Neural Computing and Applications
https://doi.org/10.1007/s00521-018-3389-1

ORIGINAL ARTICLE

Off-line Bangla handwritten word recognition: a holistic approach

Showmik Bhowmik[1] · Samir Malakar[2] · Ram Sarkar[1] · Subhadip Basu[1] · Mahantapas Kundu[1] · Mita Nasipuri[1]

Received: 9 September 2017 / Accepted: 21 February 2018

Abstract

Due to the cursive nature, segmentation of handwritten Bangla words into characters and also recognition of the same sometimes become a very challenging problem to the researchers. Presence of comparatively large character set along with modifiers, ascendants, descendants, and compound characters makes the segmentation task more complex. As holistic method avoids such character-level segmentation, it is generally useful for the recognition of words written in any such complex scripts. In the present work, a holistic handwritten word recognition method is developed using a feature descriptor, designed by combining different Elliptical, Tetragonal and Vertical pixel density histogram-based features. Recognition process is carried out separately using two classifiers, *namely* multi-layer perceptron (MLP) and support vector machine (SVM). For evaluation of the proposed method, a database of 18,000 handwritten Bangla word images, having 120 word classes, is prepared. The proposed system performs comparatively better with SVM than MLP for the prepared dataset. It has achieved 83.64% accuracy at best case and 79.38% accuracy on an average using fivefold cross-validation. The current method has also outperformed some recently reported holistic word recognition technique tested on the developed dataset. In addition to that the database, prepared in this work, is made freely available to fill the absence of a publicly available standard database for holistic Bangla word recognition.

Keywords Holistic word recognition · Handwritten word · Tetragonal feature · Elliptical feature · Vertical pixel density histogram-based feature · Bangla script

1 Introduction

Automatic recognition of handwritten text is one of the most popular areas of research in the domain of document image processing [1, 2]. The reason of its popularity lies in its wide range of applications in human society which include postal automation [3, 4], bank check processing [5, 6], form processing [7, 8], etc. Major difficulty in recognizing the handwritten text is mainly due to the varying writing styles of individuals. Even the script in which the text is written can pose additional challenges. For example, some Indic scripts like Devanagari and Bangla comprise a considerably larger character set in comparison with Roman/Latin script. Chinese, Japanese and Korean scripts also have large character sets but in these scripts characters appear isolated in the text, whereas Indic scripts are very often written in cursive manner. Therefore, development of a comprehensive and accurate handwritten text recognition system in Indic script is difficult and needs more attention from the researchers [9].

In the literature, plenty of work can be found for the recognition of words written in Arabic [10–12], Chinese/Japanese [13–15] and Roman [16–18] scripts. But in comparison with that very few attempts have been made for the recognition of words written in Bangla script. With more than 200 million speakers, Bangla is the seventh most spoken language in the world [19]. It is also the second most popular official language (out of 23 official languages) in India and the national language of Bangladesh. Besides Bangla language, Bangla script is also used to write other languages like Assamese and Manipuri. Although a significant number of work have been reported

✉ Samir Malakar
samirmalakar@ieee.org

[1] Department of Computer Science and Engineering, Jadavpur University, Kolkata, India

[2] Department of Computer Science, Asutosh College, Kolkata, India

Published online: 03 March 2018

for recognition of handwritten Bangla isolated characters [20, 21] and digits [22], very few pieces of work are there for Bangla handwritten word recognition (HWR) and even, existence of standard databases of handwritten Bangla words is also scarce.

To bridge this research gap, in this work, a novel HWR system of Bangla script is developed. For the evaluation of this system, a first-of-its-kind open-access handwritten Bangla word image database, consisting of 18,000 city name images of 120 different classes, is also prepared, which is another major contribution of this paper. The paper is organized as follows: Sect. 2 reviews the related work. Section 3 presents the description of the proposed work including database preparation, preprocessing and feature extraction. Experimental results are discussed in Sect. 4, and finally, Sect. 5 concludes the paper.

2 Related work

Handwritten text recognition including word recognition is generally performed either in online mode [23, 24] or in off-line mode [25, 26]. In online mode, some digital input devices like tabs or I-pads are used as writing medium instead of paper, and the recognition is performed simultaneously while the writing progresses. But in off-line mode, writing medium is a passive surface such as paper and simultaneous recognition is not possible. In online mode, availability of information like positional information of strokes, direction of strokes and their order [26] makes the recognition process relatively less complex in comparison with text written in off-line mode.

In the literature, two different approaches are followed to handle the problem of off-line HWR, *namely* (1) analytical approach [27, 28] and (2) holistic approach [29]. In analytical approach, a word is initially segmented into sub-units called characters and then each sub-unit is recognized sequentially in order to identify the whole word. Most of the earlier word recognition solutions have been developed based on this approach. The major problem with this approach is finding the proper segmentation points, which becomes more challenging when the handwriting is cursive in nature. On the other hand, holistic word recognition considers a word as a single and indivisible unit and thus extracts information from the whole word to recognize it. In this way, this approach avoids the segmentation issue. According to the authors in [29], holistic approach may succeed even when the writing is too poor for the identification of individual character boundary from the word, but it preserves the overall shape. It is worth mentioning here that use of this approach is restricted to the problems with fixed or limited lexicon. Besides that, this approach can also be used for the reduction in the lexicon in large vocabulary problems [29, 30]. There are many contemporary work available in the literature, where holistic approach has been followed to recognize handwritten words. For example Dasgupta et al. [26] have applied Arnold transform followed by Hough transform on the word images to get distribution of stroke orientation. Based on this information, they have recognized the word images holistically. For evaluation of their work, they have used CENPARMI database of legal amount written in English with 32 different word classes. Malakar et al. [31] have extracted several topological features either from the entire word image or from the hypothetically segmented sub-regions of a word image to recognize it holistically. They have used a database of handwritten Hindi word images having 33 different classes to assess the performance of their method. Tamen et al. [32] have described the word images at feature space using Chebyshev moments and some statistical and contour-based features. For classification, they have used a multi-classifier environment. The method is evaluated using a database of handwritten Arabic words having 21 different word classes, which is basically a subset of IFN/ENIT [33] database.

Similar to other languages, most of the earlier work on Bangla HWR are purely segmentation based [3, 4]. As mentioned previously, the major problem with this approach is finding proper segmentation points. Due to the presence of noise, touching character(s), uneven space among the characters in a word, these techniques may end up with over- or under-segmented characters. To overcome this problem, character-based Hidden Markov Model (HMM) is used in [34, 35]. In these methods, a two-stage recognition scheme is introduced for words written in Bangla. Initially, a given word image is segmented into three zones, viz., upper, lower and middle. The character components in the segmented zones are recognized separately, and the zone recognition results are combined to generate the final word recognition score. The character-based HMM is used to recognize the middle zone. The problem of presegmentation of characters from the word images is solved to some extent by this approach, but it introduces zone-level segmentation which may also become very much error prone for unconstrained handwritten words.

However, in the literature, few attempts can be found toward the development of a holistic Bangla HWR scheme. Some of those have followed a lexicon-based holistic approach [36–38], whereas others have focused on employing feature descriptor for that purpose. In [39], Histogram of Oriented Gradient (HOG) feature descriptor is used for holistic handwritten Bangla word recognition, whereas in [40, 41] Elliptical and convex hull-based features are used, respectively, for the same. Recently, in [42, 43], gradient-based feature descriptor is used for holistic Bangla HWR. In the later one, a memetic algorithm-based feature selection technique is also proposed to enhance the recognition result. However, most of these methods are evaluated on a very small dataset, so the robustness of these methods cannot be assured.

3 Proposed work

In the present work, a novel shape-based feature descriptor called *Tetragonal feature* is used along with *Vertical pixel density histogram-based feature* and our previously introduced *Elliptical feature* [40] to capture the shape or geometric nature of a handwritten word image in order to recognize it in a holistic manner. Recognition process separately employs two well-known classifiers, viz., MLP [44] and SVM [45]. Each step of the proposed work is elaborated in the subsequent sections.

3.1 Database description

One of the main reasons for the slow progress of research on HWR for regional languages is the unavailability of suitable databases. As holistic HWR systems are generally developed for specific applications, these mainly deal with limited lexicons. To carry out training and evaluation of such systems, some handwritten word databases like IFN/ENIT (in Arabic script) [33] and CENPERMI (in Roman script) [46] are made available to the research community either on-demand basis or through subscription charges. However, no such standard database comprising Bangla handwritten words is available in the literature, which can be used to train and evaluate a holistic HWR system.

To address this need, in the present work, a database of 18,000 handwritten Bangla word images, written by around 300 different native writers belonging to different age groups, sex and educational backgrounds, is prepared. Word images in the current database contain the names of 120 different cities in West Bengal, a state of India, with 150 samples for each city name. The city names listed in Table 1 are chosen based on their population and the literacy rate. The present database includes almost all urban regions of West Bengal. Handwritten words in this work were collected in A4 size datasheets containing a grid of 10 rows and 3–5 columns depending upon the word length. The writers were asked to write each city name inside the rectangular boxes only. Such datasheets are then scanned using a flat-bed scanner with a resolution of 300 dpi and are stored as 24 bit BMP image file. From each such image of the datasheets, handwritten word images are cropped automatically. More detail description regarding the height, width, number of ascendants and descendants of the word images present in each word class is given in a file named "ReadMeCMATERdb2.1.2.pdf" along with the database. This information will help the researchers to get an idea about how much intra- and/or inter-class shape and size variation their method may tolerate. More information regarding the samples of the present dataset is given in Tables 2 and 3.

