Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 143 (2018) 603–610

Procedia Computer Science

www.elsevier.com/locate/procedia

8th International Conference on Advances in Computing and Communication (ICACC-2018)

EkushNet: Using Convolutional Neural Network for Bangla Handwritten Recognition

AKM Shahariar Azad Rabby[a]*, Sadeka Haque[a]†, Sheikh Abujar[a] and Syed Akhter Hossain[a]

[a]*Dept. of Computer Science and Engineering, Daffodil International University, Dhanmondi, Dhaka-1205, Bangladesh*

Abstract

EkushNet is the first research which can recognize Bangla handwritten basic characters, digits, modifiers, and compound characters. Handwritten recognition is one of the most interesting issue in present time due to its variant applications and help to make the old form and information digitization and reliable. In spite of, there is no single model which can classify all types of Bangla characters. One of most common reason conducting with handwritten scripts is big challenge because of every person has unique style to write and also has different shape and size. Therefore, EkushNet is proposed a model which help to recognize Bangla handwritten 50 basic characters, 10 digits, 10 modifiers and 52 mostly used compound characters. The proposed model train and validate with Ekush [1] dataset and cross-validated with CMATERdb [2] dataset. The proposed method is shown satisfactory recognition accuracy 97.73% for Ekush dataset, and 95.01% cross-validation accuracy on CMATERdb dataset, which is so far, the best accuracy for Bangla character recognition.

Keywords:: Bangla handwritten; Data Science; Machine Learning; Deep Learning; Computer Vision; Patteran Recognition;

* AKM Shahariar Azad Rabby.
E-mail address: azad15-5424@diu.edu.bd
† Sadeka Haque
E-mail address: sadeka15-5210@diu.edu.bd

1877-0509 © 2018 The Authors. Published by Elsevier B.V.
This is an open access article under the CC BY-NC-ND license (https://creativecommons.org/licenses/by-nc-nd/4.0/)
Selection and peer-review under responsibility of the scientific committee of the 8th International Conference on Advances in Computing and Communication (ICACC-2018).
10.1016/j.procs.2018.10.437

1. Introduction

Automatic handwritten recognition is one of the most important research fields in recent years for its different application such as OCR which helps to recognize the character from images. Over the last few years, information has transferred from handwritten hardcopy documents to digital file formats, which is more reliable. However, this system to handle new forms of document. But till now a large number of older documents are written by hand. The challenge is attempting to convert them if follow tradition method to copy the document by manual typing. Because it takes a long time and needs a huge number of manpower. But OCR process can help to digitize old information with less time and less manpower. For that reason, a robust model of handwritten character recognition plays an important role. In spite of this Bangla handwritten character recognition has no strong model that helps to build robust Bangla OCR. Because of yet there is no model which can classify all kind of characters (basic character, numeral, modifiers, compound character). Many works have been done but those are concentrated for digit [3] or basic characters [4] or compound characters [5]. Dealing with handwritten character is complicated because of different shape and style. And the arrangement of Bangla character is complex due to its alignment and many of them are similar apart from compound characters that complement other basic characters. Fig 1 is representing an example of Bangla characters.

