

## Handwritten Bangla Character Recognition in Machine-printed Forms using Gradient Information and Haar Wavelet

Sekhar Mandal, Sanjib Sur, Avishek Dan  
Department of Computer Science & Technology  
Bengal Engineering & Science University, Shibpur  
India

Partha Bhowmick  
Computer Science and Engineering Department  
IIT Kharagpur, India

**Abstract**—A robust and efficient algorithm to recognize handwritten Bangla (Bengali) characters in machine-printed forms is proposed. It is based on the combination of gradient features and Haar wavelet coefficients. The gradient feature is used to capture local characteristics, and for its sensitivity to the usual deformation and idiosyncrasy of handwritten characters, wavelet transform is used for multi-resolution analysis of character images. Such a strategy with combined features captures adequate global characteristics in different scales. Two feature-combination schemes are devised and tested on test images of 4372 instances of 49 characters and 10 numerals, after being trained by a set of  $59 \times 25 = 1475$  images. Finally, a  $k$ -NN classifier is used for the character recognition, which shows 87.65% and 88.95% recognition accuracies for the two schemes.

**Keywords**—Handwritten Bangla; Handwritten character recognition; Machine-printed forms; Haar wavelet; Document image processing.

### I. INTRODUCTION

The basic hurdle in any form processing system is in recognizing the handwritten characters contained in the form. Several works have been done on the recognition of English handwritten characters. Softwares are also available to process forms filled up in English language. But no significant work or software is noticed till date to process forms filled up in Bengali (also called ‘Bangla’) language, which is the one of most spoken-and-written language in India (ranking 2nd, with Indian speakers’ percentage 8.11%, according to 2001 census).

The challenge in handwritten character recognition is mainly caused by the large variation of individual writing styles [11]. Hence, robust feature extraction is very important to improve the performance of a handwritten character recognition system. Again, the difficulty with Bengali character recognition is the large number of classes that exist in Bengali. There are 10 numerals, 49 basic characters, and more than 150 compound characters present in this language. Added to this, the presence of vowel modifiers complicate things to a large extent.

Many feature have been developed in the last few decades. M. Shi *et al.* [12] studied the use of gradient and curvature of gray-scale character images to improve the recognition accuracy. They presented three procedures to estimate the curvature of gray-scale curves, based on curvature coefficient, bi-quadratic interpolation, and gradient-vector interpolation. They composed a feature vector of the gradient and the curvature by simple concatenation and cross product. Results show that the direction of gradient is necessary for shape discrimination and the composite features by the cross product to achieve a higher recognition rate. S.-W. Lee [7] utilized the coefficients of wavelet transform as a feature for character recognition. They suggested that character images of different resolutions characterize different structures of the character. This method actually denotes a global feature in multi-resolution analysis. G.A. Fink *et al.* [4] have proposed an online Bangla handwriting recognition that considers cursively written words instead of isolated characters. It uses a sub-stroke level feature representation of the script and a writing model based on hidden Markov models. Mane and Ragha [9] have proposed an elastic image matching technique for recognition of off-line isolated English handwritten digits by matching against a sequence of templates. During pre-processing, it reduces some undesirable variability by filtering, normalization, segmentation, etc.; template matching is based on Euclidean and Mahalanobis similarity measures.

In this paper, we propose a hybrid technique based on *gradient feature* and coefficients of *wavelet transform* for recognition of characters inside a form. Generally speaking, the gradient feature represents *local characteristics* of a character image properly, but it is sensitive to the deformation of handwritten characters. And the gradient feature becomes more and more vulnerable with the increasing deformation and idiosyncrasy typically found in a Bengali handwriting. On the contrary, the wavelet transform represents the character image for multi-resolution analysis and keeps adequate *global characteristics* of a character

