A HIGH PERFORMANCE DOMAIN SPECIFIC OCR FOR BANGLA SCRIPT

Md. Abul Hasnat S. M. Murtoza Habib Mumit Khan
Center for Research on Bangla Language Processing,
Department of Computer Science and Engineering,
BRAC University, 66 Mohakhali, Dhaka, Bangladesh
e-mail: mhasnat@gmail.com, murtoza@gmail.com, mumit@bracu.ac.bd

Abstract-Research on recognizing Bengali script has been started since mid 1980's. A variety of different techniques have been applied and the performance is examined. In this paper we present a high performance domain specific OCR for recognizing Bengali script. We select the training data set from the script of the specified domain. We choose Hidden Markov Model (HMM) for character classification due to its simple and straightforward way of representation. We examine the primary error types that mainly occurred at preprocessing level and carefully handled those errors by adding special error correcting module as a part of recognizer. Finally we added a dictionary and some error specific rules to correct the probable errors after the word formation is done. The entire technique significantly increases the performance of the OCR for a specific domain to a great extent.

I. INTRODUCTION

Methodically, character recognition is a subset of the pattern recognition area. However, it was character recognition that gave the incentives for making pattern recognition and image analysis matured fields of science [1]. In this literature we are considering the area of off-line character recognition. In a broad sense, from top level we can classify the area of Character Recognition into two divisions called machine printed and handwritten character recognition. So, these two categories can be termed as the two domains (group of documents that share similar layout structure) for Character Recognition. Now if we consider only the machine printed text document image with an additional specification of Bengali script then the number of domain will narrow down. We have observed that in Bengali scripts there exists a variety of documents that includes letters, text books, novels, official document, legacy document, newspapers, magazines, data entry form etc. From our experiment on different domain of documents we have seen that a universal technique with some common parameters may not fit perfect for all kinds of document. Based on this observation we feel that rather than a common technique with massive complexity and insignificant performance, we should choose a specific technique with adaptive parameters for a fixed domain or a set of domains to increase the performance significantly. This decision motivates us to think about domain specific solution for OCR. So, we choose our objective as to increase the performance of the OCR for a certain domain.

Research on OCR systems for recognizing Bangla characters have been started since mid 1980's, and a variety of different approaches were applied to examine the performance compared to the prior research results [2-13]. So far two implemented version of OCR for Bangla character recognition is reported [14, 15]. Among the research efforts, some were through the complete OCR system and the rest of the efforts were specific to the different level of OCR system like preprocessing, feature extraction, classification and post-processing. In the next section we will elaborately discuss about some of these research approaches those are relevant to our interest. The listed classifiers used in these research works are Nearest Neighbor Classifier [2], feature based tree classifier [3, 5, 7], template matching [5, 7], distance based classifier [8], Neural Network [9, 10, 12, 15] and Hidden Markov Model [13]. We have examined the performance of several classifier and we choose HMM (Hidden Markov Model) based classifier compare to others because of the following reasons:

- Simple and straightforward way of representation that provides the opportunity of dynamic and time imperceptible training at user end.
- Segmentation free approach.
- Shows significant performance for trained characters.

The performance of an OCR significantly depends on the performance of the recognizer. HMM (Hidden Markov Model) is a widely used classifier mostly for Speech and Handwriting recognition. Very few research works can be found on the usage of HMM for printed character recognition [13]. On the other hand we have reported very persuade performance for some domain specific HMM based recognizer [16, 17]. These considerations greatly motivate us to perform our research and development of an OCR for Bangla using HMM technique. We tested our implemented OCR for different types of document Images and recorded the results. The result shows that most of the errors occurred for the segmentation problem that happened due to the errors at multiple stages at preprocessing step. Based on our overall analysis we find out a specific data set for training. We performed error analysis and based on these errors we changed our training methodology and also placed an error correcting module at the end of basic character recognition result. After the basic word formation is done, we added a dictionary look-up based post-processor that provides up to a certain number of suggestions for the erroneous words. The combination of all these approaches leads to the increase of performance to a great extent. To the best of our knowledge

this is the first reported attempt on domain specific OCR for recognizing Bangla text Image.

