2023 26th International Conference on Computer and Information Technology (ICCIT) | 979-8-3503-5901-5/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICCIT60459.2023.10441393

# Open Vocabulary Word Recognition From Transcribed Bangla Texts

Faias Satter*, Sk. Md. Masudul Ahsan‡
Department of Computer Science and Engineering
Khulna University of Engineering & Technology
Khulna-9203, Bangladesh
Email: *faiaspromit@gmail.com ‡smahsan@cse.kuet.ac.bd

***Abstract*—An optical character recognition (OCR) can scan a paper and extract text using technology, making people's jobs easier. While various OCR systems are available in the software industry, finding a reliable equivalent solution for Bangla takes much work. When it comes to handwritten texts, the situation is much more unusual. Recognizing words from word images is the most critical stage in any OCR process. It is the second stage after segmenting words from text pictures. If this stage fails, the overall performance of the OCR will be poor, regardless of how well the other phases perform. This study aims to recognize words using deep learning in a handwritten Bangla word image. Three object detection models, SSD with MobileNetV2, Faster R-CNN with InceptionResNetV2, and an ensemble model of these two, have been used to train and test handwritten word images. A modified Non-Maximum Suppression has been introduced to enhance the effectiveness of the models' results. A customized dataset of 9841 handwritten Bangla word images has been compiled, featuring diverse handwriting styles from various individuals. All three models' performances have been checked against the test dataset, and the ensemble model has been the most impressive, with an F1-score of 92.61%. Also, at the word level, the ensemble model correctly recognizes 96.12% of the words to some extent. The system can be further improved by introducing a post-processing phase to correct errors generated by the system.**

***Index Terms*—Bangla OCR, Handwritten Bangla word recognition, Deep learning, Optical character recognition (OCR), Word segmentation, Word image recognition, Ensemble model, Character recognition, Bangla script, Word-level recognition**

## I. Introduction

One of the computer vision applications is optical character recognition (OCR), which translates images of typed/handwritten/printed text into machine-encoded text. OCR is commonly used to digitize texts in images to be electronically modified, searched, stored more compactly, and used in machine processes such as artificial intelligence, machine translation, text-to-speech, and document analysis and information extraction. However, there is currently no good OCR for the Bangla language. The most important stage in every OCR system is recognizing words from texts. It is the second stage after word segmentation from the texts. As a result, the authors intend to create a word recognition system specifically for recognizing handwritten Bangla words from photos of paper documents.

The authors aim to recognize words from an image of handwritten Bangla words by detecting the characters using three separate object detection models. The first one is Single Shot MultiBox Detector(SSD) [1], a deep learning-based object detection model that recognizes characters using MobileNetV2 [2] as a feature extractor. The second one is Faster R-CNN [3], with InceptionResNetV2 [4] as the feature extractor. The third one is an ensemble model, which considers both models' results and gives a prediction about the one with the highest confidence score.

After obtaining the models' predictions, a Non-Maximum Suppression algorithm is applied to refine the results for accuracy. The authors aim to propose modifications to the standard Non-Maximum Suppression algorithm to enhance its effectiveness. Subsequent to applying the modified Non-Maximum Suppression to the models' predictions, a process of rearrangement and character processing is undertaken to generate the final text representation of the words. However, handwritten Bangla characters are more complex than English characters. Many handwritten Bangla alphabets look so similar that it is difficult to distinguish them as unique characters. Moreover, characters in Bangla are connected by a line within a word called Matra, so they do not have separate spaces inside a handwritten Bangla word like in English. So, detecting different characters is a big challenge. Nevertheless, some characters can be written in multiple ways. This complication becomes more problematic if the work also includes conjunct characters. Also, different people have different handwriting styles and sizes. These limitations make developing a word recognition system for Bangla handwriting more complicated.

Most of the recent deep learning-based works done in this field have been on closed vocabulary, where the number of unique words has been fixed. The reason is the tendency to recognize the whole word as an individual entity. The authors intend to develop an open-vocabulary system with no limit to the number of unique words as it detects and recognizes characters from the word images and then tweaks and re-arranges the characters to give the word result in text. This process unseals the vocabulary limit, ensuring the system's effectiveness regardless of how many new/undiscovered words are introduced after training. Contrary to the limited works done so far on word recognition that typically focus on open vocabulary, where characters must be segmented from the word first, the authors have pursued a segmentation-free approach. This is particularly significant for Bangla handwriting, where a segmentation-based approach can be challenging due to its cursive nature.

