# Minimally Segmenting High Performance Bangla Optical Character Recognition Using Kohonen Network

Adnan Mohammad Shoeb Shatil and Mumit Khan
*Computer Science and Engineering, BRAC University, Dhaka, Bangladesh*
*shoeb_shatil@yahoo.com, mumit@bracu.ac.bd*

## Abstract

*This paper presents a method to use Kohonen neural network based classifier in Bangla Optical Character Recognition (OCR) system, providing much higher performance than the traditional neural network based ones. It describes how Bangla characters are processed, trained and then recognized with the use of a Kohonen network. While there have been significant efforts in using the various types of Artificial Neural Networks (ANN) in optical character recognition, this is the first published account of using a segmentation-free optical character recognition system for Bangla using a Kohonen network. The methodology presented here assumes that the OCR pre-processor has minimally segmented the input words into easily segmentable chunks, and presenting each of these as images to the classification engine described here. The size and the font face used to render the characters are also significant in both training and classification. The images are first converted into grayscale and then to binary images; these images are then scaled to a fit a pre-determined area with a fixed but significant number of pixels. The feature vectors are then extracted from the rectangular pixel map, which in this case is simply a series of 0s and 1s of fixed length. Finally, a Kohonen neural network is chosen for the training and classification process. Although the steps are simple, and the simplest network is chosen for the training and recognition process, the resulting classifier is accurate to better than 98%, depending on the quality of the input images.*

## 1. Introduction

Bangla is the 4th most widely spoken language with more than 200 million speakers, concentrated in Bangladesh and in the Indian state of West Bengal [1]. The Bangla script, used to write Bangla, is even more predominant as it is used to write a few other Indic languages such as Assamese and Monipuri as well. Even with the large number of users of Bangla script, there is a dearth of high-performance optical character recognition (OCR) engine for it. The reason is multifold, ranging from the complexity of the script to the lack of research and development that has been done to date. The script complexity poses a formidable challenge in segmenting the Bangla text into its constituent parts, compared to a process that is trivially easy for many of the western scripts, such as Latin [3-4]. Coupled with the segmentation problem, the classifier designs have typically been based on simple feed-forward and back-propagation neural networks, and as a result, there is a lack of high performance OCR engine for Bangla [5-8]. This paper presents a system fashioned in such a manner that requires minimal, if any, segmentation and has good performance in recognition for both individual characters and groups of characters (words or sub-words). The recognition procedure is solved with Kohonen network [9-11]. Steps taken in Bangla character recognition in this paper are (a) printed Bangla character in taken for raw data; (b) the data is gray scaled and then converted into black and white (BW) images in the pre-processing stage; (c) pixels are grabbed and mapped into specific area and a vector is extracted from the image containing the Bangla word or character, a step considered as the processing stage; and lastly (d) Kohonen Neural Network is used for classification. Figure 1 shows a pictorial view of the OCR process.

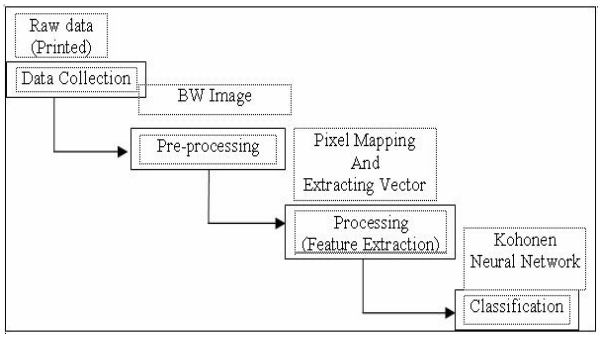


**Figure 1: Bangla character recognition procedure using Kohonen network**

Section 2 reviews some of the related work in this area, both for Bangla script and OCR processes in general. Section 3 provides a general introduction to Kohonen network, including comparison with traditional back-propagation neural networks. Section 4 provides an overview of the training and recognition steps, which includes the data collection, image processing, feature extraction and the recognition system. Section 5 describes the *open-source* implementation of the Kohonen-based Bangla OCR, followed by a discussion of the results in Section 6. The paper concludes with a look at the future in Section 7.