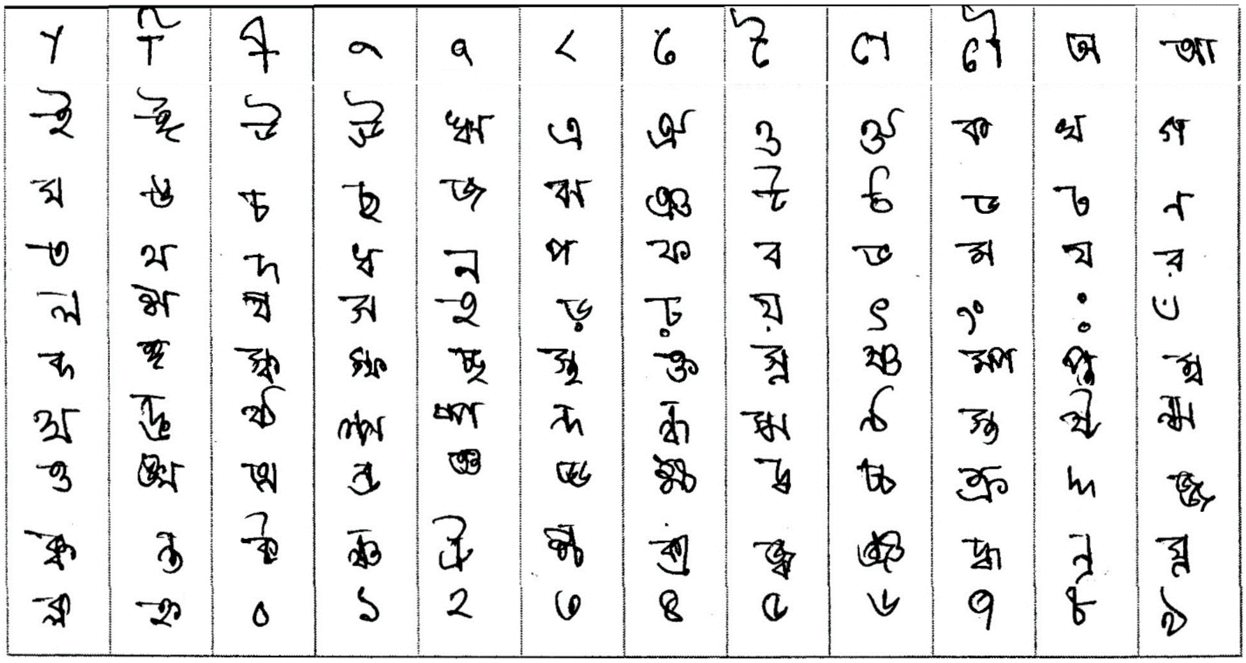


Fig 1. Example of Bangla Characters

Bangla is 4th most popular language in the world. It is the first language of Bangladesh with a rich heritage. February 21st is announced as the International Mother Language Day by UNESCO to respect the language martyrs for the Bangla language in Bangladesh in 1952. It is the second most popular language in the Indian subcontinent. So, overall About 300 million people use Bangla language as their writing and speaking purpose. Consider all that situation Bangla handwritten character recognition plays an important role to help those people in different purpose as Bangla traffic number plate recognition, automatic postal code identification, extracting data from hardcopy forms, automatic ID card reading, automatic reading of bank cheques and digitalization of documents etc.

The Convolutional Neural Network (CNN) reveal new opportunities in the field of pattern recognition for classification, which is helping numerous researchers to implement their state of the art system in solving. The CNN structure was first proposed by Fukushima et al. in 1980 [6] but it was not widely used because of the algorithm was complex. In 1990s, LeCun et al. applied a gradient-based learning algorithm to CNN and achieved successful results [7]. After that, many researchers work on it and improved CNN and made good results in pattern recognition. A few years ago, Cirean et al. [8] applied multi-column CNN to recognize digits, alpha-numerals, traffic signs, and the other object class.

One language fundamental is different from other languages like Latin scripts are differ from Bangla because Bangla comes from Sanskrit scripts. In Bangla language written scripts has 50 basic characters, 10 numerical digits, more than 200 compound characters and 10 modifiers.

2. Literature review

In past studies there are many works for recognition of handwritten character in a different language as Latin [9], Chines [10], Japanese [11] achieve great success. There are a few works are available for Bangla handwritten basic character, digit and compound character recognition, some literature has been made on Bangla characters recognition in the past years as "A complete printed Bangla OCR system" [12], "On the development of an optical character recognition (OCR) system for printed Bangla script" [13]. there are also few researches on handwritten Bangla numeral recognition that reaches to the desired recognition accuracy. Pal et al. have conducted some exploring works for recognizing handwritten Bangla characters those are "Automatic recognition of unconstrained offline Bangla hand-written numerals" [14], "A system towards Indian postal automation" [15]. And "Touching numeral segmentation using water reservoir concept" [16]. The proposed schemes are mainly based on extracted features from a concept called water reservoir. Apart from there also present several Bangla Handwritten Character Recognition and had achieved pretty good success. Halima Begum et al., "Recognition of Handwritten Bangla Characters using Gabor Filter and Artificial Neural Network" [17] works with own dataset that was collected from 95 volunteers and their proposed model achieved without feature extraction and with feature extraction around 68.9% and 79:4% of recognition rate respectively. "Recognition of Handwritten Bangla Basic Character and Digit Using Convex Hall Basic Feature" [18] achieve accuracy for Bangla characters 76.86% and Bangla numerals 99.45%. "Bangla Handwritten Character Recognition using Convolutional Neural Network" achieved 85.36% test accuracy using their own dataset. In "Handwritten Bangla Basic and Compound character recognition using MLP and SVM classifier" [19], the handwritten Bangla basic and compound character recognition using MLP and SVM classifier has been proposed and they achieved around 79.73% and 80.9% of recognition rate, respectively.

