Offline Bengali Handwritten Sentence Recognition using BiLSTM and CTC Networks

Preprint · July 2020

DOI: 10.13140/RG.2.2.20616.03846

M. A. Muhaimin Sakib
Chittagong University of Engineering & Technology
1 PUBLICATION **0** CITATIONS

SEE PROFILE

Omar Sharif
Chittagong University of Engineering & Technology
4 PUBLICATIONS **0** CITATIONS

SEE PROFILE

Moshiul Hoque
Chittagong University of Engineering & Technology
61 PUBLICATIONS **201** CITATIONS

SEE PROFILE

Offline Bengali Handwritten Sentence Recognition using BiLSTM and CTC Networks

M. A. Muhaimin Sakib, Omar Sharif, and Mohammed Moshiul Hoque

Department of Computer Science and Engineering
Chittagong University of Engineering and Technology
Chittagong-4349, Bangladesh
{muhaimin.sakib, omar.sharif, moshiul_240}@cuet.ac.bd

Abstract It is a very challenging task to recognize unconstrained Bengali handwritten text due to its cursive nature. This paper introduces an offline technique of recognizing handwritten Bengali sentences based on BiLSTM architecture and connectionist temporal classification (CTC) output layer. The traditional approach of detecting handwritten sentence recognition rises the computational complexity due to the difficulty of combining isolated characters. The proposed method only segments the sentences into words, recognizes separately and combines them to represents the output sentence. The proposed technique is evaluated with 450 Bengali sentences and achieved 92% accuracy for recognizing sentences considering 15 distinct writing styles.

Keywords: Pattern recognition, Handwritten recognition, Word separation, Feature extraction, Bidirectional long short term memory

1 Introduction

Recognizing handwriting is the specialized research issue in the scope of pattern recognition which recognizes handwritten texts by analyzing various patterns observed in hand written scripts. This is the aptitude of computers to understand and translate handwritten text into electronic form or data. The advantage of transforming physical written documents into electronic form is that digital documents are more comfortable to store, retrieve and manipulate. Besides this, it is easy to store, edit and access digital document compared to a handwritten text document. Moreover, physical documents take a tremendous amount of office space, and much valuable knowledge gets lost because documents never get translated into digital format. Although significant advancements have been made in recognizing handwritten in English and other languages, it is in the initial stage in Bengali language [8]. The process of handwritten identification can be segregated into two schemes: online and offline. In the online approach, the motion of pen tip is recorded and using those coordinates the handwriting is recognized [5]. Only an image containing the handwritten text provides the necessary information and the offline technique have to identify from the image [1]. This paper concerned on the offline approach.

A significant amount of work has conduced to recognize Bengali handwritten digits, vowels, and consonants using fuzzy logic and neural network. This research proposes a deep neural network architecture due to its large number of features consideration and better accuracy than the general neural network architectures [16]. The task of recognizing Bengali handwriting is quite challenging due to its various writing styles, shortage of resources, presence of uncorrelated segments and broken characters. Furthermore, characters in a word connected by a line called ”Matra” in Bengali handwriting which makes them more challenging to segment the words into isolated characters. To the best of our knowledge, no significant advances have been made so far in recognizing handwritten Bengali sentences. The suggested technique present in this paper can recognize Bengali handwritten sentences by segmenting it into constituent words and then recognize the individual words without segmenting them into constituent characters.

2 Related Work

There are no significant research conducted yet to recognize the Bengali handwritten sentences several approaches are already available that deals with the recognition of handwritten Bengali digits, characters and words respectively [1, 15, 17].

