	59	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ঔ	0	0	0	0	0	0	0	0	0	0	60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ক	0	0	0	0	0	0	0	0	0	0	0	59	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
খ	0	0	0	0	0	0	0	0	0	0	0	0	60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
গ	0	0	0	0	0	0	0	0	0	0	0	0	0	59	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ঘ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	59	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ঙ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

চ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ছ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	59	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
জ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ঝ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ঞ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	59	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ট	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ঠ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	59	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ড	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	59	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ঢ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	59	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ণ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	54	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
ত	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	59	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
থ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	57	0	0	1	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0
দ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ধ	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	58	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
ন	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	58	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
প	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ফ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ব	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	59	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ভ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ম	0	0	0	0	0	0	0	0	0	0	0	0	1	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	57	0	0	0	0	0	0	0	0	0	0	0	0	0	0
য	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	59	0	0	0	0	0	0	0	0	0	0	0	0	0
র	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	59	0	0	0	0	0	0	0	1	0	0	0	0
ল	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	60	0	0	0	0	0	0	0	0	0	0	0
শ	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	58	0	0	0	0	0	0	0	0	0	0
ষ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	59	0	0	0	0	0	0	0	0	0
স	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	58	0	0	0	0	0	0	0	0
হ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	60	0	0	0	0	0	0	0
য়	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	60	0	0	0	0	0	0
ড়	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	60	0	0	0	0	0
ঢ়	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	60	0	0	0	0
ৎ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	60	0	0	0
ং	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	60	0	0
ঃ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	60	0
ঁ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	58

Fig.3. Confusion Matrix of 50 Different Font Class

Table.3. Performance Comparison with Some Previous Existing Work

Methodology	Features and Classifier	Accuracy
Rahman et al. [20]	Line based 108 Geometric feature set using SVM and ANN	84.56% 75.6%
Rabby et al. [21] [22]	BornoNet, Ekushnet	98%
Rizvi et al. [23]	Zone based 26 features using SVM and ANN	87% 98.04%
Chatterjee et al. [24]	CNN	96.12%
Khandokar et al. [25]	CNN	92.91%
Proposed Work	ResNet 34+ Custom model	98.46%

Performances of various research works based on ANN and variation of CNN model are given in table 3 that shows the proposed method outperforms upon most of the previous existing work.

6. CONCLUSION AND FUTURE SCOPE

Researchers are on the move to implement variant CNN models in handwritten character recognition by predominantly changing the number of layers. Eventually, it was realized that the model accuracy can still be increased above our achieved model accuracy (98.46%) by changing the count of intermediate layers, epochs and input (document image) batch sizes, however, at the cost of higher model execution times.

Also, other hyper-parameters for fine tuning functionality of CNN model variants may include pool size and learning rate. The hybridization feature can also be incorporated by using different activation functions in different operational layers. The major strengths of using CNN model frequently by ML researchers is that it supports high noise tolerance levels with meager level of (image) data preprocessing. In future, this model is anticipated to be run upon online real-time datasets.

REFERENCES

[1] V. Patil and S. Shimpi, "Handwritten English Character Recognition Using Neural Network", *Elixir International Journal: Computer Science and Engineering*, Vol. 41, pp. 5587-5591, 2011.

[2] Y.C. Wong, L.J. Choi, R.S.S. Singh, H. Zhang and A.R. Syafeeza, "Deep Learning-Based Racing Bib Number Detection and Recognition", *Jordanian Journal of Computers and Information Technology*, Vol. 5, No. 3, pp. 181-194, 2019.

[3] G. Abandah, M. Khedher and K. Younis, "Evaluating and Selecting Features for Recognizing Handwritten Arabic Characters", Technical Report, Department of Computer Engineering, The University of Jordan, pp. 1-156, 2007.

[4] Idris Alsheikh, Mohd Masnizah and Warlina Lia, "A Review of Arabic Text Recognition Dataset", Asia-Pacific Journal of Information Technology and Multimedia, Vol. 9, pp. 69-81, 2020.

[5] Khaled S. Younis, "Arabic Hand-Written Character Recognition Based on Deep Convolutional Neural Networks", *Jordanian Journal of Computers and Information Technology*, Vol. 3, No. 3, pp. 1-13, 2017.

[6] Ahmad Hasasneh, "Arabic Sign Language Characters Recognition Based on A Deep Learning Approach and a Simple Linear Classifier", *Journal of Computers and Information Technology*, Vol. 6, No. 3, pp. 1-15, 2020.

[7] F. Yin, Q.F. Wang, X.Y. Zhang and C.L. Liu, "Chinese Handwriting Recognition Competition", *Proceedings of International Conference on Document Analysis and Recognition*, pp. 1464-1470, 2013.

[8] B. Zhu, X.D. Zhou, C.L. Liu and M. Nakagawa, "A Robust Model for On-Line Handwritten Japanese Text Recognition", *International Journal on Document Analysis and Recognition*, Vol. 13, No. 2, pp. 121-131, 2010.

