Offline Isolated Bangla Handwritten Character Recognition Using Spatial Relationships

Fahim Irfan Alam, Bithi Banik
University of Chittagong
Chittagong, Bangladesh
Email: fahim@cu.ac.bd, bithi_cse@yahoo.com

***Abstract*—Handwritten character recognition is considered to be one of the most fascinating and interesting field of research in image processing and pattern recognition. Due to the various challenges associated with it, intensive research works are currently in progress for constructing algorithms that produce better recognition accuracy. This paper proposes an algorithm that recognizes offline isolated Bangla handwritten characters using spatial relationships between any foreground pixels with the background pixels. The algorithm starts with eliminating unwanted noises from scanned images, performing normalization of size and gradually progress toward constructing feature vector representation for the characters using zoning along with spatial relationships in terms of directional relationships. The constructed feature vectors for each individual Bangla character are learned into a neural network which later classifies new instance of Bangla character. The promising preliminary experimental results indicate a positive potential of our algorithm.**

Keywords: Handwritten Bangla Characters, Normalization, Spatial Relationships, Directional Relationships, Feature Vectors.

I. Introduction

Optical character recognition (OCR) is concerned with a process of converting digital data such as images of handwritten, typed or printed text into a machine comprehensible format. The purpose of this conversion is to edit the texts so as to use them in various applications. One of the major applications of OCR is pattern recognition which is quite a fascinating field of artificial intelligence. By digitizing such analog documents, it is possible to construct a useful pattern through the use of machine learning techniques. The learned pattern is used later to predict or make decision on future instance of such data which makes it more generalized.

One of the most interesting OCR researches is recognizing handwritten characters. This particular problem is more challenging than the counterparts of typewritten and printed texts. There are numerous uses of handwritten character recognition in the fields of image processing and pattern recognition [1][2]. It establishes an interface between the machine and human in order for paving a spectacular gateway to feed the handwritten characters into the machine but process them in a machine-encoded way. Intensive research works are currently in being progress and researchers are trying to devise new methods in order to bring sound recognition accuracy while keeping the processing time to minimum [3].

Handwritten character recognition comes in two forms: on-line and off-line. In on-line character recognition, computers recognize the characters as they are drawn through the use of two dimensional coordinates of successive points which are represented as a function of time for detecting the dynamic motion [4]. In off-line character recognition, the handwritten characters are made appropriate to feed into computers by scanning them through digital scanner machines. Unlike on-line characters, the off-line character recognition systems concern with the fixed static shape of the characters [5]. Due to its various applications such as mail sorting, postal address recognition, bank document processing, rigorous research works are currently under investigation [6].

Working with handwritten characters comes with number of challenging issues. As the characters are first scanned to computers, it is an imperative concern to deal with the unwanted signals that are also transmitted along with the original documents. These unwanted signals, called as noises are associated with the drawn characters that affects the recognition model and therefore results in wrongful prediction for futures instances of test data. Also, broken strokes in the characters sometimes confuse the model and if the strokes are not connected, the model cannot differentiate those with the correct instance. Along with these, another important issue is to interact with variations in handwritten texts as different people use different styles and strokes while writing texts by hand. The recognition model should be able to differentiate among various styling patterns and come up with a standard formation for the same character.

A few studies on off-line recognition of handwritten Bangla characters are available in the literature. A two-stage process for recognizing offline handwritten bangla characters was presented in [7]. Recognition based on stroke features was also presented in [8] in which a total of 10 features were extracted for a character image and later an MLP classifier was trained using those feature vectors. A recognition system based on 5 high-level features on unique characteristics of bangla characters were proposed in [9]. Another literature review can be found in [10] where the classifier was made on significant curvature events like curvature maxima, curvature minima and inflexion points.

The difficulties and challenges of handwritten bangla character recognition are associated with the character themselves. The extreme cursive nature makes it very difficult to uniquely describe a character. Even if the characters are isolated, the similar patterns and large symbol set impose the challenges of distinguishing them. In this paper, we were highly moti-

978-1-4799-0400-6/13/$31.00 ©2013 IEEE

vated to experiment with the relationships that exist in each character which can be an important criteria of developing a representation scheme.