In this paper we will briefly present the complete methodology of the OCR technique with the probable errors encountered during recognition and also the solution of these errors using several techniques. In the rest of the paper at section 2 we briefly discuss about the related works, at section 3 we describe the methodology with several sub-sections, at section 4 we will perform result analysis and at last we end up with the conclusion.

II. Related Works

We briefly discuss some related work in this section. A great amount of work has been done by B. B. Chaudhuri and U. Pal since mid 1990's. Following them some other researchers have come up with a variety of innovative ideas. The relevant works are briefly discussed below.

Reference [3] described a complete OCR system for Bangla. A detail description of the characteristics of Bangla text is discussed here. They used a combination of template and feature matching approach for recognizing the character shapes. They used stroke features from each character and used a feature based tree classifier for character recognition. They classified the character set as basic and compound character. They used a simple dictionary lookup for OCR error correction.

Reference [4] used a technique for OCR error detection and correction on Bangla language. They used two separate lexicons of root word and suffixes. These errors are corrected by a fast dictionary access technique.

Reference [5] described their approach to recognize both Bangla and Devnagari scripts. At preprocessing level they applied Hough transform to find the skew angle, page layout analysis to handle multiple columns of text line and graphics, text line zoning technique for character segmentation. Unlike [3] they grouped characters into three classes named basic, modifier and compound character. They used feature based approach for basic and modifier character recognition and a combination of feature based template matching approach for the compound character recognition. However, recognition and error handling are almost similar as the approach described at [4].

Reference [7] presented a Complete Printed Bangla OCR System where they discussed about the difficulties encountered in Bengali script. In their approach the basic and modified characters are recognized by a structural-feature-based tree classifier and the compound characters are recognized by a tree classifier followed by template-matching approach. They used character unigram statistics to make the tree classifier efficient and several heuristics to speed up the template matching approach. A dictionary-based error-correction scheme has been used as a post processor.

Reference [8] give a brief overview of OCR research on Indian languages and also provide a substantial description of their work. They used a hybrid approach to recognize the parts of the conjunct that form part of a character class. To classify the segmented images into known classes, they used a set of filters and two distance based classifiers. They presented a two level partitioning scheme and search algorithm for the correction of optically read characters.

Reference [9] described their approach to recognize only the basic characters [3, 5]. They applied thinning and scaling at preprocessing level and multi-layered feed-forward back propagation neural network for character recognition purpose.

Reference [10] used curvature properties as local feature that is acquired from the slope distribution of chain code representation of each character. Their classification strategy was using Neural Network approach where for training and recognition conventional back propagation algorithm was performed.

Reference [12] presented a minimally segmented OCR where scaled the segmented image into a predetermined area. They extracted feature vector from a rectangular pixel map as a series of 0s and 1s. Finally they used a Kohonen neural network based classifier for character recognition.

The most recent work has been reported by M. A. Hasnat et al. [13] where they presented segmentation free OCR using Hidden Markov Model (HMM) based Recognizer. They applied Discrete Cosine Transform (DCT) to calculate the feature values. Their approach was to build separate model for each segmented primitive. Their approach shows significant performance for trained character.

So far we have seen two implementations of OCR for Bangla. They are BOCRA [14] and Apona Pathak [15]. BOCRA targets a very specific type of OCR problem, namely where the input images are high quality scans of high quality printed text written in a single font in a uniform point size. Apona Pathak has the ability to handle multiple font and size. However the performance of both applications suffers greatly from segmentation error.

III. Methodology

The procedural block diagram of the OCR system is shown in Fig. 1. This diagram is quite straightforward where each block can be further divided into several internal steps. We made our efforts in each subsection to minimize the errors based on our error analysis. The next sub-sections will elaborately describe each block.