## 2. Literature review

The first published accounts of complete Bangla OCR systems started emerging back in mid 1980's, and various efforts to create a working OCR for Bangla has continued since [2-8]. However, there really has not been much in the way of performance enhancements or innovative schemes discussed in the literature. Some of these efforts have concentrated on specific type of recognition, such as printed digits [5]. The authors propose converting the initial 256x256 image to 40x40 pixels, and the resulting *eigendigits* or *eigenspace* calculation is very compute intensive. It is also quite sensitive to the shape of the digits, and the performance degrades significantly as it tries to recognize digits printed with different typefaces; and even more so if this approach is attempted for the entire character set. Using neural networks and even for Kohonen neural network the performance can be improved significantly. The work presented in [6] takes a much better approach, but the problem with segmentation creates a challenge in improving its accuracy and performance. The training set increases significantly if the consonants with dependent vowels are trained, which degrades the performance; the performance takes another hit if the neural network contains hidden layers as well. This procedure can be greatly simplified using a Kohonen network as it has just one layer and much less time is consumed for training and recognition. Reference 7 presents a reasonable system for both isolated and continuous Bangla character recognition, with good preprocessing and feature extraction. However, after separating each character, it functions similar to [6], except that the feature is extracted from each character using an 8-directional Freeman chain code. This feature is then trained using a feed-forward neural network for recognition. The error tolerance is unfortunately rather low for this scheme, and without an adaptive error correction scheme, the accuracy may not acceptable. After the pre- and post-processing stages, a Kohonen network can perform better than feed forward network and even back propagation algorithm. Reference 12 presents an innovative method to recognize off-line Bangla hand printed numeric characters. The authors take different sets of Bangla characters of different sizes and the respective features, and compute wavelet transforms at different resolution levels. Then they use a multilayer perceptron and majority voting approach for training and recognition. The accuracy level is 97.16% for different numerical characters. But the training procedure takes significant time to complete, and the iteration steps are quite large as well. As can be seen later in this paper, the accuracy can easily be greater than 98%, and quite often 100%, when a Kohonen network is used instead.

## 3. Theory of Kohonen network

The Kohonen Neural Network, named after its creator Tuevo Kohonen, is the simplest network with only two layers of neurons – one input and one output. In this section, we examine the Kohonen network architecture and its implementation. See [11] for a thorough introduction to the Kohonen Neural Network.

In understanding the Kohonen network architecture, it is instructive to see how it differs from the feedforward backpropagation neural network architectures. The first difference is that the Kohonen network does not contain hidden layers; secondly, the Kohonen network training and recognition processes are significantly different; thirdly, it does not use an activation function and, finally, the Kohonen network does not use any bias weight in the network.

Given an input pattern to recognizing, only a single neuron is selected as the "winner", which in turnds becomes the output of the Kohonen network. The difference in training is that the Kohonen network is trained in an unsupervised mode, which is the most significant difference between this and a feedforward backpropagation network. [11]

So for a given pattern we can easily find out the vector and can send it through the Kohonen Neural Network. And for that particular pattern any one of the neuron will be fired. As for all input pattern the weight is normalized so the input pattern will be calculated with the normalized weight. As a result the fired neuron is the best answer for that particular input pattern.

## 4. Recognition steps and procedure

### 4.1. Data collection

The input date is first resized to 250 X 250 pixels to satisfy the procedure, regardless of whether it's an image of a single character or a word. The system was trained with both computer-generated images and scanned images of text. It should be noted that no skew correction was done, so the scanning process is expected to be of high quality.

### 4.2. Image processing

After collecting the image for further procedure it has to be formed suitable for Kohonen network. Generally collected Images are in RGB form. So we need to convert it into Binary image. The procedure of converting RGB to Grayscale and then Grayscale to Binary uses the Otsu algorithm [13].

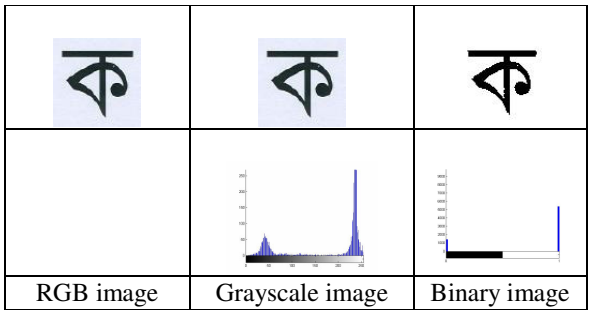


**Figure 3: converting RGB to binary image and necessary histograms**

The image above represents the Bangla character 'Ka'. And the evaluation from RGB image to gray scale image are shown in the consecutively.

