Genetic Algorithm Based Global and Local Feature Selection Approach for Handwritten Numeral Recognition

Sagnik Pal Chowdhury, Ritwika Majumdar, Sandeep Kumar, Pawan Kumar Singh, and Ram Sarkar

1 Introduction

Handwritten numeral recognition has been a widely acknowledged research field since the 1980s. In the domain of pattern recognition and image processing, this problem is considered amongst the major benchmark research problems. Development in this area is integral to the enhancement of man-machine interface. The recognition of handwritten numerals greatly differs from that of printed numerals. Printed numerals are uniform in size, shape and position in a given font. Whereas the same cannot be assured for its handwritten counterparts as all the said parameters of the writing vary from one entity to another depending on the writing styles, background and educational qualifications of the individuals. Hence, handwriting is, in general, non-uniform in nature. As a result, detection of handwritten numerals is amongst the most challenging yet popular research areas which have attracted researchers since long.

Handwriting numeral recognition can be broadly divided into two categories—Offline and Online. Offline numeral recognition utilizes a raster image taken from different digital input sources. Binarization of the image is done through appropriate threshold techniques on the basis of the grayscale patterns, in such a way that the image pixels can be either '1' (foreground) or '0' (background). In online recognition,

S. P. Chowdhury · R. Majumdar · S. Kumar
Department of Computer Science and Technology, Indian Institute of Engineering Science and Technology, Shibpur, Howrah 711103, West Bengal, India

P. K. Singh (✉)
Department of Information Technology, Jadavpur University, Jadavpur University Second Campus, Plot No. 8, Salt Lake Bypass, LB Block, Sector III, Salt Lake City, Kolkata 700106, West Bengal, India

R. Sarkar
Department of Computer Science and Engineering, Jadavpur University, 188, Raja S.C. Mullick Road, Kolkata 700032, West Bengal, India

D. Oliva et al. (eds.), *Metaheuristics in Machine Learning: Theory and Applications*, Studies in Computational Intelligence 967,
https://doi.org/10.1007/978-3-030-70542-8_30

the system is fed the current data and the recognition (of an input numeral) can be carried out simultaneously, if necessary. A stylus, in contact with a pressure sensitive input device, sends a string of (x, y) coordinate pairs. The recognition of offline handwritten numerals is trickier than the online case because of the noise introduced in the acquisition of image. Another reason is when an image of a handwritten document is acquired then information like stroke sequence and pen positions are lost. The sequential and dynamic information are extracted from these pen movements which, in turn, act as significant features for solving online numeral recognition problem. On the contrary, the absence of such information makes the task of offline recognition a challenging one.

In our study, we deal with the offline handwritten numeral methods, where a device optically scans the writing which is then saved as an image. The recognition is dedicated to automation in processing bank cheques, sorting postal mails and packages, through the use of pin codes, evaluation of examination tests having multiple choice questions and so on. Due to this vast difference in the handwriting styles, there arise some key challenges in creating correct and precise recognition systems. Hence, feature extraction develops into an arduous task, thus necessitating the need for researchers to find methods to recognize handwritten digits with utmost accuracy. As a result, this problem is still an existing research area discovering methods to increase accuracy in recognition [1, 2].

Previous works on recognition of handwritten digits dealt with the *Roman* [3] script, mainly related to *English*, some European languages and Asian languages like *Chinese* [4]. In India, languages like *Devanagari*, *Bangla*, *Odia* and *Tamil* have started to gain traction [5].

Hindi and *Bangla* are the two most popularly spoken languages in India. *Hindi* ranks third in the world in terms of the number of speakers. *Bangla*, the second most popular language in India, is also the national language of Bangladesh. It also ranks sixth in the world having about 207 million speakers [6]. In India, *Bangla* is one of the commonly spoken languages in the eastern states. Despite all the popularly, there has been very few research works, as observed in literature, carried out in the field of handwritten numeral recognition of these two languages. In view of the said facts, the present work concentrates on the recognition of *Bangla* and *Hindi* handwritten numerals. Due to our colonial past as well as the diversity of languages/scripts in India, *English* is used as a binding language. After *Hindi*, *English* ranks second as the most spoken language in the Indian sub-continent. Having about 341 million speakers, it is also the fourth largest spoken language in the world. Most of the official works are performed using both regional language and *English* language. *English* is also taught in most schools, colleges and universities. So, in the present work, we have also considered *English* numerals (written in *Roman* script) too. Handwritten samples of *Bangla*, *Hindi* and *Roman* numerals considered in the present work are shown in Fig. 1.

Generally, the steps in handwritten numeral recognition system are preprocessing, followed by feature extraction and classification. Document pages containing handwritten numerals are first scanned. From there, each individual numeral is extracted. After that these extracted digits are filtered (to remove any noisy

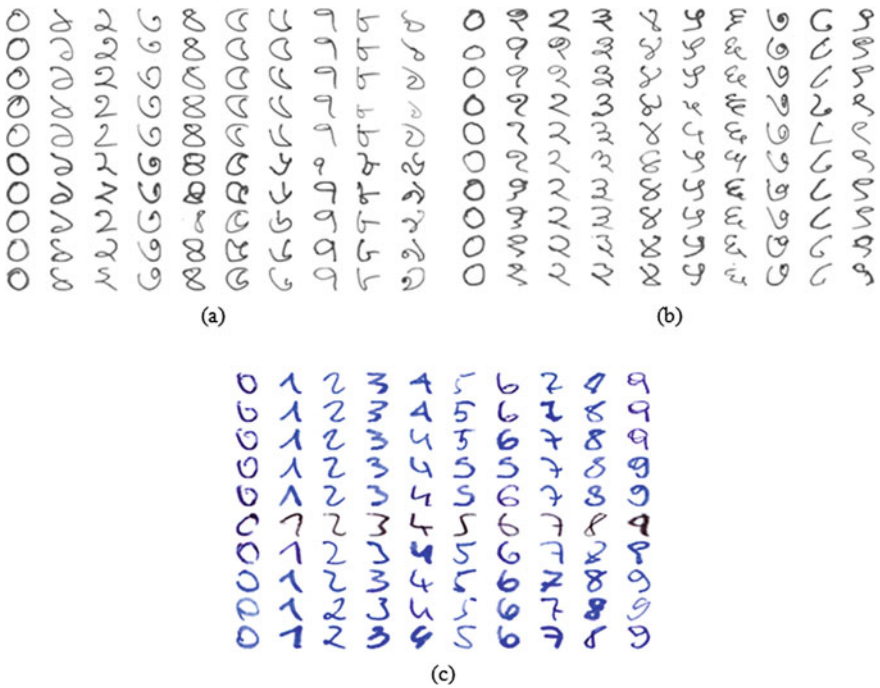

Fig. 1 Examples of numeral sample written in: **a** *Bangla*, **b** *Devanagari*, and **c** *Roman* scripts

pixels) and resized to serve as the input data for our experiment. Next, feature extraction is done via the Histogram of Gradients (HOG) technique and local distance-based features. At first, features based on HOG are extracted from the handwritten numerals and the classification is done using Multi-Layer Perceptron (MLP) classifier. The confusion matrix generated, is analyzed to investigate the maximum number of misclassifications found among the numerals. Based on this matrix, the numerals showing the maximum number of misclassifications among each other are grouped together. The feature extraction is once again performed using the combination of HOG and local distance-based features. The running time and cost of a recognition method is increased due to each feature that is used for classification. Hence, we strive to create and execute a system which has minimized feature sets.