To support this goal, the authors have created a vast dataset of handwritten Bangla word images where many words are non-words(e.g., words with combinations of characters that lack actual meaning and do not exist in the dictionary). This system aims to present the authors' efforts to develop a bespoke dataset and enhance word recognition performance from handwritten Bangla words. Although there is yet to be a high-quality OCR for Bangla like there is for English, Chinese, or Arabic, the authors hope their system will be a milestone toward meeting the demand for reliable OCR in Bangla. The scope of this paper encompasses the recognition of words in handwritten Bangla word images.

## II. Related Works

Alom et al. [5] employed a Deep Convolutional Neural Network (DCNN) for Bangla character recognition. Their experiments showed that modern DCNN models like DenseNet, FractalNet, and ResNet outperformed other classical methods. Roy et al. [6] employed deep learning algorithms to recognize handwritten Bangla conjunct characters. They developed a novel DCNN using supervised layerwise training. They significantly reduced the error rate from 19% to 9.67%. Fardous et al. [7] developed a CNN-based model to distinguish handwritten solitary Bangla conjunct letters. They evaluated the model's performance by training it on CMATERdb 3.1.3.3. Purkayastha et al. [8] proposed a DCNN technique for recognizing Bangla handwritten characters. They tested their system on the BanglaLekha-Isolated dataset, achieving impressive accuracies of 89.93% on practically all Bangla characters(80 character classes). However, the model's main challenge was distinguishing similar-form characters, resulting in most recognition mistakes. Ghosh et al. [9] presented a model for Bangla handwritten character recognition using the MobileNetV1 architecture. Their system recognized 231 classes, including 171 conjuncts, 50 basics, and 10 numerals, achieving an overall accuracy of 96.46%. Adole et al. [10] investigated the application of Faster R-CNN InceptionResNetV2 on offline Kanji handwriting characters. They explored the effectiveness of this deep learning model for recognizing Kanji characters in handwritten form.

Islam et al. [11] introduced a Real-time Bangla license plate recognition system based on Faster R-CNN and SSD. The characters on the license plate are individually detected and then rearranged in their correct order. Their model achieved a precision of 91.67% for character detection on the license plate. Abdullah et al. [12] developed a YOLO-based three-stage network for recognizing Bangla license plates in Dhaka. They used YOLOv3 to locate plates and ResNet-20-based CNN to recognize characters. Their method achieved over 92.7% accuracy in character identification.

Sadaf et al. [13] introduced an end-to-end system for Bangla word recognition, leveraging pre-trained CNN architectures and recurrent units, including Puigcerver's CRNN and Flor's Gated-CNN. Their approach achieved superior performance with a CER of 12.83% and a WER of 36.01%. Dutta et al. [14] introduced an effective CNN-RNN hybrid architecture for lexicon-free handwritten text recognition in Devanagari and Bangla, surpassing prior methods on the RoyDB test set. Das et al. [15] introduced 'H-WordNet,' a holistic handwritten word recognition approach using a streamlined convolutional neural network. By sidestepping character-level segmentation and handcrafted features, the model achieved a significant parameter reduction while outperforming recent methods. Evaluation on a Bangla word database demonstrated H-WordNet's effectiveness with a recognition accuracy of 96.17%.

Majid et al. [16] demonstrated a framework for offline handwriting recognition using character spotting and autonomous tagging, which works for any alphabetic script. Character spotting was built on the idea of object detection to find character elements in unsegmented word images. In character spotting, they applied two Faster R-CNN with VGG-16 for the characters and the diacritics. Hasan et al. [17] used an object detection model, SSD with MobileNetV2, as a feature extractor to recognize characters from handwritten Bangla words. Their work did not include any conjunct character, and their approach correctly recognized only 39.67% of the word images.

## III. Proposed Methodology

### A. Working Procedure

The system's overall operation can be broken down into many steps. The working process is illustrated in Fig. 1.

