Offline Bangla Handwritten Text Recognition: A Comprehensive Study of Various Deep Learning Approaches

Farhan Sadaf*, S. M. Taslim Uddin Raju†, and Abdul Muntakim‡
Department of Computer Science and Engineering
Khulna University of Engineering & Technology
Khulna-9203, Bangladesh
imfarhansadaf@gmail.com*, taslimuddinraju7864@gmail.com†, basitmuntakim@gmail.com‡

2021 3rd International Conference on Electrical & Electronic Engineering (ICEEE) | 978-1-6654-8281-3/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICEEE54059.2021.9718890

***Abstract*—Offline Handwritten Text Recognition (HTR) is a technique for translating handwritten images into digitally editable text format. Due to the presence of cursive letters, punctuation marks, and compound characters, it is more complex to recognize Bangla handwritten text. Over the years, several approaches to the optical model of the HTR system have been developed, including Hidden Markov Model (HMM) or deep learning techniques such as Convolutional Recurrent Neural Networks (CRNN), and current state-of-the-art Gated-CNN based architectures. Despite this, there are relatively limited works available for Bangla word recognition. In this paper, we introduce an end-to-end system for Bangla word recognition. We used a variety of popular pre-trained CNN architectures, including Xception, MobileNet, and DenseNet, followed by recurrent units such as LSTM or GRU. Furthermore, we experimented with Puigcerver's CRNN based and Flor's Gated-CNN based optical model architectulimited works available in Bangla.res. Flor architecture provided the highest recognition rate in our experiment, with a CER of 12.83% and a WER of 36.01%.**

***Index Terms*—Bangla Handwritten Text Recognition; Deep Learning Techniques; CRNN; Gated-CNN; Connectionist Temporal Classification (CTC) loss; Vanilla Beam Search.**

I. Introduction

A Handwritten Text Recognition system transcribes cursive handwritten characters or phrases to a computer-readable digital representation. Offline HTR has attracted intense academic and commercial interest over the years due to its vast application, such as reading postal addresses, bank checks, verifying signatures, writer and license plate recognition, digitalizing historical records and manuscripts, and so on. Recognition of Bangla text is much harder than other languages like English because of the shape difference of the Bangla characters and their complexity. The Bangla script has 39 consonants, 11 vowels, 10 modifiers and 334 compound characters, this makes recognition much harder.

Numerous HTR models have been built over the years, with Hidden Markov Models (HMM) [1] being one of the first used recognition techniques. However, because of the Markovian assumption that each observation is dependent on the current state, HMM was unable to utilize the context information, especially in a long text sequence. In recent years, deep learning-based models such as Convolutional Recurrent Neural Networks (CRNN) improved the recognition rate a lot compared to other traditional architectures for HTR. Gated Recurrent Units (GRUs) and Long Short Term Memories (LSTMs) are commonly used sequence decoders in CRNN. However, while processing very long texts, the recurrent layers continue to suffer from vanishing gradient issues. Millions of trainable parameters are common in high-performance models, which increases computing costs. Recently, Gated-CNN models achieved a higher accuracy rate with a lower number of trainable parameters than CRNN architectures [2].

Even though there has been a significant advancement in the field of HTR, there isn't a lot of work available for Bangla text recognition. The majority of these studies focus on character-level recognition [3]. In this study, we propose an end-to-end system for Bangla text recognition from images on word-level. We compared several deep learning-based CRNN and Gated-CNN architectures on the BanglaWriting dataset. We adopted popular pre-trained CNN architectures including Xception, MobileNet, and DenseNet with LSTM or GRU as recurrent layers. We implemented two commonly used English HTR optical models, one is a CRNN-based design suggested by Puigcerver [4], and the other is based on Flor's [5] Gated-CNN approach.

The paper is organized in the following sequence. Section II discusses some previous studies in this field. Section III discusses the proposed methodology. The results of the experiment is described in Section IV. Finally, in Section V, conclusions are drawn.

II. Literature Review

A few notable works are available for Bangla HTR; most of those works concentrate primarily on Bangla numeric and basic character recognition. Pal and Chaudhuri [6] developed a new approach for handwritten Bangla character recognition that was based on feature extraction. The proposed approach was evaluated on data collected from persons of various backgrounds, with an overall recognition accuracy of about 91.98% using the concept of water overflow from the reservoir.

Purkaystha et al. [3] performed a deep convolutional neural network to create a Bangla handwritten character recognition

model. On the BanglaLekha-Isolated dataset, they achieved 98.66% accuracy on numerals, 94.99% accuracy on vowels, 91.23% accuracy on alphabets, 91.60% accuracy on compound letters (20 character classes), and 89.93% accuracy on nearly all Bangla characters (80 character classes).