On the other hand, we also have to include ample feature sets to obtain high recognition rates under challenging conditions. This has led to the development of a variety of optimization techniques for finding an "optimal" subset of features from a larger set of possible features. Machine learning techniques, which generate useful classification procedures, can be significantly improved if we can effectively find "optimal" feature subsets from the point of view of size and performance. In our research, we have used an adaptive feature selection strategy using GA which can significantly reduce the number of features required for *intra-numeral* classification of handwritten numerals formed within the groups. The MLP classifier is again used

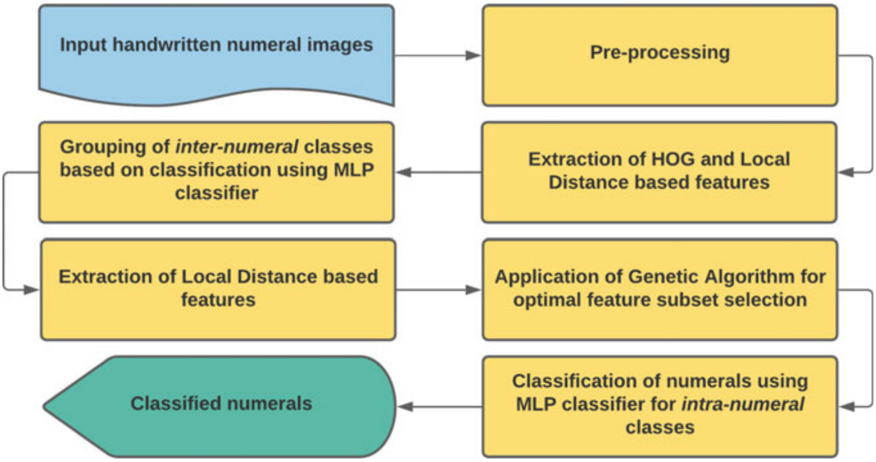

Fig. 2 Flowchart of the proposed handwritten numeral recognition system

for the classification purpose. On obtaining the optimal feature sets using GA for each *intra-numeral* group, the numerals are separately classified to their respective numeral classes. The schematic diagram representing the key steps of the proposed approach is shown in Fig. 2. This proposed system is applied for three most popular scripts used in Indian subcontinent *viz*., *Devanagari*, *Bangla*, and *Roman*. The key advantage of this approach includes the ability to accommodate two important criteria such as number of features and accuracy of classifiers in order to adequately represent the handwritten numeral recognition problem.

The rest of the paper is organized as follows: Sect. 2 describes a brief study of previous methods related to handwritten numeral recognition methods of *Bangla*, *Devanagari* and *Roman* scripts. Section 3 explains the collection process of input numeral databases for evaluating the proposed work. Section 4 presents the proposed methodology for recognition of handwritten numerals whereas experimental results are reported in Sect. 5. The final summary of the work and some possible future scopes are mentioned in Sect. 6.

2 Literature Survey

Digit recognition is a subfield of character recognition and a subject of substantial importance since the early years of research in the field of handwriting recognition. There are a lot of methodologies to solve this problem, as proposed in the literature. Most of the knowledge provided by these investigations may also be applied to character and word recognition. In this section, we review some of the research articles related to the handwritten recognition of *Bangla*, *Devanagari* and *Roman* scripts.

The method proposed by Basu et al. [7] divided the digit image into 9 overlapping sub-images of fixed size. Then, from each of these sub-images, they computed locally, the longest run-based feature. A MLP based classifier was used to test this approach on *Bangla* numeral dataset and achieved a recognition accuracy of 96.65%. To recognize handwritten *Bangla* numerals, Wen et al. [8] incorporated a kernel and Bayesian Discriminant based technique, whereas Nasir et al. [9] recommended a hybrid system for the same to be used for automated postal system. This carried out feature extraction using *k*-means clustering, Bayes theorem and maximum of posteriori and finally using Support Vector Machine (SVM) classifier. Surinta et al. [10] proposed the usage of a feature-set, for example, the outline of the handwritten image computed using 8-directional codes, distance measured between black pixels and hotspots, and the intensity of pixel points of small blocks. Now, these features were fed to a nonlinear SVM classifier individually, and the final conclusion was obtained on the basis of majority voting. For handwritten *Bangla* numeral recognition, a framework is presented in the study of Khan et al. [11], using Sparse Representation Classifier. To classify the *Bangla* digits, this classifier is applied on the image zone density, which is an image domain statistical feature extracted from the character image. This unique method for *Bangla* Optical Character Recognition (OCR) displays an outstanding accuracy of 94% on the off-line handwritten *Bangla* digit database *CMATERdb* 3.1.1. Akhand et al. in [12] investigated a Convolutional Neural Network (CNN) based handwritten *Bangla* numeral recognition system. The proposed system uses moderate preprocessing technique on the images of handwritten numbers by generating patterns from them, after which CNN is used to classify individual numerals. It does not utilize any feature extraction methods which is generally seen in other related works. Recently, Singh et al. in [13] reported a comprehensive survey especially for *Bangla* handwritten numeral recognition.

The first research report on the recognition of *Devanagari* numerals was issued in 1977 [14]. However, for the next 30 years, less number of significant works has been reported. Some researchers have proposed various methods in recent times on handwritten *Devanagari* characters. Bhattacharaya et al. [15] put forward a classification approach based on MLP neural network for *Devanagari* numerals with an accuracy of 91.28%. They employed multi-resolution features based on wavelet transforms. Hanmndlu and Murthy [16] proposed a fuzzy model-based recognition for handwritten *Devanagari* numbers, where normalized distance was used as a feature for individual boxes and this resulted in 92.67% accuracy. For isolated handwritten *Devanagari* numerals, Singh et al. in [17] suggested an automatic recognition system. Feature extraction methods are based on the topological and geometrical properties of the character and the structure of the character image. Based on different levels of granularity, they have used the recursive subdivision of the handwritten image to extract the features. At each level, there are vertical and horizontal lines which split the handwritten image into 4 quadrants of sub-images, consisting of nearly the same quantity of foreground pixels. The intersection of these lines denotes a point, on the basis which, features are extracted. This image division method results in 4 and 16 sub-images. On each level initially, the features are calculated and the SVM Classifier is used to determine the highest recognition rate. The subsequent outcomes are compared with the Quadratic and *k*-NN classifier. Another system for the recognition of isolated handwritten *Devanagari* numerals has been proposed by Aggarwal et al. [18]. The recommended method divides the sample image into sub-blocks. In these sub-blocks the strength of gradient is accumulated in 8 standard directions in which gradient direction is broken down resulting in a feature vector with a dimensionality of 200. For the classification, SVM classifier is utilized. In the research carried out by Singh et al. [19], a robust offline *Devanagari* handwritten recognition system is introduced using an amalgamation of global and local features. Structural features like cross point, endpoint, 'C' shaped structure, 'U' shaped structure, loop centroid, and inverted C shaped structure constitute global features. On the other hand, the zone-wise calculated distance of thinned image from geometric centroid and histogram-based features are combined to form the local features. Since the numerals are written by hand, there is bound to be variation in writing style. As a pre-processing step prior to feature extraction, this variation is managed by size normalization and normalization to constant thickness. For the classifier to be used for recognition, they have employed an Artificial Neural Network with an average accuracy rate of 95% or higher. Another feature extraction method as recommended by Prabhanjan et al. [20] is the Uniform Local Binary Pattern (ULBP) operator. Though this operator exhibited good performance in texture classification and object recognition, it is not used in *Devanagari* handwritten character/digit recognition. This suggested method works by extracting both the local and global features, and is carried out in two steps. The first step is noise-removal and this pre-processed image is converted to binary image and normalized to a fixed size of 48 by 48. The second step sees the application of the ULBP operator to the image in order to extract global features. The input image is thereafter split into 9 blocks and to extract local features, the operator is applied on each of the blocks. Lastly, global and local features are used for training the SVM classifier. Recently, Singh et al. [21] proposed a new feature extraction method known as *Regional Weighted Run Length* (*RWRL*) having a dimension of 196 elements, for handwritten *Devanagari* numeral recognition.