There are several methods for recognizing handwritten characters. Some methods are used fuzzy logic [10, 6] while others use neural networks [3, 7]. There are also methods using genetic algorithms described in [9, 11]. Alom et al. [3, 2] used different deep learning techniques to recognize Bengali handwritten digits and characters. Digit recognition system gets maximum 98.78% accuracy by using CNN with dropout and Gabor filters. Deep convolution neural networks can recognize characters with approximately 98% accuracy. The system proposed by Adak et al. [1] is achieved 86.96% accuracy in detecting handwritten cursive Bangla words. Performance of this system can be enhanced by employing the recurrent neural network. A new method can identify handwritten Bengali words using a set of fuzzy linguistic rules [6]. The online handwritten recognition system has proposed by Bhattacharya et al. [5], which used HMM-based classifier and N-pen++ features. This method achieved 92.33% accuracy with 20000 words. Recent work focused on LSTM and CNN models to classify handwritten English words [4]. This work used CNN model to classify words and LSTM model to segment the characters. An automated system that detects English sentence by analyzing gestures of the mouse in real-time based on ANN with notable accuracy shown in [13]. Graves et al. [7] presented an alternative method for sequence labelling tasks where data have long-range bidirectional interdependencies and hard to segment based on novel type RNN technique. Majority of the existing work focused on recognition of Bengali digits, characters or words. None of the previous work focused to recognise Bengali handwritten sentences. This work proposes a machine learning-based framework that can recognize the handwritten Bengali sentences.

3 Proposed Handwriting Recognizer

Fig. 1 represents the proposed framework of Bengali handwritten sentence recognition. The framework consists of two main parts: learning and recognition. Both of them consists of several smaller parts which explained below.

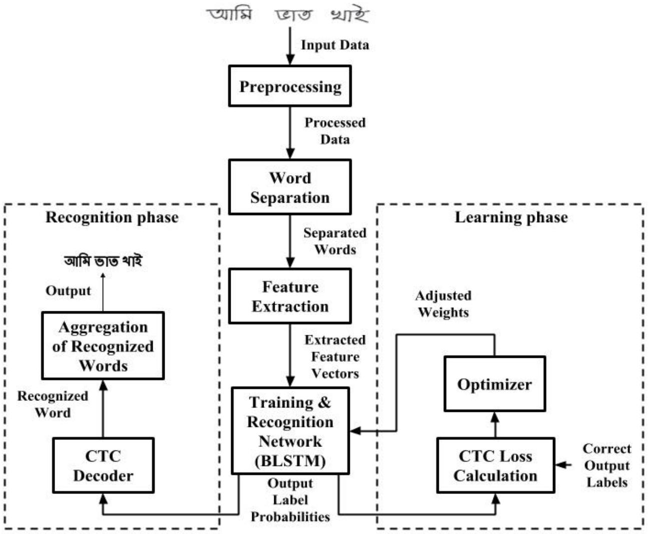


Figure 1. Proposed framework for recognizing Bengali handwritten sentences

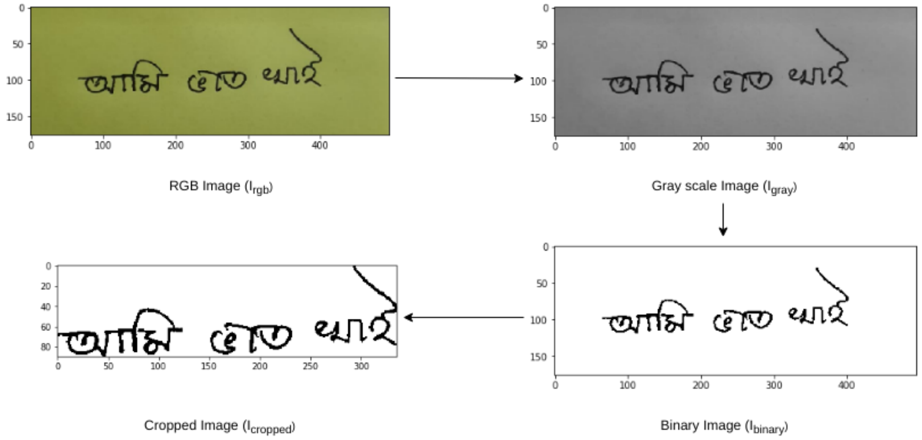
3.1 Preprocessing

RGB image of the handwritten sentence takes as the input of the suggested technique. The input image is further processed to extract relevant features by applying following steps. **Step 1:** the input RGB image I_{rgb} is translated to gray-scale image I_{gray} [14]. **Step 2:** the gray-scale image I_{gray} is transformed to binary image I_{binary} using Otsu's method [12]. **Step 3:** the binary image I_{binary} is cropped so that there are no redundant pixels at the beginning, end, top or bottom of the image. These steps are summarized in Fig. 2.