In the article, a study on recognition of handwritten Bangla basic characters is presented. Our work is based on a large database of real-life handwritten samples. We proposes an algorithm for Bangla handwritten character recognition that uses zone density feature and most importantly, spatial relationships in terms of directional relationships among each foreground pixel with background pixels to characterize each individual Bangla character and later the feature descriptions are learned into a neural network model. The learned neural network later is used to recognize future instances of test data. During the simulation it is observed that the spatial relationships provide useful information to describe the shape of a character in a better way.

The rest of this article is organized as follows. In Section II, the database used for training and test of the proposed recognition methods has been described along with some unique characteristics of Bangla characters. Our recognition algorithm is described in Section III. Experimental results are presented in Section IV. Finally, concluding remarks are given in Section V.

II. Handwritten Bangla Character Database

In this section, we will give a brief description of the bangla character database that we have worked with in this paper. Bangle is a language mixed of syllabic and alphabetic scripts. The characteristics of Bangla characters are complicated as the characters hugely vary in size and form. The origin of the script of Bangla is an ancient Indian script called Brahmi. Unlike English alphabet, the Bangla alphabet has no equivalent capital forms.

The complications that we encountered during the processing of the Bangla scripts are mainly due to the relatively large symbol set which come with 11 vowels (Shoroborno) and 39 consonants (Benjonborno). These 50 characters have unique shapes of each which is a prime challenge lies in our recognition task. Moreover, three unique characteristics are associated with the bangla characters which imposes yet more challenges in the recognition tasks. Those are:

- Matra: A horizontal line segment in the upper part of the symbol (shown in Table I(a))
- Upper part: Sometimes, a portion of a character extend vertically above the horizontal line segment (shown in Table I(b))
- Disjoint section: Some symbols have multiple continuous sections that are completely disjoint ((shown in Table I(c))

A. Data Collection and Preparation

For collecting data to be used in our research, we have used a moderately large database that we have created with the help of our laboratory co-workers. We collected samples from different people in the University by requesting them to fill out a form with the 50 individual characters. The form

TABLE I
Properties of Bangla Characters: (a) Matra (First Column) (b) Upper Part (Second Column) (c) Disjoint Section(Third Column)

consists of individual boxes to contain the characters. Subjects wrote one character per box. Subjects were requested to write in their natural style which reflected in our dataset. In some cases, we requested the same subject to write again so that we can observe the changes in handwriting style done by the same person. As the subjects used their natural style, we end up with a complex dataset in nature. We realized the characters that we have to work with vary dramatically in size, style and angle. These variations imposed the critical challenges in our research. A sample set of three instances is given in Figure 1.

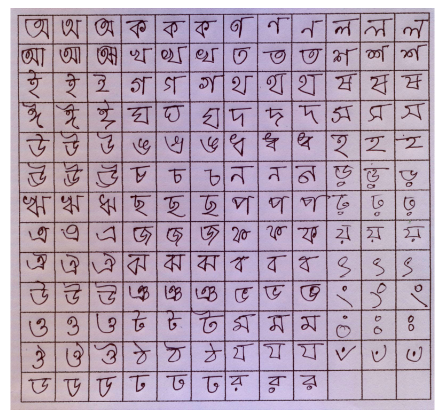


Fig. 1. Sample of Handwritten Bangla script

B. Data Preparation

After we completed collecting filled out forms from our subjects, we started preparing to process those forms in computer. First, we scanned the forms at 300 d.p.i resolution using a standard scanner. Then, we created separate jpeg images for every isolated character. These jpeg image files are later to be used for our research. A total of 200 samples for each character were taken which resulted in a dataset of 10000 image files containing isolated bangle characters.

The entire dataset will be divided into two separate sets: training and test set in order to perform cross-validation which will be explained in Section IV.

III. Proposed Framework

As mentioned before, our proposed method consists of two major steps: training and testing. The training of Bangla characters is one of the most important phases of our recognition task as it requires a suitable and appropriate way to handle and learn the complicated nature of the characters. After we obtain the scanned image containing the filled form, we require to segment every individual handwritten character. In the segmentation stage, an image of sequence of characters is decomposed into sub-images of individual character. In the proposed system, the pre-processed input image is segmented into isolated characters by assigning a number to each character using a labeling process. This labeling provides information about number of characters in the image. Before learning the characters, it is indeed very important to first carry our some preprocessing tasks in order to facilitate the learning phase.