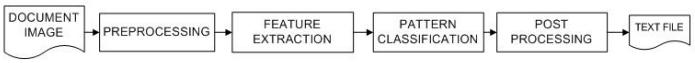

Fig. 1: Block Diagram of OCR system

A. Preprocessing

Preprocessing is the fundamental and very important stage. Lots of errors occurred at this step and for this reason enormous number of research work is going in this area which leads to a separate area of research called Document Image Analysis. Possible errors that may occur at this stage are briefly discussed at [1, 16]. During this research work we were aware of these errors and tried to overcome these errors at best. In this stage we perform the followings tasks:

1) Image Acquisition and Binarization: We used a flatbed scanner for Image acquisition and digitization. We are able to

process text images at any format. Then we applied traditional method to convert the color image to grayscale image if necessary. Next we perform thresholding operation to produce a binary image whose one state will indicate the printed text while the complementary state will correspond to the background. We experimented several traditional methods [18] and Otsu method [19] for thresholding and applied these methods based on domain type.

2) Noise elimination: We learned about different types of noise, their sources, and effects from the papers [1, 20]. From our observation we identified that for printed document the majority of the noises are the salt and pepper noises and the background noise. For background noise removal we used connected component information and eliminate the noise using statistical analysis. For other type of noise removal and smoothing we used wiener and median filters [21].

3) Skew detection and correction: We considered two methods [6, 11] for this purpose and we followed the approach discussed at [11]. First we identified the upper envelope and then we applied Radon transform to the upper envelope to get the skew angle. We applied generic rotation algorithm for skew correction and then applied bi-cubic interpolation.

4) Line, word and character level segmentation: We have studied several segmentation approaches discussed at [3, 5, 7-10, 12]. From implementation perspective we observed that, most of the errors occurred at character level segmentation. Line and word level segmentation failed due to the presence of noise which gives wrong estimation of the histogram projection profile. However character level segmentation mostly suffers from joining error (fail to establish a boundary where there should be one) and splitting error (mistakenly introduce a boundary where there should not be one). Considering all these we made our effort up to a minimal segmentation [12] and we resolved these issues during classification. Finally we used a simple technique similar to [3]. Fig. 2 shows the segmented units.

Fig. 3: Segmentation Result

The output from this block is a set of segmented image which is provided as an input to the next block where each one is processed separately.

B. Feature Extraction

At this stage we divide each segmented character image into several frames of fixed length (e.g. 8 pixels). Then we applied DCT (Discrete Cosine Transform) calculation over each pixel of the frames. We followed the similar technique described at [13]. The extracted feature for each character image is written into a file in a specific format. This process is shown in Fig. 3.

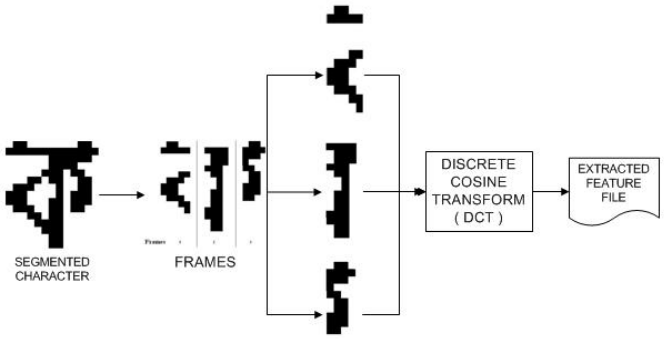

Fig 3: Feature Extraction process

C. Pattern Classification

This stage describes the training and recognition methodology. The extracted features for each segmented character are considered as the input for this stage. We followed almost similar strategy discussed at [13] for classification, however we did not limit ourselves on several issues like training from multiple samples and also the trained data representation using a fixed prototype model. We introduced the concept of dynamic training at any level of recognition and dynamic prototyping as well. For the recognition process we create a temporary model from the feature file of each character image and simply pass the model to the recognizer for classification. Like [13] we also used HTK recognizer [22] for our research and implementation.