3.2 Word Separation

The column wise density of black pixel (d_{col}) is considered to separate words in an image which is done in following two steps.

- Step 1: a sliding window of height (H = image height), width ($W = 1$ pixel) is taken. This window is slided over the image (I) to right for calculating black pixel counts (C_{black}) in each window position. Pseudo-code for this step is shown in Algorithm 1.



RGB Image (I_{rgb})

Gray scale Image (I_{gray})

Cropped Image ($I_{cropped}$)

Binary Image (I_{binary})

Figure 2. Output images after preprocessing the sentence 'আমি ভাত খাই '

Algorithm 1 Count the number of black pixels in a column

```
1: Initialize variables, $i \leftarrow 0, L \leftarrow image\_width$
2: Calculate $C_{black}$ in $i^{th}$ column
3: if $i = L - 1$ then
4:     exit
5: else
6:     $i \leftarrow i + 1$
7:     go to step 2
8: end if
```

- Step 2: count the number of consecutive window positions C_{window} where $C_{black} = 0$. The image is segmented if the following equation holds,

$$C_{window} \geq T_{max} \tag{1}$$

Here T_{max} is the maximum value C_{window} can have before we segment the image. The success of these algorithms depends on the following two assumptions,

$$D_{min} \geq T_{max}, D_{max} < T_{max} \tag{2}$$

Here, D_{min} is the minimum distance between consecutive words and D_{max} is the maximum distance between consecutive characters. Algorithm 2 shows the pseudo-code for performing segmentation and the word separation process is depicted in Fig. 3

3.3 Feature Extraction

The feature vectors enumerated for each word image W_{img}. Nine features are calculated in each window position by using a sliding window along the width of the image.The mean number of black pixels at each window position, the centre

Algorithm 2 Word Segmentation

```
1:  Initialize variables, $i \leftarrow 0, C_{window} \leftarrow 0$
2:  $C_{black} \leftarrow$ *no. of black pixels at $i^{th}$ column*
3:  **if** $C_{black} = 0$ **then**
4:      $C_{window} \geq C_{window} + 1$
5:  **else**
6:      $C_{window} \leftarrow 0$
7:  **end if**
8:  **if** $C_{window} = T_{max}$ **then**
9:      *segment the image*
10: **else**
11:     $i \leftarrow i + 1$
12: **end if**
13: **if** $i = image\_width$ **then**
14:     *exit*
15: **else**
16:     *go to step* 2
17: **end if**
```

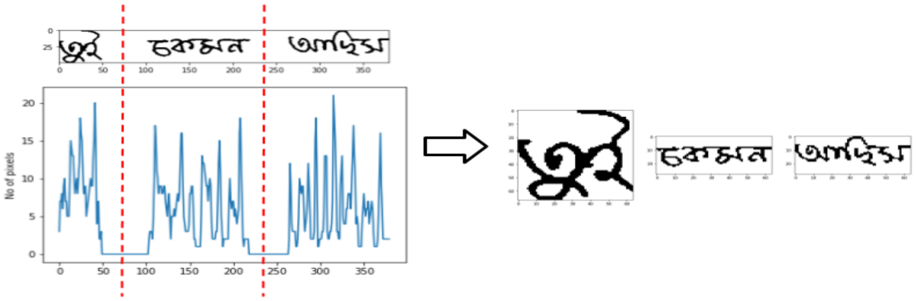


Figure 3. Segmentation of S_{img} (left image) into W_{img} (right image).