A. Smoothing

As mentioned in section I, it is very common for images to contain noises which results in loss of significant information from the contents and as a result, the model fails to represent every details of an image. The scanning of the images introduce 'speckle' noises which increases the mean grey level of a local area and therefore degrades the quality of an image. To handle this problem, we used a method described in [11] which is called the adaptive weighted median filter (AWMF). This method introduces the weight coefficients to the well known median filter and by proper adjustments of the weight coefficients, the filter method smoothes each point of the image and as a result, the noises are suppressed with edges and other significant features preserved. As this technique can detect the subtle gray scale variation, it is indeed very a very appropriate method for our character recognition problem.

B. Normalization

After we perform the smoothing operation, we normalize the image size into 32 by 32 pixels. Also, the character is transferred to the top left corner of the image.

C. Binarization

Next, we binarize the normalized image using Otsu's global thresholding technique explained in [12]. With this method, a histogram shape-based thresholding is performed which reduces a gray level image to a binary image. An optimal threshold is calculated and therefore, the resulting image produces two classes of pixels: foreground and background.

D. Feature Extraction

The feature description for the individual characters is very important as it is used to learn the properties of every character. In our paper, we have used two kinds of features to represent a handwritten character: (1) Zone density feature and (2) Directional features.

1) Zonal Density Computation: To compute the density of a handwritten character, at first each binarized image I is divided into 16 blocks of size 8×8 pixels. These blocks, referred to as Zones $Z_i; i = 1, \dots\ 16$ each of which contain the pixels $Pj; j = 1, \dots\ 64$. These individual zones describe separate portions of a character. As the image is in binary form, therefore, we can denote the foreground pixels as P_f where $P_f = 1$ and background pixels as P_b where $P_b = 0$. By representing the characters through zones, it is possible to represent the details of a character in a more descriptive way. Next, we calculate the zone density for each individual zone by

$$Density_{Z_i} = \frac{\#\text{ of foreground pixels in } Z_i}{\#\text{ of pixels in } Z_i} \tag{1}$$

This gives us 16 zoning density features for every character.

2) Directional Features Computation: Next, we compute the spatial relationships that exist in each character shape. We will determine the directional relationships among the foreground pixels with the background pixels. We chose this approach because this high-level interpretation of pixel arrangements will provide more semantics into our learning framework which will facilitate the recognition process.

To learn the spatial relationships, we create a directional spatial template $T^{d_w}_{(P_f,P_k)}$ (where P_f is a foreground pixel and P_k is an adjacent neighbor pixel of P_f) for each directional relationship $d_w \in \{d_1, d_2, \dots\ \} \subset (\pi, 2\pi]$. The directions are East, North-East, North, North-West, West, South-West, South, South-East as shown in Fig. 2. This template also to be referred as a 3×3 congruity matrix expresses the influence of neighborhood around the foreground pixel.

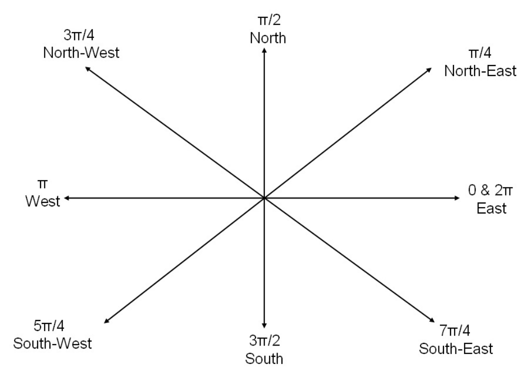


Fig. 2. Cardinal Directions over $(\pi, 2\pi]$

The entries of a directional spatial template is formed in the following way:

- The middle value of the template will always be labeled as P_f as it denotes the foreground pixel for which we are computing the contiguity.
- Angle is calculated for P_f from the starting angle θ with respect to d_w. The entries of the template corresponding to adjacent neighboring pixels that fall in $\theta{+}45$ and $\theta{-}45$ are labeled as 1. Entries corresponding to pixel falling in the direction d_w is labeled as 2 as it contains the summed up value of the two other pixels which are 1.
- For all other pixels P_k, the label is simply 0.