A handwritten Bangla elemental and compound character recognition model using multilayer perceptron (MLP) and support vector classification (SVM) classifier was proposed by Das et al. [7]. They achieved around 79.73% and 80.9% of recognition rate for MLP and SVM, respectively.

Alom et al. [8] evaluated a set of state-of-the-art deep convolutional neural networks (DCNNs) on the application of Bangla handwritten character recognition. For constructing a real-life application on an automatic HBCR system, they validated the strengths of DCNN models like DenseNet, FractalNet, and ResNet over other common object recognition models where DenseNet achieved the highest level of accuracy.

Rabby et al. [9] introduced the EkushNet model to recognize 50 fundamental letters, 10 digits, 10 modifiers, and 52 often used compound characters in Bangla handwriting. The suggested technique has a recognition accuracy of 97.73 percent for the Ekush dataset and a cross-validation accuracy of 95.01 percent for the CMATERdb dataset.

Most of the previous works on Bangla text recognition are based on character-level classification. In this paper, we propose an end-to-end Bangla HTR system that recognizes text from word-level images.

III. Proposed Methodology

Our approach uses word-level images to recognize handwritten Bangla texts. Each word image is preprocessed before being fed into an optical model, where the loss is calculated and trained using the CTC [10] loss function. The output of the optical model is then decoded to obtain digital Bangla texts. Figure 1 demonstrates the suggested system flow.

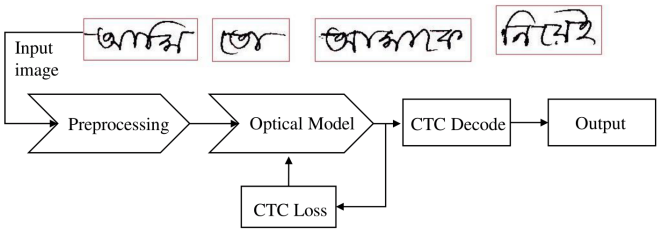

Fig. 1: The proposed system in a nutshell.

A. Dataset

We used the BanglaWriting dataset [11] to carry out this experiment.The dataset contains single-page handwritings of 260 individuals, where each page includes bounding-boxes of each word as well as their Unicode representation. There are 21,234 words in this dataset, with 5,470 of them being unique. However, we used 20,736 word images to perform the experiment, yielding 91 distinct Bangla characters.

B. Preprocessing

We excluded images with word sizes of more than 10 characters. The following strategies have been adopted to reduce variations in images and writing:

- Reshaping: each word image is first reshaped into a width of 256 by a height of 64 pixels with padding. For pre-trained models, an RGB image of 3 channels was used.
- Normalization: each reshaped image is then normalized in the range of [0, 1], since data normalization ensures that every input parameter has a consistent data distribution.

$$Normalize(D) = \begin{bmatrix} d_{11} & \cdots & d_{1m} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & d_{nm} \end{bmatrix} /255.0 \qquad (1)$$

- Augmentation: in order to increase the variety of trainable data and avoid overfitting during the training of optical models.

C. Optical Models

Handwritten text recognition systems explored in this paper operates in three steps: i) input image is fed into the CNN layers to extract features, ii) the information from CNN is then propagated via the RNN layers, which map the characteristics in both directions through the sequence, and lastly iii), the Connectionist Temporal Classification (CTC) generates the loss value while training and decodes the optical model output into the final text for model inference.

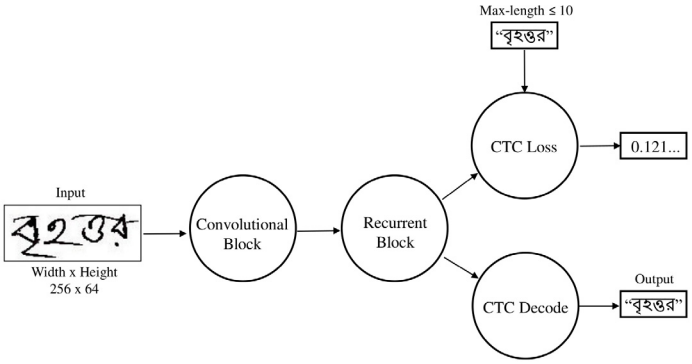

Fig. 2: A diagram of the data flow through a simple CRNN network.

1) Pre-trained models: We experimented with three different pre-trained CNN architectures as baseline CNN layers and a stack of bidirectional RNN layers. The input word image of shape $256 \times 64 \times 3$ is passed to the baseline CNN block. We adjusted the pre-trained convolutional models by removing several layers to ensure the propagation of 32 time-steps in RNN layers. The optical module of our HTR system's recurrent block is composed of 2 BGRU or BLSTM with dropout (rate 0.5) alternating by a dense layer. For the first RNN layer, the number of hidden units is set to 128 and 64 for the second, and Units in the dense layer are equal to the charset size plus one (CTC blank symbol).