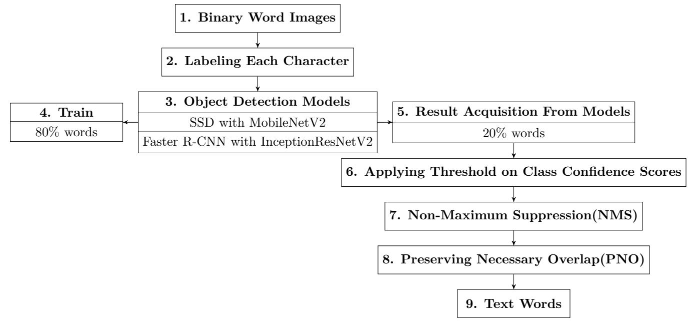



Fig. 1. Flowchart of the working procedure

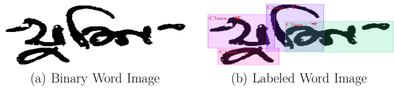


Fig. 2. Binary image and character labels example

*1) Binary Word Images:* The handwritten Bangla words are taken input as binary images where character pixels are black and background pixels are white.(Fig. 2a)

*2) Labeling Each Character:* All the characters/letters, including the vowels/diacritics from the word images, are labeled according to their respective classes. A total of 88 distinct classes(TABLE I), including the numbers, punctuation marks, and seventeen most used conjunct characters [18], have been used here for labeling. Fig. 2b shows how each character from the word images is labeled. Characters have been labeled after a rectangle bounding box has bounded them.

*3) Object Detection Models:* A total of 9841-word images have been labeled for this research, showcasing an open vocabulary. Among these images, nearly 80% have been used to train the object detection models, while the remaining 20% have been allocated for manipulating the models' results and evaluating the system's performance.

*4) Train:* After gathering the training input data, the next step is to create the training configuration and label map. Two independent object-detection models, SSD with MobileNetV2 and Faster R-CNN with InceptionResNetV2, are employed in parallel for training. Tensorflow detection model zoo [19] is used to download the pre-trained models. Each model is fine-tuned using empirically determined optimum parameter values to accomplish its prescribed duties efficiently(TABLE II).

*5) Result Acquisition From Models:* After conducting the models on images, characters are localized and recognized. The models generate rectangular bounding boxes, character classes, and corresponding class confidence scores. In this step, three different sets of results have been obtained. The first set of results is from the SSD model, the second is from the Faster R-CNN model, and the third is an ensemble result achieved through a multi-model ensemble approach. This ensemble result is obtained by merging the outputs of both the SSD and Faster R-CNN models and taking the best result for each character from the two. It results in a more comprehensive and accurate detection outcome. In Fig. 3a, for the word '১২', SSD detects '১হ'(Fig. 3b) where '১' has a confidence score of 89% and 'হ' has 82%. Similarly, Faster R-CNN detects the word as '১২' (Fig. 3c) with a confidence score of 85% and 87%. Ensemble model takes '১' from SSD and '২' from Faster R-CNN as they have the highest scores and gives '১২'(Fig. 3d) as result.

Table I: Different classes used in this research

| | |
|---|---|
| Vowels | অ, ই, ঈ, উ, ঊ, ঋ, এ, ঐ, ও, ঔ |
| Consonants | 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, ড়, ঢ়, য়, ৎ, ঃ, ং, ঁ |
| Numbers | ০, ১, ২, ৩, ৪, ৫, ৬, ৭, ৮, ৯ |
| Diacritics | া, ি, ী, ে, ু, ূ,ৈ, ৌ, ৃ, ্য, ্র |
| Punctuations | । |
| Conjuncts | ক্ষ, ন্ত, ত্র, ঙ্গ, স্থ, স্ব, ক্ত, স্ত, ন্দ, চ্ছ, দ্ধ, ন্ত্র, ত্ত, ষ্ট, ন্ন, ল্ল, স্প |

Table II: Training parameters applied on different models

| Model | SSD with MobileNetV2 | Faster R-CNN with InceptionResNetV2 |
|---|---|---|
| Batch Size | 16 | 1 |
| No. of Classes | 88 | 88 |
| Initial Learning Rate | 0.08 | 0.008 |
| Optimizer | Momentum Optimizer | Momentum Optimizer |
| Epochs | 50000 | 50000 |