Database nomenclature is an important task for future reference by the research community. The database prepared in this work is given the name "CMATERdb2.1.2" based on our predefined naming convention [47]. One of the main contributions of the current work is that the database is made freely available to the entire research community of the said domain. To access the database refer to the link given in [48].

3.2 Preprocessing

Before feature extraction, all the word images are smoothed using disk filter with radius 4 to remove noise and then binarized using global histogram-based Otsu method [49]. After that, two morphological operations, viz., erosion and dilation [50] are performed on the binarized word images using a 3×3 structuring element to remove the isolated foreground[1] pixels as well as to smooth the contours of the object regions. These operations also help in filling up the holes, created during binarization, within the object regions.

3.3 Feature extraction

The proposed feature descriptor is designed with the intension of capturing shape information of a given word image. In the field of object recognition, shape information of an object is heavily relied upon for the discrimination purpose [51]. But most of these shape descriptors are computationally expensive and require extra memory space for further processing. Keeping that in mind, in the present work, shape information of a handwritten word is obtained in terms of *local distribution of foreground pixels* along with *presence* and *position of ascendant(s) and descendent (s)*. For that, from a given word image several *Elliptical* [40], *Tetragonal* and *Vertical pixel density histogram-based* features are extracted. The proposed descriptor is not only simple to compute but also effective in recognition of large number of handwritten word classes which are truly complex in nature.

[1] By foreground pixels in a word image, we mean object pixels only and the rest of the pixels are considered as representing the background. In this paper, we have followed this convention.

Table 1 City names, used in the database, written in Bangla along with their class numbers

Class #	Name	Class #	Name	Class #	Name	Class #	Name
1	আলিপুর	31	বনগাঁ	61	এগরা	91	কোন্নগর
2	বালুরঘাট	32	বানপুর	62	ফুলিয়া	92	কুলটি
3	বাঁকুড়া	33	বাঁশবেড়িয়া	63	গঙ্গারামপুর	93	লালগোলা
4	বারাসাত	34	বাঁশড়া	64	গারুলিয়া	94	মধ্যমগ্রাম
5	বর্ধমান	35	ব্যারাকপুর	65	গায়েশপুর	95	মহেশতলা
6	বহরমপুর	36	বরানগর	66	ঘাটাল	96	মেমারি
7	চুঁচুড়া	37	বারুইপুর	67	গোবরডাঙ্গা	97	মুর্শিদাবাদ
8	কোচবিহার	38	বসিরহাট	68	গুসকরা	98	নবদ্বীপ
9	দার্জিলিং	39	বেলেডাঙা	69	হাবড়া	99	নৈহাটী
10	ইংলিশবাজার	40	বেলেঘরিয়া	70	হলদিয়া	100	নলহাটি
11	হাওড়া	41	ভাটপাড়া	71	হালিশহর	101	ঔরঙ্গাবাদ
12	জলপাইগুড়ি	42	বিরাটি	72	হিজুলি	102	পানিহাটি
13	কলকাতা	43	বীরনাগার	73	ইছাপুর	103	পলাশী
14	কৃষ্ণনগর	44	বিষ্ণুপুর	74	ইসলামপুর	104	রামপুরহাট
15	মালদা	45	বোলপুর	75	জামুরিয়া	105	রানাঘাট
16	মেদিনীপুর	46	বজবজ	76	জঙ্গীপুর	106	রিষড়া
17	পুরুলিয়া	47	চাকদহ	77	ঝাড়গ্রাম	107	সাঁইথিয়া
18	রায়গঞ্জ	48	চাঁপদানি	78	কাজোড়া	108	শান্তিপুর
19	সিউড়ি	49	চন্দননগর	79	কালনা	109	শিবপুর
20	তমলুক	50	চিত্তরঞ্জন	80	কল্যাণী	110	শিলিগুড়ি
21	আগারপাড়া	51	দাঁইহাট	81	কামারহাটী	111	শ্যামনগর
22	আজিমগঞ্জ	52	ডালখোলা	82	কাঁচরাপাড়া	112	সোদপুর
23	আরামবাগ	53	ডানকুনি	83	কান্দি	113	সোনামুখী
24	আসানসোল	54	ধুলিয়ান	84	কাঁকশা	114	সোনারপুর
25	অশোকনগর	55	ধূপগুড়ি	85	কাঁথি	115	শ্রীরামপুর
26	বাদকুল্লা	56	দিনহাটা	86	করিমপুর	116	তারকেশ্বর
27	বৈদ্যবাটী	57	ডোমকল	87	কাটোয়া	117	টিটাগড়
28	বলরামপুর	58	দুবরাজপুর	88	খড়্গপুর	118	উখরা
29	বালি	59	দমদম	89	খড়দহ	119	উলুবেড়িয়া
30	ব্যান্ডেল	60	দুর্গাপুর	90	কোলাঘাট	120	উত্তরপাড়া

3.3.1 Elliptical features

An ellipse is a curved line forming a closed loop, where the sum of the distances from two points (foci) to every other points on the curve is constant. Ellipse can also be defined parametrically as well as nonparametrically.

The parametric definition of an ellipse is,

$$X = C_x + a\cos(t) \tag{1}$$

$$Y = C_y + b\sin(t) \tag{2}$$

where $\left(C_x, C_y\right)$ is the coordinate of the center of ellipse, a is a constant representing the length of its radius along X-axis., b is a constant representing the length of its radius along Y-axis and t is a parameter such that $0 < t < 2\pi$. Equation of an ellipse in nonparametric or Cartesian form is defined as,

$$\frac{(X - C_x)^2}{a^2} + \frac{\left(Y - C_y\right)^2}{b^2} = 1 \tag{3}$$

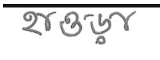
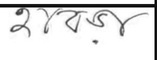
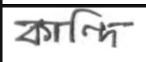
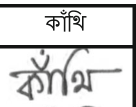
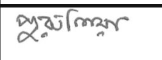
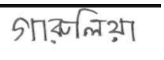
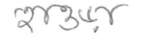
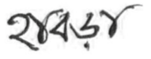
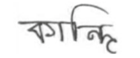
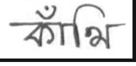
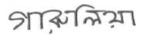
The nonparametric equation of the ellipse is used here for feature computation.

3.3.1.1 Fitting of ellipses on the word image

In this work, some hypothetical concentric ellipses are first fitted on a word image. To decide the center of these ellipses, the center of gravity of a word image is computed as,

$$C_x = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{4}$$

Table 2 Sample word classes having high shape similarity

হাওড়া	হাবড়া	কান্দি	কাঁথি	পুরুলিয়া	গারুলিয়া
					
					

$$C_y = \frac{1}{N}\sum_{i=1}^{N} y_i \qquad (5)$$

where (x_i, y_i) is the coordinate of the ith foreground pixel and N represents the total number of foreground pixels. The constants a and b are computed as follows,

$$a = \min\{(C_x - X_L), (X_R - C_x)\} \qquad (6)$$

$$b = \min\{(C_y - Y_B), (Y_T - C_y)\} \qquad (7)$$

where (X_L, Y_B) and (X_R, Y_T) are the coordinates of the top-left and bottom-right corners of the minimum bounding box of a word image, respectively (see Fig. 1).

While fitting a hypothetical ellipse within a boundary of a word image, our objective is to take most of the foreground pixels inside the ellipse. For that reason, the center of gravity of a word image is considered as the center of the ellipse, as it always resides in the dense foreground region of the image. Now as the center of gravity for a word image may always not be the geometric center of its bounding box, choosing maximum value instead of minimum in Eqs. (6) and (7) may cause a situation, when a portion of the hypothetical ellipse may lie outside the image bounding box. That empty portion of the ellipse will not provide any useful shape information of that word.

3.3.1.2 Computation of feature values Elliptical features over an entire word image are computed in following two ways:

Concentric ellipses Initially, three hypothetical concentric ellipses are considered on a given word image (see Fig. 2). Then from those three ellipses four Elliptical features are computed. Let us assume that the outermost ellipse is marked as first, the next ellipse is marked as second and the innermost ellipse is marked as third. Radii of the ellipses are computed as follows,

$$a_i = \frac{a}{2^{(i-1)}} \qquad (8)$$

$$b_i = \frac{b}{2^{(i-1)}} \qquad (9)$$

where $i = 1, 2, 3$.

Here a_i and b_i represent the parameters a and b of ith ellipse. From each of the ellipses, following *four* feature values representing the *number of foreground pixels* over different local regions of the word image are computed. The local regions considered here are: (i) outside the first ellipse (but within the minimum bounding box of the word image), (ii) *between first and second ellipses*, (iii) *between second and third ellipses* and (iv) *inside the third ellipse*. As different words have different shapes, there may be noticeable differences in the foreground pixel distribution at various local regions created by those concentric ellipses. Our endeavor is to use these differences which eventually help classifier in recognizing the word images.