1) Training: For training we create a separate model for each of the training character or symbol from the training data set. We estimated all around 650 training data unit (primitives and compounds) into the training data set based on our analysis on the OCR performance. This large amount of training data unit ensures the error tolerance at recognition. These samples are considered as the primitives for any trained OCR. We proposed dynamic training which enables us to train the OCR even after observing the recognition result and hence further improve the performance. We choose prototypes dynamically for the initialization of each model where each prototype contains the proper HMM model parameters like: Number of states, observation and transition matrix. HTK re-estimates the model parameters using this prototype and the extracted features.

Data Set for Training: In our training data set initially we considered only the alphabets of Bangla character set with the traditional segmentation method, but the recognition performance was not considerable. Then we added the compound characters into the training set and we obtain a good performance. However with this database the system was yet suffering from segmentation error occurred at the places of the vowel and consonant modifiers. So, finally we have taken the minimal segmentation approach [12] and added the characters with the vowel and consonant modifiers into the training set. During training, we must associate the appropriate Unicode character in the same order as they appear in the image.

2) Recognition: The recognition process is quite straightforward. The classifier temporary creates a model for

each minimally segmented character and recognizes this model using Viterbi decoder [13, 22].

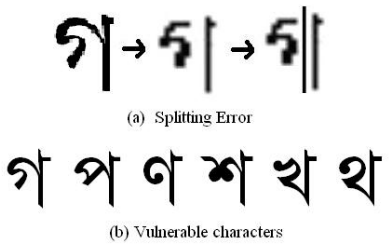

Fig. 4: (a) example of splitting error (b) vulnerable basic characters

We added a simple error correcting module at the end of basic character and symbol recognition that corrects the splitting errors which occurred due to the combination of over thresholding and segmentation problem. We identified that the characters that do not have full matra (baseline) over it, mostly suffers from this error. An example of the splitting error and also the list of vulnerable basic characters are shown in Fig. 4. We observed that the second part of each of these broken characters is classified as Bangla aakar (a vowel modifier) and the first part is misclassified or not classified. To solve this problem we trained the broken first part with special symbols and keep the second part as it is. After the recognizer classified the basic characters or symbols we resolved these erroneous issues in this module by a special table lookup. The output of this module is the words formed after the character level classification. Here we are considering this module as a part of our recognizer.

D. Post Processing

In this stage we used a suggestion based spelling checker for correcting the erroneously recognized words. We applied a technique based on the concept of the spelling checker proposed by Naushad et al. [23]. However instead of a phonetic encoding table we used a table that actually codes the graphitic symbols. We assign same code to those characters that are visually almost similar. Here we encode the characters based on the possible errors encountered at our observation on the OCR result without spell checker. In our approach rather than replacing the erroneous word we would like to provide a certain number of suggestions for that word.

IV. RESULT ANALYSIS

We tested the performance of the classifier in several domains with the specified training data set and we obtained an average of almost 98% accuracy of the classifier for properly binarized image and segmented characters. However from our analysis we are aware that once a split, join, or misalignment error is present in the output of the segmentation stage, even otherwise perfect classifiers will generally fail [16]. Our error correcting module and the postprocessor are capable to handle 14% - 17% of these errors depending on the different domains. Table I gives a clear scenario of the reported error, error tolerance rate and final accuracy at different level for different domain.