For the recognition of handwritten *Roman* numerals, Cao et al. [22] propose zone-based direction histogram feature. This research was mainly driven by two-stage classifier scheme consisting of two different neural networks. The lowest error rate was 0.17% with 14.5% rejection rate. To recognize offline handwritten *English* digits, Prasad et al. in [23] used a rotation invariant feature extraction scheme. Hybrid feature extraction method comprises features due to moment of inertia and projection features. They used a Hidden Markov Model (HMM) based classifier as the recognizer. Salouan et al. [24] presented isolated handwritten *Roman* numerals recognition on the basis of the combination of zoning method, Radon transform, Hough transform and Gabor filter. The performances of their individual and combined features are compared with respect to their accuracy and time complexity. The initial comparison is obtained between four hybrid methods used to extract the features from numbers. First, the zoning is combined with Radon transform, for the second, it is combined with Hough transform, for the third, it is next combined with Gabor and for the fourth, combined with all these three descriptors. On the other hand, the other comparison is implemented between three classifiers—first one is neuronal or the MLP classifier, second is probabilistic or HMM and the third classifier is a combination of the first and second classifiers. Qacimy et al. [25] examine the effectiveness of the four Discrete Cosine Transform (DCT) based feature extraction methodologies—first, the DCT upper left corner (ULC) coefficients, second, DCT zigzag coefficients, third, block based DCT ULC coefficients and finally, block based DCT zigzag coefficients. Feature extraction was conducted on the MNIST database [26] by inputting the coefficients of each DCT variant to the SVM classifier. It was determined that the recognition accuracy of block based DCT zigzag feature extraction was higher at a rate of 98.76%.

It can be seen from the above literature analysis that a lot of works have been done for numerals written in a single script. However, some works, described in [27–31] have been performed for numerals written in multiple scripts. This is done in order to develop script invariant methods for handwritten numeral recognition. Furthermore, few works have also been reported based on feature selection for the aforementioned problem. In the year 2018, Ghosh et al. [32] proposed a feature selection method known as Histogram-Based Multi-objective GA (HMOGA) for handwritten *Devanagari* numeral recognition This feature selection approach was improved by Guha et al. [33], by introducing a modified version of HMOGA named Memory-Based HMOGA (M-HMOGA) to solve this handwritten digit classification problem for *Bangla*, *Devanagari* and *Roman* scripts. Ghosh et al. [34] tried an innovative approach by applying union-based ensemble approach of three popular filter methods, namely Mutual Information (MI), ReliefF and Chi-square to reduce the feature dimension of HOG descriptor extracted form *Bangla*, *Hindi* and *Telugu* script numerals. Few works, described in [35–37], have also been reported for feature selection in handwritten numeral recognition from multiple scripts.

Recently, some researchers are also moving towards deep learning approach to tackle this state-of-the-art problem. Mukhoti et al. in [38] used deep learning to classify handwritten digits in *Bangla* and *Hindi* scripts. But there are limitations using this technique as it requires very large amount of data in order to perform better than other techniques. It is also extremely expensive to train due to complex data models. Moreover, deep learning requires expensive GPUs and hundreds of machines. Keeping this in mind, in the work done by Ghosh et al. [39], a script invariant feature vector is designed based on the concept of the DAISY descriptor and applied on handwritten digits written in four different scripts namely *Arabic*, *Bangla*, *Devanagari* and *Roman*. It is found to be computationally inexpensive approach when compared to other state-of-the-art prevalent deep learning architectures like Long Short Term Memory (LSTM) networks or (CNN). Motivated by the above facts, we are also using traditional machine learning techniques to tackle this problem of handwritten digit recognition.

3 Collection and Pre-processing of Numeral Databases

Handwritten sample collection, containing differences in handwriting styles, is the main objective of gathering data. This is required in order to get an accurate assessment of any feature extraction methodology. 10,000 samples of handwritten numerals are considered for *Roman* script taken from the benchmark HDRC 2013 [40] database. Even though there are standard benchmark databases for *Bangla* and *Devanagari* numerals [25], the sample size is small. Hence, we have prepared an in-house database of 10,000 handwritten numerals written in *Bangla* and *Devanagari*. Each person is asked to write 4 samples of numerals (0–9) each, and a total of about 250 people are involved in the data collection process who belong to varying age, sex, educational qualification, profession etc. Samples are collected on datasheets consisting of pre-defined rectangular grids in which they had to write the numerals using a blue or black colored pen. A datasheet sample containing *Bangla* numerals is demonstrated in Fig. 3. Each datasheet contains exactly 15 samples of handwritten numerals (0 through 9). For the present work, we have considered 10,000 numerals for every script *namely*, *Devanagari*, *Bangla* and *Roman*.

3.1 Pre-processing

Pre-processing involves the initial processing of the image, so that it can be used to make the further processing easier for an input to the recognition system. At first, the datasheets containing *Bangla* or *Devanaagri* numerals are captured on a flatbed scanner with a resolution of 600 dpi and are stored in bitmap file format. Now,

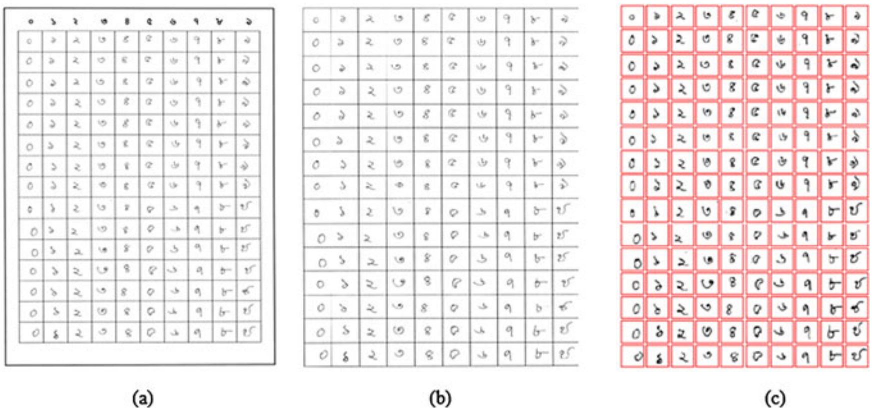

Fig. 3 **a** Scanned datasheet containing handwritten *Bangla* numerals; **b** Image obtained after trimming the outer frame and column headers; **c** Image divided into 10 × 15 cells to get 150 isolated numeral images

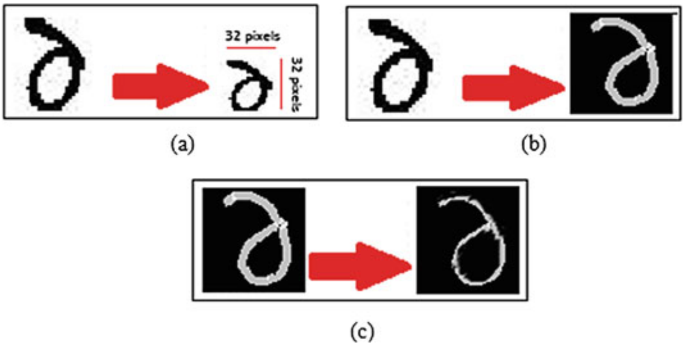

Fig. 4 Illustration of: **a** image resizing, **b** binarizing, and **c** image thinning for handwritten *Bangla* numeral image '1'

there are four below mentioned sub-processes involved for the processing of input numerals.

3.2 Extraction of Individual Digits

Firstly, we get rid of the noise using Gaussian filter [41]. Then, both the column headers (located across the top of the datasheet) as well as outer frame margin are removed. This is done by dividing the frame margins into pre-defined cells of size 10 × 15 (shown in Fig. 3). The individual digits obtained are saved as "B_data#####.bmp" or "D_data#####.bmp" depending upon *Bangla* or *Devanaagri* numerals respectively. Here, "#####" represents the naming scheme of the numeral images.

3.3 Image Resizing

The input numeral images may be of varying sizes which can affect the recognition results. Therefore, the minimally bounding rectangular box of each numeral image is normalized to 32 × 32 pixels separately, as shown in Fig. 4a.

3.4 Image Binarization

The process of binarization converts a grayscale image to its binary counterpart using Otsu's binarization methodology [41]. This turns out to be useful when features are extracted using the feature extraction module. An example is illustrated in Fig. 4b.