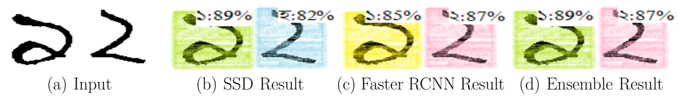


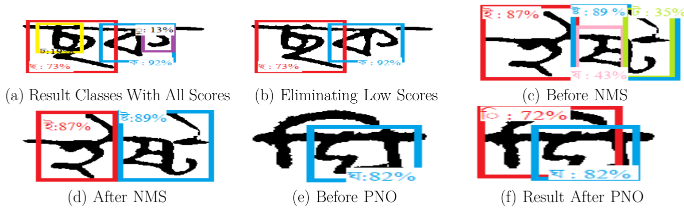
Fig. 3. Detected classes and confidence scores by SSD, Faster R-CNN, and Ensemble Model

*6) Applying Threshold on Class Confidence Scores:* Each of the three models gives more than a hundred predictions for each character in the word. To refine these predictions, Non-Maximum Suppression(NMS) is applied first, reducing the number of predictions to a few for every character. Because, after training with 50,000 epochs, any class identified with a confidence score less than 20% can be considered meaningless. For that, all classes with a confidence score of less than 20%(empirically decided) are eliminated. Here in Fig. 4b, the characters' predicted classes, confidence scores, and bounding boxes by SSD are shown.

*7) Non-Maximum Suppression(NMS):* Even after eliminating classes with a confidence score of less than 20%, another round of NMS is applied to further refine the classes. This is necessary because, in the remaining results provided by the models, there may still be different predictions for the same character. For example, in Fig. 4c, SSD predicts three different classes 'ষ,' 'ট,' 'ষ্ট' for the same character 'ষ্ট'. A new term Overlap Value(OV) has been introduced to detect overlap between them, and then NMS has been used to eliminate the ones with the lowest confidence scores. For each overlapping character, more than one character will have some overlapping portions. OV is used to determine to how much extent the characters overlap. The smaller the value of OV for a character, the more overlap there is with another character.

$$\text{OV} = \frac{\text{Character's Personal Area}}{\text{Common Area of All Overlapping Characters}} \quad (1)$$

The authors empirically decided that if the value of OV for any overlapping character is less than 2, more than one class strongly share a common area. The class



(a) Result Classes With All Scores (b) Eliminating Low Scores (c) Before NMS (d) After NMS (e) Before PNO (f) Result After PNO

Fig. 4. Results of SSD before and after eliminating classes with lower confidence scores, applying NMS and PNO

with the highest confidence score will be selected in this case. In this way, the models detect a single class for one character/diacritic. In Fig. 4c, the bounding boxes of 'ষ,' 'ট,' 'ষ্ট' have an area of 0.04, 0.07, and 0.14 respectively.'ষ্ট' has a common area of 0.03 with 'ষ' and 0.065 with 'ট'. The OV for 'ষ', and 'ট' is respectively 1.33 and 1.08. As both of their value is less than 2, it means both have lots of common/overlapping portions with 'ষ্ট'. It concludes that NMS needs to eliminate two out of three as all three are different results for one character. 'ষ,' 'ট,' and 'ষ্ট' have confidence scores of 43%, 35%, and 89% respectively. So, 'ষ্ট' has been chosen from these three as it has the highest confidence score. (Fig. 4d)

*8) Preserving Necessary Overlap(PNO):* There are some cases where NMS is troublesome rather than a blessing. E.g., in Bangla, there are some diacritics such as 'ি,' ' ী,' ' ৌ', for which overlapping is obvious. In these cases, if NMS is applied as usual, either a consonant or the diacritic will be eliminated. That is why, whenever these diacritics are involved along with the blue labeled consonants/conjuncts from TABLE I in any overlapping issue, the threshold value for OV is decreased to 0.5 instead of 2(empirically decided).In Fig.4e, the bounding boxes of 'ি' and 'ঘ' have an area of 0.20 and 0.13 and their common area is 0.09. 'ি' and 'ঘ' have OV of 2.22 and 1.44. After NMS, 'ি' is eliminated as they share an OV value less than 2. However, decreasing the OV threshold to 0.5 helps preserve both 'ি' and 'ঘ'. (Fig. 4f) Also, another similar case is with ' ঁ' and characters which have some extended part over matra like 'ট,' 'ঠ,' and 'ষ্ট'. In this case, the OV threshold is also 0.5. Bangla diacritics like 'ে,' 'ৈ' could also be added in this step, but in different handwriting style, these diacritics look very much like the Bangla consonants 'ট,' 'ঢ,' 'চ'. This makes the models detect at least two classes for one character. That is why these diacritics are not added in this step despite creating significant overlaps in some cases.