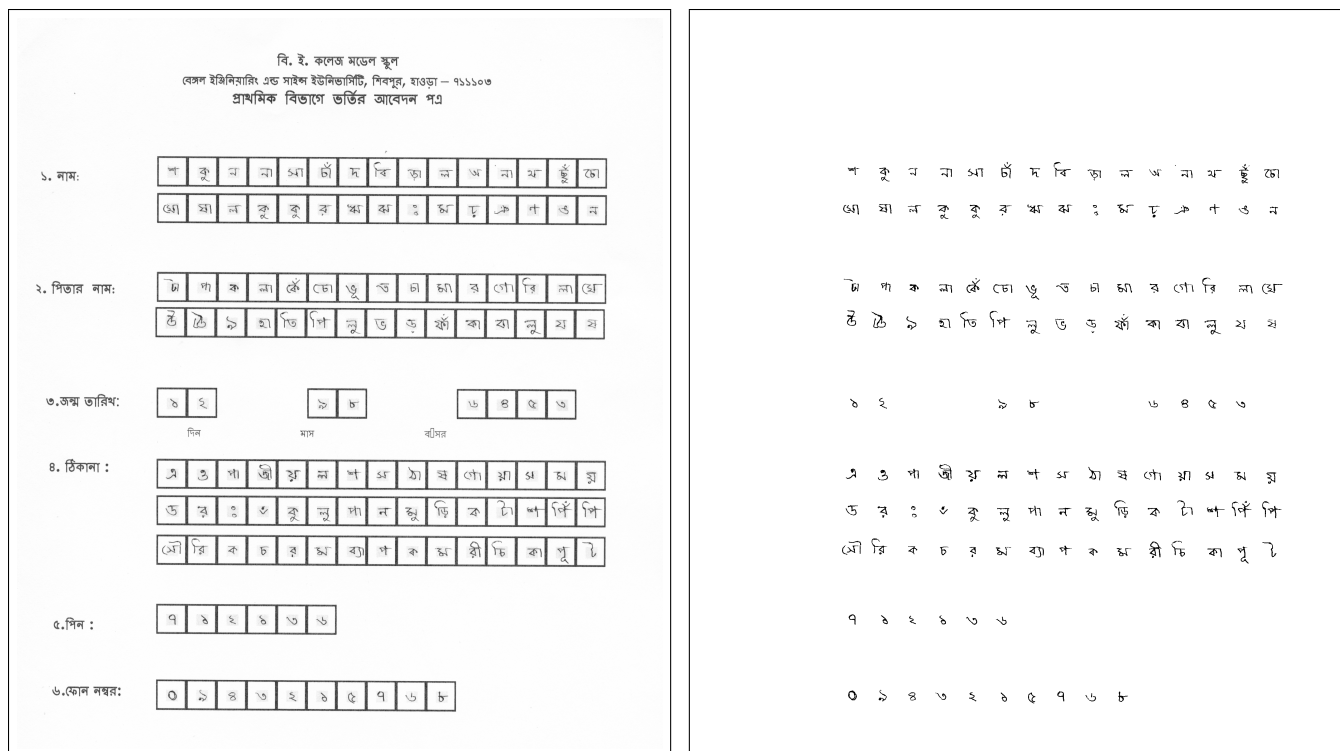


Figure 1. Left: A sample handwritten form with machine-printed boxes containing handwritten characters. Right: Segmentation of handwritten characters by our algorithm.

image in different scales. Hence, in order to improve the discriminating power of a feature, it is better to compose local and global characteristics into a combined feature. We have conducted experiments on the recognition of basic Bengali characters to evaluate the performance of our two schemes based on 1-NN classifier. And the accuracies are found to be 87.65% and 88.95%, which is quite encouraging.

The rest of the paper is organized as follows. Section II describes the preprocessing and extraction of characters from handwritten forms. Section III describes the generation of combined features in detail. Section IV briefs the character recognition using 1-NN classifier. Experimental results are given in Section V and the concluding notes in Section VI.

## II. PREPROCESSING OF FORMS

The initial task in any form processing system is to scan the hard-copies of forms to produce gray-scale digital images. An example of scanned form used in our experimentation is shown in Fig. 1. The gray-scale image is converted into a binary image, since the handwritten data is independent of the color of the ink used to fill up the form. We have used Otsu's binarization method [10] to convert gray-scale images into their binary forms.

### A. Form Segmentation

The form segmentation is first performed by us on the aforesaid binary image to yield a form that contains hand-

written characters inside the form. The printed lines and characters inside the form are deleted to get an image consisting of *only* handwritten characters. For this purpose, morphological closing and opening operations are performed using different structuring elements to detect broken characters, broken lines, horizontal and vertical lines, etc.