Outermost ellipse After computation of the feature values from three hypothetical concentric ellipses, outermost ellipse is considered for computation of another *five* feature values. These are (i) *number of foreground pixels on boundary of the ellipse*, (ii) *number of foreground pixels along the axis parallel to X-axis*, (iii) *number of foreground pixels along the axis parallel to Y-axis*, (iv) *ratio of foreground pixels and background pixels inside the ellipse and* (v) *ratio of foreground pixels inside and outside of the ellipse* (not exceeding the minimum bounding box of the word).

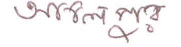
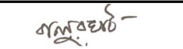
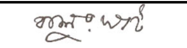
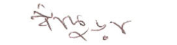
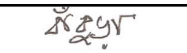
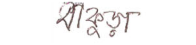
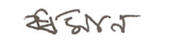
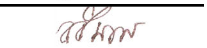
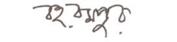
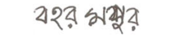
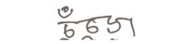
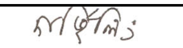
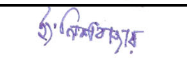
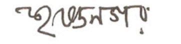
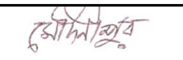
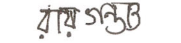
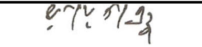
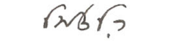
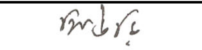
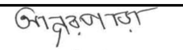
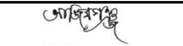
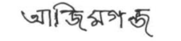
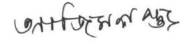
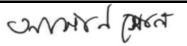
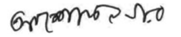
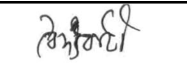
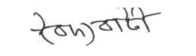
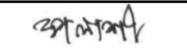
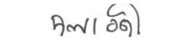
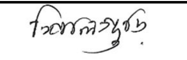
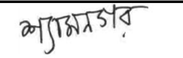
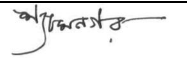
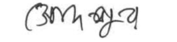
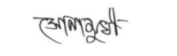
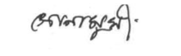
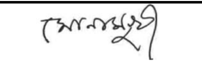
To acquire more information about the shape of a word image, region inside the outermost ellipse is divided into four sub-regions depending on the center and two foci points of the ellipse as shown in Fig. 3. The foci points $(C_x + \delta, C_y)$ and $(C_x - \delta, C_y)$ for the outermost ellipse are computed using the foci distance δ computed as follows,

$$\delta = \sqrt{|a^2 - b^2|} \qquad (10)$$

Finally, *the number of foreground pixels* inside each of these four sub-regions is computed. Thus, in total, *nine* feature values are computed from the outermost ellipse.

From Eq. 10, it is clear that computation of foci points requires the values of a and b, which are directly related to width and height of a given word image, respectively. Widths of the word images vary from one class to another depending on the number of characters in a word and the writing styles of various writers. Thus the value of a would also vary over the different word classes. Similarly, heights

Table 3 Examples of some cursive word samples reflecting the shape variation present in the prepared database

Sl. #	Word Class	Sample 1	Sample 2	Sample 3
1	আলিপুর			
2	বালুরঘাট			
3	বাঁকুড়া			
4	বারাসাত			
5	বর্ধমান			
6	বহরমপুর			
7	চুঁচুড়া			
8	দার্জিলিং			
9	ইংলিশবাজার			
10	জলপাইগুড়ি			
11	কলকাতা			
12	কৃষ্ণনগর			
13	মেদিনীপুর			
14	পুরুলিয়া			
15	রায়গঞ্জ			
16	সিউড়ি			
17	আগারপাড়া			
18	আজিমগঞ্জ			
19	আসানসোল			
20	অশোকনগর			
21	বৈদ্যবাটী			
22	পলাশী			
23	শান্তিপুর			
24	শিলিগুড়ি			
25	শ্যামনগর			
26	সোদপুর			
27	সোনামুখী			

of the word images differ from one word class to another because of the same logic and also due to the presence of ascendants and descendants. So, the value of b also varies for different word types. As the values of these two constants get changed over the word classes, the location of the foci points would also vary accordingly. That means the pixel distribution at various local regions created by those foci points would also differ significantly from one word class to another, which helps during classification.

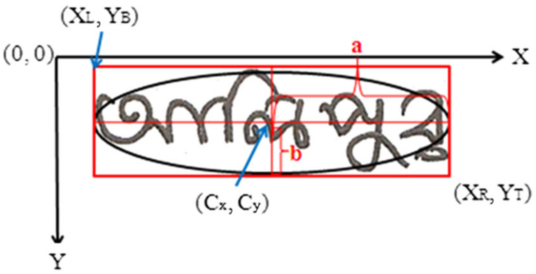


Fig. 1 Illustration of outermost hypothetical ellipse fitted on a sample word image and the related parameters. Here (C_x, C_y) represents the center of gravity of the word image and the center of the hypothetical ellipse. (X_L, Y_B) and (X_R, Y_T) represent the coordinates of the upper-left and lower-right corner of the bounding box of the word image, respectively. a and b are the length of radius along X and Y-axes, respectively

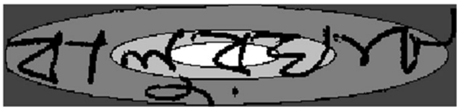


Fig. 2 Three concentric ellipses fit on a word image

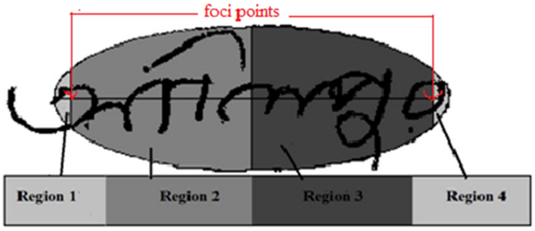


Fig. 3 Illustration of four sub-regions generated based on the foci points by considering the outermost ellipse

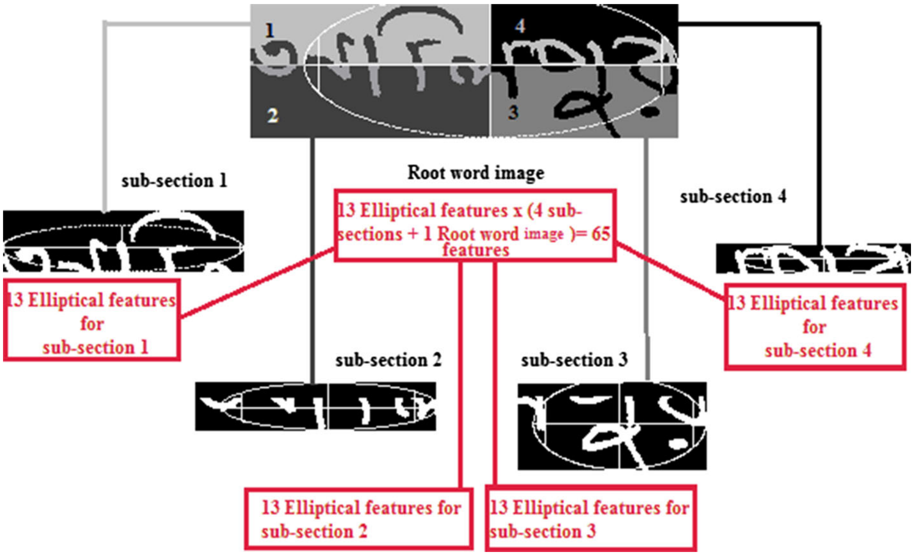
Therefore, from each word image, 13 (i.e., 4 from concentric ellipses and 9 from outermost Ellipse) global feature values are extracted. To get additional local information, a word image is further divided into four small sub-parts depending on the center of the ellipse and then from each sub-part, same set of feature values, as mentioned earlier, is computed. Therefore, in total 65 (i.e., 5 × 13) Elliptical features are computed from a particular word image (see Fig. 4). All these feature values are suitably normalized before feeding them to classifier.

3.3.2 Tetragonal features

In Euclidian plane geometry, a tetragon can be defined as a polygon with four sides and four corners. It can be concave or convex as well. In the present method, two convex tetragons are considered, the first one, whose two corner points (upper and lower) vary from one word type to another, is named as *Flexible Tetragon* (see Fig. 5a) and the second one, whose all four corner points are fixed or rigid, is named as *Rigid Tetragon* (see Fig. 5b). The main motive behind using *Flexible Tetragon* is to capture the geometrical dissimilarity among the words belonging to different classes, whereas *Rigid Tetragon* is used to estimate the variation present in the local pixel distribution of a word images, inside a fixed shaped region.