The following is a noteworthy point—the current process is applicable for extraction of local distance-based features.

3.5 Image Thinning

The final step is used to reduce a handwritten digit of thick lines into thinner lines, thus making it easier for its feature extraction, as shown in Fig. 4c. In the present work, this technique is applied before the extraction of local distance-based features.

4 Design of Feature Set

After pre-processing, the extraction of features is done to estimate the most pertinent characters of individual numeral classes to be used for the recognition stage. These features which have been extracted for our can be grouped into two main classes: global features and local features. Global features define the image in its entirety. Local features are extracted from the sub-regions or local regions and describe the most important sub regions in the image. The accuracy of the recognition process is improved by combining these local and global features subject to increase in computational overheads.

4.1 Global Features

For detecting objects from images, Dalal and Triggs [42] initial explanation of the HOG descriptor was principally concentrated on recognition of pedestrians. The elementary guiding thought is that the outline and appearance of the object within an image can be described by the intensity gradient distribution or the edge directions. The HOG descriptor is processed by dividing an image into smaller component regions and for each of these regions, the gradient and orientation are computed. The histogram buckets are evenly spaced over 0–180° or 0–360° on the basis of signed or unsigned gradient values usage. The features are produced by combining the histogram of all the component regions. HOG features are well suited for this kind of challenge since it functions on the localized cells. It can also be used to describe the outline and appearance of the handwritten digits in the given context. Our study considers 8 buckets over 7×7 blocks for feature extraction, thus resulting in a 392-dimensional ($8 * 7 * 7 = 392$) feature vector. The HOG transformed images for handwritten *Devanagari* numerals '0' and '3' are shown in Fig. 5.

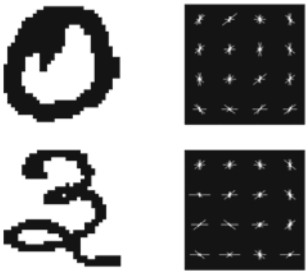

Fig. 5 Illustration of HOG feature descriptor (images on the left side are the original *Devanagari* numerals '0' and '3' whereas the right side shows their corresponding HOG transformed images)

4.2 *Local Distance Based Features*

These features mainly consider the distance of the first foreground pixel from the outer edge of the numeral calculated in 8 different directions as shown in the Fig. 5. Since all the numeral images are normalized to 32 × 32 pixels, the whole length is pigeonholed into 4 bins that is 0–7, 8–15, 16–23 and 24–31. Now, if we encounter a foreground pixel in the 1st bin, then the number of these foreground pixels present is calculated and stored in that bin. The same procedure is repeated for the other bins as well. After this process is done for one row (or column), we move on to the next row (or column) approaching in clockwise direction and the same procedure of counting foreground pixels for each bin is again repeated. Since we discretize these 32 values into 4 bins based on their count, thus a total of 32 (4*8) element feature vector is extracted using local distance features. Figure 6 shows an example of the estimation of the local distance features for handwritten *Roman* numeral '6'.

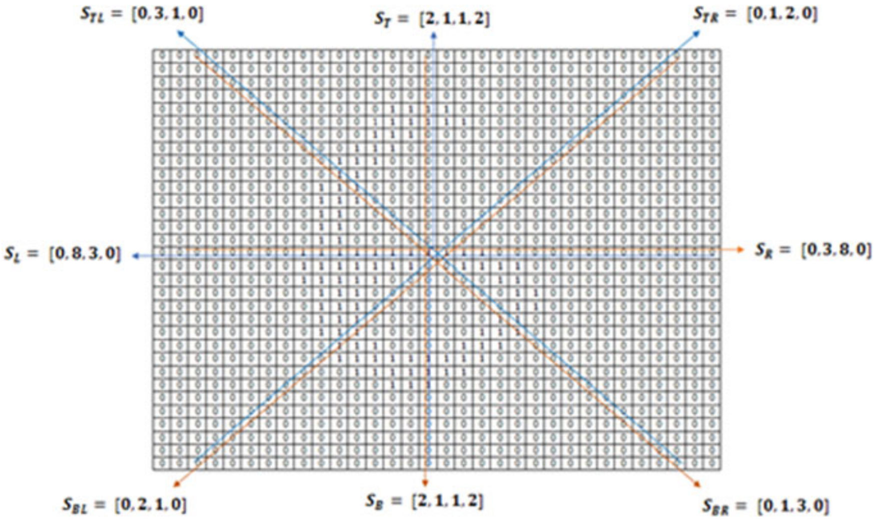

Fig. 6 Computation of local distance features for sample handwritten *Roman* numeral '6'. (S_T, S_B, S_R, S_L, S_{TR}, S_{TL}, S_{BR}, S_{BL} indicate the features values considered in all eight directions respectively)

4.3 Selection of Optimal Feature Subset Using GA

The final confusion matrix produced as a result of the classification stage using MLP classifier, shows instances of overlap and misclassification among similar shaped numerals. The outputted confusion matrix is further examined in order to find the numeral classes having the highest number of classifications. These numerals are considered similar to each other in terms of discriminating features and grouped together. Subsequently, both HOG and local distance features are used to precisely classify numerals within these smaller groups of numerals. This methodology gives rise to a large number of such features, hence the task of exhaustive search for an optimal set of local features becomes a cumbersome one. To counter this situation, we choose GA based feature selection method, which lets us recognize an optimal set from both the global and local feature sets, thereby leading to better recognition performance. Objective of feature selection includes removal of irrelevant and misleading features, reduction of time required for learning the classification function, increasing overall accuracy of recognition and to preserve only the relevant features which provide broad understanding of every input pattern class. There are various traditional methods of feature selection used in the literature like Sequential Forward Selection (SFS), Sequential Backward Selection (SBS), Exhaustive search and GA.

In our work, we have used GA owing to the following advantages. It is not advisable to carry out extensive search for large feature space as the time complexity is high. SFS and SBS consider all the features to find the optimal set simultaneously but in these algorithms if a feature is deleted from the set, then the chances of that feature getting selected is zero, resulting in removal of some discriminating feature in some cases. GA differs from all these methods due to the ability to evolve optimal features from the selected features and a good exploration of search space for newer and better solutions. The evolving process is made possible using GA operators such as selection, crossover and mutation. The process continues until the fitness criteria is met to the user's preference or if the number of iterations specified by the user is over. The various parameters of GA used in the present work are listed in Table 1.

GA is a meta-heuristic iterative process of improvement which incorporates evolution features like Darwin's principle of survival of the fittest [43], crossover and mutation. It is based on the premise that evolution arises due to the need for search of optimal solution set. Much like the biological process of adaptation, GA makes use of the historical runs to forecast the future solutions, in order to achieve optimal performance. In a specific chromosome, each bit is assigned a value of '1', if the feature is selected on the basis of its fitness; otherwise it is assigned a value of '0'. The flowchart of GA used in our current research is illustrated in Fig. 7.

