Applied Soft Computing 12 (2012) 1592–1606

Contents lists available at SciVerse ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

A genetic algorithm based region sampling for selection of local features in handwritten digit recognition application

Nibaran Das, Ram Sarkar, Subhadip Basu, Mahantapas Kundu, Mita Nasipuri *, Dipak Kumar Basu

Computer Science & Engineering Department, Jadavpur University, Kolkata 700032, India

ARTICLE INFO

Article history:
Received 4 November 2010
Received in revised form 27 October 2011
Accepted 21 November 2011
Available online 8 January 2012

Keywords:
Variable sized local regions
Region sampling
Genetic algorithm
Optimal local regions
Feature selection
N-Quality consensus

ABSTRACT

Identification of local regions from where optimal discriminating features can be extracted is one of the major tasks in the area of pattern recognition. To locate such regions different kind of region sampling techniques are used in the literature. There is no standard methodology to identify exactly such regions. Here we have proposed a methodology where local regions of varying heights and widths are created dynamically. Genetic algorithm (GA) is then applied on these local regions to sample the optimal set of local regions from where an optimal feature set can be extracted that has the best discriminating features. We have evaluated the proposed methodology on a data set of handwritten Bangla digits. In the present work, we have randomly generated seven sets of local regions and from every set, GA selects an optimal group of local regions which produces best recognition performance with a support vector machine (SVM) based classifier. Other popular optimization techniques like simulated annealing (SA) and hill climbing (HC) have also been evaluated with the same data set and maximum recognition accuracies were found to be 97%, 96.7% and 96.7% for GA, SA and HC, respectively. We have also compared the performance of the present technique with those of other zone based techniques on the same database.

1. Introduction

To recognize different pattern classes, human cognition system has an inherent ability to identify the local regions, where the pattern classes differ significantly. This inherent ability of human being may be imitated in any pattern recognition system by incorporating the ability of locating the regions which contain the maximum discriminating information among the pattern classes. For example, the shapes of *Bangla* digit '১' (one) [Fig. 1 (a)] and '৯' (nine) [Fig. 1(b)], are almost similar except the regions in the lower middle portion of the digit images which is shown in Fig. 1(c). Therefore any person has to observe closely the lower portion of the digit images to distinguish these two *Bangla* digits.

The simplest way to identify the regions containing maximum discriminatory information is to divide the pattern image into a fixed number of equal sized regions. These regions may have some overlap with each other. For each such region, features (often called local features) are extracted. These local regions are then sampled randomly to produce various subsets of them. The recognition performance is evaluated with feature set formed with the features of each of those subsets. The subset, producing the best result, may be considered as an optimal set of local regions where the pattern classes differ significantly. Handwritten character recognition is a typical example of a real world pattern recognition problem which requires modeling of cognitive abilities of human being [1]. Some works have already been done in the field of handwritten character recognition following the above mentioned principles [2–7]. In [2], Rajashekararadhya et al. proposed an efficient zone based feature extraction technique for handwritten *Kannada*, *Telugu*, *Tamil* and *Malayalam* numeral recognition. In that paper, they divided the characters into some $M \times N$ zones and centroid and distance features were extracted from the zones. They obtained recognition accuracies of 99%, 99%, 96%, and 95% for *Kannada*, *Telugu*, *Tamil*, and *Malayalam* numerals, respectively, with nearest neighbor classifiers. In our earlier work [3], Basu et al. divided the digit image into 9 overlapping fixed sized *sub-images* and from each of these sub-images, longest run based features were locally computed. They used multi layer perceptron (MLP) based classifier and tested their methodology on *Bangla* numeral dataset with recognition accuracy of 96.65%. In the paper [4], Cao et al. used zone based direction histogram features for recognition of handwritten *Roman* numerals. The work was primarily motivated by two stage classifier scheme comprising of two different neural networks. They obtained lowest error rate of 0.17% with rejection rate of 14.5%. All the methods described in [2–4] used specific number of fixed sized windows covering the whole digit image. But a fixed sized window may contain some ambiguous information about the pattern shapes besides the discriminating ones, which may have an adverse effect on the performance of the recognition system. In the paper [5], Park et al.

* Corresponding author. Tel.: +91 33 24146766/+91 9831128131.
E-mail address: mitanasipuri@yahoo.com (M. Nasipuri).