3.3.2.1 Flexible Tetragon Before extraction of the Tetragonal features, a Flexible Tetragon for a given word image is drawn hypothetically. Four vertices p_1, p_2, p_3 and p_4 of the hypothetical tetragon are estimated as follows. Point p_1 is the location of first foreground pixel from left on the upper boundary of the bounding box of a word image. Similarly p_3 corresponds to the first foreground pixel from left on the lower boundary, whereas, p_2 and p_4 are mid-points on the left and right sides of the bounding box,

Fig. 4 Illustration of local elliptical feature extraction process from a given word image


1
4
2
3
Root word image
sub-section 1
sub-section 4
13 Elliptical features x (4 sub-sections + 1 Root word image)= 65 features
13 Elliptical features for sub-section 1
13 Elliptical features for sub-section 4
sub-section 2
sub-section 3
13 Elliptical features for sub-section 2
13 Elliptical features for sub-section 3

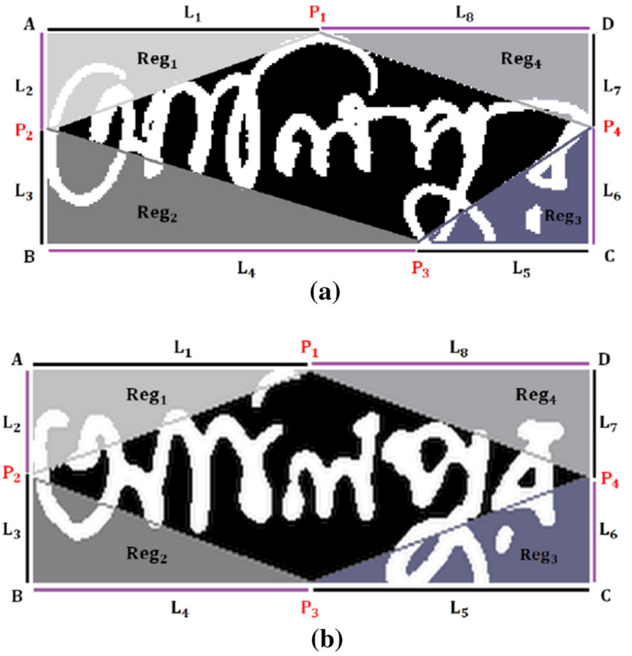


Fig. 5 Illustration of hypothetical tetragons, drawn on a given word image. Here P_i represents the *i*th corner of a hypothetical tetragon. L_is are the sides representing bases and heights of the right-angled triangles created by a hypothetical tetragon with the word boundary. Reg_i is the region, within the word boundary covered by the *i*th triangle. **a** Flexible Tetragon. **b** Rigid Tetragon

respectively. After the estimation of those four points, four hypothetical lines are constructed from p_1 to p_2, p_2 to p_3, p_3 to p_4 and p_4 to p_1. Figure 5a shows a hypothetical Flexible Tetragon on a given word image.

Feature computation using Flexible Tetragon After designing the hypothetical convex tetragon, two types of features are computed, *namely* (1) *Pixel distribution* and (2) *Geometric*.

Pixel distribution features A hypothetical tetragon divides the minimum bounding box of a word image into five sub-regions. This includes four sub-regions, created outside the tetragon named as Reg_1, Reg_2, Reg_3, Reg_4 and one sub-region inside the tetragon (see Fig. 5a). From this tetragon, following six Pixel distribution features are computed: (i) *Number of foreground pixels in* Reg_1, (ii) *Number of foreground pixels in* Reg_2, (iii) *Number of foreground pixels in* Reg_3,(iv) *Number of foreground pixels in* Reg_4, (v) *Number of foreground pixels inside the tetragon* and (vi) *Number of foreground pixels along the boundary of the tetragon.* All these feature values are normalized by the total number of foreground pixels in a given word image.

Geometric features Following two types of geometric features are considered for the present work: (i) *Area-based* and (ii) *Angle-based.*

(i) Area-based features

In this work, eight Area-based features are computed. Out of that four feature values are computed by estimating areas of the four outer sub-regions created by the tetragon and the minimum bounding box of a word image, which are basically right-angled triangles (see Fig. 5a) and rest of the feature values are computed by considering the ratio of the area of each outer region and the area of the inside region.

Area of a right-angled triangle can be computed by the following equation,

$$\text{Area} = \frac{1}{2} \times \text{base} \times \text{height} \tag{11}$$

In this work, area of a sub-region is computed as follows,

$$\text{Area}(\text{Reg}_i) = \frac{1}{2} \times L_{2\times i} \times L_{((2\times i)-1)} \tag{12}$$

where $i = 1, 2, 3, 4$. Here L_is are the sides representing bases and heights of the right-angled triangles. Area of the inside region is computed as follows,

$$\text{Area (inside region of a tetragon)} \\ = (H_{\text{BB}} \times W_{\text{BB}}) - \sum_{i=1}^{4} \text{Area}(\text{Reg}_i) \tag{13}$$

where H_{BB} and W_{BB} represent the height and width of a word bounding box.

First four Area-based feature values are normalized by the area of the minimum bounding box of the word.

(ii) Angle-based features

Here also, four different feature values are computed by estimating the angles created at the four corners of a hypothetical tetragon (see Fig. 5a). These angles are named as $\angle p1p2p3$, $\angle p2p3p4$, $\angle p3p4p1$ and $\angle p4p1p2$ which are computed as follows,

$$\angle p1p2p3 = 180^\circ - (\angle Ap2p1 + \angle Bp2p3) \tag{14}$$

$$\angle p2p3p4 = 180^\circ - (\angle p2p3B + \angle p4p3C) \tag{15}$$

$$\angle p3p4p1 = 180^\circ - (\angle p3p4C + \angle p1p4D) \tag{16}$$

$$\angle p4p1p2 = 180^\circ - (\angle p4p1D + \angle p2p1A) \tag{17}$$

To solve Eq. 14, first, the values of $\angle Ap2p1$ and $\angle Bp2p3$ are computed as follows,

$$\angle Ap2p1 = \arctan\left(\frac{L_1}{L_2}\right) \tag{18}$$

$$\angle Bp2p3 = \arctan\left(\frac{L_4}{L_3}\right) \tag{19}$$

In a similar way, Eqs. 15, 16 and 17 are computed. These feature values are normalized by dividing them with 360, as the sum of four interior angles of a tetragon is 360°.

Therefore, from each word image 18 (i.e., 6 + 12) features are extracted globally. To get local information, a given word image is further divided into four sub-regions depending on the center of gravity of the image and from each sub-region same set of features are computed. Thus, in total 90 (i.e., 18 × (4 local + 1 global)) features are estimated from a given word image. Figure 6 shows how the geometrical nature of the Flexible Tetragon gets changed over different classes of words.

As presence and position of the ascendants and descendants in a word play a very significant role in defining its shape, the above-mentioned geometrical properties of the hypothetical Flexible Tetragon reflect the shape information almost accurately. In addition to that they also vary over words from different word classes (see Fig. 6).

3.3.2.2 Rigid Tetragon For a given word image, a hypothetical Rigid Tetragon is computed by estimating its four corner points $p1$, $p2$, $p3$ and $p4$ as the midpoints of upper, leftmost, lower and rightmost edges of the word bounding box, respectively (see Fig. 5b).

Feature computation using Rigid Tetragon Rigid Tetragon helps in utilizing the information variation in terms of pixel distribution in different local regions of the word images belonging to different word classes. In this feature computation, four feature values are extracted from each outer region such as (1) *number of foreground pixels* (normalized by total number of foreground pixels in the word image), (2) *ratio between the number of foreground pixels and background pixels*, (3) *ratio between the number of foreground pixels present in the considered region and the number of foreground pixel present inside the rigid tetragon*, (4) *ratio between the number of foreground pixels and the area of the considered region*. From the inner region of the rigid tetragon, two more feature values such as (1) *number of foreground pixels* (normalized by total number of foreground pixels in the word image) and (2) *ratio between the number of foreground pixels and background pixels* are computed. Finally, *number of foreground pixels along boundary of the tetragon* (normalized by total number of foreground pixels in the word image) is also estimated.

Therefore, from a single image initially, 19 (i.e., (4 × 4) + 2 + 1) feature values are extracted globally. After that, to get local information, the word image is further divided into four sub-images depending on the center of gravity of the image and from each sub-image same set of Rigid Tetragonal features are computed. Thus, in total 95 (i.e., 19 × (4 local + 1 global)) feature values are estimated. That means from a given word image all total 185 (90 feature values using Flexible Tetragon + 95 feature values using Rigid Tetragon) Tetragonal features are computed. Figure 7 shows the Tetragonal features computed from a single sub-image.