Dividing the pattern image into a fixed set of identically divided regions is the easiest way to detect the regions with highest data discrimination. There are regions

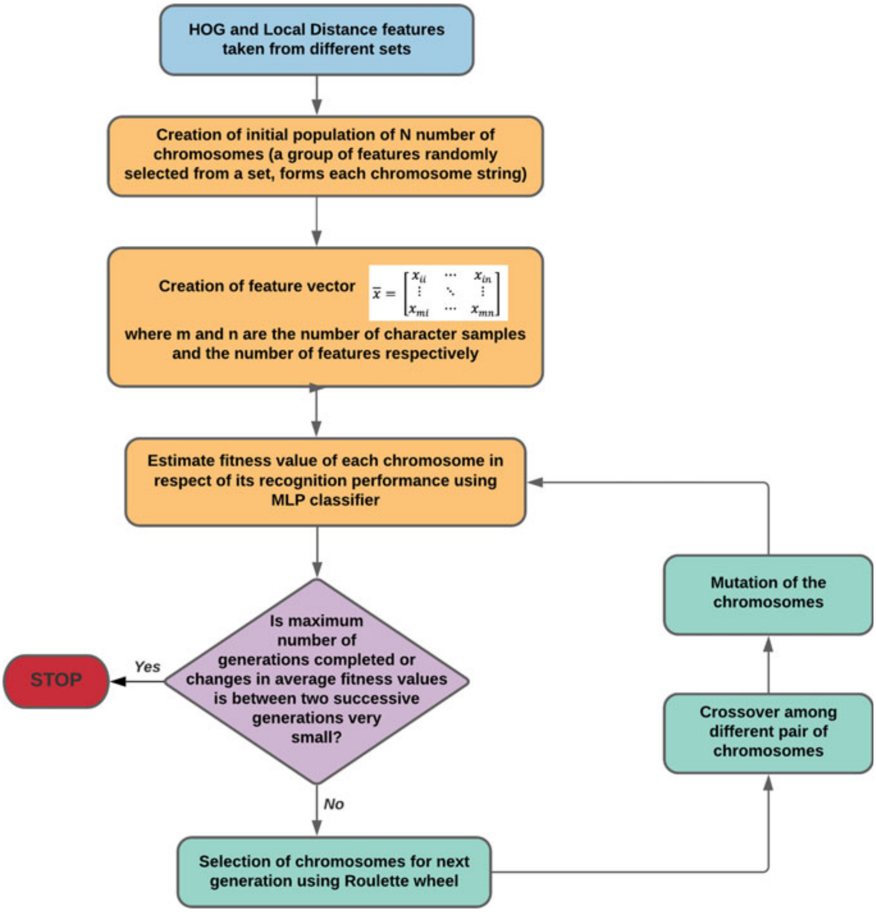

Fig. 7 Flowchart of the feature sampling technique using GA

Table 1 Values of parameters for GA used in the present work

GA parameters	Value
Population size	20
Initial population	100
Iterations	100
Selection method	Roulette-wheel
Crossover probability	0.85
Crossover point	Random
Mutation probability	0.15

which may overlap with each other, for such instances, local features are extracted. The global features along with the local features are combined and sampled randomly to generate different subsets. The effectiveness of the recognition methodology is assessed with each of these generated subsets. The subset, giving rise to the best outcome (fitness), can be assumed as an ideal set of features where the pattern classes can be distinguished considerably [44]. On the basis of these feature set outcomes, GA is applied to obtain the ideal local regions sets.

Each candidate solution or chromosome can be thought of as an *n* bit binary vector, where each bit represents a particular feature of a digit image. The initial population is created by generating random vectors, consists of 0 and 1 s. As per the standard definition of GA, if a feature is selected on the basis of its fitness, the bit is assigned a value of '1', else it is '0'. The initial population or generation consists of 100 such random vectors.

The fitness of the GA algorithm is calculated by training the MLP classifier using the training generated by taking the features selected (set as '1') in the given vector and it is then checked against the verified data set. The MLP classifier determines the fitness value of each chromosome. The top 20 vectors with the highest fitness values are used to obtain the next generation of chromosomes. These undergo crossover and mutation in the subsequent runs of the GA and another set of 100 vectors are further created. The percentage of correctly classified data is the fitness value. This is repeated for all the vectors and the best 20 of these is run for the next iteration of GA. The fitness is calculated once again for these vectors, to get the best 20 out of them and the process continues, until the number of specified iterations is completed (which, in our case is 100) or there is no significant change in result in the subsequent iterations of GA.

For the current research, we have followed a two-level classification for handwritten digit identification of three aforementioned scripts *namely*, *Roman*, *Devanagari* and *Bangla*. For *English* numerals, we start by grouping at the first classification level where we have used 424 features. Of these, 392 are the HOG features while the rest 32 are local features. Upon completion of the first level of classification, we obtain certain groups of similar numerals. There are four such groups *namely*, EG1 (8, 9), EG2 (3, 4, 7), EG3 (5, 6, 0) and EG4 (2, 1) which are also illustrated in Fig. 8a. The numbers of features selected by GA are used for their *intra-group* classification. These feature vectors consist of 180, 167, 171 and 167 elements for EG1, EG2, EG3 and EG4 groups respectively. These grouping so occurred since numerals within a particular group have similar shape structure. For example, the upper halves of the numerals '8' and '9' are almost identical and as such, these were grouped together as EG1. Similar trends are observed amongst the other group members as well.

In case of *Bangla* numerals, we employ 424 features in the first level of classification. Following this first level of classification, we obtain five groups of similar numerals *namely*, BG1 (1, 2, 9), BG2 (0, 3, 5, 6), BG3(4), BG4(7) and BG5(8). This grouping is demonstrated in Fig. 8b. It can be observed from Fig. 8b that the first two groups *namely*, BG1 and BG2 contain three and four numeral classes respectively whereas the other three groups involve only one individual numeral as their member.

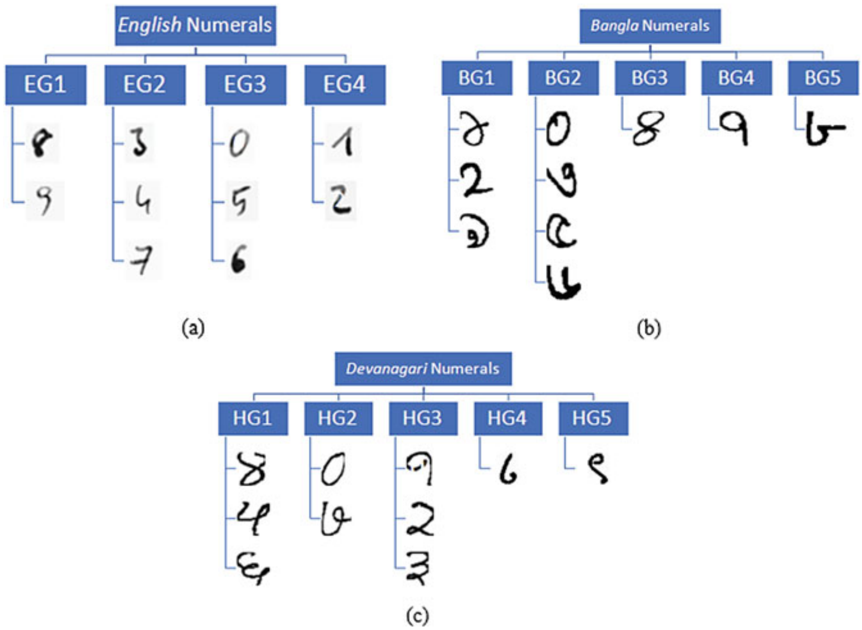

Fig. 8 Illustration of *inter-class* grouping of: **a** *English*, **b** *Bangla* and **c** *Devanagari* handwritten numerals in the first level of classification after the application of both global and local features

The numbers of features selected by GA are used for the first two *intra-group* classifications which are 180 and 183 respectively. On the other hand, the two groupings so occurred since numerals within a group have similar characteristics. For example, the *Bangla* numerals '1' and '9' are almost fully identical except for the bottom left quadrant. Again, *Bangla* numerals '1', '2' and '9' collectively have analogous shaped right halves. Hence, these are grouped together as BG1.

Considering *Devanagari* numerals, the first classification level consists of 424 features in a similar manner. Due to similarity in the shapes of some numeral classes, a second-level of classification is carried out wherein we obtain certain groups of similar numerals. There are five such groups *namely*, HG1 (4, 5, 6), HG2 (0, 7), HG3 (1, 2, 3), HG4(8) and HG5(9) which is also shown in Fig. 8c. The first three groups (HG1, HG2, and HG3) contain three, two and three members in their grouping whereas the remaining two groups have one numeral each. The numbers of features selected by GA are used for their *intra-group* classifications which are found to be 185, 187 and 178 respectively. For *Devanagari* numerals '1', '2' and '3', it is observed that these numerals have rounded upper halves, which make them identically shaped in the upper quadrant. Hence, this grouping occurred due to such similar characteristics. As a result, these were grouped together as HG3.