1568-4946/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.asoc.2011.11.030

(a) Sample image of Bangla digit 'one'

(b) Sample image of Bangla digit 'nine'

(c) Overlapped images of Bangla digits 'one' and 'nine'

Fig. 1. Shape similarities between Bangla digits 'one'and 'nine'.

described a hierarchical feature space based on the zones, created by quin-tree based zoning scheme. The variable sized zones were created dynamically based on the centroids of the contours of the characters/character segments within the parent zone. Thus, zones of variable sizes covering the entire image were created for every character of the database. Histogram based gradient and moment based projection features were calculated from those zones and recognition accuracy of 98.2% was achieved on NIST digit database. In the approaches [6–8] windows of varying sizes covering the whole image were dynamically created on the basis of the center of gravity (CG) of black pixels in the entire character region or its sub-regions (windows) and from each of those regions, four longest run features were extracted. Those features along with another set of four longest run features computed globally from the entire character image were used for recognition of *Bangla* alphabet in [6] and multi-script numerals in [7,8]. This approach gave better result over [3] on the same *Bangla* numeral dataset due to dynamic creation of non-empty windows, i.e. each of the windows contains some character shape information. In [9], authors tried to partition the whole digit image into a fixed number of regions from which local features were extracted. Here they used techniques based on fuzzy logic and GA to adjust the partitioning lines, so that an optimum recognition performance was achieved. However, none of the above mentioned methods tries to find out the set of local regions, where the character shapes differ most. But in the paper [10], Das et al. described a method to identify appropriate group of windows out of the 16 overlapping fixed size windows covering the whole character image in order to decrease computational cost and achieve high recognition performance. In that attempt, genetic algorithm (GA) was used to select the set of windows identifying optimal local regions in the digit images which contain discriminating information among the ten digit classes of *Bangla* numerals. However, the method used fixed sized windows for extraction of local features and it suffered from limitation of fixed sized windows already discussed in early part of this section.

In the present work, we have proposed a technique in which variable sized local regions are created randomly and they do not have any fixed height, width or aspect ratio. A certain percentage of overlap among neighboring regions may be allowed. As the number of such local regions created in this manner is large, an exhaustive search for an optimal set of local regions is a mammoth task. To alleviate this problem, we have used GA based region sampling method to select an optimal set of these local regions which will provide better recognition performance.

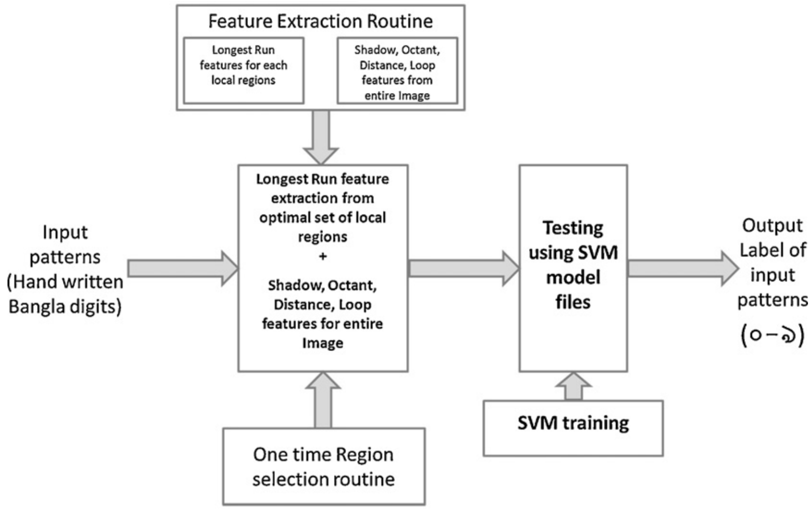
GA is a heuristic iterative search algorithm based on the mechanism of natural genetics derived from Darwin's principle of *survival of the fittest*. It efficiently utilizes the history of the previous runs to predict new solution set (population) with an objective of improved performance. GA is different from other traditional search methods because it uses directed random choice to conduct a highly exploitative search through a coding of parameter space. To locate better solutions, GA uses crossover techniques by which useful information may be exchanged among different solutions. Further, using mutation, it is possible to reintroduce lost genetic trails, which also prevents GA from sticking at a local optimum point. To evaluate the effectiveness of the present approach, we have tested our method on *Bangla* digit images which is a subset o*f Bangla character set*. We have randomly created several sets of local regions with varying constraints like percentage of the pattern image area covered by local regions, percentage of overlap among neighboring local regions, etc. For each set, longest run features [11] are extracted for each of the local regions. Based on these extracted local features, together with some global features, GA is applied to search for the optimal set of local regions from which best discriminating information can be obtained. Due to probabilistic development of the solution, GA does not ensure optimality within fixed number of runs. So we have used a five quality consensus among the five optimal sets of local regions selected by five different runs of GA to obtain the best optimal set of local regions. This results in high recognition rate with lesser number of features. The local regions thus selected are finally validated from a fuzzy grey level template, which is created using the combination of all digit images in the training set. We have also evaluated other popular optimization techniques such as simulated annealing (SA) and hill climbing (HC) using the same database to establish the superiority of GA.