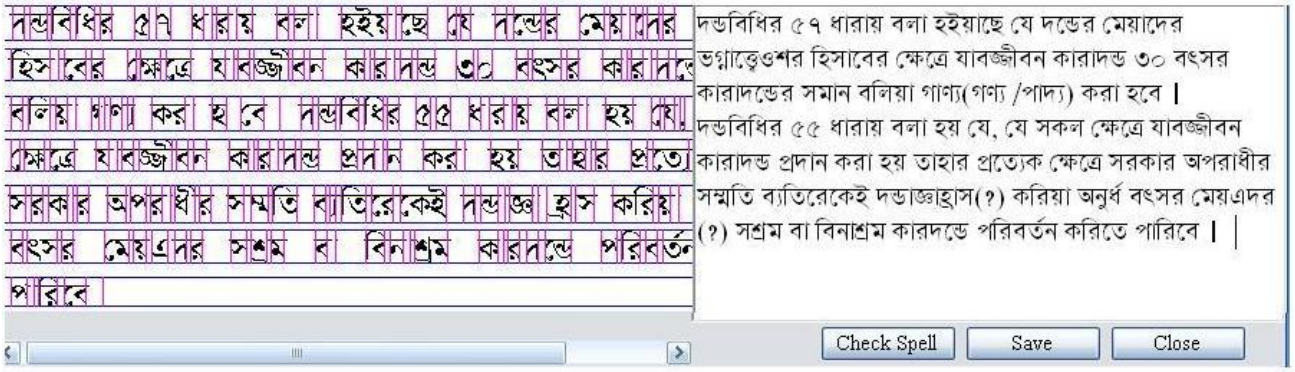

Fig. 5: a) Domain: Legal document

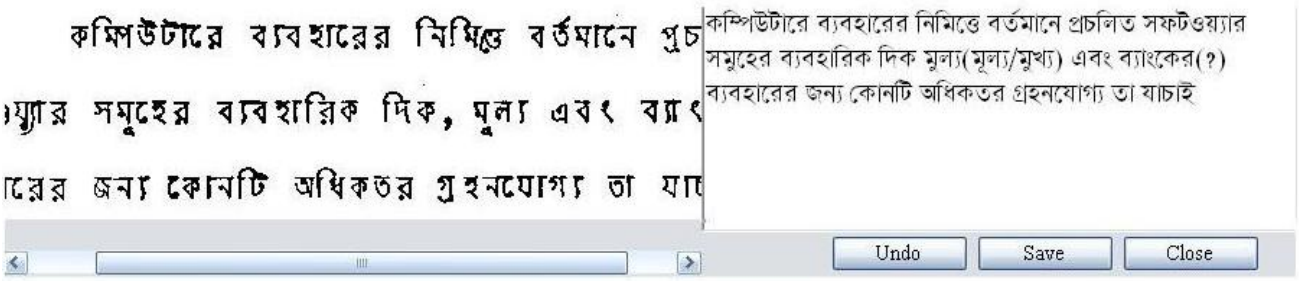

Fig. 5: b) Domain: Typewriting document

Fig. 5: Performance of the OCR for two different domain (a & b) document images.

TABLE I
LIST OF DOMAIN SPECIFIC PERFORMANCE

Domain Name	Classification Accuracy	Segmentation Error Rate	Total Error	Error Tolerance	Final Accuracy
Legal Document	98%	16%	18%	16%	98%
Typewriting	96%	15%	19%	16%	95%
Printed Article	97%	13%	16%	14%	98%

Fig. 5 shows the screen shot of the implemented version of our proposed OCR with the results.

V. CONCLUSION

This paper presents a complete Bangla OCR for domain specific document images. At different stages we tested several methods and choose the appropriate one for our purpose. We have done a complete analysis of the possible errors. Proper identification of the errors helps us to take right decisions to correct those errors at different stages. As a complete solution, the OCR shows high performance for specific domains. We have shown that at classification level we obtain massive accuracy, however segmentation problem degrades the accuracy and at the end we put our efforts to achieve the high accuracy by introducing an error correcting module with the recognizer and a suggestion based post processor.

VI. REFERENCE

[1] Line Eikvil, "Optical Character Recognition", "citeseer.ist.psu.edu/142042.html".