5 Experimental Analysis

In this section, we present the detail experimental results to illustrate the suitability of the proposed approach to handwritten numeral recognition. All the experiments are implemented in MATLAB 2015a under a Windows 8 environment on an Intel Core2 Duo 2.4 GHz processor with 4 GB of RAM and performed on gray-scale digit images. The recognition accuracy, used as assessment criteria for measuring the recognition performance of the proposed system, is expressed as follows:

$$Recognition\ Accuracy(\%) = \frac{\#Correctly\ classifieddigits}{\#Total\ digits} \times 100\% \quad (1)$$

A classification scheme modelled on neural-networks is devised for the task of classification. The MLP is a type of Artificial Neural Network (ANN). MLP classifier (as described in [45]) has been identified to be used, because of its recognized abilities to generalize and imbibe human behaviour by modelling the biological neural networks of a human. MLP is a feed-forward layered network of artificial neurons. A sigmoid function of the weighted sum of inputs is calculated by each artificial neuron in the MLP. An MLP contains the following layers: 1 input layer, 1 output layer and numerous hidden/intermediate layers. For our research, we had 1 hidden layer and the number of iterations for our MLP classifier was 100. For example, an artificial neuron combines its input signals by a weighted sum. The output is a single numerical figure, calculated from an 'activation' function, nearly modelling a 'firing' of a coarsely modelled biological neuron. Each neuron has synaptic weight coefficients of its own, with each neuron having an operating activation function. For the classification of handwritten numerals, we have to design the MLP, where the Back Propagation (BP) learning algorithm has a learning rate (η) of 0.8 and momentum term (α) of 0.7 being used here with different amounts of neurons in its hidden/intermediate layers.

We work with a training set of 6000 samples and another test set of 4000 samples, chosen for handwritten *Bangla*, *Devanagari* and *Roman* numerals; we consider equal quantities of digit samples from each class. The designed HOG feature set along with local distance-based features are then applied on the three script numerals and the MLP classifier is used for the identification purpose. The confusion matrix generated for handwritten *Bangla*, *Devanagari* and *Roman* numerals are reported in Tables 2, 3 and 4 respectively. Average recognition accuracies scored for all the three script numerals using MLP classifier are also illustrated in Table 5. It can be seen from Table 5 that the best recognition accuracies of 93.5, 83.98 and 88.83% are achieved for handwritten *Bangla*, *Devanagari* and *Roman* numerals respectively.

Based on the confusion matrices produced as a result of classification of handwritten numerals, the grouping of similar and overlapping numerals are performed. It can be examined from Table 2 that the *Bangla* numerals '1', '2' and '9' are confused among each other and can be placed into one group. Similarly, the numerals '0', '3',

Table 2 Confusion matrix produced by MLP classifier for handwritten *Bangla* numerals using both global and local features

Classified as —>	a	b	c	d	e	f	g	h	i	j
a = '0'	590	0	0	3	0	4	0	2	0	1
b = '1'	0	561	14	0	5	0	1	0	0	19
c = '2'	0	0	581	1	5	0	6	0	7	0
d = '3'	10	4	0	520	3	14	38	0	6	5
e = '4'	0	1	3	0	576	0	7	11	2	0
f = '5'	19	7	0	5	11	489	54	2	10	3
g = '6'	0	3	0	8	0	4	582	0	3	0
h = '7'	2	5	3	0	16	2	1	565	2	4
i = '8'	0	1	3	0	1	0	2	0	593	0
j = '9'	0	28	3	0	9	0	2	4	1	553

Table 3 Confusion matrix produced by MLP classifier for handwritten *Devanagari* numerals using both global and local features

Classified as —>	a	b	c	d	e	f	g	h	i	j
a = '0'	401	8	17	40	20	14	26	58	4	12
b = '1'	2	367	54	37	24	5	43	9	28	31
c = '2'	3	10	502	49	2	6	8	4	7	9
d = '3'	8	16	72	488	3	2	0	4	6	1
e = '4'	3	1	2	4	542	5	17	25	0	1
f = '5'	3	0	1	0	2	544	45	2	3	0
g = '6'	0	0	1	0	1	9	559	0	12	18
h = '7'	58	21	0	4	0	7	2	492	0	16
i = '8'	0	7	0	1	0	4	0	1	588	0
j = '9'	1	37	12	0	0	1	0	1	0	548

'5' and '6' can be placed into another group. The remaining numerals (i.e., '4', '7' and '8') are kept as singletons. In the case of *Devanagari* numerals, the maximum confusion is seen among the numerals '4', '5' and '6' due to which we keep them in one group. For the same reason, the second grouping is formed by taking the numerals '0' and '7' whereas the third group consists of the numerals '1', '2' and '3'. The remaining numerals '8' and '9' are set aside as singletons. In a similar way, four main groups are formed for handwritten *Roman* numerals by observing the number of misclassifications. The first group consists of the numerals '8', '9'; the second group consists of the numerals '3', '4' and '7'. The third grouping is prepared for the numerals '0', '5', '6' whereas the final group contains the numerals '1' and '2'. Here, no numeral is kept as singleton.

Table 4 Confusion matrix produced by MLP classifier for handwritten *Roman* numerals using both global and local features

Classified as —>	a	b	c	d	e	f	g	h	i	j
a = '0'	591	1	1	0	3	0	4	0	0	0
b = '1'	14	569	8	2	3	0	0	1	0	3
c = '2'	5	84	492	0	3	0	7	5	3	1
d = '3'	3	0	9	543	0	4	6	9	21	5
e = '4'	0	7	6	3	567	1	11	1	0	4
f = '5'	14	0	0	12	1	537	23	11	0	2
g = '6'	14	1	3	2	5	79	492	1	2	1
h = '7'	1	4	2	18	7	0	5	534	2	29
i = '8'	7	2	0	2	3	17	1	0	545	23
j = '9'	21	2	9	28	5	8	0	13	54	460

Table 5 Average recognition accuracy achieved for handwritten *Bangla, Devanagari* and *Roman* numerals using both global and local features

	Bangla	*Devanagari*	*Roman*
Total number of instances	6000	6000	6000
Correctly classified instances	5610	5031	5330
Incorrectly classified instances	390	969	670
Recognition accuracy (%)	93.50	83.98	88.83

Now, for the classification of *intra*-numeral classes present in each group for the three scripts, the combined feature set consisting of 424 features (HOG and local distance features) is applied on the individual groups of numeral classes written in one of the three scripts. The numerals that are selected as singletons would not undergo this procedure. GA, as described in Sect. 4.3, is then implemented as a feature selection procedure. The summary of the results after applying GA for all groupings of the three script numerals is shown in Table 6. The final results for all three script numerals are shown in Table 7. It can be observed from Table 7 that overall recognition accuracies of 98.49, 97.65 and 97.96% are attained for *Bangla*, *Devanagari* and *Roman* scripts respectively which are found to be much higher than the previous recognition accuracies reported in Table 5.