### B. Character Segmentation

There are three zones in a Bengali character: *top*, *middle*, and *lower*. The *upper baseline* demarcates the separating line between the top and the middle zones, whereas the *lower baseline* demarcates that between the middle and the lower zones. Precise segmentation of these zones is done for extraction of basic characters. First, the three zones are identified using morphological opening operation, and uppermost and lowermost elements are segmented. The major task is to detect and segment middle-zone elements. This is done by an intelligent vertical scan starting from the lower baseline of each character. While scanning, each pixel along the scan line is marked as 'visited'. If the scan encounters any obstruction (object pixel) in its path, then it moves either left or right depending on the type of the neighbor pixel. The scan stops when both the right and the left pixels are already 'visited', and then a new scan starts from the next position from the lower baseline again. During the scan, the instantaneous vertical distance from the lower



Figure 2. Vertical scan (shown in red) done by our algorithm to delete vowel modifiers.

baseline is computed, and if the vertical distance is found to exceed the median height of the handwritten character set, then a cut is performed vertically. The algorithm for vertical scan is as follows.

**begin**

Var: LP = left ( $90^\circ$  w.r.t. vertical) neighboring pixel

Var: RP = right ( $90^\circ$  w.r.t. vertical) neighboring pixel

```

1: for each row
2:   for each column
3:      $x \leftarrow 1$ 
4:     move upward from an unvisited pixel at baseline
5:     if a pixel  $p$  is unvisited & white
6:       then mark  $p$  as VISITED;  $x \leftarrow x + 1$ 
7:     else look at LP
8:     if LP is WHITE
9:       then move upward and goto Step 8
10:    else look at RP
11:    if RP is WHITE
12:      then move upward and goto Step 8
13:    else if ( $x \geq MedianHeight$ )
14:      then make a vertical cut (segmented)
end

```

The cut separates the basic character from the vowel modifier as shown in the Fig. 2. After the middle zone basic characters are separated, those are sent to the recognizer engine for classification and editable production.

### III. FEATURE EXTRACTION

The feature extraction stage consists of three major steps: *gradient computation*, *wavelet transform*, and *feature combination*, which are explained below.

#### A. Gradient Computation

The gray-scale image of each handwritten character is normalized into  $64 \times 64$  size. The *Sobel gradient operators* [5] are then used to compute the gradient. The Sobel operators are two  $3 \times 3$  weighted masks to compute the gradient components in horizontal and vertical directions (Fig. 3). Two gradient components at  $(i, j)$  are computed as

$$g_v(i, j) = +f(i-1, j+1) + 2f(i, j+1) + f(i+1, j+1) - f(i-1, j-1) - 2f(i, j-1) - f(i+1, j-1)$$

and

$$g_h(i, j) = +f(i-1, j-1) + 2f(i-1, j) + f(i-1, j+1) - f(i+1, j-1) - 2f(i+1, j) - f(i+1, j+1).$$

|    |   |   |
|----|---|---|
| -1 | 0 | 1 |
| -2 | 0 | 2 |
| -1 | 0 | 1 |

|    |    |    |
|----|----|----|
| 1  | 2  | 1  |
| 0  | 0  | 0  |
| -1 | -2 | -1 |

Figure 3. Sobel operators. Left: vertical; Right: horizontal.

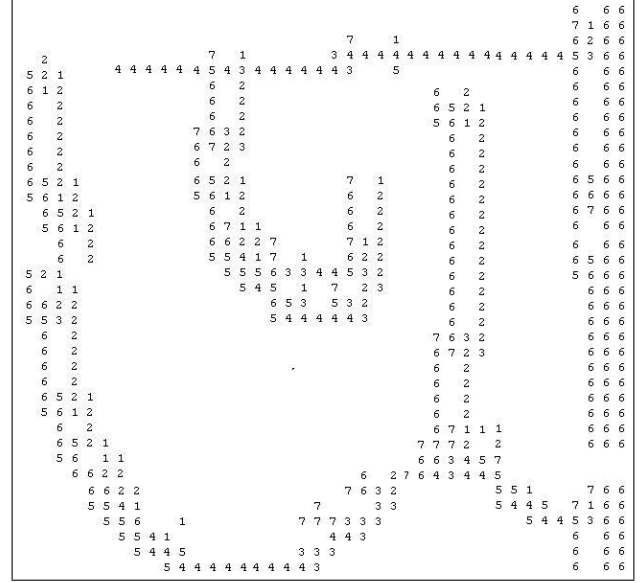


Figure 4. A Bengali character represented by eight direction codes (0, 1, ..., 7), with the '0' information omitted (white fields inside) for sake of clarity.