of gravity of the pixels, the second-order vertical moment of the centre of gravity and the positions of the lowermost and uppermost black pixels are calculated by using Eqs. 3-6 respectively.

$$f_1(t) = \frac{1}{m} \sum_{y=1}^{m} p(t, y) \tag{3}$$

$$f_2(t) = \frac{1}{m} \sum_{y=1}^{m} y.p(t, y) \tag{4}$$

$$f_3(t) = \frac{1}{m^2} \sum_{y=1}^{m} y^2.p(t, y) \tag{5}$$

$$f_4(t) = y_u, f_5(t) = y_l \tag{6}$$

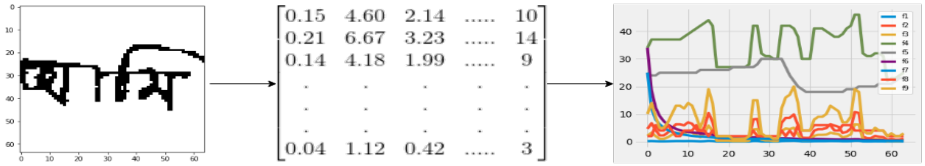
Eqs. 7-9 calculates gradient of uppermost and lowermost black pixels, number of black-white transitions and number of black pixels inside the window respectively.

$$f_6(t) = \frac{f_4(t)}{t}, f_7(t) = \frac{f_5(t)}{t} \tag{7}$$

$$f_8(t) = \sum_{y=1}^{m}(p(t,y) \oplus p(t,y-1)) \tag{8}$$

$$f_9(t) = \sum_{y=1}^{m} p(t,y) \tag{9}$$

Figure 4 illustrates the visual representation of features.



(a) Sample word image

(b) Tabular representation of the features

(c) Graphical representation of the features

Figure 4. Extracted feature vectors form input image of word 'আমি '.

3.4 Training and Recognition Network

The input to the BiLSTM network is a (64 × 9) matrix. This layer produces a probability distribution for each of the 59(+blank) character present in Bengali alphabet. Therefore, the output of this network is a 64 × 60 matrix which is fed to the CTC layer. Fig. 5 illustrates the input and corresponding output of the BiLSTM network.

Learning Phase The system learns using the output of the BiLSTM network and weights of the network are adjusted to enhance the recognition performance. This phase consist of two basic parts: CTC loss and optimizer.

- **CTC loss Calculation:** CTC loss function calculate the loss using the output matrix of the BiLSTM network and the correct output labels. The loss value is calculated as the negative *log* of the sum of probability of all paths that correspond to the correct sequence of labels.

$$loss = -\ln \sum_{i=1}^{x} p(q_i) \tag{10}$$

  Loss value is calculated by equation 10 where $q_1, q_2, q_3..., q_x$ are all the paths that correspond to correct sequence of labels.

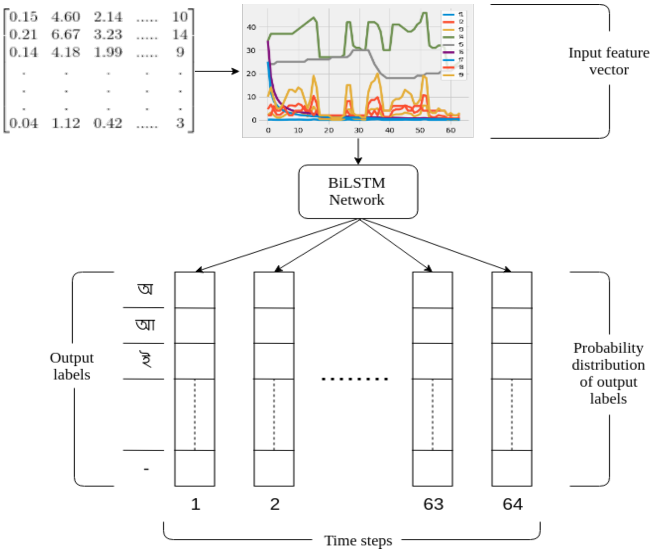


Figure 5. Input and output of BiLSTM network.