*9) Text Words:* The text word is produced by processing and rearranging detected characters according to their outcome. Characters are rearranged based on their co-ordinate value. Some characters and diacritics such as ' ে', 'ি,' 'ৈ,' 'ৌ,' '৭' are written before their corresponding character but typed after it. 'আ', once again, was not one of the identified classes. When the characters 'অ' and 'া' are found together, they are substituted with 'আ' [17]. This type of exception is taken into account while reordering the characters.

## IV. Results

### A. Datasets

The authors created a customized dataset of 9841 handwritten Bangla word images with one or multiple characters for the Word Recognition stage. 70% of the images of the dataset are collected from different people's handwriting, and 30% of the word images are collected from the CMATERdb 3.1.1 [20], CMATERdb 2.1.2.1, CMATERdb 3.1.3.1 [21], CMATERdb 3.1.3.3 [22]. The inclusion of various CMATERdb datasets is to make the customized dataset more versatile. After collecting the images of papers containing many words from different writers, the words have been segmented and binarized [23]so that each image includes only a word with one or more characters/classes. Even after segmenting the words, the word images have been manually fine-tuned to eliminate irrelevant information besides word texts. Then the characters from the word images are labeled manually according to their corresponding classes.

Out of 9841-word images from the customized dataset, 7701 images are used for training and 2140 for testing. In the 7701 train images, there is a total of 18526 character/diacritic/number/punctuation instances. The words for train images are selected in such a way that the frequent Bangla characters used daily have the most occurrences. In the 2140 test images, there are 5451 character/diacritic/number/punctuation instances. The words for test images are selected in such a way that every class has at least 50 instances so that the result does not create any biases.

### B. Evaluation Metrics

Various metrics such as Recall(2), Precision(3), and F-measure (F1-score)(4) are used to measure the system's recognition rate accuracy on character level. Macro-averaging has been employed to summarize the models' performance across different character classes.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{2}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{3}$$

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}$$

Here, True Positive(TP) refers to a character class that is correctly identified and recognized. A false negative(FN) indicates that a class/character that is present was not detected; something was overlooked. False Positive(FP) occurs when a character in an image is correctly identified but incorrectly classified.

### C. Experimental Results on Character Level

In this experiment, the test images are tested separately by the three different models, and Recall, Precision, and F1-score are measured and reported in TABLE III.

Here, both of the models have their limitations. Overall, SSD gives better results than Faster R-CNN, but there are cases where SSD fails frequently, e.g., SSD classifies 'ঐ' every time as 'এ'. Also, SSD often fails to classify the numbers correctly. Faster R-CNN does not have difficulty classifying these characters but gives poor results where diacritics are involved. However, the ensemble model takes both of their results and gives the best result with the highest confidence score, and that is why the ensemble model has an improved result over these two. The ensemble model having a recall of 96.83% means only 3.17% of the actual classes were not identified. It shows that the proposed model significantly overcame the challenges of the cursive nature of Bangla handwriting.

Satter [23] introduced a similar approach using both models where no ensemble method was used, and SSD and Faster R-CNN had an F1-score of 74.04% and 83.72%. Majid et al. [16] had a similar work using two Faster R-CNNs, and they found a recall and precision of 91.24% and 88.25% after merging both the models' result to form a transcription. However, it's worth noting that the F1-score they reported was formulated using the geometric mean of precision and recall. If they had calculated the F1-score using (4) (harmonic mean of precision and recall), it would have been 89.72%. Hasan et al. [17] worked on a similar approach using SSD; their F1-score for character recognition was 84.21%, and they did not even include any conjunct character. All of their results have been shown in TABLE III to display a comparison between the proposed methodology and some of the existing methodologies.