2. Present work

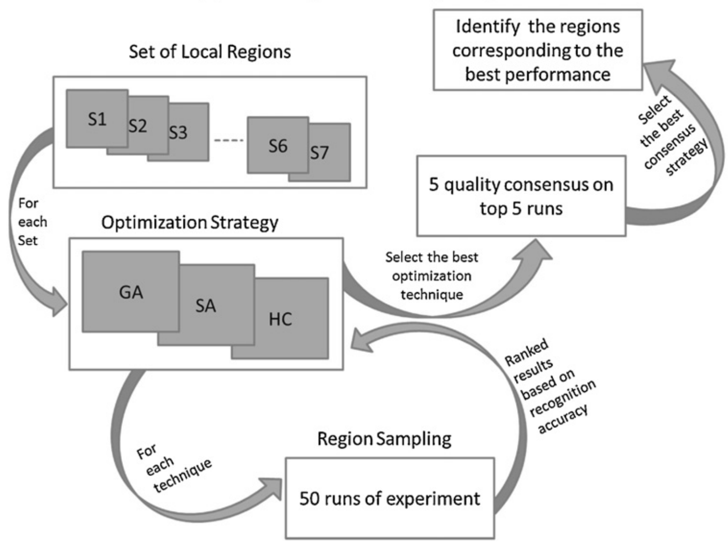
As discussed before, objectives of the current work are: (1) to design an optimum strategy to select local regions containing maximum discriminating information about the pattern classes of the application domain, (2) to select the best optimization technique among GA, SA and HC, (3) to develop a consensus strategy to select the final set of local regions among the top experimental runs, and (4) to evaluate the performance of the system with handwritten digit samples of *Bangla* script. A diagram of the integrated system is shown in Fig. 2(a). Fig. 2(b) shows a block diagram of the *one-time window selection strategy* (shown as a single module in Fig. 2(a)) employed during the training phase.

2.1. Definitions and notation

Let, $I_{M,N}$ be a digital image defined as a 2D array of dimension $M \times N$, such that, $I_{M,N} = \left\{ f(i,j) | 0 \le i \le M-1,\ 0 \le j \le N-1 \right\}$ where $f(i,j)$ is the intensity value at the coordinate (i,j). For a binary



(a). Block diagram of the integrated system



(b). Block diagram of the one-time region selection strategy

Fig. 2. Integrated system developed under the present work.

image, $f(i,j) \in \{0,1\}$. A local region is nothing but a sub-image of $I_{M,N}$ which may be defined as $J_k(i_{1k}, j_{1k}, i_{2k}, j_{2k})$, where (i_{1k}, j_{1k}) and (i_{2k}, j_{2k}) are the top left and bottom-right coordinates of the sub-image and $i_{2k} > i_{1k}, j_{2k} > j_{1k}, \quad 0 \le i_{1k}, i_{2k} \le M-1$ and $0 \le j_{1k}, j_{2k} \le N-1$. A set $P = \{J_0, J_1, J_2, J_3, \ldots, J_L\}$ is defined as a random collection of L local regions of $I_{M,N}$ such that $\{J_0 \cup J_1 \cup J_2 \cup \ldots J_L\} \subseteq I_{M,N}$. P will be subsequently referred as a set of local regions. We also define S_i as a subset of $P(S_i \subseteq P)$ that maximizes recognition performance for a given training/test data set.

A grey level template T is formed by averaging k number of binary patterns of dimension $M \times N$. Each pixel position $(x, y), \quad 0 \le x \le M-1, 0 \le y \le N-1$, of the template T, represents the average pixel value $\mu(x, y)$, which is estimated from the pixel values of all the binary patterns as follows:

$$\mu(x,y) = 255\left\{1 - \frac{1}{k}\sum_{j=0}^{k-1} f_j(x,y)\right\} \tag{1}$$

where $f_j(x, y)$ represents the pixel value at (x, y) position of jth binary pattern and $f_j(x,y) \in \{0,1\}$ (here 0 represents background and 1 represents foreground).

2.2. Methodology

In our present work we have considered handwritten *Bangla* digit patterns, a subset of *Bangla* characters, for validating the effectiveness of our methodology. Fig. 3 shows some handwritten samples of *Bangla* digit patterns.

Our work is divided into several steps as discussed below:

1. Overlapping/non-overlapping local regions (J_k) with varying sizes are created randomly. Different sets of local regions, P_j, $j = 1, 2, \ldots, N$ are then randomly selected from those local regions such that the union of the areas covered by the local regions of each set P_j is at least a certain percentage of the overall area of the digit image $I_{M,N}$, i.e. $P_j = \{J_k | k = 1, 2, \ldots, L$ and $(\cup_{k=0}^{L} A|J_k|) \ge \delta A \left|I_{M,N}\right|\}$ where $A|.|$ denotes area and δ denotes a thresholding value.
2. Different groups of local regions S_i are randomly selected from each P_j(i.e. $S_i \subseteq P_j$) to construct different chromosome strings forming the initial population of GA. Each of the selected local regions is treated as a gene in the chromosome.