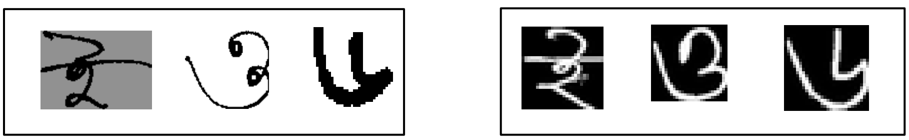
3. Proposed methodology

The proposed "EkushNet" CNN model has many steps as described below.

3.1. Datasets

The proposed model used a new dataset Ekush for training and most popular CMATERdb for cross-validation. CMATERdb database contains a total of 21,000 characters image where 15,000 for Bangla alphabets and 6,000 images of Bangla digits. CMATERdb dataset images are 32 pixels in both height and weight.

The Ekush dataset has total 368,776 images where 155,570 alphabets, 151,607 compound characters, 30830 digits and 30769 modifiers. Ekush dataset's image resolution depends on character size. Most of the images have less padding with a black background while the character in white. Fig 2 showing an example of both datasets.



(a) (b)

Fig 2. Example of dataset (a) CMATERdb, (b) Ekush

3.2. Preparation of dataset

Data preparation plays an important role in deep learning. Data is everywhere, however, the problem is the lack of processed data. Proposed EkushNet model used Ekush dataset and CMATERdb dataset.

CMATERdb dataset images background is white and characters are black. Firstly, we inverted all the images to make the background black and character to white. Black pixels represent the value 0, which reduce lots of computation.

The images of Ekush dataset are different in height and width to reduce unnecessary information. Then both datasets resized into 28 x 28 pixels. All of these 28 x 28 pixels are stored into a CSV file to speed up the calculation by reducing the hard disk read time. The CSV has 785 values in each row, where 784 (28 x 28) represent the characters pixel value and 1 value represent the label or class of the characters.

During training time, the image pixel values are normalized using Minmax (1) normalizer. This normalization method linearly transforms the value to map between 0 to 1 where the maximum value is 1 and the minimum value is 0. This method reduces the effect of lighting difference. Also, CNN performs better on 0-1 data than on 0-255.

$$Z_i = \frac{X_i - \text{minmum}(X)}{\text{maximum }(X) - \text{minmum}(X)} \quad (1)$$

Then converted all the data label into one hot encoding.

3.3. EkushNet architect

Proposed EkushNet used a multilayer CNN for classifying Bangla Handwritten Characters. This model used convolution, Max pooling layer, fully connected dense layer and some regularization method like batch normalization [20] and dropout [21].

Layer 1 and 2 are a convolutional layer with a filter size of 32 and kernel size of 5, these two layers also use ReLU (2) activation with the same padding. The output of these layer later connected with max pooling layer 3 followed by 25% dropouts layer 4.

$$ReLU(X) = MAX(0, X) \quad (2)$$

The output of layer 4 than goes into two separate layers- layer 5 and layer 11. Layer 5 is a convolution with 64 filters followed by a batch normalization layer 7. Layer 8 is also a convolution with 64 filters with kernel size of 5 with a batch normalization layer-8, max pooling layer-9 and 20% dropout layer – 10.

Layer 11, a convolutional layer which takes input from layer 4. This layer has 64 filters with 3x3 kernel size. Layer 12 has the same configuration of layer 11, connected with a max pooling layer 13 with 25% dropouts – layer 14.