Table III: Comparative Evaluation: Recall, Precision, and F1-Score of the proposed method on character level and comparison with existing approaches

| Approach | Model | Recall (%) | Precision (%) | F1 score (%) |
|---|---|---|---|---|
| **Proposed** | SSD with MobileNetV2 | 95.88 | 87.62 | 91.19 |
| | Faster R-CNN with InceptionResNetV2 | 92.52 | 86.51 | 88.25 |
| | **Ensemble model** | **96.83** | **89.42** | **92.61** |
| Satter [23] | SSD with MobileNetV2 | 88.02 | 68.3 | 74.04 |
| | Faster R-CNN with InceptionResNetV2 | 92.18 | 80.20 | 83.72 |
| Majid et al. [16] | Faster R-CNN with VGG-16 | 91.24 | 88.25 | 89.72 |
| Hasan et al. [17] | SSD with MobileNetV2 | 80.43 | 88.26 | 84.21 |

### D. Experimental Results on Word Level

Levenshtein Distance [23] is used to measure the models' performance on word level on 2140 test words. 100% accuracy means the predicted and annotated words are precisely the same. 75% accuracy means if there are four annotated classes in the actual output, the model correctly predicts 3 of the classes. TABLE IV shows the models' cumulative accuracy using Levenshtein Distance. TABLE V shows the effectiveness of the NMS and PNO steps of the proposed methodology by showing the result if they were omitted.

Some word images in the test dataset contain only a single character. So, if a model fails to detect/classify that character, the word recognition accuracy directly falls into the 0% accuracy category. That is why there are many words with 0% recognition accuracy. Also, there are many small words (e.g., 'এই,' 'ছন্দ,' 'যে' ). If a model wrongly classifies any character, the word is 50% accurate. These types of instances create misinterpretations about the models' performance on the word level. This misinterpretation is visible if the character recognition result is compared with this result.

Hossain et al. [24] found a word recognition accuracy of 75% when handwriting is poor. Also, they tested against only 100-word images and did not include any conjuncts and their system highly relied on the alignment of the Matra Row. On the other hand, the proposed method does not have any reliance on the Matra Row's alignment and character sizes, and the ensemble model can recognize 96.12% of the test words to some extent.

## V. Conclusion

OCR systems use computer vision to recognize and transform word images into editable text. In this research, a word recognition system for open vocabulary handwritten Bangla words is created, which uses a bespoke dataset and deep learning models to recognize words from images. However, the system has limitations too, e.g., only 17 conjunct characters have been included here, but almost 200 more conjunct characters can be included. Besides, Bangla has more punctuation marks than '।'. Also, a post-processing technique can be introduced to correct the errors generated in the texts.

Table IV: Accuracy on word level for different models

| Minimum Accuracy | % of Words (SSD) | % of Words (Faster R-CNN) | **% of Words (Ensemble Model)** |
|---|---|---|---|
| 100 | 70.89 | 64.07 | **72.52** |
| >= 75 | 77.57 | 70.93 | **78.56** |
| >= 50 | 92.29 | 88.83 | **93.04** |
| >= 25 | 95.84 | 93.60 | **96.12** |
| >= 0 | 100 | 100 | **100** |

Table V: Accuracy on word level for different models if NMS and PNO were omitted

| Scenario | If only PNO was omitted | | If both NMS and PNO were omitted | |
|---|---|---|---|---|
| Minimum Accuracy | % of Words (SSD) | % of Words (Faster R-CNN) | % of Words (SSD) | % of Words (Faster R-CNN) |
| 100 | 68.64 | 62.43 | 30.79 | 13.97 |
| >= 75 | 76.36 | 70.28 | 33.13 | 16.45 |
| >= 50 | 92.38 | 88.79 | 50.23 | 28.64 |
| >= 25 | 96.12 | 93.97 | 58.64 | 36.45 |
| >= 0 | 100 | 100 | 100 | 100 |

The object detection models in the proposed method have been trained with a wide variety of handwriting. Also, it is not limited to any closed vocabulary. So it can recognize different people's writing. The most promising fact about the proposed method is that the ensemble method takes the result of two parallelly trained object detection models and gives the best result. The improved result of the ensemble model shows a new pathway for other researchers. The dataset used here can help future researchers as creating and labeling a huge dataset takes a lot of time. Also, the introduction of NMS and PNO helps improve the result.

The system's three outputs have been compared, and the performance has been measured on character and word level. The recognized words can be further used to develop a post-processing word correction for them. The existing system may help transition to a more established OCR system.

## References


[1] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European Conference on Computer Vision.* Springer, 2016, pp. 21–37.

[2] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[3] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[4] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, no. 1, 2017.

[5] 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. 2017, 2017.

[6] S. Roy, N. Das, M. Kundu, and M. Nasipuri, "Handwritten isolated bangla compound character recognition: A new benchmark using a novel deep learning approach," *Pattern Recognition Letters*, vol. 90, pp. 15–21, 2017.