The directional spatial templates for each direction d_w are:

$$T^{East}_{(P_f,P_k)} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & P_f & 2 \\ 0 & 0 & 1 \end{pmatrix} \quad T^{North-East}_{(P_f,P_k)} = \begin{pmatrix} 0 & 1 & 2 \\ 0 & P_f & 1 \\ 0 & 0 & 0 \end{pmatrix}$$

$$T^{North}_{(P_f,P_k)} = \begin{pmatrix} 1 & 2 & 1 \\ 0 & P_f & 0 \\ 0 & 0 & 0 \end{pmatrix} \; T^{North-West}_{(P_f,P_k)} = \begin{pmatrix} 2 & 1 & 0 \\ 1 & P_f & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

$$T^{West}_{(P_f,P_k)} = \begin{pmatrix} 1 & 0 & 0 \\ 2 & P_f & 0 \\ 1 & 0 & 0 \end{pmatrix} \; T^{South-West}_{(P_f,P_k)} = \begin{pmatrix} 0 & 0 & 0 \\ 1 & P_f & 0 \\ 2 & 1 & 0 \end{pmatrix}$$

$$T^{South}_{(P_f,P_k)} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & P_f & 0 \\ 1 & 2 & 1 \end{pmatrix} \; T^{South-East}_{(P_f,P_k)} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & P_f & 1 \\ 0 & 1 & 2 \end{pmatrix}$$

After we create the directional spatial templates for every direction d_w, we look at the pixel values that fall in $\theta + 45$ and $\theta - 45$. If the pixel is a background, we keep the corresponding entry as 1 and change it to a 0 if the pixel is a foreground. Next, we add those entries (including the one that represents the direction d_w and we obtain the feature value $F^{d_w}_{P_f,P_k}$ for the foreground pixel P_f at the direction d_w. After we obtain the feature values for every foreground pixel in the zone, for every direction d_w, we construct a zonal feature value $ZF^{d_w}_{P_f,P_k}$ by simply adding $F^{d_w}_{P_f,P_k}$. We use the following equation:

$$ZF^{d_w}_{P_f,P_k} = \sum F^{d_w}_{P_f,P_k} \tag{2}$$

This equation gives us 8 feature values for every Z_i. After we obtain all the zonal feature values, we end up with $16 \times 8 = 128$ feature values for every character. Therefore, to represent each handwritten character, we have 16 zoning density features $+$ 128 zonal directional features $= 144$ features

E. Designing Classifier

At this point, we have feature representation of the individual handwritten characters, each consisting of 144 features. Next, these features need to be learned into a classifier which can classify the future instances of test handwritten data. In this paper, we have used the well-known Multilayer Perceptron (MLP), a feedforward artificial neural network model that maps sets of input data onto a set of appropriate output. For training purposes, the backpropagation algorithm [13] can be used. But the performance of this algorithm critically depends on the choices of several parameters, namely; the learning rate and the momentum factor. For a careful selection of these parameters, we have used a modified version of the algorithm known as adaptive backpropagation algorithm [14] which uses self-adaptive learning rates.

The hidden layers used the log sigmoid as the activation function. The number of input neurons is determined by the length of each feature vector which is 144 in our case. The total numbers of characters determines the number of neurons in the output layer which is 50 for our problem (11 vowels and 39 consonants).The performance function is measured by the mean square function.

F. Recognizing Test Data

The section explains the steps of performing recognition for new and future test data. When a new image of a handwritten character is given, it is first filtered to eliminate noises and the normalized image is converted into its equivalent binary form. From the binary image, the feature vectors are computed using the methodology explained in the Section 3-D and the equations 1 and 2.

After the feature vector is generated for the test character, it is feed forwarded into our training model which can classify it according to the 50 classes for which it was learned with the representation characteristics.

IV. Experimental Results

In this section, we present our preliminary experiments on some digital color images. We will investigate the performance of our proposed algorithm by implementing the individual steps of it. As mentioned in Section II, we have a dataset of 10000 images upon which we performed the learning procedure. Table II shows the results for a training image that has been normalized, filtered and converted into a corresponding binary image.

The computed feature vector which consists of 144 features (both zonal density and directional relationships) is trained into the MLP. As mentioned in the previous section, the structure of the MLP includes an input layer with 144 input nodes, two hidden layers each with 100 neurons each and an output layer with 50 neurons. The back propagation method with momentum and adaptive learning rate and log-sigmoid transfer functions are used for neural network training.