At layer 15 both layer 10 and 14 are added together and pass into a new convolutional layer with 64 filters and 3 x 3 kernel size which is connected to a max pooling layer 16 with 25% dropout layer 18.

After all of these 18 operations, the output is flatten into an array and pass through a fully connected dense layer 20 with 2048 hidden units and regularized with 25% dropout. The output of the layer 21 connected with a fully connected dense layer 22 with 122 nodes with SoftMax (3) activation which is also the output layer for the model. Code and details about EkushNet can be found on https://github.com/shahararrabby/EkushNet. Fig 3 showing the proposed EkushNet architect.

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \; for\; j = 1, \dots k \quad (3)$$

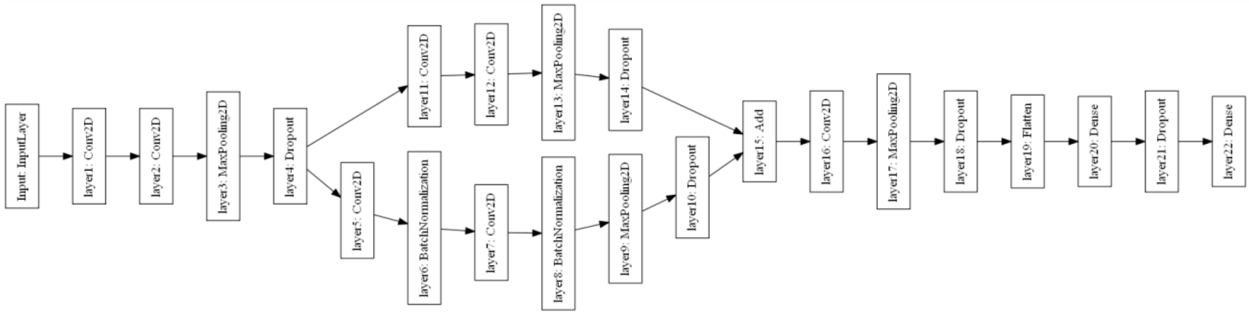


Fig 3. Architecture of EkushNet

Table 1. EkushNet model architect summary.

Layer No (type)	Output Shape	Parameter	Connected to	Layer No (type)	Output Shape	Paramet er	Connected to
(Input Layer)	28, 28, 1	0	-	13 (MaxPooling2D)	7, 7, 64	0	12
1 (Conv2D)	28, 28, 32	832	Input Layer	10 (Dropout)	7, 7, 64	0	9
2 (Conv2D)	28, 28, 32	25632	1	14 (Dropout)	7, 7, 64	0	13
3 (MaxPooling2D)	14, 14, 32	0	2	15 (Add)	7, 7, 64	0	10, 14
4 (Dropout)	14, 14, 32	0	3	16 (Comv2d)	7, 7, 64	36928	15
5 (Conv2D)	14, 14, 64	51264	4	17 (MaxPooling2D)	3, 3, 64	0	16
6 (Batch Normalization)	14, 14, 64	256	5	18 (Dropout)	3, 3, 64	0	17
7 (Conv2D)	14, 14, 64	36928	6	19 (Flatten)	576	0	18
11 (Conv2D)	14, 14, 64	51264	4	20 (Dense)	2048	1181696	19
8 (Conv2D)	14, 14, 64	256	7	21 (Dropout)	2048	0	20
12 (Conv2D)	14, 14, 64	36928	11	22 (Dense)	122	249978	21
9 (MaxPooling2D)	7, 7, 64	0	8				

Total parameters: 1,671,962
Trainable parameters: 1,671,706
Non-trainable parameters: 256

3.4. Optimizer and Learning rate

Optimization algorithms help CNN algorithms to minimize the error. Proposed Ekush net used Adam optimizer [22]. Adam optimization algorithm that can be used to update network weights iteratively in training data. Adam is an update of extension to stochastic gradient descent algorithm. For its better performance, it is widely used in computer vision researches. Proposed EkushNet used Adam optimizer (4) with a learning rate of 0.001.