TABLE I: Benchmarks for different HTR architectures along with different RNN categories for pre-trained CNN baseline architectures.

Model			Validation		Test	
Baseline	RNN	# of params	CER%	WER%	CER%	WER%
Xception	LSTM	20.9 M	26.03	58.62	25.78	57.72
	GRU	19.3 M	27.63	60.86	28.46	62.86
MobileNet	LSTM	2.5 M	35.51	69.96	37.76	72.27
	GRU	1.9 M	45.23	77.93	47.13	78.78
DenseNet121	LSTM	5.8 M	17.97	47.17	18.68	48.63
	GRU	4.7 M	18.99	49.46	19.88	50.56
Puigcerver		8.3 M	17.01	40.29	16.94	40.92
Flor		**0.8 M**	**13.58**	**37.39**	**12.83**	**36.01**

a) MobileNet:
MobileNet [12] is lightweight convolutional architecture pre-trained on ImageNet dataset. We used layers till the 5^{th} depthwise convolution block out of 13 depthwise separable convolution blocks.

b) Xception:
Xception [13] is a convolutional neural network architecture based on convolution layers that are depthwise separable. It's an extreme version of another popular pre-trained convolutional architecture, Inception. We used layers till 'block13_sepconv2_act' named in Keras framework.

c) DenseNet121:
DenseNet121 [14] is also a convolutional architecture pre-trained on the ImageNet dataset in which each layer is connected to deeper layers. We used the first 2 dense blocks out of 4 dense blocks of this architecture.

2) Puigcerver: Puigcerver's architecture [4] adopts a traditional CRNN approach, with a large number of trainable parameters. The convolutional block of this architecture includes five 2D convolutional layers with 3×3 kernels, and the number of filters per layer equals 16n at the n^{th} layer. Five bidirectional LSTMs with dropout (rate 0.5) in LSTM cells make up the recurrent block. In all LSTMs, the number of hidden units is specified to 256.

3) Flor: Flor's architecture [5] aims to have higher precision with fewer trainable parameters based on Gated-CNN approach. The convolutional block is comprised of five mini-blocks, each having traditional and gated convolutions, followed by a convolutional layer at the end. The recurrent block of this architecture consists in 2 BGRU each with 128 hidden units with dropout (probability 0.5), ending with a dense layer.

D. Loss calculation and decoding

We used Connectionist Temporal Classification (CTC) function [10] to calculate loss and minimize validation loss while training. The loss is computed by adding all of the scores of every possible alignment of the ground truth text, regardless of where it appears in the image. Also, Vanilla Beam Search algorithm [15] was used for decoding the ground truth text.

IV. Experimental Analysis

A. Experimental Setup

The optical model was implemented using Tensorflow, Numpy, and Python. We used the Albumentation [16] library to augment the dataset. The training of the model was done with the goal of minimizing the validation loss value, which was determined using the CTC loss function [10]. We divided the dataset into training (18,663 words), validation (829 words), and test (1,244 words) subsets. In addition, if the validation loss value did not improve after 5 epochs, the early stopping technique was applied to stop training the model and reduce learning rate on plateau (factor 0.2) technique was used with 20 epochs. In inference mode (beam width = 10), the Vanilla Beam Search algorithm [15] was implemented as the CTC decode function. We also used the Adam optimizer with a learning rate of 0.001 and 32 samples per step in mini-batches.

B. Evaluation Metrics

The performance of our word-level HTR system was measured using the two most often used evaluation metrics, WER and CER. i) Word Error Rate (WER): defined as the minimum number of words that need to be inserted, substituted, or deleted to match the recognition output with the corresponding reference ground truth text, divided by the total number of words in the reference transcripts. ii) Character Error Rate (CER): defined similarly as WER but at the character level. For both characters and words, these metrics were determined using the Levenshtein Distance [17] between ground truth and predictions.

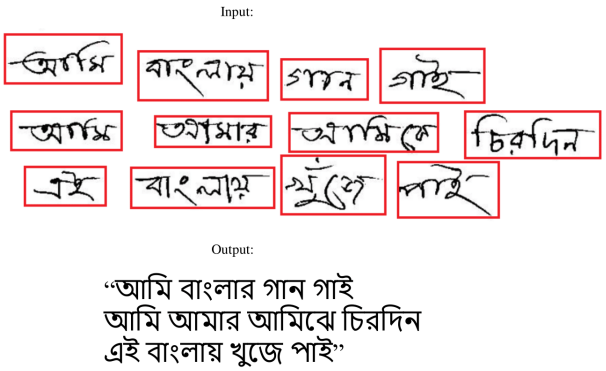

Fig. 3: A handwritten image with word-level bounding boxes and its corresponding output using Flor optical model architecture.