### 4.3. Feature extraction

We consider this feature extraction as a most important part of the character recognition procedure. Here we are creating vectors from as image (binary image). We have a sampled are with limited number of square pixels. But before mapping into the sampled area we grab all the pixels from the image and create groups. Each group finds and defines the probability of making squares based on the majority of its pixels combination. Then the whole image is mapped to sampled area and thus we find an abstract of that image. The shape may be changed a bit as we are reducing the pixel of the image. But it is not a major issue rather what comes out from that sampled are is mandatory. This is because the sampled area forms vector for that particular image. The steps are described below:

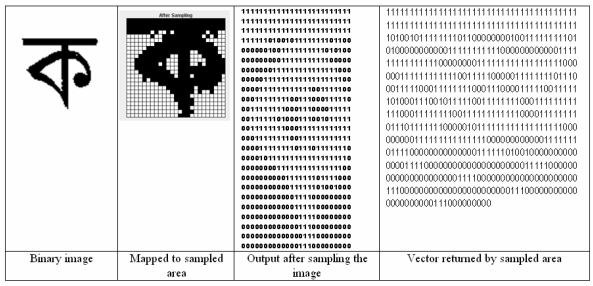


**Figure 4: Vector collection procedure from binary image**

### 4.4. Recognition procedure

We did lots of activities in pre-processing stages and as well as in processing stage. The main idea is to make it simple and acceptable for Kohonen Neural Network. And as there are no hidden layers, we defined the input neuron and output neuron numbers. For example, in our system we consider 10 Bangla digits, 11 vowels, 36 consonants all together 57 characters. So we fixed the number of output neuron is 57. As our vector length is 625 so our input layer has 625 neurons. But in our output layer the number of neuron depends on the number of character trained with the network. As we take 625 as input and n for output character, we can draw our suitable Kohonen Neural Network as Figure 5.

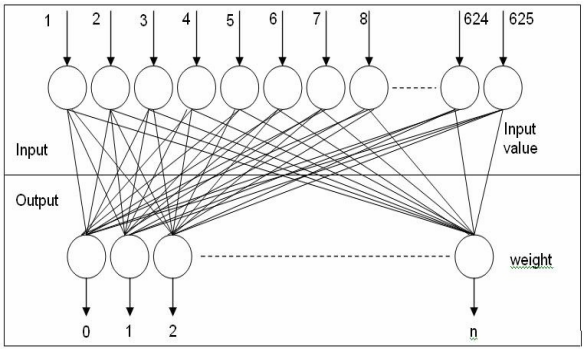


**Figure 5: Kohonen Neural Network design for Bangla Character Recognition**

We discussed about Kohonen network in section 2. And rests of our processes are considered exactly by those methods and steps.

## 5. Implementation

The *open source* OCR system has been implemented in Java, and some screen shots of its usage can be seen in the Figures 6a and 6b. Both the individual characters and group of characters (Bangla words) are sampled, trained and recognized with the system. It has image field, character database, model database based on character(s), fixed pixel area, vector field and finally the text area for containing recognized character.

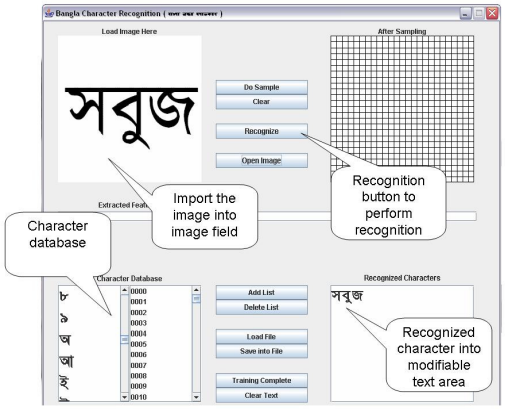


**Figure 6a: character recognition with the system**

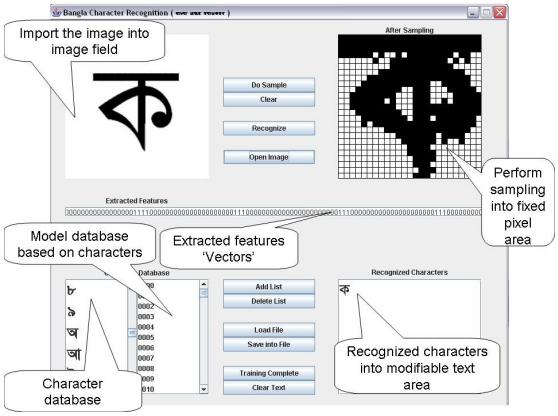


**Figure 6b: Bangla word recognition procedure**

## 6. Results

Table 1 shows the accuracy results for different input data. The data is considered as trained characters or words, untrained similar fonts (*SolaimanLipi* is the trained font and similar font is *Dhanshiri*), scanned documents are of *SolaimanLipi* fonts and lastly irregular shape fonts (some big or small fonts).

**Table 1: Accuracy rates corresponding to different sectors**

| *Character/words* | *Rate of accuracy with Kohonen network* |
|---|---|
| Trained | 100% |
| Untrained similar fonts | 99% |
| Scanned documents | 99% |
| Irregular font size | 98% |

A brief comparison between Kohonen network and neural network is given below; with the data for the neural network implementation is taken from [6]. Here the two different types of fonts with a font size of 14 are considered as training set and the result is given in Table 2.