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt[1]{\hat{v}_t + \varepsilon}} \hat{m}_t \qquad (4)$$

To calculate the error for optimizing algorithm we used categorical cross entropy (5) function. Recent research shows that cross entropy performs better than other function like classification error and mean squared error etc[23].

$$L_i = -\sum_j t_{i,\ j} \log(p_{i,\ j}) \qquad (5)$$

Learning rate is one of the most important hyper-parameters to tune for training convolutional neural networks. If the learning rate is low the classification is more accurate but optimizer will take more time to reach the global optima by reducing the loss. And if the learning rate is high the accuracy may not converge also some time may diverge. So, choosing the best learning rate is more difficult. To overcome this challenge, we use automatic learning rate reduction method [24]. For faster computation, we set a higher learning rate of 0.001 which is atomically reduced by monitoring the validation accuracy.

3.5. Data augmentation

Deep learning technique performs better if it finds more data. For this reason, data augmentation helps to produce more data artificially. For handwriting characters, recognition data augmentation helps more because a single person can write a character in a different variation. For data augmentation, the images are shifted randomly 10% in height or width or both, also 10% rotation and 10% zoom the images.

3.6. Training the model

The EkushNet model was trained on Ekush dataset with a batch size of 86. After 50 epochs the model got good accuracy. The automatic learning rate reduction formula helps the optimizer to converge faster by reducing the learning rate. End of the training the learning rate reduced by 0.001 to $1.5 x 10^{-5}$.

4. Model performance

EkushNet was trained and validated on Ekush dataset and cross-validated using CMATERdb datasets and give a promising result on a train set, test set, and validation set.

4.1. Train, Test and Validation split

To measuring the model performance, train test and validation split were created. The training set is used to train the model with the known output. Validation set used to check model performance during training time and help the model to tune the hyper-parameters. And test data used to check the final model performance after training.

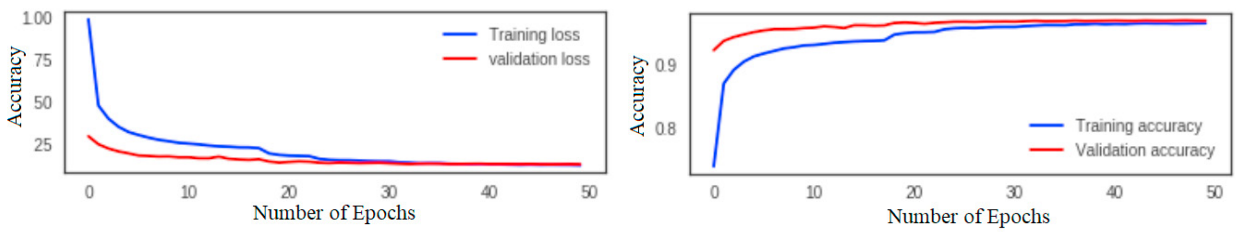
For training and validation purpose we used the Ekush dataset. The Ekush dataset has 367,018 characters images. 55,052 characters 15% of total used in validation and 311,966 characters 85% used to train the model. For making the training data equal from all types of characters, we take 85% of modifiers, basic characters, compound characters, and digits separately than concatenated them and make the training set and add other 15% on the validation set.

For testing the model CMATERdb was used which is completely comes from different distribution. CMATERdb has 15,000 basic characters and digits but no compound characters. All 15,000 images were used to measure model performances.

4.2. Model performance

After 50 epochs proposed model got 96.90% accuracy on the training set and 97.73% accuracy on a validation set of Ekush datasets. After training the model it was tested on CMATERdb dataset and got 95.01 % accuracy.

Analyzing the result and confusion Metrix, found that model performs very well to classify handwritten characters. Many Bangla characters are so similar and sometimes it is harder for a human to recognize the character. In those case, the model did the pretty good job. Fig 4 showing the training and validation loss and accuracy.