Tetragonal features estimate the true shape structure of a given word image not only by considering the geometrical properties of a hypothetically drawn tetragon over it but

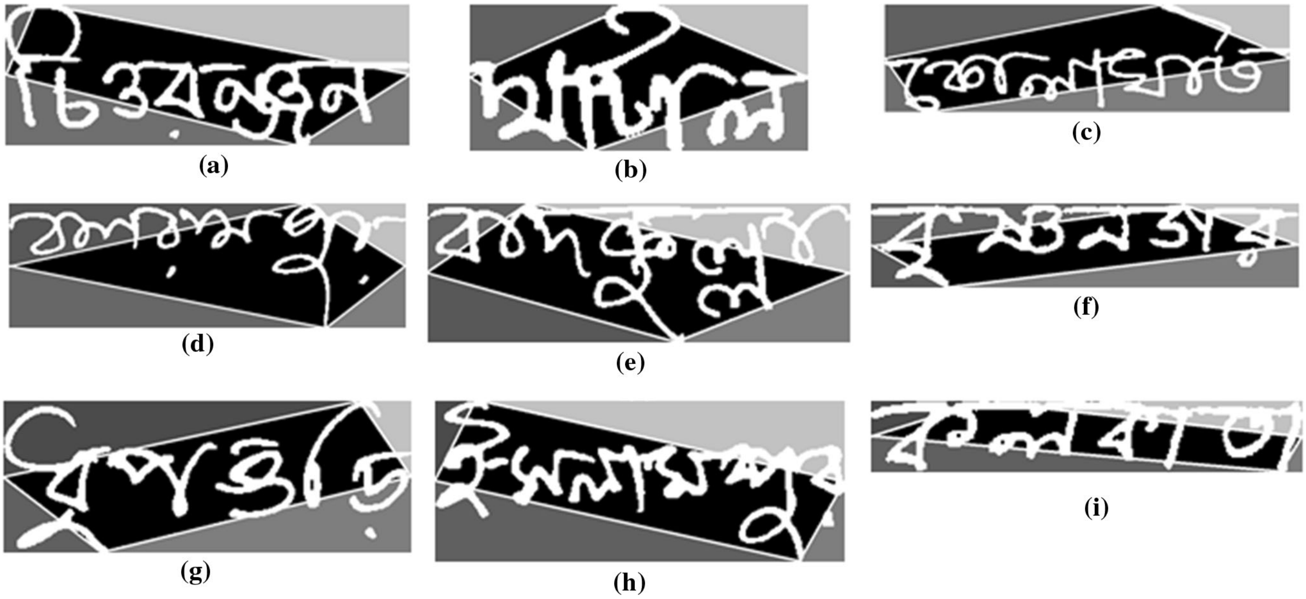


Fig. 6 Flexible Tetragons around the word images: **a**–**c** ascendant in left, middle and right respectively, **d**–**f** descendant in right, middle and left respectively, **g**, **h** both ascendant and descendant, **i** no ascendant and descendant

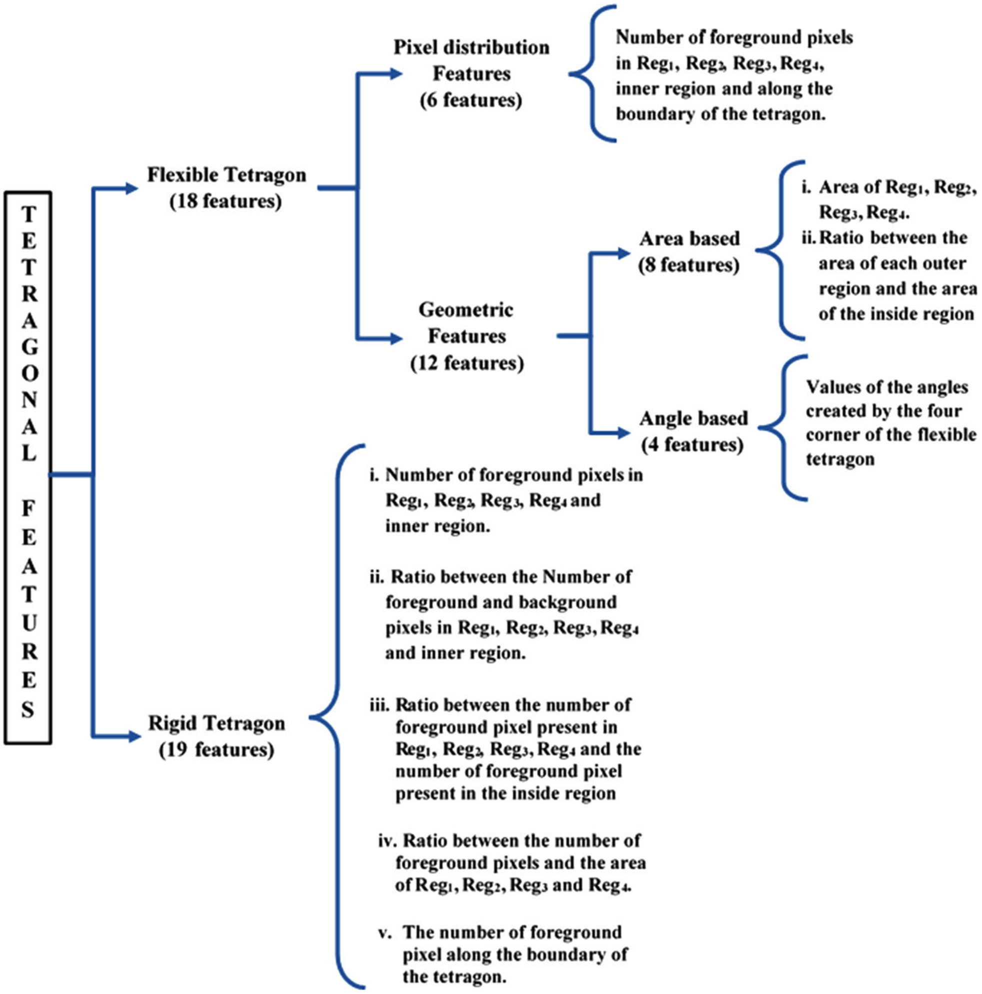

TETRAGONAL FEATURES
Flexible Tetragon (18 features)
Pixel distribution Features (6 features)
Number of foreground pixels in Reg₁, Reg₂, Reg₃, Reg₄, inner region and along the boundary of the tetragon.
Geometric Features (12 features)
Area based (8 features)
i. Area of Reg₁, Reg₂, Reg₃, Reg₄.
ii. Ratio between the area of each outer region and the area of the inside region
Angle based (4 features)
Values of the angles created by the four corner of the flexible tetragon
Rigid Tetragon (19 features)
i. Number of foreground pixels in Reg₁, Reg₂, Reg₃, Reg₄ and inner region.
ii. Ratio between the Number of foreground and background pixels in Reg₁, Reg₂, Reg₃, Reg₄ and inner region.
iii. Ratio between the number of foreground pixel present in Reg₁, Reg₂, Reg₃, Reg₄ and the number of foreground pixel present in the inside region
iv. Ratio between the number of foreground pixels and the area of Reg₁, Reg₂, Reg₃ and Reg₄.
v. The number of foreground pixel along the boundary of the tetragon.

Fig. 7 Hierarchical description of different types of tetragonal features used in our current work for word classification

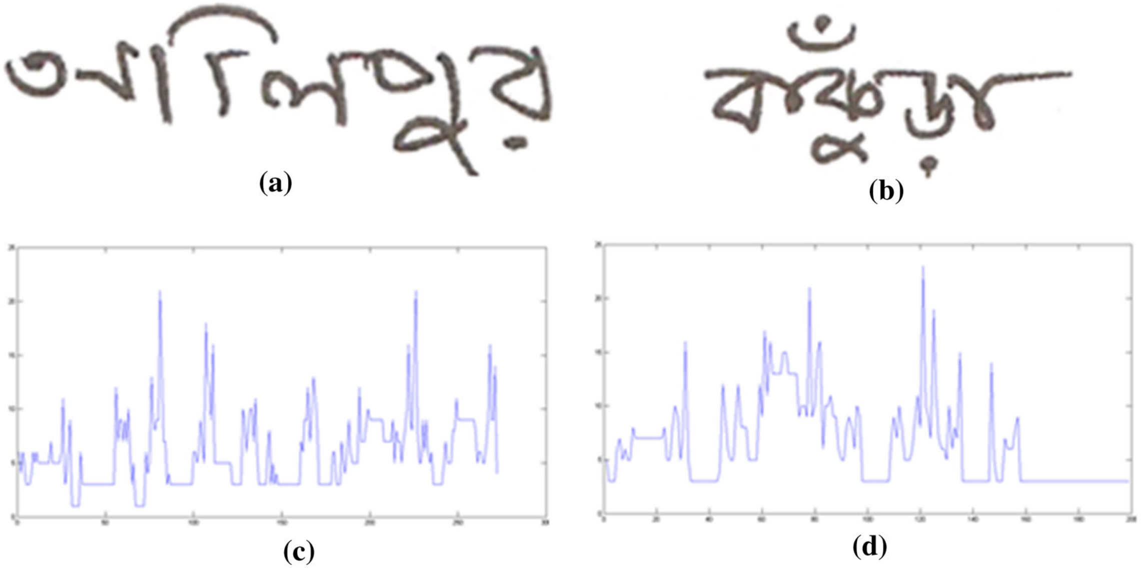


Fig. 8 Vertical pixel density histogram of (**a**, **b**) are shown in (**c**, **d**) respectively

Table 4 Threefold cross-validation result using SVM

Fold #	Number of training samples	Number of test samples	Accuracy in training data (in %)	Accuracy in test data (in %)
1	12,000	6000	98.26	76.18
2			98.05	**80.72**
3			**98.37**	73.18

Bold values indicate the best score

also by estimating the local pixel distribution at various regions created by these tetragons.

3.3.3 Vertical pixel density histogram-based feature

To compute these feature values, vertical pixel density histogram of a given word image is estimated by counting the number of data pixels present at each column of the word image. Figure 8 presents the vertical pixel density histogram of two handwritten Bangla words. After computing the vertical pixel density histogram, *the number of valleys* and *the number of peaks* are computed as two feature values. As in Bangla, words are generally written from left to right direction, vertical pixel density histogram of a word image will be more useful to capture the shape information in comparison with the horizontal pixel density histogram. This is why, in the present work, the vertical pixel density histogram of the word images is considered.