[2] A. K. Roy and B. Chatterjee, "Design of a Nearest Neighbor Classifier for Bengali Character Recognition", J. IETE, vol. 30, 1984.

[3] U. Pal and B. B. Chaudhuri, "OCR in Bangla: An Indo-Bangladeshi Language", Proc. of 12th Int. Conf. on Pattern Recognition, IEEE Computer Society Press, pp. 269-274, 1994.

[4] B. B. Chaudhuri and U. Pal, "OCR Error Detection and correction of an Inflectional Indian Language Script", Proceedings of ICPR, 1996.

[5] B. B. Chaudhuri and U. Pal, "An OCR System To Read Two Indian Language Scripts: Bangla And Devnagari (Hindi)", Proc. Fourth ICDAR, 1997.

[6] B.B. Chaudhuri and U. Pal, "Skew Angle Detection Of Digitized Indian Script Documents", IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 19, pp.182-186, 1997.

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

[8] Veena Bansal and R.M.K. Sinha, A Devanagari OCR and A Brief Overview of OCR Research for Indian Scripts in Proceedings of STRANS01, held at IIT Kanpur, 2001.

[9] A. A. Chowdhury, Ejaj Ahmed, S. Ahmed, S. Hossain and C. M. Rahman, "Optical Character Recognition of Bangla Characters using neural network: A better approach". 2nd ICEE 2002, Khulna, Bangladesh.

[10] J. U. Mahmud, M. F. Raihan and C. M. Rahman, "A Complete OCR System for Continuous Bangla Characters", Proc. of the Conf. on Convergent Technologies, 2003.

[11] S. M. Murtoza Habib, Nawsher Ahmed Noor and Mumit Khan, Skew correction of Bangla script using Radon Transform, Proc. of 9th ICCIT, 2006.

[12] S. M. Shoeb Shatil and Mumit Khan, "Minimally Segmenting High Performance Bangla OCR using Kohonen Network", Proc. of 9th ICCIT, 2006.

[13] Md. Abul Hasnat, S. M. Murtoza Habib, and Mumit Khan, Segmentation free Bangla OCR using HMM: Training and Recognition, Proc. of 1st DCCA2007, Irbid, Jordan, 2007.

[14] http://bocra.sourceforge.net/doc/ last accessed Oct 22, 2007.

[15] http://www.apona-bd.com/apona-pathak/bangla-ocr-apona-pathak.html last accessed Oct 22, 2007.

[16] 16. A. Kornai and K. Mohiuddin and S. Connell, "An HMM-Based Legal Amount Field OCR System for Checks", 1995 IEEE International Conference on Systems, Man and Cybernetics, Vancouver BC, October 1995, 2800-2805.

[17] 17. A. Kornai, Experimental HMM-based postal OCR system, Proc. Int. Conf. Acoustics, Speech, Signal Processing, Munich, Germany, Vol. 4, 3177-3180, 1997.

[18] 18. John C. Russ, "The image processing handbook", CRC Press, Boca Raton, FL, USA, 1998.

[19] 19. N. Otsu, "A Threshold Selection Method from Gray-Level Histograms", IEEE Transactions on Systems, Man, and Cybernetics, 1979.

[20] 20. Yan Solihin and C.G. Leedham, "Noise and Background Removal from Handwriting Images", Proc. of the IASTED Int. Conf. on Intelligent Information Systems , 1997.

[21] 21. Tinku Acharya and Ajoy K. Ray, "Image Processing, Principles and Applications", John Wiley & Sons, Inc., Hoboken, New Jersey, 2005.

[22] 22. The HTK Book available at http://htk.eng.cam.ac.uk/docs/docs.shtml

[23] 23. Naushad UzZaman and Mumit Khan, "A Double Metaphone Encoding for Bangla and its Application in Spelling Checker", Proc. 2005 IEEE Int. Conf. on Natural Language Processing and Knowledge Engineering, Wuhan, China, October 30 - November 1, 2005.