(a) (b)

Fig 4. (a) Training and validation loss. (b) Training and validation accuracy

4.3. Result compression

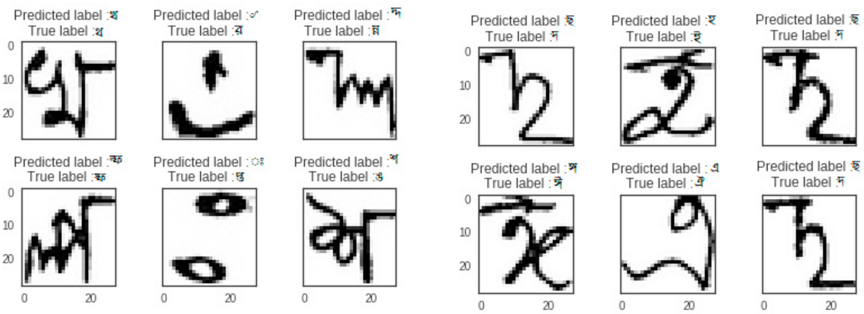
There is no previous work which classifies all types of Bangla characters (basic Characters, numerical characters, compound characters). Table 3 showing some comparison between some previous work on different Bangla characters.

Table 2. Compression between some previous work.

Work	Accuracy	Layer No (type)	Accuracy
Recognition of Handwritten Bangla Characters Using Gabor Filter and Artificial Neural Network [12]	79.4%	HMM-Based Online Handwritten Bangla Character Recognition using Dirichlet Distributions [26]	91.85%
Recognition of Bangla handwritten basic characters and digits using convex hull-based feature set [13]	76.86%	Bangla Hand-Written Character Recognition Using Support Vector Machine [27]	93.43%
Handwritten Bangla Character Recognition Using Neural Network [25]	84.00%	Bengali handwritten character recognition using Modified syntactic method [28]	95.00%
Bangla Handwritten Character Recognition using Convolutional Neural Network [4]	85.36%	**EkushNet (Proposed)**	**97.73% (Ekush)** **95.01% (CMATERdb)**

5. Error remark

Exploring the error from test set and cross-validation set we considered that model is successfully classified 53,803 characters out of 55,052. On Fig 5 (a) top 6 error from test set and Fig 5(b) top 6 error from cross-validation are set shown. From this top 12 images, it is clear that model performs well and top error is caused from incorrectly labeled datasets.



(a) (b)

Fig 5: Error for (a) Validation set (b) Test set

6. Conclusion and future work

To consider the model we understand that Convolutional Neural Network can achieve better performance to classify and recognize Bangla handwritten characters, digits and compound characters. If we had more support as high configuration computer, GPU then the accuracy would have been better as this model gave the result. Experiments on a large dataset Ekush which was helped to build the robustness of this method for Bangla handwritten character recognition.

In future with more resources and bigger CNN architecture, researchers can achieve a better result and improve the state of the art scale for Bangla handwritten basic characters and digit and all compound characters recognition.

References

{1] Ekush: A multipurpose and multitype comprehensive database for Online Off-line Bangla Handwritten Characters, Website: https://github.com/shahariarrabby/Ekush. Last access: 20 Jun. 18

[2] R. Sarkar, N. Das, S. Basu, M. Kundu, M. Nasipuri, and D. K. Basu, "Cmaterdb1: a database of unconstrained handwritten Bangla and Bangla–English mixed script document image," International Journal on Document Analysis and Recognition (IJDAR), vol. 15, no. 1, pp. 71–83, 2012.

[3] Alom, Md. Zahangir & Sidike, Paheding & M. Taha, Tarek & Asari, Vijayan. (2017). Handwritten Bangla Digit Recognition Using Deep Learning.

[4] Md. M. Rahman, M. A. H. Akhand, S. Islam, P. C. Shill, "Bangla Handwritten Character Recognition using Convolutional Neural Network," I. J. Image, Graphics and Signal Processing, vol. 8, pp. 42-49, July 15, 2015.

[5] Saikat Roy, et al. "Handwritten Isolated Bangla Compound Character Recognition: a new benchmark using a novel deep learning approach". Pattern Recognition Letters, Elsevier, Vol. 90, Pages 15-21, 2017.

[6] Fukushima, Kunihiko. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biological Cybernetics, 36(4):193–202, 1980.

[7] LeCun, Yann, Bottou, Leon, Bengio, Yoshua, and Haffner, ´Patrick. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11):2278–2324, 1998a.

[8] Ciresan, D. and Meier, U. Multi-column deep neural networks for offline handwritten Chinese character classification. In International Joint Conference on Neural Networks (IJCNN), pp. 1–6, July 2015.