The performance of the training process through MLP classifier is measured by calculating Mean Square Error (MSE) [15] and we record the number of epochs that were needed to achieve our performance goal regarding MSE. Fig. 3 illustrates the plot between MSE and number of epochs in our training phase.

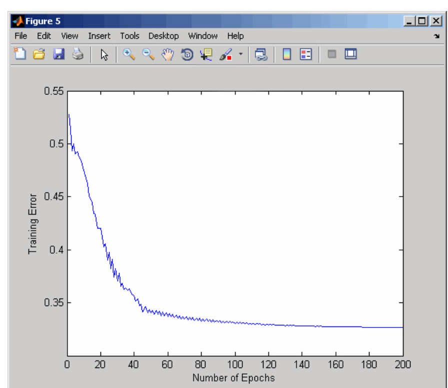
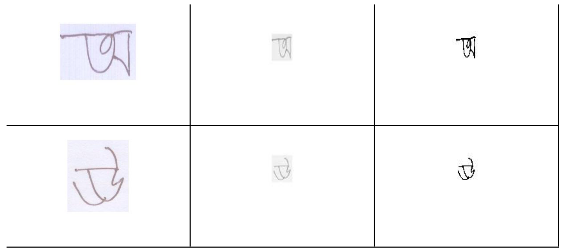


Fig. 3. The variation of MSE with training Epochs

Next, we focus on determining recognition accuracy of the algorithm on our dataset. We attempted to avoid the case where the recognition produces highest accuracy on the training data but does not work well in case of unseen, new

TABLE II
ORIGINAL CHARACTER (FIRST COLUMN), NORMALIZED & FILTERED CHARACTER (SECOND COLUMN) AND THE BINARIZED CHARACTER (THIRD COLUMN).



data. This overfitting problem restricts the determination of recognition accuracy of a model. To handle this situation, we perform cross-validation which is a generally applicable and very useful technique for recognition accuracy estimation. It consists of partitioning a data set T_r into n subsets T_{r_i} and then running the training algorithm n times, each time using a different training set T_r-Tr_i and validating the results on T_{r_i}. Based on the cross-validation approach, we tested on each character by taking 100 sample values from the test set. In this way, a total of 5000 test cases were found.

To measure accuracy, we used three parameters namely recall/True Positive Rate (TPR), precision and False Positive Rate (FPR). Recall/TPR is the measurement which is the fraction of relevant or correct instances that are retrieved over the entire dataset. Precision is the fraction of retrieved instances that are relevant. FPR is the rate of negative examples incorrectly labeled as positive.

$$Recall/TPR = \frac{TP}{TP+FN},\ Precision = \frac{TP}{TP+FP},\ FPR = \frac{FP}{FP+TN}$$

where *TP* = correctly classified characters, *FN* = positive examples incorrectly labeled as negative, *FP* = Misclassified characters and *TN* = negative examples correctly labeled as negative

Table III shows the recognition results for the test images. Each character was tested for 100 times to measure its class. Among the test cases both positive and negative examples were provided. Based on the recognition results for test data, we have plotted an ROC curve between average Recall/TPR and FPR among groups of characters. The groups were chosen based on range of TPR and FPR values. The graph is illustrated in Fig.4 for the values shown in Table IV.

TABLE IV
AVERAGE TPR AND FPR VALUES

Range of TPR	Average TPR	Average FPR
86-88.99	86.75	11.77
89-91.99	91.22	12.58
92-94.99	93.44	16.46
95-97.99	96.69	18.06
98-100.00	98.89	47.74

We made an performance comparison of our method with two other existing bangla handwritten character recognition methods. The details of the comparative study is illustrated in Table V.

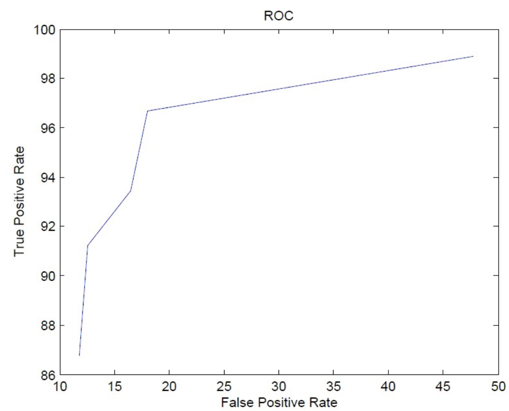


Fig. 4. ROC Curve

TABLE V
COMPARATIVE ASSESSMENT

method	Extracted Features	Classifier	Accuracy
Subhadip Basu et al.[16]	24 shadow features, 16 centroid features, 36 longest-run features	MLP	80.76% (Test)
S. K. Parui et al.[17]	Different strokes for 50 character classes	HMM	84.6%
Our proposed approach	144 feature (zonal density and directional relationship)	MLP	88.64%