[9] Mayur Bhargab Bora, Dinthisrang Daimary, Khwairakpam Amitab and Debdatta Kandar, "Handwritten Character Recognition from Images using CNN-ECOC", *Procedia Computer Science*, Vol. 167, pp. 2403-2409, 2020

[10] N. Das, R. Sarkar, S. Basu, M. Kundu, M. Nasipuri and D.K. Basu, "A Genetic Algorithm Based Region Sampling for Selection of Local Features in Handwritten Digit Recognition Application", *Applied Soft Computing*, Vol. 12, pp. 1592-1606, 2012.

[11] N. Das, J.M. Reddy, R. Sarkar, S. Basu, M. Kundu, M. Nasipuri and D.K. Basu, "A Statistical-Topological Feature Combination for Recognition of Handwritten Numerals", *Applied Soft Computing*, Vol. 12, pp. 2486-2495, 2012.

[12] N. Das, K. Acharya, R. Sarkar, S. Basu, M. Kundu and M. Nasipuri, "A Novel GA-SVM Based Multistage Approach for Recognition of Handwritten Bangla Compound Characters", *Proceedings of the International Conference on Information Systems Design and Intelligent Applications*, pp. 145-152, 2012.

[13] N. Das, S. Basu, R. Sarkar, M. Kundu, M. Nasipuri and D.K. Basu, "Handwritten Bangla Compound Character Recognition: Potential Challenges and Probable Solution", *Proceedings of the International Conference on Artificial Intelligence*, pp. 1901-1913, 2009.

[14] N. Das, S. Basu, R. Sarkar, M. Kundu, M. Nasipuri and D.K. Basu, "An Improved Feature Descriptor for Recognition of Handwritten Bangla Alphabet", *Proceedings of the International Conference on Signal and Image Processing*, pp. 451-454, 2009.

[15] N. Das, K. Acharya, R. Sarkar, S. Basu, M. Kundu and M. Nasipuri, "A Benchmark Data Base of Isolated Bangla Handwritten Compound Characters", *International Journal on Document Analysis and Recognition*, Vol. 17, pp. 413-431, 2014.

[16] B.B. Chaudhuri and U. Pal, "A Complete Printed Bangla OCR System", *Pattern Recognition*, Vol. 31, pp. 531-549, 1998.

[17] U. Pal, A. Belad and C.H. Choisy "Touching Numeral Segmentation using Water Reservoir Concept", *Pattern Recognition Letters*, Vol. 24, pp. 261-272, 2003.

[18] T.K. Bhowmik, U. Bhattacharya and S.K. Parui, "Recognition of Bangla Handwritten Characters using An MLP Classifier Based on Stroke Features", *Proceedings of the International Conference on Neural Information*, pp. 814-819, 2004.

[19] S. Basu, N. Das, R. Sarkar and D.K. Basu, "Handwritten Bangla Alphabet Recognition using An MLP Based Classifier", *Proceedings of the International Conference on Artificial Intelligence*, pp. 1-8, 2012.

[20] M.D. Rahman and P.C. Shill, "Bangla Handwritten Character Recognition using Convolutional Neural Network", *International Journal on Image, Graphics and Signal Processing*, Vol. 8, pp. 42-49, 2015.

[21] A.K.M.S. Rabby, S. Azad, M.S. Haque and S.A. Hossain, "Bornonet: Bangla Handwritten Characters Recognition using Convolutional Neural Network", *Procedia Computer Science*, Vol. 143, pp. 528-535, 2018.

[22] A.K.M.S. Rabby, S. Azad, M.S. Haque and S.A. Hossain, "Ekushnet: Using Convolutional Neural Network for Bangla Handwritten Recognition", *Procedia Computer Science*, Vol. 143, pp. 603-610, 2018.

[23] M.A.I. Rizvi, K. Deb, M.I. Khan, M.M.S. Kowsar and T. Khanam, "A Comparative Study on Handwritten Bangla Character Recognition", *Turkish Journal of Electrical Engineering and Computer Sciences*, Vol. 27, No. 4, pp. 3195-3207, 2019.

[24] S. Chatterjee, R.K. Dutta and S. Roy, "Bengali Handwritten Character Classification using Transfer Learning on Deep Convolutional Network", *Proceedings of the International Conference on Intelligent Human Computer Interaction*, pp. 138-148, 2019.

[25] I. Khandokar, M. Hasan, F. Ernawan, S. Islam and M.N. Kabir, "Handwritten Character Recognition using Convolutional Neural Network", *Journal of Physics: Conference Series*, Vol. 1918, No. 4, pp. 593-611, 2021.

[26] A. Sayeed, J. Shin and M.M. Hasan, "BengaliNet: A Low-Cost Novel Convolutional Neural Network for Bengali Handwritten Characters Recognition", *Applied Science*, Vol. 11, pp. 6845-6854, 2021.
[27] Alejandro Baldominos, Yago Saez and Pedro Isasi, "Evolutionary Convolutional Neural Networks: an application to Handwriting Recognition", *Neurocomputing*, Vol. 49, pp. 1-16, 2017.
[28] S.L. Chooi and Aimi Syamimi Binti A.B. Ghafar, "Handwritten Character Recognition using Convolutional Neural Network", *Progress in Engineering Application and Technology*, Vol. 2, No. 1, pp. 593-611, 2021.