The gradient strength and direction are computed as

$$G(i, j) = \sqrt{g_v^2(i, j) + g_h^2(i, j)}, \quad \theta = \arctan \frac{g_v(i, j)}{g_h(i, j)}. \quad (1)$$

The direction of gradient is divided into eight equi-length ranges:  $[0^\circ, 45^\circ)$ ,  $[45^\circ, 90^\circ)$ , ...,  $[315^\circ, 360^\circ)$ . These eight directional ranges are mapped to eight direction codes, which are integers starting from 0 ( $[0^\circ, 45^\circ)$ ) and ending at 7 ( $[315^\circ, 360^\circ)$ ). Figure 4 shows a basic Bengali character represented by these direction codes obtained by the Sobel gradient operators.

#### B. Wavelet Transform

Wavelet transform (WT) can be regarded as a transformation that maps a signal to the multi-resolution representation [8]. The coefficients of wavelet transform for a character image give us a scale-invariant representation in multi-resolution analysis. The *continuous wavelet transform* (CWT) decomposes  $f(t)$  by a set of basic functions, namely  $\psi_{a,b}(t)$ , called *wavelets*. Mathematically, a wavelet is defined as  $\psi_{a,b}(t) = \frac{1}{\sqrt{a}}\psi\left(\frac{t-b}{a}\right)$ . And we define a wavelet transform as

$$W(a, b) = \int_t f(t) \frac{1}{\sqrt{a}} \psi\left(\frac{t-b}{a}\right) dt \quad (2)$$

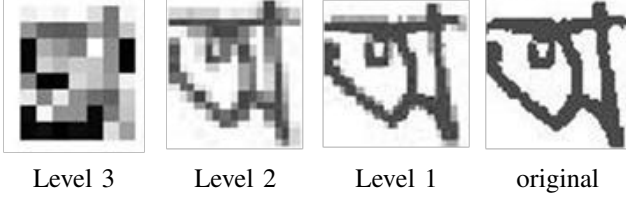


Figure 5. An example of wavelet transforms for a character image up to Level 3.

where  $a$  represents the *scale factor* and  $b$  represents the *translation factor*. For every  $(a, b)$ , we have a wavelet transform coefficient  $W$ , representing how much the scaled wavelet is similar to the function  $f(t)$  at  $t = b/a$ .

In the proposed work, we have used *Haar wavelet* as it is easy to implement and low computational cost. Haar wavelet is defined as

$$\psi(t) = \begin{cases} 1 & \text{if } t \in [0, \frac{1}{2}) \\ -1 & \text{if } t \in [\frac{1}{2}, 1) \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

The character image was decomposed using *Mallat's pyramid algorithm* [8]. An example of Level-3 wavelet transform of the first basic Bengali handwritten character image is shown in Fig. 5.

### C. Feature Combination

Two feature combination schemes are developed and tested by us. Their brief overview is given below.

**Combination Scheme 1.** Each character image (binary and normalized, as mentioned in Sec. III-A) is decomposed up to Level 3 using Haar wavelet. The character image is divided into  $8 \times 8 = 64$  blocks, each block containing  $8 \times 8$  pixels. The directions corresponding to each of these 64 blocks are computed. We get 8 direction values for these blocks (as explained in Sec. III-A). We further reduce them from 8 to 4 directions in total, by considering  $d = d \bmod 4$  so that the larger between two opposite directions, namely  $d$  and  $d \bmod 4$ , map to the smaller between them; i.e., 0 and 4 are mapped to 0 only, 1 and 5 mapped to 1 only, and so on. Thus, finally we get 4 direction codes for these blocks; and as there are 64 blocks, we get a total of  $64 \times 4 = 256$  features. The coefficients of Level-3 Haar wavelet are directly used as the features without any further operation. There are 64 coefficients in total for Level-3 Haar wavelet transformation. Hence, the total feature size in Scheme 1 becomes  $256 + 64 = 320$ .

**Combination Scheme 2.** Scheme 2 is similar to Scheme 1 in its feature domain. The difference is that Scheme 2 only adds the coefficients of Level-2 Haar wavelet transformation directly along with the 320 features used in Scheme 1. Hence, Scheme 2 contains the following features inside a feature vector: (i) the direction ranges mapped to 4 direction values:  $\{0, 1, 2, 3\}$ , and so we get  $4 \times 64 = 256$  features,

along with  $16 \times 16 = 256$  Level-2 Haar coefficients and  $8 \times 8 = 64$  Level-3 Haar coefficients. Hence, the total size of feature vectors in Scheme 2 is  $256 + 256 + 64 = 576$ .