V. CONCLUSION

In this paper, we proposed an algorithm to support the recognition Bangla handwritten characters. Due to the complicated shape characteristics of Bangla characters, this paper attempted to represent the details associated with each character in order for producing a correct recognition. We proposed zonal representation of characters through which we measured the zonal density and directional feature computations that represent the spatial relationships among the foreground and the background pixels. The inclusion of spatial relationships provided more high-level semantics to the characters through which the complex representation was handled. Finally, an

TABLE III
SAMPLE TEST IMAGES AND THEIR CORRESPONDING PERFORMANCE MEASURES (IN %

Class	TP	FP	TN	FN	Recall/ TPR	FPR	Precision	Accuracy	Class	TP	FP	TN	FN	Recall/ TPR	FPR	Precision	Accuracy
অ	80	18	1	1	98.77	94.74	81.63	81.00	আ	80	3	16	1	98.77	15.79	96.39	96.00
ই	78	7	13	2	97.50	35.00	91.76	91.00	ঈ	81	5	13	1	98.78	27.78	94.19	94.00
উ	85	2	13	0	100.00	13.33	97.70	98.00	ঊ	85	2	10	3	96.59	16.67	97.70	95.00
ঋ	71	9	13	7	91.03	40.91	88.75	84.00	এ	80	18	1	1	98.77	94.74	81.63	81.00
ঐ	77	4	16	3	96.25	20.00	95.06	93.00	ও	82	0	14	4	95.35	0.00	100.00	96.00
ঔ	79	2	12	7	91.86	14.29	97.53	91.00	ক	80	0	14	6	93.02	0.00	100.00	94.00
খ	77	9	11	3	96.25	45.00	89.53	88.00	গ	80	4	12	4	95.24	25.00	95.24	92.00
ঘ	80	0	13	7	91.95	0.00	100.00	93.00	ঙ	80	1	17	2	97.56	5.56	98.77	97.00
চ	81	5	10	4	95.29	33.33	94.19	91.00	ছ	82	16	1	1	98.80	94.12	83.67	83.00
জ	76	9	13	2	97.44	40.91	89.41	89.00	ঝ	75	0	19	6	92.59	0.00	100.00	94.00
ঞ	79	1	15	5	94.05	6.25	98.75	94.00	ট	81	0	14	5	94.19	0.00	100.00	95.00
ঠ	82	3	9	6	93.18	25.00	96.47	91.00	ড	84	2	11	3	96.55	15.38	97.67	95.00
ঢ	80	4	10	6	93.02	28.57	95.24	90.00	ণ	79	1	12	8	90.80	7.69	98.75	91.00
ত	80	3	16	1	98.77	15.79	96.39	96.00	থ	82	3	14	1	98.80	17.65	96.47	96.00
দ	82	3	13	2	97.62	18.75	96.47	95.00	ধ	64	7	25	4	94.12	21.88	90.14	89.00
ন	72	3	14	11	86.75	17.65	96.00	86.00	প	81	4	11	4	95.29	26.67	95.29	92.00
ফ	80	7	12	1	98.77	36.84	91.95	92.00	ব	84	1	12	3	96.55	7.69	98.82	96.00
ভ	81	17	1	1	98.78	94.44	82.65	82.00	ম	78	6	14	2	97.50	30.00	92.86	92.00
য	83	3	9	5	94.32	25.00	96.51	92.00	র	76	0	16	8	90.48	0.00	100.00	92.00
ল	82	1	14	3	96.47	6.67	98.80	96.00	শ	79	4	15	2	97.53	21.05	95.18	94.00
ষ	82	2	10	6	93.18	16.67	97.62	92.00	স	85	0	13	2	97.70	0.00	100.00	98.00
হ	83	1	13	3	96.51	7.14	98.81	96.00	ড়	77	7	10	6	92.77	41.18	91.67	87.00
ঢ়	72	1	16	11	86.75	5.88	98.63	88.00	য়	77	6	14	3	96.25	30.00	92.77	91.00
ং	82	1	15	2	97.62	6.25	98.80	97.00	ৎ	79	4	16	1	98.75	20.00	95.18	95.00
ঃ	81	1	15	3	96.43	6.25	98.78	96.00	ঁ	81	0	17	2	97.59	0.00	100.00	98.00