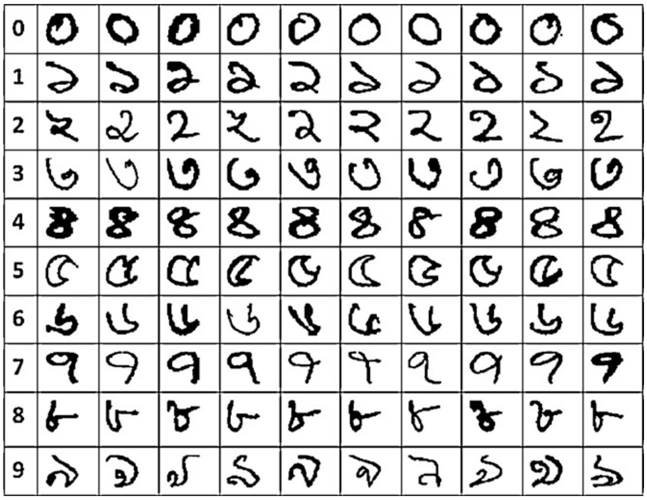


Fig. 3. Some handwritten samples of digits of *Bangla* script.

3. For each chromosome, local features are extracted from local regions represented by the genes of that chromosome. Global features are generated from the whole digit image.
4. With the feature set obtained in step 3, a SVM classifier is trained and tested on handwritten *Bangla* digit images. The recognition performance on the test set is considered as an estimate of fitness value of that chromosome.
5. Steps 3 and 4 are repeated for all chromosomes.
6. GA produces chromosomes for the subsequent generations using its operators, *crossover* and *mutation*, and heuristically searches optimized set of local regions based on the fitness values of the chromosomes.

Fig. 4 describes the flow of steps to be performed during the operation of GA.

2.3. Random creation of the sets of local regions

In our experiment, local regions on the character pattern are created by randomly generating the coordinates of the local region J_k, i.e. $(i_{1k}, j_{1k}, i_{2k}, j_{2k})$ using a program. During the creation of local regions we have considered the following constraints.

a) The maximum and minimum sizes of the windows are restricted by two threshold values such that $\delta_1 A|I_{M,N}| \leq A|J_i| \leq \delta_2 A|I_{M,N}|$, where δ_1, δ_2 are two thresholding parameters such that $0 \leq \delta_1, \delta_2 \leq 1$, $\delta_1 \leq \delta_2$.
b) During the creation of sets, overlap among local regions is not allowed for some sets, i.e. $A|J_i| \cap A|J_k| = \varnothing \forall i, k$ or overlap up to a certain degree is allowed for rest of the sets; i.e. $A|J_i| \cap A|J_k| \leq \delta_3 A|J_i| \quad \forall i, k$, where δ_3 is a thresholding parameter.
c) The union of the areas covered by all the created local regions should not be less than a certain fraction of the overall area of the image; i.e. $\cup_{i=0}^{L} A|J_i| \geq \delta_4 A|I_{M,N}|$, where δ_4 is another thresholding parameter (Table 1).

2.4. Design of feature set

Choice of suitable feature set for pattern classes is a domain specific design issue [12]. In the present work, two different feature sets are designed for classification of handwritten *Bangla* digit patterns.