- **Optimizer:** The optimizer try to minimize loss as well as adjust the weights of the BiLSTM network by considering the calculated CTC loss. The adjusted weights of the network is enables it to perform better.

Recognition Phase In preprocessing step, a sentence image S_{img} is converted into several word images $w_1, w_2, ..., w_l$. The proposed system is recognize these word images w_i to produce final output which is the recognised sentence. The recognition is done in two steps: CTC decoding and aggregation.

- **CTC Decoder:** An unknown input sequence u represents the probability distribution of output labels, produced by the BiLSTM network. This is translated by CTC decoder which choose the sequence of labels L^* with the maximum conditional probability calculated by equation 11 in BiLSTM network.

$$L^* = \max_L p(L|u) \tag{11}$$

  This sequence of labels represented by L^* represents our recognised word.
- **Aggregation of Words:** the recognized words $L^*_{w1}, L^*_{w2}, ..., L^*_{wl}$ are concatenated using a space character (G) in between them to form final output sentence F_s of length S_l. Where S_l represents the number of words the sentence consists of. Using equation 12 we get our final final sentence by aggregating words.

$$F_s = \begin{cases} \sum \mathrm{L}^*_{wi} + \mathrm{G} & \text{if } i < S_l \\ \sum \mathrm{L}^*_{wi} & \text{if } i = S_l \end{cases} \tag{12}$$

4 Evaluation Measures

The proposed system is evaluated at word level as well as sentence level in terms of edit distance, segmentation error rate, label error rate, word accuracy, and sentence accuracy respectively.

- **Edit Distance:** It is the deviation of output sequence ($w_1w_2..w_n$) from the actual sequence of labels ($g_1g_2..g_n$). It can be expressed as in Eq. 13.

$$E_d = \gamma(w_1w_2..w_n, g_1g_2..g_n) \tag{13}$$

  This returns the minimum number of substitution, insertion or deletion required to convert $w_1w_2..w_n$ to $g_1g_2..g_n$.
- **Segmentation Error Rate:** It denotes the percentage of sentences that the system failed to correctly segment. If total number of sentence is T_s and number of correctly segmented sentence is C_s then segmentation error S_e is given by Eq. 14.

$$S_e = \frac{C_s}{T_s} \tag{14}$$

- **Label Error Rate:** It represents the percentage of characters that the system failed to correctly identify in a sentence. If R is a sample set of recognized words and X is the set of actual words then label error rate (L_e) is found by Eq. 15.

$$L_e = \sum_{x \in R, y \in X} \frac{\gamma(x, y)}{len(y)} \tag{15}$$

  Here, $len(y)$ gives the number of characters in word y.
- **Word Accuracy:** It gives the percentage of words that the system able to correctly predict. A word w is considered correctly predicted if $E_d = 0$.
- **Sentence Accuracy:** It represents the percentage of correctly predicted sentences. Sentence accuracy S_a is calculated by the ratio of correctly predicted sentence C_p and total number of sentences T_s.

5 Experimental Results

Due to scarcity of resources and benchmark dataset on Bengali handwritten we developed a dataset. The summary of the dataset shown in the Table 1.

The proposed system trained the classifier on the word level. The BiLSTM-CTC model used in this experimentation consists of three layers, each with 64 BiLSTM cells. Each BiLSTM cell consists of 512 hidden nodes. The network is trained for 4000 iterations on the training set before evaluation. We used RMSProp optimizer to adjust the weights of our network, and for decoding the, we used CTC beam search decoding algorithm with a beamwidth of 80. Table 1 also indicates that the overall accuracy of the proposed technique at the sentence level. The result indicates that the system can predict the Bengali handwritten sentences with 92% accuracy.