## IV. CLASSIFICATION

The classification of feature vectors is conducted by using 1-NN (nearest neighbor) classifier. In case of a tie, we use 3-NN (and 5-NN for a tie in 3-NN) classifier. For a tie in 5-NN classifier, the character is left as unclassified. Let the training set of feature vectors whose classes are known *a priori*, be  $V = \{\mathbf{v}_i^{(j)} \mid i = 1, 2, 3, \dots, n_j; j = 1, 2, 3, \dots, m\}$ , where  $\mathbf{v}_i^{(j)}$  denotes the  $i$ th test vector that belongs to Class  $C_j$ ,  $n_j$  is the number of training vectors from  $C_j$ , and  $m$  denotes the number of classes. For 1-NN, an input test vector  $\mathbf{x}$  is classified to Class  $C_k$  according to the following decision function:

$$\mathbf{x} \in C_k \text{ if } \|\mathbf{x} - \mathbf{v}_i^k\|^{\frac{1}{2}} = \min_{\substack{1 \leq i \leq n_j \\ 1 \leq j \leq m}} \|\mathbf{x} - \mathbf{v}_i^j\|^{\frac{1}{2}}, \quad (4)$$

where  $\|\mathbf{x} - \mathbf{v}_i^j\|^{\frac{1}{2}}$  denotes the Euclidian distance between the test vector,  $\mathbf{x}$ , and the training vector,  $\mathbf{v}_i^j$ . The database characters are used to prepare the training samples. The classification produces reasonable accuracy in Bengali character recognition.

## V. EXPERIMENTAL RESULTS

We have prepared our own database of basic Bengali character set using random handwriting samples of people around, capable of writing the basic language in forms. The number of Bengali alpha-numeric characters used for training is 49, each having samples of 25, thus giving the total number of characters in training data set is  $59 \times 25 = 1475$ . We have conducted experiments to demonstrate the performance of the proposed form processing scheme. Both the computational cost and the recognition accuracy of Bengali basic character set have been tested on our test data sets. With the aforesaid training set, the test set consists of 4372 instances of alpha-numeral characters. There are 83 to 112 characters in each form, and 46 forms in total in this test set.

### A. Computational Cost

We have estimated the computational time of processing and extracting features from characters and the classification time. Our CPU is Core 2 Duo 2.67GHz with 1024MB RAM. The results are shown in Table I. The computational time shown in the 3rd column is the total time for processing, feature extraction, and classification for 59 characters.

### B. Recognition Accuracy Experiment

The performance of our algorithms in terms of recognition accuracy is shown in the Fig. 7. The average accuracies of Scheme 1 and Scheme 2 are 87.65% and 88.95% respectively.