[9]] E. Kavallieratou, N. Liolios, E. Koutsogeorgos, N. Fakotakis, G. Kokkinakis, "The GRUHD Database of Greek Unconstrained Handwriting", ICDAR, 2001, pp. 561-565.

[10] F. Yin, Q.-F. Wang, X.-Y. Zhang, and C.-L. Liu. ICDAR 2013 Chinese Handwriting Recognition Competition. 2013 12th International Conference on Document Analysis and Recognition (ICDAR), pages 1464–1470, 2013

[11] B. Zhu, X.-D. Zhou, C.-L. Liu, and M. Nakagawa. A robust model for on-line handwritten Japanese text recognition. IJDAR, 13(2):121–131, Jan. 2010.

[12] B. B. Chaudhuri, U. Pal, "A complete printed Bangla OCR system," Pattern Recognition, vol. 31, pp. 531–549, 1998.

[13] U. Pal, "On the development of an optical character recognition (OCR) system for printed Bangla script," Ph.D. Thesis, 1997.

[14] U. Pal, B. B. Chaudhuri, "Automatic recognition of unconstrained offline Bangla hand-written numerals," T. Tan, Y. Shi, W. Gao (Eds.), Advances in Multimodal Interfaces, Lecture Notes in Computer Science, vol. 1948, Springer, Berlin, pp. 371–378, 2000.

[15] K. Roy, S. Vajda, U. Pal, B.B. Chaudhuri, "A system towards Indian postal automation," Proceedings of the Ninth International Workshop on Frontiers in Handwritten Recognition (IWFHR-9), pp. 580–585, October 2004.

[16] U. Pal, A. Belad, Ch. Choisy, "Touching numeral segmentation using water reservoir concept," Pattern Recognition Lett. vol. 24, pp. 261-272, 2003.

[17] Halima Begum et al, Recognition of Handwritten Bangla Characters using Gabor Filter and Artificial Neural Network, International Journal of Computer Technology & Applications, Vol 8(5),618-621, ISSN:2229-6093.

[18] Nibaran Das, Sandip Pramanik, "Recognition of Handwritten Bangla Basic Character and Digit Using Convex Hall Basic Feature". 2009 International Conference on Artificial Intelligence and Pattern Recognition(AIPR-09)

[19] N. Das 1, B. Das, R. Sarkar, S. Basu, M. Kundu, M. Nasipuri, "Handwritten Bangla Basic and Compound character recognition using MLP and SVM classifier," Journal of Computing, vol. 2, Feb. 2010.

[20] Sergey Ioffe, Christian Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift", arXiv:1502.03167 [cs.LG]

[21] Srivastava, Nitish & Hinton, Geoffrey & Krizhevsky, Alex & Sutskever, Ilya & Salakhutdinov, Ruslan. (2014).Dropout: A Simple Way to Prevent Neural Networks from Overfitting. Journal of Machine Learning Research.15.1929-1958.

[22] Kingma, Diederik P. and Ba, Jimmy. Adam: A Method for Stochastic Optimization. arXiv:1412.6980 [cs.LG], December 2014.

[23] Katarzyna Janocha and Wojciech Marian Czarnecki. On loss functions for deep neural networks in classification. Arxiv, abs/1702.05659, 2017

[24] Schaul, Tom, Zhang, Sixin, and LeCun, Yann. No more pesky learning rates. arXiv preprint arXiv:1206.1106,2012.

[25] Md. Alamgir Badsha et al, Handwritten Bangla Character Recognition Using Neural Network, International Journal of Advanced Research in Computer Science and Software Engineering 2 (11), November- 2012, pp. 307-312.

[26] Chandan Biswas et al, HMM-Based Online Handwritten Bangla Character Recognition using Dirichlet Distributions, 978-0-7695-4774-9/12 $26.00 © 2012 IEEE DOI 10.1109/ICFHR.2012.213

[27] Riasat Azim et al, Bangla Hand-Written Character Recognition Using Support Vector Machine, International Journal of Engineering Works vol. 3, Issue 6, PP. 36-46, June 2016.

[28] Mohammad Badiul Islam et al, Bengali handwritten character recognition using modified syntactic method, NCCPB-2005 © Independent University, Bangladesh