Despite the fact that we have achieved convincing results, there are still some misclassifications that occurred during the experimentation. Figure 9 shows some sample images of misclassified numerals. It can be seen from Fig. 9a, b that the *Bangla* numerals '0' and '5' show confusion in recognition. Similarly, *Bangla* numerals '5' and '6' also showed a lot of confusion due to unusual handwriting styles (refer to Fig. 9c, d). For *Devanagari* script, a significant amount of confusion is found among the numerals '1', '2' and '9'. For illustration, see Fig. 9e–h. Similarly, a substantial amount of *Roman* numerals '1', '3', '5' and '9' have been seen as misclassified with the numerals '7', '5', '3' and '4' respectively (shown in Fig. 9i–l). These misclassifications arise due to the fact that there are structural similarities between these

Table 6 Recogntion accuracies of the *intra-numeral* classes achieved by MLP classifier

Script	Group	Numerals in the group	Number of features selected by GA	Total number of instances	Correctly classified instances	Incorrectly classified instances	Recognition accuracy (%)
Bangla	BG1	1, 2, 9	180	1800	1760	40	97.8
	BG2	0, 3, 5, 6	183	2400	2370	30	98.75
Roman	RG1	8, 9	180	1200	1186	14	98.8
	RG2	3, 4, 7	167	1800	1775	25	98.6
	RG3	0, 5, 6	171	1800	1758	42	97.67
	RG4	1, 2	167	1200	1159	41	96.6
Devanagari	DG1	4, 5, 6	185	1800	1763	37	97.93
	DG2	0, 7	187	1200	1157	43	96.42
	DG3	1, 2, 3	178	1800	1744	56	96.89

Table 7 Overall summary of the results for handwritten *Bangla, Devanagari* and *Roman* numerals (the overall accuracy is obtained by performing the weighted average of recognition accuracies achieved for each individual group)

Script	Numeral classes	Recognition accuracy (%)	Overall accuracy (%)
Bangla	1, 2, 9	97.8	98.49
	0, 3, 5, 6	98.75	
	4	98.83	
	7		
	8		
Roman	8, 9	98.8	97.96
	3, 4, 7	98.6	
	5, 6, 0	97.67	
	2, 1	96.6	
Devanagari	4, 5, 6	97.93	97.65
	0, 7	96.42	
	1, 2, 3	96.89	
	8	99.61	
	9		

numerals, which were created due to peculiar styles of handwriting styles of different sets of people.

In the current research, we have compared our recommended approach with other contemporary numeral recognition techniques, as shown below in Table 8. We can observe that our recommended methodology performs better than most of the prior numeral recognition techniques, thus proving that our present procedure is well applicable for various scripts.

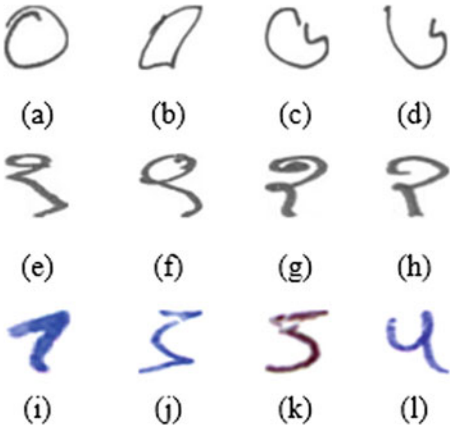

Fig. 9 Sample numeral images misclassified by the present technique of: **a** *Bangla* numeral '0' (misclassified as '5'), **b** *Bangla* numeral '5' (misclassified as '0'), **c** *Bangla* numeral '5' (misclassified as '6'), **d** *Bangla* numeral '6' (misclassified as '5'), **e** *Devanagari* numeral '1'(misclassified as '9'), **f** *Devanagari* numeral '9'(misclassified as '1'), **g** *Devanagari* numeral '1'(misclassified as '2'), **h** *Devanagari* numeral '2'(misclassified as '1'), **i** *Roman* numeral '1' (misclassified as '7'), **j** *Roman* numeral '3' (misclassified as '5'), **k** *Roman* numeral '5' (misclassified as '3'), **l** *Roman* numeral '9' (misclassified as '4')

6 Conclusion

Recognition of handwritten digits is one of the most interesting and challenging areas of research in the field of pattern recognition and image processing. Several research works have been accomplished for non-Indian scripts focusing on the development of new techniques with the aim of achieving higher recognition accuracy in this area. Still there exists a shortage of interest in researching identification of numerals written in different Indian scripts. The present work reports the recognition of numerals for three widely spoken official Indian scripts written in *Bangla*, *Devanagari* and *Roman*. For our study, we have used global and local features in tandem in order to identify handwritten digits. The extraction of global features is done using HOG descriptor whereas the local distance based features are measured as the local features. The proposed system applies the global features for grouping of the numerals which are structurally similar in nature. Then, the mix of both local and global features is taken for the selection of optimal subset of features using GA. Finally, these optimal set of features are employed for the classification of the *intra-numeral* classes using the MLP classifier. Applying this proposed technique, we have achieved impressive recognition accuracies of **98.49%, 97.65%** and **97.96%** for *Bangla*, *Devanagari* and *Roman* numerals respectively on a database of 10,000 handwritten numerals considered per script. Our research findings can be applied to other less-researched digit recognition areas of Indian scripts. Further study can be carried out by integrating HOG and local distance-based features along with other texture-based features with the objective of obtaining greater classification accuracy.

Table 8 Comparative study of proposed methodology with some state-of-the-art techniques (present recognition accuracies are highlighted in bold style)

Researchers	Numeral script	Database used	Size of the database	Feature set used	Classifier	Recognition accuracy (%)
Basu et al. [7]	*Bangla*	*CMATERdb* 3.1.1 [7]	6000	Shadow, centroid and longest run feature	MLP	96.67
Nasir et al. [9]	*Bangla*	Own database	300	Hybridization of k-means clustering and Bayes' Theorem	SVM	99.33
Surinta et al. [10]	*Bangla*	Own database	10, 920	Feature and Pixel based features	SVM	96.8
Khan et al. [11]	*Bangla*	*CMATERdb* 3.1.1 [7]	6000	Zone density feature extraction	SRC	94
Proposed methodology	*Bangla*	**Own database**	**10,000**	**GA based HOG and local distance feature selection**	**MLP**	**98.49**
Bhattacharaya et al. [15]	*Devanagari*	Own database	Not known	Multi-resolution features based on wavelet transform	MLP	91.28
Hanmandlu et al. [16]	*Devanagari*	Own database	Not known	Vector distances	Fuzzy model	92.67
Singh et al. [19]	*Devanagari*	Own database	3,000	Global, local and Profile based features	ANN	95
Proposed methodology	*Devanagari*	**Own database**	**10,000**	**GA based HOG and local distance feature selection**	**MLP**	**97.65**
Prasad et al. [23]	*Roman*	CENPARMI	500	Moment of Inertia based features	HMM	91.2

(continued)

Table 8 (continued)

Researchers	Numeral script	Database used	Size of the database	Feature set used	Classifier	Recognition accuracy (%)
Salouan et al. [24]	*Roman*	Own database	3,000	Radon Transform, Hough transform and Gabor Filter	MLP and HMM	96.60
Qacimy et al. [25]	*Roman*	MNIST	10,000	Discrete Cosine Transform	SVM	96.61
Proposed methodology	*Roman*	**HDRC 2013** [29]	**10,000**	**GA based HOG and local distance feature selection**	**MLP**	**97.96**