Table 5 Threefold cross-validation result using MLP

Fold #	Number of training samples	Number of test samples	Accuracy in training data (in %)	Accuracy in test data (in %)
1	12,000	6000	96.92	75.33
2			96.39	**79.87**
3			**96.92**	72.67

Bold values indicate the best score

Table 6 Fivefold cross-validation result using SVM

Fold #	Number of training samples	Number of test samples	Accuracy in training data (in %)	Accuracy in test data (in %)
1	14,400	3600	97.69	79.44
2			97.79	77.33
3			97.73	**83.64**
4			**98.08**	79.3
5			97.86	77.19

Bold values indicate the best score

Table 7 Fivefold cross-validation result using MLP

Fold #	Number of training samples	Number of test samples	Accuracy in training data (in %)	Accuracy in test data (in %)
1	14,400	3600	96.23	77.89
2			96.60	75.50
3			96.32	**81.72**
4			**96.65**	77.75
5			96.58	73.75

Bold values indicate the best score

4 Experimental evaluation

Proposed HWR method is implemented on a machine with Intel$^{®}$Core(TM)i3-5010U@2.10 GHz as the CPU with 4 GB RAM. The current method is evaluated with a dataset comprising 120 handwritten city names of West Bengal (a state of India). Here the classification is carried out separately using two well-known classifiers MLP and SVM. For that purpose, a machine learning tool called WEKA is used [52]. All the steps of the current experiments along with error case analysis are described in Sect. 4.1 and finally in Sect. 4.2, comparison of the current work with *state-of-the-art* feature descriptors as well as methods is presented.

4.1 Experimental results and error analysis

The present experiment is carried out by following fivefold and threefold cross-validation schemes using both MLP and SVM classifiers. MLP classifier in this experiment has 185 neurons in its hidden layer (only one hidden layer is considered here) and trained with learning rate (η) = 0.3 and momentum term (α) = 0.2 for each fold using 1000 iterations. For SVM classifier, a polynomial kernel is used. All these parameter values are chosen experimentally. In fivefold cross-validation, 14,400 images are used for training the model and rest 3600 images are used for testing, whereas, 12,000 and 6000 word images are used for training and testing, respectively, for each fold in threefold cross-validation. Detailed results using MLP and SVM in threefold cross-validation are given in Tables 4 and 5, respectively. Outcomes of the fivefold cross-validation scheme using MLP and SVM are given in Tables 6 and 7, respectively.

4.1.1 System performance with increasing number of word classes

In the present experiment, performance of the proposed system is also observed when number of word classes increased gradually. For this purpose, entire experiment is carried out using threefold and fivefold cross-validation

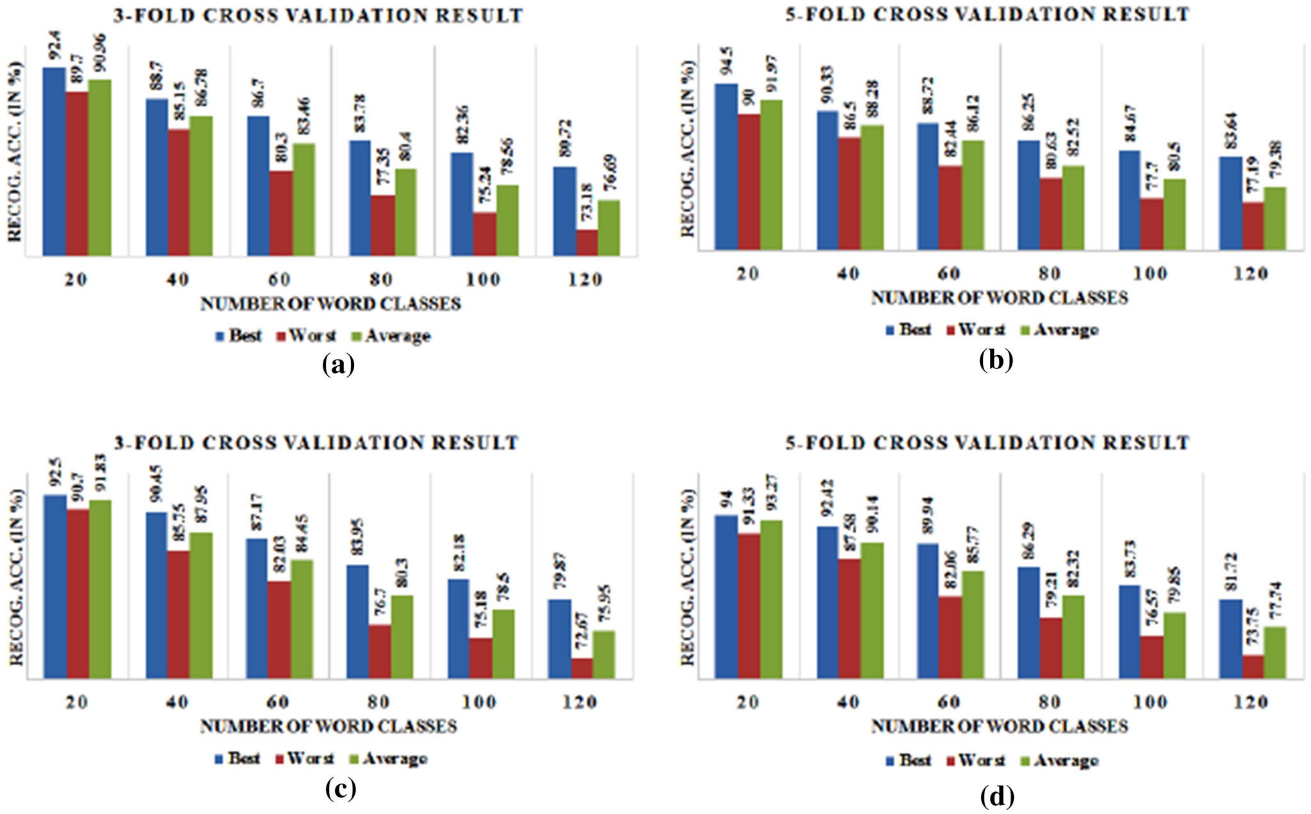


Fig. 9 Performance of the system for different number of word classes using SVM (**a**, **b**) and MLP (**c**, **d**)

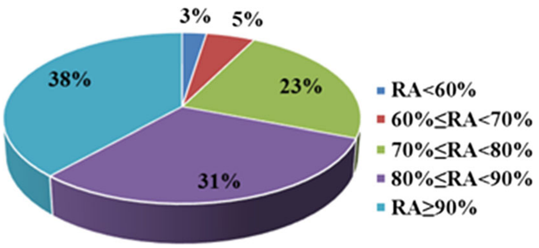


Fig. 10 Number of word classes (in percentage) having recognition accuracy (RA) within a given range

schemes with MLP and SVM classifiers following the previously specified setup for 20, 40, 60, 80, 100 and 120 word classes. Results of these experiments are shown in Fig. 9.

From the above experiments it is observed that in entire dataset (of 120 classes), SVM performs well in comparison with MLP. Thus rest of the discussions are presented based on the results obtained using SVM with fivefold cross-validation.

4.1.2 Error case analysis

The proposed system has achieved impressive results for most of the word classes except a few. A detailed analysis of the result is given in Figs. 10 and 11.

Our analysis revels that the probable reasons behind these misclassifications could be (1) *spelling disparity*, (2) *complex shape* and (3) *similar shaped words from different classes.*

1. Spelling disparity

The disparity in spellings within the same word class causes some alarming shape varieties. As the present work estimates some shape-based features, this variation of shapes due to wrong spellings surely misleads the classifier to identify the true class of the word samples. Figure 12 illustrates such spelling disparities found in our database.

2. Complex shape

Large skew of the word images and complex shape angle can be another two reasons for such misclassifications observed in the current work (see Fig. 13). Since during the feature extraction, the main focus is given on various local regions of any word image, the expected local feature values of the samples belonging to the same word class, as shown in Fig. 13, would differ significantly due to the above reasons.