Table 1. Dataset Statistics

	Training	Testing	Accuracy
Number of sentences	300	150	92%
Number of words	930	465	
Avg. no. of words per sentence	3.7	3.5	
Writing Styles	15	15	
Age range	19 - 24	19-24	

The system also evaluated by considering different sentence lengths and different writing styles to see how accuracy varies depending on various contexts. The performance of the system for four different sentence lengths shown in Fig. 6. This result indicates that as the length of the sentence increases, segmentation error also increases and accuracy decreases as expected.

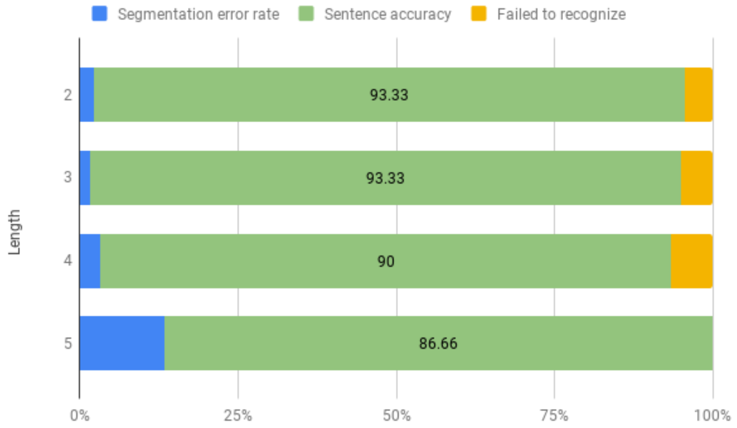
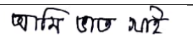
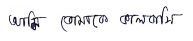
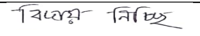
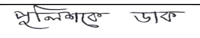
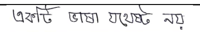
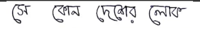
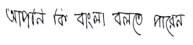
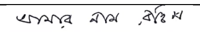
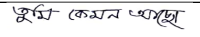
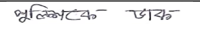
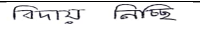
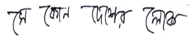
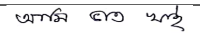


Figure 6. Accuracy versus sentence length.

Table 2 shows samples of 15 different writing styles and their corresponding accuracy. We can see that the system performs well for most of the writing styles. For the styles 10 and 14, the system recognized only 70% and 50% of the sentences respectively.

Table 2: Performance of the system for various writing styles.

Style	Sample images	Recognized sentence	Word accuracy	Sentence accuracy
1		আমি ভাত খাই	100%	100%

2		খটি তোমাকে ভালবাসি	96.66%	90%
3		বিশের নিচ্ছি	91.66%	80%
4		পুলিশকে ডাক	100%	100%
5		একটি ভাষা যথেষ্ট নয়	100%	100%
6		সে কোন দেশের লোক	100%	100%
7		আপনি কি বাংলা বলতে পারেন	100%	100%
8		মাফ করবেন	100%	100%
9		আমার নাম রহিম	100%	90%
10		তুমি কেমন আছো	100%	70%
11		পুলিশকে ডাক	100%	100%
12		বিদায় নিচ্ছি	100%	100%
13		মাফ করবেন	100%	100%
14		সে কেমন তোোন লোক	79.63%	50%
15		আমি ভাত খাই	100%	100%

A comparative analysis is performed between the proposed technique and CNN. The CNN architecture consisted of 3 convolution, 2 max pooling, 2 fully connected and a dropout layers. Table 3 displays the analysis of comparison

Table 3. Comparative analysis between BiLSTM-CTC and CNN at word level.

Architecture	Total words	Correctly predicted	Word accuracy
BiLSTM-CTC	465	378	81.29%
CNN	465	351	75.48%

The result shows that BiLSTM-CTC networks are more accurate to predict the Bengali handwritten sentences than CNN. In BiLSTM-CTC, the output nodes and labels are in equal numbers for the Bengali language. On the other hand, the CNN output layer considers as many nodes as there are words in Bengali vocabulary which can be quite large.

6 Conclusion

This work presents a deep neural network-based technique that can recognize the Bengali handwritten sentences. The evaluation results indicated that BiLSTM-CTC network is more effective in recognizing Bengali handwritten sentences.