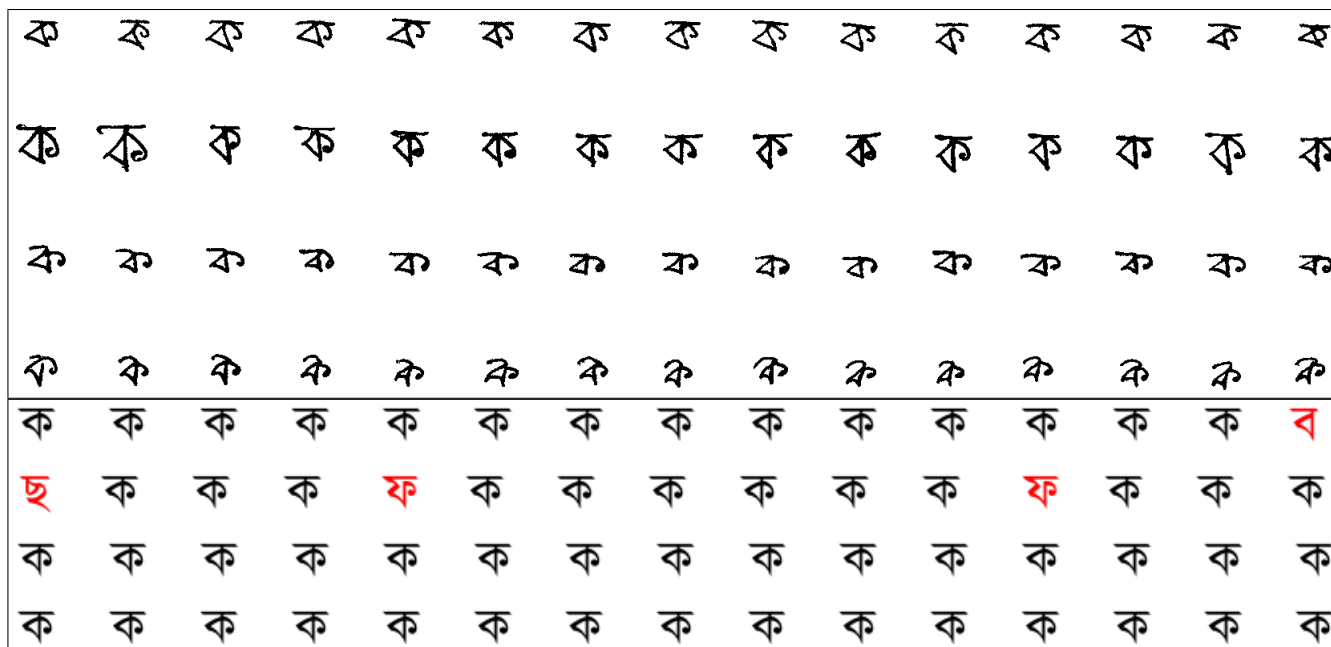


Figure 6. Top: Sample test images of 'ka' (the first Bangla consonant). Bottom: Recognition by Scheme 1; 36 out of 40 characters here are rightly classified, and 4 characters misclassified (15th, 16th, 20th, and 27th; shown in red).

Table I  
SUMMARY OF TEST RESULTS.

| Feature set | Dimension | Time (in secs.) |
|-------------|-----------|-----------------|
| Scheme 1    | 320       | 2.8650          |
| Scheme 2    | 576       | 4.3107          |

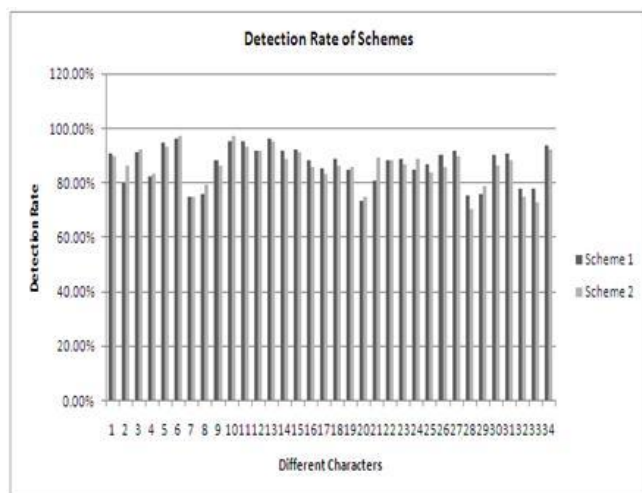


Figure 7. Recognition rates of different characters in Dataset 1 using Scheme 1 and Scheme 2.

The accuracy of the experiment can be further increased by increasing number of sample characters in the database. We have taken a total of 50 samples each of 34 basic characters. So, the total number of samples in the database

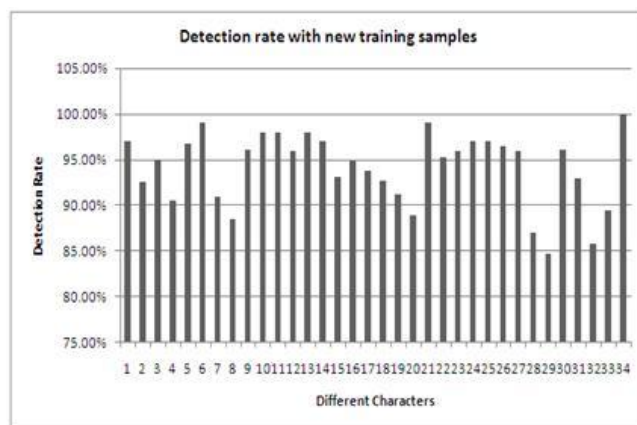


Figure 8. Recognition rates of different characters using Scheme 1 on Dataset 2.

is now  $34 \times 50 = 1700$ . It is quite encouraging to see that the detection rate is appreciably high even when we have taken feature vectors of length 320, i.e., Scheme 1. The plots showing recognition accuracies are presented in Fig. 8. As evident from these plots, the average accuracy in some cases is around 95%, which tells about the strength of feature set used by us.