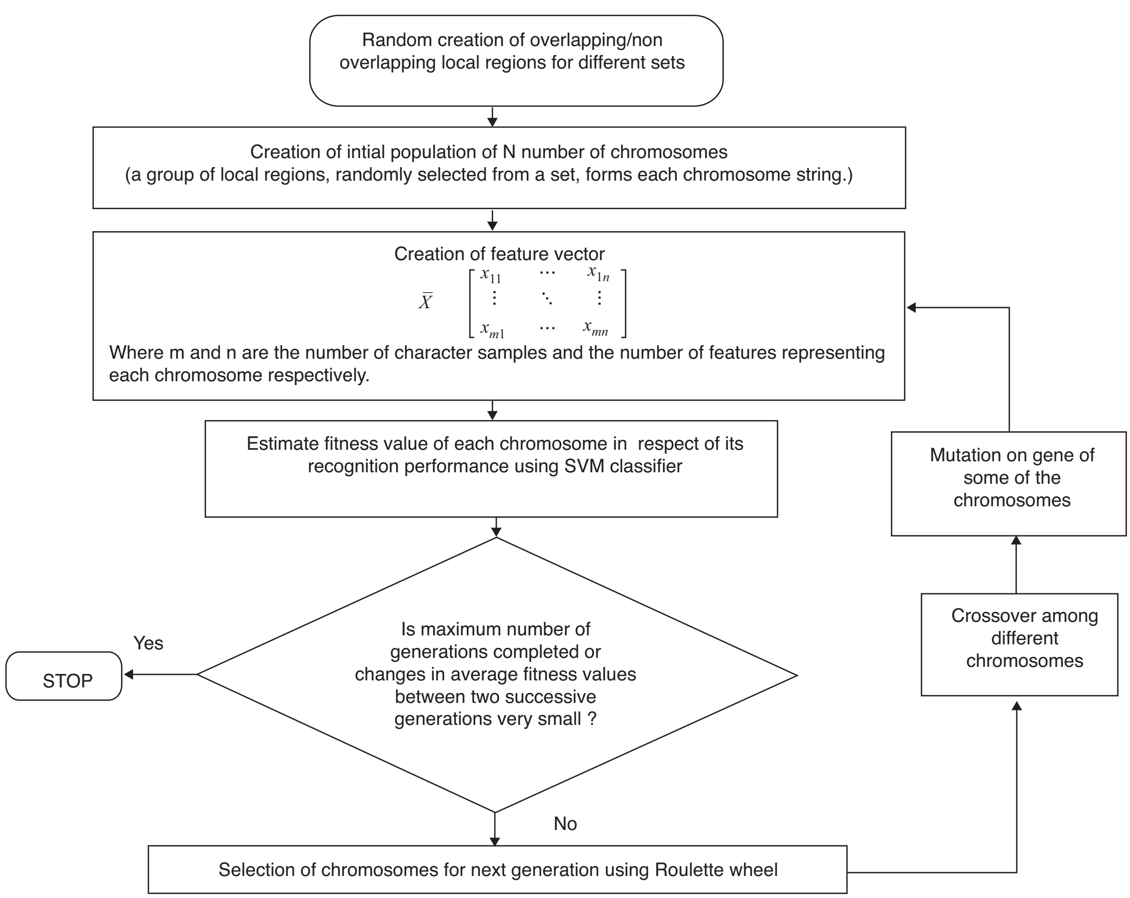


Fig. 4. Flow diagram of the region sampling technique using GA.

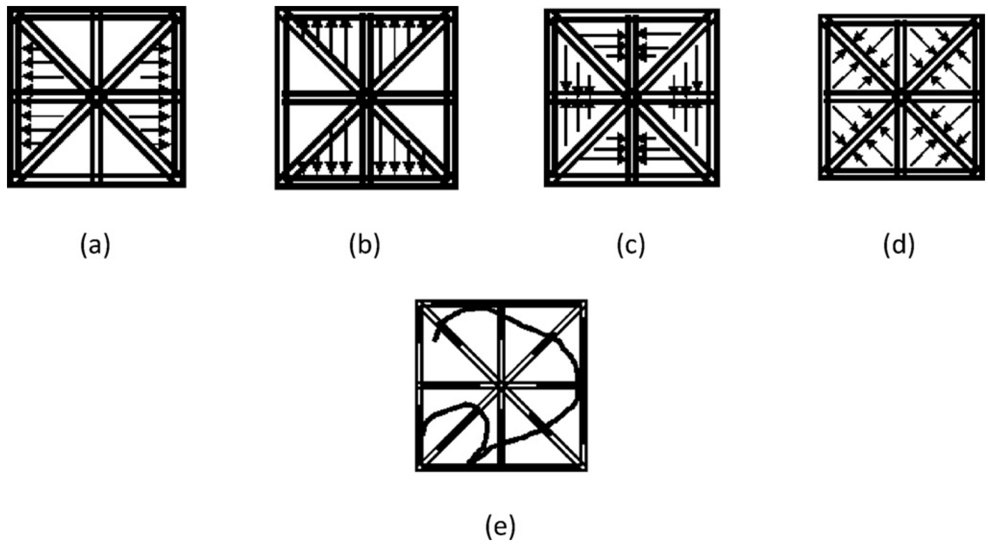


Fig. 5. An illustration for shadow features.

Depending upon the area of coverage on the digit image, features are classified as global and local. Global features are extracted from the whole digit image whereas local features represent the digit shape information within local regions. The total number of global features is fixed, but the number of local features varies as the number of local regions encoded by the genes of a chromosome of GA is not fixed. All these feature extraction methods assume that the digit image is a binary one. We have considered both global and local features so that the feature set should be capable of supplying complementary information about the digit patterns. The different types of features used here are described below.

2.4.1. Global feature set

It consists of 53 features in all. These features are formed with 24 modified shadow features [11], 16 octant centroid features [11], 8 distance based features [6], 1 feature representing the number of loops and 4 longest run features [11] computed on the entire digit image. For extraction of features, each digit image is so-scaled that the minimal bounding box enclosing it is of square shape. Descriptions of all those features are given in the following subsections.

2.4.1.1. Modified shadow features. Shadow features are computed by considering the lengths of projections of the digit image on the three sides of each octant dividing triangles within the minimal bounding box enclosing the digit image. Considering all the 8 octant dividing triangles of the minimal bounding box, 24 shadow features in all, are extracted from each digit image. Each value of the shadow features so computed is to be normalized by dividing it with the maximum possible length of the projections on the respective side. Fig. 5(a–e) illustrates the shadow features.

2.4.1.2. Octant centroid features. Coordinates of centroids of black pixels in all the 8 octants of a digit image are considered to add 16 features in all to the feature set. Each co-ordinate value is normalized by dividing it with the length of the side of the bounding square. An illustration of octant centroid features is shown in Fig. 6.

Table 1
Thresholding parameters used during the creation of windows.