C. Results

Table I shows the error rates of different optical model architectures used in this study. Among the pre-trained baseline CNN architectures, Densenet121 with LSTM recurrent block achieves the best result with 18.68% CER and 48.63% WER on the test set. Puigcerver's architecture, which is a popular CRNN architecture with a high recognition rate for English HTR, bettered the results, achieving 16.94% CER and 40.92% WER. But Flor architecture has the best recognition rate for Bangla characters with CER of 12.83% and WER of 36.01%.

Another significant need for deep neural networks is the architecture's complexity, which has an impact on the model's size and decoding time. Flor architecture stands out the most, having the lowest number of parameters (0.8 million) compared to all other architectures, thus requiring less computational resources and providing faster text recognition.

V. Conclusions

In this study, we demonstrated an end-to-end HTR system that detects Bangla text from handwritten word images. The benchmark experiment adopted the same methodology for optical models under the BanglaWriting dataset. For CRNN based approach, we used three distinct pre-trained convolutional architectures (MobileNet, Xception, DenseNet121) and implemented an architecture proposed by Puigcerver. And for the Gated-CNN approach, we used Flor architecture. The Flor architecture is the lightest having the least amount of parameters. Flor's optical model achieves the best results with 12.83% CER and 36.01% WER. Error rates in Bangla handwriting recognition are higher compared to English because of the existence of cursive punctuation marks and connected letters in the Bangla language. In the future, we would like to develop an optical model specifically built for Bangla handwriting recognition which will hopefully provide better recognition rates than the current ones we used. We intend to incorporate language models and spelling corrections in order to further reduce the error produced by the optical model.

References

[1] S. R. Eddy, "Hidden markov models," *Current opinion in structural biology*, vol. 6, no. 3, pp. 361–365, 1996.

[2] T. Bluche and R. Messina, "Gated convolutional recurrent neural networks for multilingual handwriting recognition," in *2017 14th IAPR international conference on document analysis and recognition (ICDAR)*, vol. 1. IEEE, 2017, pp. 646–651.

[3] B. Purkaystha, T. Datta, and M. S. Islam, "Bengali handwritten character recognition using deep convolutional neural network," in *2017 20th International conference of computer and information technology (ICCIT)*. IEEE, 2017, pp. 1–5.

[4] J. Puigcerver, "Are multidimensional recurrent layers really necessary for handwritten text recognition?" in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 1. IEEE, 2017, pp. 67–72.

[5] A. F. de Sousa Neto, B. L. D. Bezerra, A. H. Toselli, and E. B. Lima, "Htr-flor: a deep learning system for offline handwritten text recognition," in *2020 33rd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*. IEEE, 2020, pp. 54–61.

[6] U. Pal and B. Chaudhuri, "Automatic recognition of unconstrained off-line bangla handwritten numerals," in *International Conference on Multimodal Interfaces*. Springer, 2000, pp. 371–378.

[7] N. Das, B. Das, R. Sarkar, S. Basu, M. Kundu, and M. Nasipuri, "Handwritten bangla basic and compound character recognition using mlp and svm classifier," *arXiv preprint arXiv:1002.4040*, 2010.

[8] M. Z. Alom, P. Sidike, M. Hasan, T. M. Taha, and V. K. Asari, "Handwritten bangla character recognition using the state-of-the-art deep convolutional neural networks," *Computational intelligence and neuroscience*, vol. 2018, 2018.

[9] A. S. A. Rabby, S. Haque, S. Abujar, and S. A. Hossain, "Ekushnet: using convolutional neural network for bangla handwritten recognition," *Procedia computer science*, vol. 143, pp. 603–610, 2018.

[10] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 369–376.

[11] M. F. Mridha, A. Q. Ohi, M. A. Ali, M. I. Emon, and M. M. Kabir, "Banglawriting: A multi-purpose offline bangla handwriting dataset," *Data in Brief*, vol. 34, p. 106633, 2021.

[12] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[13] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1251–1258.

[14] F. Iandola, M. Moskewicz, S. Karayev, R. Girshick, T. Darrell, and K. Keutzer, "Densenet: Implementing efficient convnet descriptor pyramids," *arXiv preprint arXiv:1404.1869*, 2014.

[15] K. Hwang and W. Sung, "Character-level incremental speech recognition with recurrent neural networks," in *2016 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2016, pp. 5335–5339.

[16] A. Buslaev, V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, and A. A. Kalinin, "Albumentations: fast and flexible image augmentations," *Information*, vol. 11, no. 2, p. 125, 2020.

[17] V. I. Levenshtein *et al.*, "Binary codes capable of correcting deletions, insertions, and reversals," in *Soviet physics doklady*, vol. 10, no. 8. Soviet Union, 1966, pp. 707–710.