3. Similar shaped words from different classes

Along with various geometrical features, some pixel count features are also estimated from the word images in

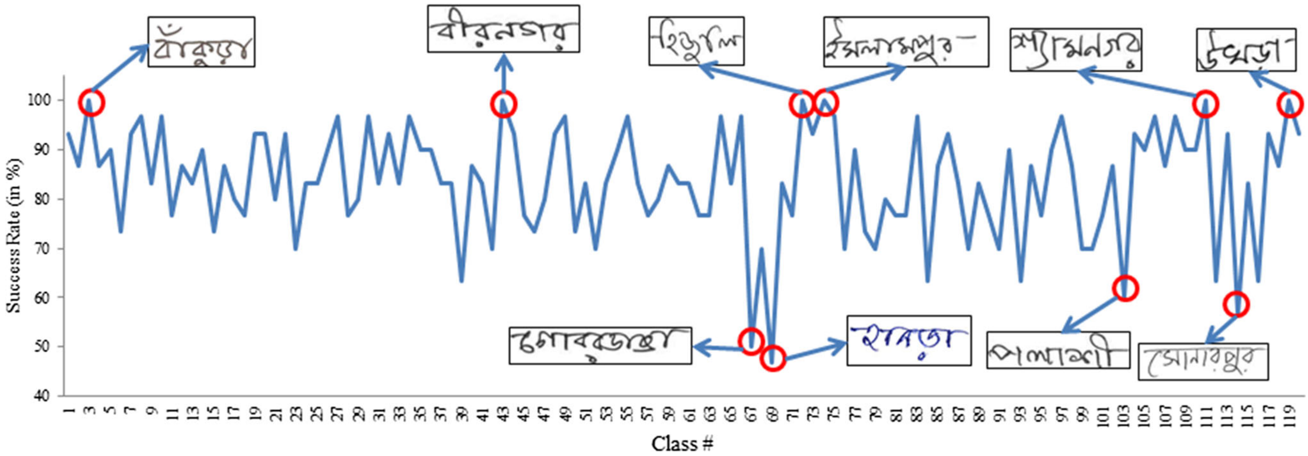


Fig. 11 Class-wise recognition performance is shown with sample word images from classes having top-5 and worst-4 recognition accuracy

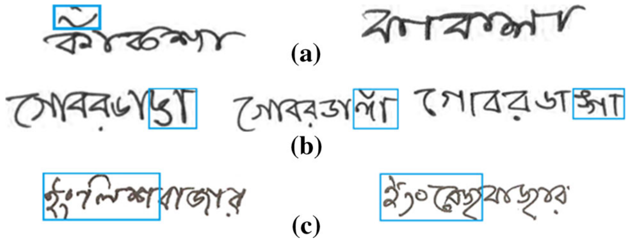


Fig. 12 Illustration of spelling disparities **a**, **b** show disparity caused by miss-spelling, **c** is basically alternative spelling

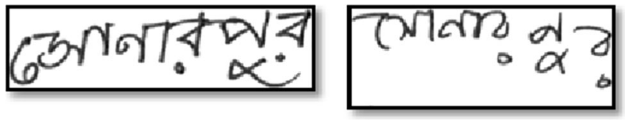


Fig. 13 Illustration of words with skew variation

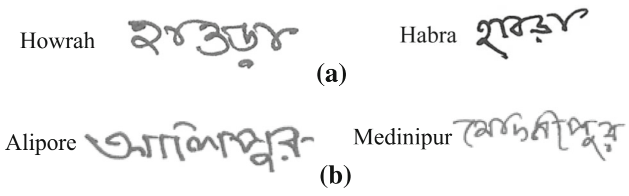


Fig. 14 Some of the word classes having high shape similarity

this work. Thus, in few cases, it is observed that word images with high shape resemblance end up with almost same feature values, which also results in the misclassification. Figure 14 shows some most confusing word classes.

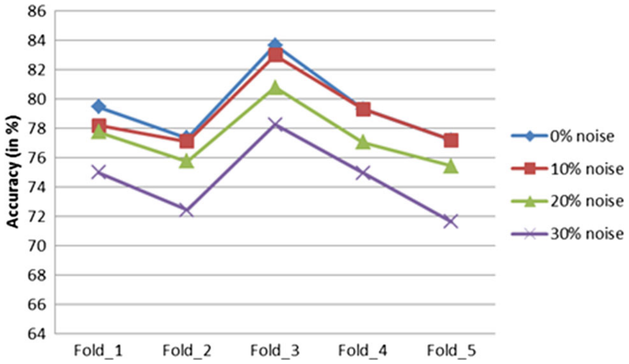


Fig. 16 Fivefold cross-validation results of the proposed system using SVM for data with different level of embedded Gaussian noise

4.1.3 Experiment with noisy data

Performance of the proposed system is also observed while dealing with noisy data. For that purpose, we have synthetically created noisy data by adding Gaussian noise with different levels to the word images (see Fig. 15). Due to the addition of 10%, 20% and 30% noise, a reduction of 0.43%, 2.04% and 4.93% in average recognition rate, respectively, is observed in each fold (see Fig. 16).

4.2 Performance comparison

Recognition performances of each category of elementary feature descriptors, described in Sect. 3.3, are evaluated and given in Table 8. Next, the performance of the

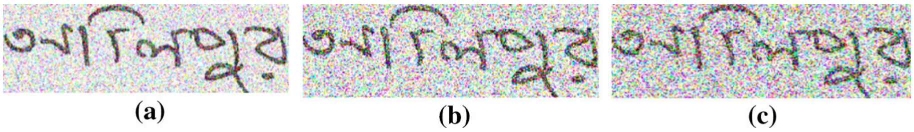


Fig. 15 Sample of handwritten Bangla word image with 10%, 20% and 30% embedded Gaussian noise, respectively

Table 8 Performance comparison of different feature descriptors used in this work on CMATERdb2.1.2

Feature descriptors	Feature dimension	Accuracy (in %)		
		Best	Worst	Avg
Elliptical	65	59.97	53.56	56.5
Tetragonal	185	76.42	70	72.74
Elliptical and tetragonal	250 (65 + 185)	82.61	74.61	77.96
Elliptical, tetragonal and vertical pixel density histogram-based features (Proposed)	**252 (250 + 2)**	**83.64**	**77.19**	**79.38**

Bold values indicate the best score

Table 9 Performance comparison of proposed feature descriptor with the *state-of-the-art* feature descriptors on CMATERdb2.1.2

Feature descriptors	Feature dimension	Avg. feature computation time (in s)	Accuracy (in %)		
			Best	Worst	Avg
LGH [34]	786	0.7199	83.4	76.10	78.55
PHOG [35]	672	0.1721	78.42	73.47	75.18
G-PHOG [53]	720	0.1759	72.53	67.14	78.93
Proposed descriptor	**252**	**0.1158**	**83.64**	**77.19**	**79.38**

Bold values indicate the best score

Table 10 Performance comparison of proposed method with some of the *state-of-the-art* holistic HWR methods on CMATERdb2.1.2

Method with year of publication	Maximum accuracy	Minimum accuracy	Average accuracy	Standard deviation
Bhowmik et al. [40], 2014	58.85	52.82	55.97	2.9
Dasgupta et al. [26], 2016	76.69	68.56	72.71	2.81
Malakar et al. [31], 2017	73.03	67.42	69.72	2.099
Barua et al. [42], 2017	83.06	75.75	79.06	2.72
Proposed method	**83.64**	**77.19**	**79.38**	**2.33**

Bold values indicate the best score

combined feature descriptor is compared with some *state-of-the-art* feature descriptors used for Bangla word recognition [53]. This comparison is performed in terms of recognition accuracy, feature dimension and feature computation time (see Table 9). Feature descriptors considered for comparison include, viz., Local Gradient of Histogram (LGH), Pyramid Histogram of Oriented Gradient (PHOG) and combination of GABOR and PHOG called G_PHOG [53]. Although the recognition accuracy achieved by LGH is very close to the proposed technique but if we consider the other parameters, it is observed that the proposed technique outperforms the LGH and others.

Finally, the proposed method is compared with some recently published holistic word recognition methods [26, 31, 36, 42]. Work reported in [36, 42] has dealt with Bangla HWR, whereas the remaining two methods have developed for the recognition of handwritten English [26] and Hindi [31] words. For the comparison, our fivefold cross-validations result is considered. The best, worst, average case recognition accuracies along with feature dimension and the classifier used by these techniques are summarized in Table 10. It also includes deviation from average recognition rate. From the table, it is clear that the present technique outperforms the said methods. Besides the results, we would also like to mention here that, in the literature no holistic Bangla word recognition work has been reported, where such a large number of word classes are considered.

5 Conclusion

In the present work, a holistic HWR scheme is developed for the recognition of handwritten Bangla words. For that purpose, a shape-based feature descriptor which is a combination of Elliptical, Tetragonal and Vertical pixel density histogram-based features is designed. Recognition process is carried out using two well-known classifiers, viz., MLP and SVM. But the proposed method performs comparably better with SVM than MLP. In holistic word recognition approach, as a given word image is considered as a single and indivisible unit, shape dissimilarity of the words belonging to different classes can be very effective during recognition. This very fact motivates us to design the proposed feature descriptor. For the evaluation purpose, a database of 18,000 Bangla handwritten word images belonging to 120 different word classes is also prepared and it is also made freely available to the research community.

Although the proposed method is currently used for recognition of handwritten Bangla words but as it emphasizes on shape-level information, it may equally be useful for recognition of handwritten words written in other scripts. In addition to that, no skew or slant correction has been undertaken at the preprocessing stage (which is very common in handwritten word recognition) but still it has achieved reasonable accuracy. Thus in future, inclusion of a suitable skew and slant correction module can make it more effective.

Compliance with ethical standards

Conflict of interest We declare that we do not have any conflict of interest.