Symbol	Purpose
δ_1	Defines minimum size of a window
δ_2	Defines maximum size of a window
δ_3	Defines maximum overlap among local regions
δ_4	Defines lower limit of image area coverage by the local regions

2.4.1.3. Distance based features. To compute the distance-based features we have partitioned the digit images in four quadrants. For each quadrant, maximum horizontal and diagonal distances from image boundary to character boundary have been calculated which are also normalized. Thus, for four quadrants, $2 \times 4 = 8$, distance features have been calculated in all. The features are shown in Fig. 7.

2.4.1.4. Longest run features. Four longest run features are computed row wise, column wise and along two major diagonals of a digit image/sub-image. For row wise longest run feature, we compute for each row, the length of the longest bar that fits consecutive black pixels along each row in a rectangular sub-image region. Then we calculate the sum of all those lengths as row wise longest run feature for the image. The three other longest run features are computed in the same way but along the column wise and two major diagonal directions within the rectangle separately. Each of these feature values are normalized by dividing it with a factor $M \times N$, where M and N represent the height and width of the digit image, respectively. An illustration of longest run features is shown in Fig. 8.

2.4.1.5. Loop feature. The region totally enclosed by a digit pattern or part of a digit pattern is called a loop. The number of loop(s) varies from one digit to another in *Bangla* script. We have considered the number of loop(s) in a digit pattern as a global feature. Very small

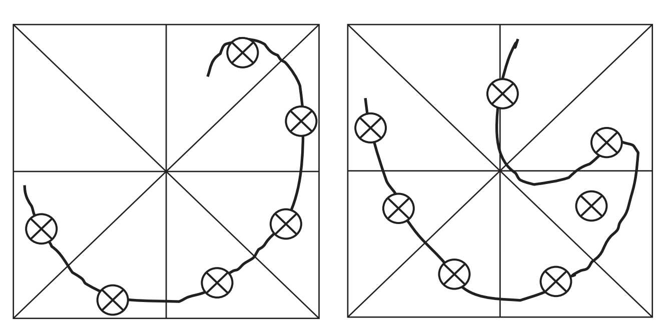


Fig. 6. An illustration of the 16 centroid features.

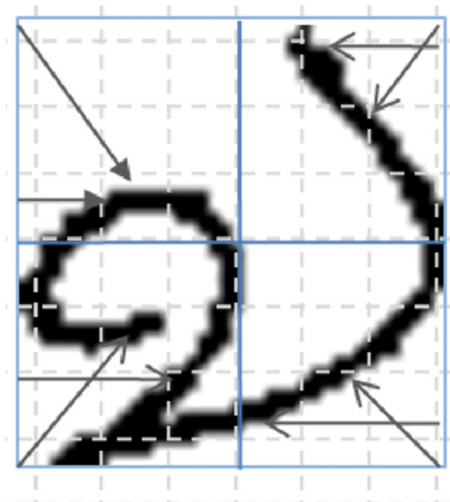


Fig. 7. An illustration of distance based feature.

					Length of Longest Run
0	1	1	1	0	3
0	0	0	1	1	2
0	0	0	0	1	1
0	1	1	0	1	2
1	0	0	1	1	2
1	1	1	1	0	4
					Sum = 14

Fig. 8. An illustration for computation of the row wise longest run feature for a portion of a binary image enclosed within a rectangular region.

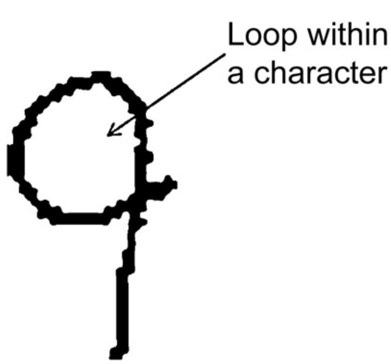


Fig. 9. Loop feature for *Bangla* digit 'seven'.

loops having less than 20 pixels are rejected as noise. Fig. 9 shows the existence of a loop for *Bangla* digit '7'.

2.4.2. Local feature set

From each of the local regions created following the criteria given in Section 2.3, 4 longest run features are extracted. Thus, if m number of local regions is created for a set, $4 \times m$ longest- run features are computed locally from each character image. All these extracted local features are not considered simultaneously for the recognition purpose because a subset of the local regions is selected by GA during experiments, which will be discussed in Section 2.5.

2.5. The SVM classifier

SVM [12,13] is primarily a binary classification methodology, which has been used for pattern recognition and regression tasks. In general, SVM uses structural risk minimization principle (SRM) [13] with structural learning theory to solve a problem. An SVM, employed as a binary pattern classifier, constructs an optimal hyper-plane for maximizing the margin of separation between the positive and negative data sets of pattern classes. Though SVM is mainly designed for binary pattern classification, but multi class pattern recognition problem can also be solved by combining several binary SVM classifiers. Among different regulations available for combining a number of binary SVM classifiers "one versus one" (OVO) and "one versus all"(OVA) are widely used among pattern recognition community. To solve k class problem, $k(k-1)/2$ numbers of binary SVM classifiers is constructed in OVO methodology. On the other hand OVA method requires k number of binary SVM classifiers to solve the same problem. None of the methods is completely superior to each other. For our case, we have used OVA for recognition purpose due to its lesser complexity over the other.