**Table 2: Performance comparison between Kohonen Network and Neural Network**

| *Typeface* | *Accuracy rate* | |
|---|---|---|
| | *Kohonen network* | *Neural network* |
| *Sutonny* | 97.6% | 94.37% |
| *Sulekha* | 96.2% | 91.25% |

Both the results demonstrate that the OCR recognition engine based on Kohonen network shows very good promise indeed, especially as compared to Neural network based ones. Not only is the accuracy rate consistently higher, the time performance to train and recognize are better as Kohonen networks do not have hidden layers.

## 7. Conclusion

Two of the bottlenecks in creating a high-performance and accurate Bangla OCR have been the lack of accurate segmentation and the use of neural networks with many hidden layers in training and recognition. This paper presents a Kohonen network based OCR for Bangla script that requires little or no segmentation. The performance of the Kohonen recognition engine is high because the network does not use hidden layers. However, if the input is not segmented at all, then the number of input neurons will simply negate any performance increase offered by the Kohonen network. The solution is to segment the input is easily segmentable *chunks*, such as sub-words that have clearly distinguishable histogram profiles, and use that as the input instead. These *chunks* must also be trained, and for that one must do a statistical analysis using a text corpus.

## 8. Acknowledgement

This work has been partially supported through PAN Localization Project (www.PANL10n.net) grant from the International Development Research Center, Ottawa, Canada, administered through Center for Research in Urdu Language Processing, National University of computer and Emerging Sciences, Pakistan.

## 9. References

[1] B. Comrie, ed., "The World's Major Languages", Oxford University Press, New York, 1987.

[2] U. Garain and B. B. Chaudhuri, "Segmentation of Touching Characters in Printed Devnagari and Bangla Scripts Using Fuzzy Multifactorial Analysis", *IEEE Transactions On Systems, Man, And Cybernetics—Part C: Applications And Reviews, Vol. 32*, No. 4, November 2002.

[3] B. B. Chaudhuri and U. Pal, "A complete Bangla OCR System", *Pattern Recognition, Vol 31*, 1998.

[4] B. B. Chaudhuri and U. Pal, "OCR in Bangla: an Indo-Bangladeshi Language", *Pattern Recognition, Vol. 2*, 1994.

[5] M.A. Hasan, P.P. Paul, M.W. Islam, S. Biswas, S. Ahmed, "Bangla digit recognition using Eigen digits".

[6] A.A. Chowdhury, E. Ahmed, S. Ahmed, S. Hossain, C.M. Rahman, "Optical Character Recognition of Bangla Characters using neural network: A better approach", *Proc. 2nd International Conference on Electrical Engineering (ICEE)*, Dhaka, 2002.

[7] J.U. Mahmud, M.F. Raihan and C.M. Rahman, "A Complete OCR System for continuous Bengali Character", *TENCON 2003, Conference on Convergent Technologies for Asia-Pacific Region*, 15-17 Oct. 2003.

[8] M. R. Hasan, M.A. Haque and S.U.F. Malik, "Bangla Optical Character Recognition System", *Proc. 2nd International Conference on Computer and Information Technology, ICCIT 1999*, Dhaka, 1999.

[9] M.T. Hagan, H.B. Demuth and M. Beale, "Neural Network Design", PWS Publishing Company, 1995.

[10] R.C. Gonzalez and R.E. Woods, *Digital Image Processing*, second edition, Pearson Education, 2003.

[11] J. Heaton, *Introduction to Neural Network in Java*, HR publication, 2002.

[12] U. Bhattacharya, B. B. Chaudhuri, "A Majority Voting Scheme for Multiresolution Recognition of Hand printed Numeral", *Document Analysis and Recognition, Proceedings, Seventh International Conference*, 3-6 August 2003.

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

[14] A.K. Jain and T. Taxt, "Feature Extraction Methods for Character Recognition – A Survey", *Pattern Recognition, Vol. 29, No. 4*, 1996, pp. 641-662.

[15] J.C. Perez, E. Vidal and L. Sanchez, "Simple and Effective Feature Extraction for Optical Character Recognition", *Selected Papers From the 5th Spanish Symposium on Pattern Recognition and Images Analysis: Advances in Pattern Recognition and Applications*, Valencia, Spain, 1994.

[16] Z. Lu, I. Bazzi and R. Schwartz, "A Robust, Language Independent OCR System", *27th Advances in Computer-Assisted Recognition SPIE*, 1999.