References

1. R. Hussain, A. Raza, I. Siddiqi, K. Khurshid, C. Djeddi, A comprehensive survey of handwritten document benchmarks: structure, usage and evaluation. EURASIP J. Image Video Process. **2015**(1), 46 (2015)
2. U. Bhattacharya, B.B. Chaudhuri, Handwritten numeral databases of Indian scripts and multistage recognition of mixed numerals. IEEE Trans. Pattern Anal. Mach. Intell. **31**(3), 444–457 (2008)
3. U. Pal, N. Sharma, T. Wakabayashi, F. Kimura, Handwritten numeral recognition of six popular Indian scripts, in *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, vol. 2. (IEEE, 2007), pp. 749–753
4. R.M. Bozinovic, S.N. Srihari, Off-line cursive script word recognition. IEEE Trans. Pattern Anal. Mach. Intell. **11**(1), 68–83 (1989)
5. P.K. Wong, C. Chan, Off-line handwritten Chinese character recognition as a compound Bayes decision problem. IEEE Trans. Pattern Anal. Mach. Intell. **20**(9), 1016–1023 (1998)
6. U. Pal, B.B. Chaudhuri, Indian script character recognition: a survey. Pattern Recogn. **37**(9), 1887–1899 (2004)
7. S. Basu, N. Das, R. Sarkar, M. Kundu, M. Nasipuri, D.K. Basu, An MLP based approach for recognition of handwritten Bangla' numerals. arXiv preprint arXiv:1203.0876
8. Y. Wen, L. He, A classifier for Bangla handwritten numeral recognition. Exp. Syst. Appl. **39**(1), 948–953 (2012)
9. M.K. Nasir, M.S. Uddin, Hand written Bangla numerals recognition for automated postal system. IOSR J. Comput. Eng. (IOSR-JCE) **8**(6), 43–48 (2013)
10. O. Surinta, L. Schomaker, M. Wiering, A comparison of feature and pixel-based methods for recognizing handwritten bangla digits, in *2013 12th International Conference on Document Analysis and Recognition* (IEEE, 2013), pp. 165–169
11. H.A. Khan, A. Al Helal, K.I. Ahmed, Handwritten Bangla digit recognition using sparse representation classifier, in *2014 International Conference on Informatics, Electronics and Vision (ICIEV)* (IEEE, 2014), pp. 1–6
12. M.A.H. Akhand, M. Ahmed, M.H. Rahman, Convolutional neural network based handwritten Bengali and Bengali–English mixed numeral recognition. Int. J. Image Graph. Sig. Process. **8**(9), 40 (2016)
13. P.K. Singh, R. Sarkar, M. Nasipuri, A comprehensive survey on Bangla handwritten numeral recognition. Int. J. Appl. Pattern Recogn. **5**(1), 55–71 (2018)
14. I.K. Sethi, B. Chatterjee, Machine recognition of constrained hand printed Devanagari. Pattern Recogn. **9**(2), 69–75 (1977)
15. U. Bhattacharya, B.B. Chaudhuri, R. Ghosh, M. Ghosh, On recognition of handwritten Devnagari numerals, in *Proceedings of the Workshop on Learning Algorithms for Pattern Recognition (in conjunction with the 18th Australian Joint Conference on Artificial Intelligence)*, Sydney, p. 1–7 (2005)
16. M. Hanmandlu, O.R. Murthy, Fuzzy model based recognition of handwritten numerals. Pattern Recogn. **40**(6), 1840–1854 (2007)
17. M.J.K. Singh, R. Dhir, R. Rani, Performance comparison of devanagari handwritten numerals recognition. Int. J. Comput. Appl. **22 (**2011)
18. A. Aggarwal, R.R. Renudhir, Recognition of Devanagari handwritten numerals using gradient features and SVM. Int. J. Comput. Appl. **48**(8), 39–44 (2012)
19. P. Singh, A. Verma, N.S. Chaudhari, Handwritten Devnagari digit recognition using fusion of global and local features. Int. J. Comput. Appl. **89**(1) (2014)
20. S. Prabhanjan, R. Dinesh, Handwritten devanagari characters and numeral recognition using multi-region uniform local binary pattern. Int. J. Multimedia Ubiquit. Eng. **11**(3), 387–398 (2016)
21. P.K. Singh, S. Das, R. Sarkar, M. Nasipuri, Recognition of offline handwriten Devanagari numerals using regional weighted run length features, in *2016 International Conference on Computer, Electrical and Communication Engineering (ICCECE)* (IEEE, 2016), pp. 1–6

22. J. Cao, M. Ahmadi, M. Shridhar, Recognition of handwritten numerals with multiple feature and multistage classifier. Pattern Recogn. **28**(2), 153–160 (1995)
23. B.K. Prasad, G. Sanyal, A hybrid feature extraction scheme for Off-line English numeral recognition, in *International Conference for Convergence for Technology* (IEEE, 2014), pp. 1–5
24. R. Salouan, S. Safi, B. Bouikhalene, Isolated handwritten Roman numerals recognition using methods based on radon, Hough transforms and Gabor filter. Int. J. Hybrid Inf. Technol. **8**, 181–194 (2015)
25. B. El Qacimy, M.A. Kerroum, A. Hammouch, Feature extraction based on DCT for handwritten digit recognition. Int. J. Comput. Sci. Issues (IJCSI) **11**(6), 27 (2014)
26. Y. LeCun, The MNIST database of handwritten digits (1998). http://yann.lecun.com/exdb/mnist/
27. P.K. Singh, S. Das, R. Sarkar, M. Nasipuri, Recognition of handwritten Indic script numerals using Mojette transform, in *Proceedings of the First International Conference on Intelligent Computing and Communication* (Springer, Singapore, 2017), pp. 459–466
28. P.K. Singh, R. Sarkar, M. Nasipuri, A study of moment based features on handwritten digit recognition. Appl. Comput. Intell. Soft Comput. (2016)
29. P.K. Singh, S. Das, R. Sarkar, M. Nasipuri, Script invariant handwritten digit recognition using a simple feature descriptor. Int. J. Comput. Vis. Rob. **8**(5), 543–560 (2018)
30. S. Ghosh, A. Chatterjee, P.K. Singh, S. Bhowmik, R. Sarkar, Language-invariant novel feature descriptors for handwritten numeral recognition. Vis. Comput. **2020** (2020). https://doi.org/10.1007/s00371-020-01938-x
31. R. Samanta, S. Ghosh, A. Chatterjee, R. Sarkar, A novel approach towards handwritten digit recognition using refraction property of light rays. Int. J. Comput. Vis. Image Process. (IJCVIP) **10**(3), 1–17 (2020)
32. M. Ghosh, R. Guha, R. Mondal, P.K. Singh, R. Sarkar, M. Nasipuri, Feature selection using histogram-based multi-objective GA for handwritten Devanagari numeral recognition, in *Intelligent Engineering Informatics* (Springer, Singapore, 2018), pp. 471–479
33. R. Guha, M. Ghosh, P.K. Singh, R. Sarkar, M. Nasipuri, M-HMOGA: a new multi-objective feature selection algorithm for handwritten numeral classification. J. Intell. Syst. **29**(1), 1453–1467 (2019)
34. S. Ghosh, S. Bhowmik, K.K. Ghosh, R. Sarkar, S. Chakraborty, A filter ensemble feature selection method for handwritten numeral recognition. EMR **7213** (2016)
35. A. Roy, N. Das, R. Sarkar, S. Basu, M. Kundu, M. Nasipuri, An axiomatic fuzzy set theory based feature selection methodology for handwritten numeral recognition, in *ICT and Critical Infrastructure: Proceedings of the 48th Annual Convention of Computer Society of India,* vol. I. (Springer, Cham ,2014), pp. 133–140
36. S. Sarkar, M. Ghosh, A. Chatterjee, S. Malakar, R. Sarkar, An advanced particle swarm optimization based feature selection method for tri-script handwritten digit recognition, in *International Conference on Computational Intelligence, Communications, and Business Analytics* (Springer, Singapore, 2018), pp. 82–94
37. S. Chakraborty, S. Paul, R. Sarkar, M. Nasipuri, Feature map reduction in CNN for handwritten digit recognition, in *Recent Developments in Machine Learning and Data Analytics* (Springer, Singapore, 2019), pp. 143–148
38. J. Mukhoti, S. Dutta, R. Sarkar, Handwritten digit classification in Bangla and Hindi using deep learning. Appl. Artif. Intell. 1–26 (2020)
39. A. Chatterjee, S. Malakar, R. Sarkar, M. Nasipuri, Handwritten digit recognition using DAISY descriptor: a study, in *2018 Fifth International Conference on Emerging Applications of Information Technology (EAIT)* (2018), pp. 1-4
40. M. Diem, S. Fiel, A. Garz, M. Keglevic, F. Kleber, R. Sablatnig, ICDAR 2013 competition on handwritten digit recognition (HDRC 2013), in *2013 12th International Conference on Document Analysis and Recognition* (IEEE, 2013), pp. 1422–1427
41. R.C. Gonzalez, R.E. Woods, *Digital Image Processing*, vol. I (Prentice-Hall, India, 1992)
42. N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1. (IEEE, 2005), pp. 886–893