Suppose, a training data set T_D consists of pairs $\{(\mathbf{x}_i, \ \ y_i), \ \ i = 1, 2 \ldots n, \mathbf{x}_i \in \mathbb{R}^n$ and $y_i \in \{-1, 1\}$, where $\mathbf{x}_i$ denotes input feature vector for ith sample and y_i denotes the corresponding target value. For a given input pattern $\boldsymbol{x}$, the decision function of an SVM binary classifier is

$$f(x) = \text{sign}\left(\sum_{i=1}^{n} y_i \alpha_i K(x, x_i) + b\right) \tag{2}$$

where

$$\text{sign}(u) = \begin{cases} 1 & \text{for} \quad u > 0 \\ -1 & \text{for} \quad u < 0 \end{cases},$$

b is the bias, α_i is the langrage multiplier and $K(\mathbf{x},\mathbf{x}_i)$ is the kernel function.

The kernel function is used to map the input feature vector $\mathbf{x}$ into higher dimensional feature space to make them linearly separable. There are several numbers of kernels used in support vector machine models. Some of the popularly used kernel functions are shown below:

$$\text{Gaussian (radial basis function) kernel}: \ K(\mathbf{x}, \mathbf{x}_i)$$

$$= \exp(-\gamma^* ||\mathbf{x} - \mathbf{x}_i||^2) \tag{3}$$

where $\gamma = 1/2\sigma^2$ and σ is the standard deviation of the $\mathbf{x}_i$ values.

$$\text{Polynomial} \quad \text{kernel}: K(\mathbf{x}, \mathbf{x}_i) = (\mathbf{x}^T \mathbf{x}_i + 1)^d, \tag{4}$$

where d is the degree of the polynomial

$$\text{Linear kernel}: K(\mathbf{x}, \mathbf{x}_i) = \mathbf{x}^T \mathbf{x}_i \tag{5}$$

Out of these kernels, Gaussian Kernel that results into realization of the RBF network is an important one. The rationale behind the RBF network specially targets non-linearly separable patterns, which may be the most likely scenario for practical applications. Studies [14] have shown that RBF networks designed through support vector (SV) method can produce better recognition performances compared to those designed with traditional methodology for the same data set.

Radial basis functions (RBFs) [15] are traditionally used for performing multivariate interpolation in high dimensional space. The RBF neural network is an outcome of viewing the neural network design as a curve-fitting (approximation) problem in a high dimensional space. According to this view, learning is equivalent to finding a surface in a multidimensional space such that the surface provides a best fit to the training data. The criterion for "best fit" is determined on the basis of some statistical measure.

In an RBF network, Gaussian functions with different centres are used to map an input into the hidden space. One major problem of the RBF network lies with proper selection of the centres for the Gaussian functions and determination of the weight set for the output layer. All these parameters can be automatically determined when SV method is applied for realization of the RBF network with the Kernel functions of the SVM replaced with Gaussian functions. The data points selected as SVs from the training set, the product term $y_i\alpha_i$ s and the bias term b in Eq. (2) provide the centers, weight values and the bias for the RBF network, respectively.

2.6. Region sampling strategy using GA

GA [16], an intelligent optimization algorithm which is free from chances of sticking at local minima is used successfully for parameter pruning of neural networks [17], feature selections and parameter optimization of SVM in the OCR problems [18–20]. For the present work, GA is used for identifying the optimal set of local regions containing discriminating features for *Bangla* digit patterns. For systematic selection of local regions, each region is numbered uniquely for each set in this work. An optimal combination of these local regions usually does not contain all of them.

Each candidate solution or chromosome is encoded with an n bit binary string, in which each bit corresponds to one of the local regions in a particular set of local regions on a digit image. How many regions of an n bit chromosome string are activated, is decided randomly for forming the initial population. The bit in the chromosome corresponding to a region number has a value 1,if the region is selected for feature extraction; otherwise it has a value 0. Initially a population, also called generation, of candidate solutions is created by randomly generating 50 chromosomes for GA. In every subsequent runs of GA, the population size is kept fixed to 50 chromosomes. The value of n is decided during experimentations. Fig. 10 shows a population of 20 chromosomes of the GA for the Set#3 of local regions.

2.6.1. Design of Fitness function

The fitness of each chromosome of the initial population thus created is measured by computing the recognition rate of a SVM classifier on the test samples. Prior to that, the SVM classifier is to be trained with training samples. For both training and testing, besides 53 global features, longest run features, extracted from the selected local regions as encoded by that chromosome, are used as local features for each digit pattern. This process is repeated for all the chromosomes of the population.