References

1. Chacko BP, Krishnan VRV, Raju G, Anto PB (2012) Handwritten character recognition using wavelet energy and extreme learning machine. Int J Mach Learn Cybern 3(2):149–161
2. Prasad JR, Kulkarni U (2015) Gujrati character recognition using weighted k-NN and mean χ^2 distance measure. Int J Mach Learn Cybern 6(1):69–82
3. Pal U, Roy K, Kimura F (2009) A lexicon-driven handwritten city-name ecognition scheme for Indian postal automation. IEICE Trans Inf Syst 92(5):1146–1158
4. Pal U, Roy RK, Kimura F (2012) Multi-lingual city name recognition for Indian postal automation. In: 2012 international conference on frontiers in handwriting recognition (ICFHR), pp 169–173
5. Morita M, El Yacoubi A, Sabourin R, Bortolozzi F, Suen CY (2001) Handwritten month word recognition on Brazilian bank cheques. In: Sixth international conference on document analysis and recognition. Proceedings, pp 972–976
6. Bunke H, Bengio S, Vinciarelli A (2004) Offline recognition of unconstrained handwritten texts using HMMs and statistical language models. IEEE Trans Pattern Anal Mach Intell 26 (6):709–720
7. Madhvanath S, Govindaraju V, Ramanaprasad V, Lee D-S, Srihari SN (1995) Reading handwritten US census forms. In: Proceedings of the third international conference on document analysis and recognition, vol 1, pp 82–85
8. Srihari SN, Shin YC, Ramanaprasad V, Lee D-S (1995) Name and address block reader system for tax form processing. In: Proceedings of the third international conference on document analysis and recognition, vol 1, pp 5–10
9. Prasad JR, Kulkarni U (2015) Gujarati character recognition using adaptive neuro fuzzy classifier with fuzzy hedges. Int J Mach Learn Cybern 6(5):763–775
10. Broumandnia A, Shanbehzadeh J, Varnoosfaderani MR (2008) Persian/arabic handwritten word recognition using M-band packet wavelet transform. Image Vis Comput 26(6):829–842
11. El Qacimy B, Kerroum MA, Hammouch A (2015) Word-based Arabic handwritten recognition using SVM classifier with a reject option. In: 2015 15th international conference on intelligent systems design and applications (ISDA), pp 64–68
12. Dehghan M, Faez K, Ahmadi M, Shridhar M (2001) Handwritten Farsi (Arabic) word recognition: a holistic approach using discrete HMM. Pattern Recognit 34(5):1057–1065
13. Liu C-L, Koga M, Fujisawa H (2002) Lexicon-driven segmentation and recognition of handwritten character strings for Japanese address reading. IEEE Trans Pattern Anal Mach Intell 24 (11):1425–1437
14. Su T (2013) Chinese handwriting recognition: an algorithmic perspective. Springer, Berlin
15. Srihari SN, Yang X, Ball GR (2007) Offline Chinese handwriting recognition: an assessment of current technology. Front Comput Sci China 1(2):137–155
16. Koerich AL, Sabourin R, Suen CY (2005) Recognition and verification of unconstrained handwritten words. IEEE Trans Pattern Anal Mach Intell 27(10):1509–1522
17. Bunke H (2003) Recognition of cursive Roman handwriting: past, present and future. In: Seventh international conference on document analysis and recognition. Proceedings, pp 448–459
18. Bozinovic RM, Srihari SN (1989) Off-line cursive script word recognition. IEEE Trans Pattern Anal Mach Intell 11(1):68–83
19. "Bengali language". https://en.wikipedia.org/wiki/Bengali_language. Accessed 27 Dec 2017
20. Das N, Sarkar R, Basu S, Saha PK, Kundu M, Nasipuri M (2015) Handwritten Bangla character recognition using a soft computing paradigm embedded in two pass approach. Pattern Recognit 48 (6):2054–2071
21. Rahman MM, Akhand MAH, Islam S, Shill PC, Rahman MMH (2015) Bangla handwritten character recognition using convolutional neural network. Int J Image Graph Signal Process 7(8):42
22. Das N, Basu S, Saha PK, Sarkar R, Kundu M, Nasipuri M (2015) A GA based approach for selection of local features for recognition of handwritten Bangla numerals. arXiv Prepr. arXiv:1501.05495
23. Plamondon R, Srihari SN (2000) Online and off-line handwriting recognition: a comprehensive survey. IEEE Trans Pattern Anal Mach Intell 22(1):63–84
24. Tappert CC, Suen CY, Wakahara T (1990) The state of the art in online handwriting recognition. IEEE Trans Pattern Anal Mach Intell 12(8):787–808
25. Ruiz-Pinales J, Jaime-Rivas R, Castro-Bleda MJ (2007) Holistic cursive word recognition based on perceptual features. Pattern Recognit Lett 28(13):1600–1609
26. Dasgupta J, Bhattacharya K, Chanda B (2016) A holistic approach for Off-line handwritten cursive word recognition using directional feature based on Arnold transform. Pattern Recognit Lett 79:73–79
27. Koerich AL, Sabourin R, Suen CY (2003) Large vocabulary offline handwriting recognition: a survey. Pattern Anal Appl 6 (2):97–121
28. Plötz T, Fink GA (2009) Markov models for offline handwriting recognition: a survey. Int J Doc Anal Recognit 12(4):269–298
29. Madhvanath S, Govindaraju V (2001) The role of holistic paradigms in handwritten word recognition. IEEE Trans Pattern Anal Mach Intell 23(2):149–164

30. Madhvanath S, Kleinberg E, Govindaraju V (1999) Holistic verification of handwritten phrases. IEEE Trans Pattern Anal Mach Intell 21(12):1344–1356
31. Malakar S, Sharma P, Singh PK, Das M, Sarkar R, Nasipuri M (2017) A holistic approach for handwritten hindi word recognition. Int J Comput Vi. Image Process 7(1):59–78
32. Tamen Z, Drias H, Boughaci D (2017) An efficient multiple classifier system for Arabic handwritten words recognition. Pattern Recognit Lett 93:123–132
33. Pechwitz M, Maddouri SS, Märgner V, Ellouze N, Amiri H (2002) IFN/ENIT-database of handwritten Arabic words. Proc CIFED 2:127–136
34. Roy PP, Dey P, Roy S, Pal U, Kimura F (2014) A novel approach of Bangla handwritten text recognition using HMM. In: 2014 14th international conference on frontiers in handwriting recognition (ICFHR), pp 661–666
35. Roy PP, Bhunia AK, Das A, Dey P, Pal U (2016) HMM-based Indic handwritten word recognition using zone segmentation. Pattern Recognit 60:1057–1075
36. Vajda S, Roy K, Pal U, Chaudhuri BB, Belaid A (2009) Automation of Indian postal documents written in Bangla and English. Int J Pattern Recognit Artif Intell 23(8):1599–1632
37. Bhowmik TK, Roy U, Parui SK (2012) Lexicon reduction technique for Bangla handwritten word recognition. In: 2012 10th IAPR international workshop on document analysis systems (DAS), pp 195–199
38. Bhowmik TK, Parui SK, Roy U (2008) Discriminative HMM training with GA for handwritten word recognition. In: ICPR 2008. 19th international conference on pattern recognition, pp 1–4
39. Bhowmik S, Roushan MG, Sarkar R, Nasipuri M, Polley S, Malakar S (2014) Handwritten Bangla word recognition using HOG descriptor. In: Proceedings—4th international conference on emerging applications of information technology, EAIT
40. Bhowmik S, Malakar S, Sarkar R, Nasipuri M (2014) Handwritten Bangla word recognition using elliptical features. In: 2014 international conference on computational intelligence and communication networks (CICN), pp 257–261
41. Bhowmik S, Polley S, Roushan MG, Malakar S, Sarkar R, Nasipuri M (2015) A holistic word recognition technique for handwritten Bangla words. Int J Appl Pattern Recognit 2(2):142–159
42. Barua S, Malakar S, Bhowmik S, Sarkar R, Nasipuri M (2017) Bangla handwritten city name recognition using gradient-based feature, vol 515
43. Ghosh M, Malakar S, Bhowmik S, Sarkar R, Nasipuri M (2017) Memetic algorithm based feature selection for handwritten city name recognition, vol 776
44. Ban JC (2015) Neural network equations and symbolic dynamics. Int J Mach Learn Cybern 6(4):567–579
45. Li Z, Zhou M, Lin H, Pu H (2014) A two stages sparse SVM training. Int J Mach Learn Cybern 5(3):425–434
46. Liu CL, Koga M, Fujisawa H (2005) Gabor feature extraction for character recognition: comparison with gradient feature. In: Eighth international conference on document analysis and recognition (ICDAR'05), pp 121–125
47. Sarkar R, Das N, Basu S, Kundu M, Nasipuri M, Basu DK (2012) CMATERdb1: a database of unconstrained handwritten Bangla and Bangla-English mixed script document image. Int J Doc Anal Recognit 15(1):71–83
48. "CMATERdb2.1.2". https://drive.google.com/file/d/0B8rZngAQdufXemZmYlI2M2xwdXc/view?usp=sharing
49. Otsu N (1975) A threshold selection method from gray-level histograms. Automatica 11(285–296):23–27
50. Soille P (2005) Erosion and dilation. In: Morphological image analysis. Springer, pp 63–103
51. Yang M, Kpalma K, Ronsin J (2008) A survey of shape feature extraction techniques. IN-TECH
52. Smith TC, Frank E (2016) Introducing machine learning concepts with WEKA. Stat Genomics Methods Protoc 1418:353–378
53. Bhunia AK, Das A, Roy PP, Pal U (2015) A comparative study of features for handwritten Bangla text recognition. In: 2015 13th international conference on document analysis and recognition (ICDAR), pp 636–640