This architecture enables the system to recognize words without segmenting them into constituent characters. The overall accuracy of the system can be enhanced by using a different set of features and increase the number of handwriting styles in the training dataset. This present implementation can be extended to recognize handwritten paragraphs by including more features in preprocessing and segmentation phases.

References

1. Adak, C., Chaudhuri, B.B., Blumenstein, M.: Offline cursive bengali word recognition using cnns with a recurrent model. In: 2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR), pp. 429–434. IEEE (2016)
2. Alom, M.Z., Sidike, P., Hasan, M., Taha, T.M., Asari, V.K.: Handwritten bangla character recognition using the state-of-the-art deep convolutional neural networks. Computational intelligence and neuroscience **2018** (2018)
3. Alom, M.Z., Sidike, P., Taha, T.M., Asari, V.K.: Handwritten bangla digit recognition using deep learning. arXiv preprint arXiv:1705.02680 (2017)
4. Balci, B., Saadati, D., Shiferaw, D.: Handwritten text recognition using deep learning. CS231n: Convolutional Neural Networks for Visual Recognition, Stanford University, Course Project Report, Spring (2017)
5. Bhattacharya, N., Roy, P.P., Pal, U., Setua, S.K.: Online bangla handwritten word recognition. Malaysian Journal of Computer Science **31**(4), 300–310 (2018)
6. Chowdhury, K., Alam, L., Sarmin, S., Arefin, S., Hoque, M.M.: A fuzzy features based online handwritten bangla word recognition framework. In: Computer and Information Technology (ICCIT), 2015 18th International Conference on, pp. 484–489. IEEE (2015)
7. Graves, A., Liwicki, M., Fernández, S., Bertolami, R., Bunke, H., Schmidhuber, J.: A novel connectionist system for unconstrained handwriting recognition. IEEE transactions on pattern analysis and machine intelligence **31**(5), 855–868 (2009)
8. Hussain, R., Raza, A., Siddiqi, I., Khurshid, K., Djeddi, C.: A comprehensive survey of handwritten document benchmarks: structure, usage and evaluation. EURASIP Journal on Image and Video Processing **2015**(1), 46 (2015)
9. Kala, R., Vazirani, H., Shukla, A., Tiwari, R.: Offline handwriting recognition using genetic algorithm. arXiv preprint arXiv:1004.3257 (2010)
10. Malaviya, A., Peters, L.: Extracting meaningful handwriting features with fuzzy aggregation method. In: Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on, vol. 2, pp. 841–844. IEEE (1995)
11. Oliveira, L.S., Sabourin, R., Bortolozzi, F., Suen, C.Y.: Feature selection using multi-objective genetic algorithms for handwritten digit recognition. In: null, p. 10568. IEEE (2002)
12. Otsu, N.: A threshold selection method from gray-level histograms. IEEE transactions on systems, man, and cybernetics **9**(1), 62–66 (1979)
13. Parwej, F.: English sentence recognition using artificial neural network through mouse-based gestures. arXiv preprint arXiv:1301.4659 (2013)
14. Saravanan, C.: Color image to grayscale image conversion. In: 2010 Second International Conference on Computer Engineering and Applications, vol. 2, pp. 196–199. IEEE (2010)

15. Sazal, M.M.R., Biswas, S.K., Amin, M.F., Murase, K.: Bangla handwritten character recognition using deep belief network. In: 2013 International Conference on Electrical Information and Communication Technology (EICT), pp. 1–5. IEEE (2014)
16. Schmidhuber, J.: Deep learning in neural networks: An overview. Neural networks **61**, 85–117 (2015)
17. Shawon, A., Rahman, M.J.U., Mahmud, F., Zaman, M.A.: Bangla handwritten digit recognition using deep cnn for large and unbiased dataset. In: 2018 International Conference on Bangla Speech and Language Processing (ICBSLP), pp. 1–6. IEEE (2018)