[7] J. Fardous and A. Afroge, "Handwritten solitary bangla compound letter recognition using convolutional neural network," *International Journal of Image, Graphics and Signal Processing (IJIGSP)*, vol. 7, no. 12, pp. 13–20, 2015.

[8] P. Purkayastha and et al., "Deep convolutional neural network-based technique for recognizing bengali handwritten characters," *Pattern Recognition Letters*, vol. 125, pp. 27–34, 2019.

[9] D. Ghosh and et al., "Handwritten bangla character recognition using mobilenetv1," in *2018 21st International Conference on Computer and Information Technology (ICCIT)*, 2018, pp. 1–6.

[10] A. Adole, E. Edirisinghe, B. Li, and C. Bearchell, "Investigation of faster-rcnn inception resnet v2 on offline kanji handwriting characters," in *Proceedings of the 2020 International Conference on Pattern Recognition and Intelligent Systems*, 2020.

[11] M. S. Islam and A. Rasel, "Real-time bangla license plate recognition system using faster r-cnn and ssd," *2019 5th International Conference on Electrical Engineering and Information & Communication Technology (ICEEICT)*, pp. 1–5, 2019.

[12] S. Abdullah, M. M. Hasan, and S. M. S. Islam, "Yolo-based three-stage network for bangla license plate recognition in dhaka metropolitan city," in *2018 International Conference on Bangla Speech and Language Processing (ICBSLP)*, Sylhet, Bangladesh, 2018, pp. 1–6.

[13] F. Sadaf, S. M. Taslim Uddin Raju, and A. Muntakim, "Offline bangla handwritten text recognition: A comprehensive study of various deep learning approaches," in *2021 3rd International Conference on Electrical Electronic Engineering (ICEEE)*, 2021, pp. 153–156.

[14] K. Dutta, P. Krishnan, M. Mathew, and C. V. Jawahar, "Towards accurate handwritten word recognition for hindi and bangla," in *Computer Vision, Pattern Recognition, Image Processing, and Graphics*, 2018, pp. 470–480.

[15] D. Das, D. R. Nayak, R. Dash, B. Majhi, and Y.-D. Zhang, "H-wordnet: A holistic convolutional neural network approach for handwritten word recognition," *IET Image Processing*, 2020.

[16] N. Majid and E. H. Smith, "Character spotting and autonomous tagging: offline handwriting recognition for bangla, korean and other alphabetic scripts," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 25, pp. 245–263, 2022.

[17] M. S. Hasan and A. Pal, "Optical character recognition for handwritten bangla documents," 2021, unpublished undergraduate thesis. Khulna University of Engineering & Technology, Khulna, Department of Computer Science and Engineering. CSE-4000 Project/Thesis, CSER-22-53.

[18] N. Das, K. Acharya, R. Sarkar, S. Basu, M. Kundu, and M. Nasipuri, "A benchmark image database of isolated bangla handwritten compound characters," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 17, no. 4, pp. 413–431, 2014.

[19] H. Yu, C. Chen, X. Du, Y. Li, A. Rashwan, L. Hou, P. Jin, F. Yang, F. Liu, J. Kim, and J. Li, "TensorFlow Model Garden," https://github.com/tensorflow/models, 2020.

[20] N. Das, R. Sarkar, S. Basu, M. Kundu, M. Nasipuri, and D. 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.

[21] *Proceedings of the International Conference on Information Systems Design and Intelligent Applications 2012 (INDIA 2012)*, vol. 132. Visakhapatnam, India: Springer, January 2012.

[22] N. Das, K. Acharya, R. Sarkar, S. Basu, M. Kundu, and M. Nasipuri, "A benchmark image database of isolated bangla handwritten compound characters," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 17, no. 4, pp. 413–431, 2014.

[23] F. Satter, "Optical character recognition from handwritten bangla texts," 2023, unpublished undergraduate thesis. Khulna University of Engineering & Technology, Khulna, Department of Computer Science and Engineering. CSE-4000 Project/Thesis, CSER-23-64. [Online]. Available: https://github.com/FaiasPromit/Optical-Character-Recognition-From-Handwritten-Bangla-Texts.git

[24] M. T. Hossain, M. W. Hasan, and A. K. Das, "Bangla handwritten word recognition system using convolutional neural network," in *2021 15th International Conference on Ubiquitous Information Management and Communication (IMCOM)*, 2021.