After the characters are classified, we have converted them into respective Unicodes to produce the editable version of the form. The Unicode for the data fields can be used for further processing of the form.

## VI. CONCLUSION

A fully data-driven and automatic model has been built, the novelty of which lies in the feature extraction. We proposed a handwritten character recognition feature based on the combination of gradient feature and coefficients of wavelet transform. Wavelet transform has been used for the past two decades for various image processing jobs. However its application to the classification in Bengali handwritten character recognition is a new idea. Two combination schemes were proposed in this paper. The proposed handwritten character recognition is used to convert the handwritten forms into editable forms using the preprocessing and character extraction and classification techniques discussed. The result of the recognition accuracy is exciting and we hope that the recognition schemes are far better than other schemes currently available in the market to recognize Bengali handwritten characters. The forms are used because of the fact it can inherently align the characters without further re-alignment of the characters. The experimental results indicate the following points.

- 1) The direction feature is the domination characteristic in any scale image for handwritten character recognition.
- 2) The combination of local gradient feature and global feature is feasible and useful to improve the recognition accuracy.
- 3) For large inner-variations pattern recognition task, such as handwritten character recognition, it is a better way to compose different scale features into a feature vector to overcome the difficulty caused by the large inner-variations.

## REFERENCES

- [1] T. Acharya and P.S. Tsai. Computational Foundations of Image Interpolation Algorithm. ACM Ubiquity, Vol. 8. 2007.
- [2] B. B. Chaudhuri and U. Pal. A complete printed Bangla OCR system. *Pattern Recognition*, Vol. 31(5):531–549, 1998.
- [3] R. O. Duda, P.E. Hart, and D. G. Stork. Pattern Classification, 2nd Edition, Wiley Interscience, 2007.
- [4] G. A. Fink, S. Vajda, U. Bhattacharya, S. K. Parui, B. B. Chaudhuri. Online Bangla Word Recognition Using Sub-Stroke Level Features and Hidden Markov Models. In *Proc. ICFHR 2010*, pp. 393–398.
- [5] R. C. Gonzalez, R. E. Woods. Digital Image Processing, Second Edition, Prentice Hall, 2008.
- [6] Md. A. Hasnat, M. R. Chowdhury, and M. Khan. Integrating Bangla Script Recognition Support in Tesseract OCR. In *Proc. of the Conference on Language and Technology 2009 (CLT09)*, Lahore, Pakistan, 2009.
- [7] S.-W. Lee, C.-H. Kim, H. Ma, and Y. Y. Tang. Multiresolution recognition of unconstrained handwritten numerals with wavelet transform and multilayer cluster neural network. *Pattern Recognition*, 29(12):1953–1961, 1996.
- [8] S. Mallat. A theory for multi-resolution signal decomposition: The wavelet representation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 11(7):674–693, 1989.
- [9] V. Mane and L. Ragha. Handwritten digit recognition using elastic matching. In *Proc. International Conference & Workshop on Emerging Trends in Technology (ICWET '11)*. pp. 1364–1364, 2011.
- [10] N. Otsu. A threshold selection method from gray-level histogram. *IEEE Trans. SMC*, 9(1):62–66, 1979.
- [11] R. Plamondon and S. N. Srihari. On-line and off-line handwriting recognition: A comprehensive survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(1):63–84, 2000.
- [12] M. Shi, Y. Fujisawa, T. Wakabayashi, and F. Kimura. Handwritten numeral recognition using gradient and curvature of gray scale image. *Pattern Recognition*, 35(10):2051–2059, 2002.
- [13] K.P. Soman, K.I. Ramachandran, and N.G. Resmi. Insights Into Wavelets: From Theory to Practice, Third edition, Prentice-Hall, 2004.
- [14] W. Zhuang, Y. Y. Tang, and Y. Xue. Handwritten Character Recognition Using Combined Gradient and Wavelet Feature. In *Proc. International Conference on Computational Intelligence and Security*, pp. 662–667, 2006.