MLP was used to train the features which later was used to classify test instances of handwritten characters. The preliminary experiments were done on a dataset created by collecting samples from volunteers. In future, we would like to provide a more descriptive representation of handwritten characters through which complicated characters (joint characters) can be recognized and also run the algorithm in a more complex dataset.

REFERENCES

[1] S. Mori, C. Suen, and K. Yamamoto, "Historical review of ocr research and development," *Proceedings of the IEEE*, vol. 80, no. 7, pp. 1029 –1058, jul 1992.

[2] S. Impedovo, L. Ottaviano, and S. Occhinegro, "Optical character recognition - a survey," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 05, no. 01n02, pp. 1–24, 1991.

[3] V. Govindan and A. Shivaprasad, "Character recognition a review," *Pattern Recognition*, vol. 23, no. 7, pp. 671 – 683, 1990.

[4] R. Plamondon and S. N. Srihari, "On-line and off-line handwriting recognition: A comprehensive survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 1, pp. 63–84, Jan. 2000.

[5] N. Arica and F. T. Yarman-Vural, "An overview of character recognition focused on off-line handwriting," *Trans. Sys. Man Cyber Part C*, vol. 31, no. 2, pp. 216–233, May 2001.

[6] U. Bhattacharya and B. B. Chaudhuri, "Handwritten numeral databases of indian scripts and multistage recognition of mixed numerals," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 3, pp. 444–457, Mar. 2009.

[7] U. Bhattacharya, S. K. Parui, M. Shridhar, and F. Kimura, "Two-stage recognition of handwritten bangla alphanumeric characters using neural classifiers," in *IICAI'05*, 2005, pp. 1357–1376.

[8] T. Bhowmik, U. Bhattacharya, and S. Parui, "Recognition of bangla handwritten characters using an mlp classifier based on stroke features," in *Neural Information Processing*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2004, vol. 3316, pp. 814–819.

[9] M. F. A.F.R. Rahman, R. Rahman, "Recognition of handwritten bengali characters: a novel multistage approach," *Pattern Recognition*, vol. 35, no. 5, p. 9971006, May 2002.

[10] A. Dutta and S. Chaudhury, "Bengali alpha-numeric character recognition using curvature features," *Pattern Recognition*, vol. 26, no. 12, pp. 1757 – 1770, 1993.

[11] T. Loupas, W. McDicken, and P. Allan, "An adaptive weighted median filter for speckle suppression in medical ultrasonic images," *Circuits and Systems, IEEE Transactions on*, vol. 36, no. 1, pp. 129 –135, jan. 1989.

[12] N. Otsu, "A threshold selection method from gray-level histograms," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 9, no. 1, pp. 377–393, jan. 1979.

[13] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Parallel distributed processing: explorations in the microstructure of cognition, vol. 1," Cambridge, MA, USA, 1986, pp. 318–362.

[14] U. Bhattacharya and S. Parui, "Self-adaptive learning rates in backpropagation algorithm improve its function approximation performance," in *Neural Networks, 1995. Proceedings., IEEE International Conference on*, vol. 5, nov/dec 1995, pp. 2784 –2788.

[15] S. Bermejo and J. Cabestany, "Oriented principal component analysis for large margin classifiers," *Neural Networks*, vol. 14, no. 10, pp. 1447 – 1461, 2001.

[16] S. Basu, N. Das, R. Sarkar, M. Kundu, M. Nasipuri, and D. K. Basu, "Self-adaptive learning rates in backpropagation algorithm improve its function approximation performance," in *the 2nd National Conf. on Computer Processing of Bangla*, Feb 2005, pp. 285–291.

[17] S. Parui, K. Guin, U. Bhattacharya, and B. Chaudhuri, "Online handwritten bangla character recognition using hmm," in *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, 2008, pp. 1–4.