2.6.2. Creation of next generation

A new population of more promising chromosomes is to be reproduced from the earlier population after evaluation of fitness values of all chromosomes of the population. In order to preserve the primary chromosomes we have introduced elitism by retaining 60% of the chromosomes of previous generation according to their ranks based on their corresponding fitness values. Rest 40% chromosomes are selected by using roulette wheel [16], after weighting proportionally with the fitness values of the chromosomes of the existing population. Thus, 50 new chromosomes are generated from the existing population. Once a new population of 50 chromosomes is created in this way, 80% of the chromosomes are randomly selected pair wise for performing crossover operation between them. The *crossover* point is selected at the middle of each chromosome here. Fig. 11 shows the *crossover* operation between two randomly selected chromosomes.

Once *crossover* operation is completed, the current population consists of 40 chromosomes already undergone crossover and the remaining 10 unchanged chromosomes. Half of the chromosomes of this population, i.e. 25 numbers are again randomly selected for *mutation*. The exact bit of a chromosome to be mutated is also selected randomly. The *mutation* operation for a chromosome is shown in Fig. 12. In this way, a new generation or population is reproduced from the old one, which is to be evaluated again.

2.6.3. The termination criteria

After a population is evaluated on the basis of the fitness values, the stopping criterion of GA is to be checked. In this work, the stopping criterion is reached either after 50 generations have passed or the average fitness value of the current population is greater than or equal to 99% of the maximum fitness value obtained so far.

2.7. Region sampling using simulated annealing and hill climbing

SA [21] and HC [22], two other popular optimization techniques, are also used for identifying the optimal set of local regions

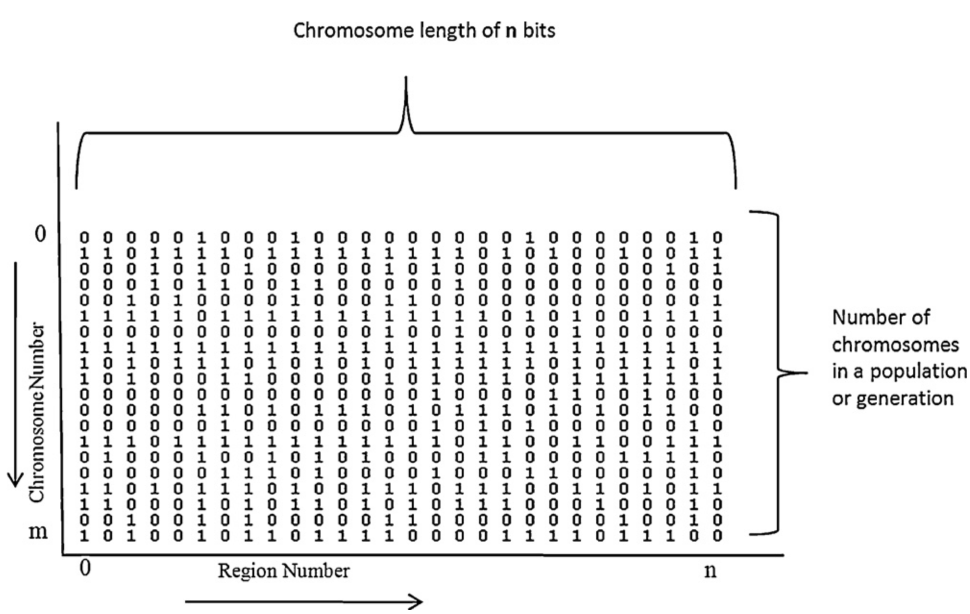


Fig. 10. Some sample chromosomes in a population of GA.

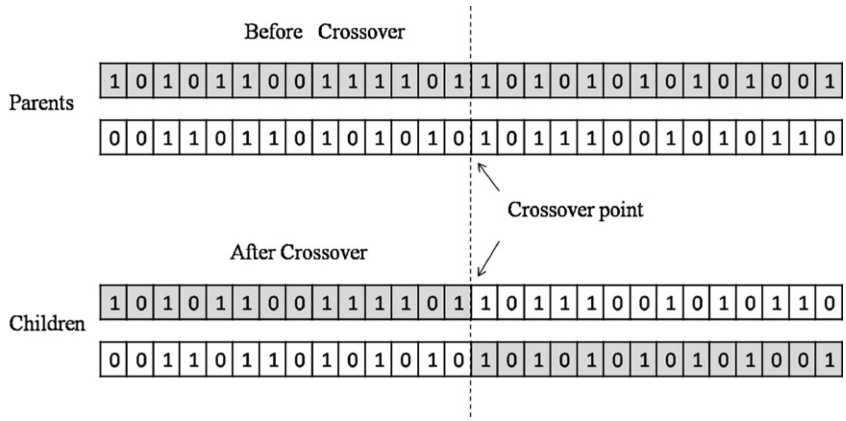


Fig. 11. Crossover operation between two randomly selected chromosomes.

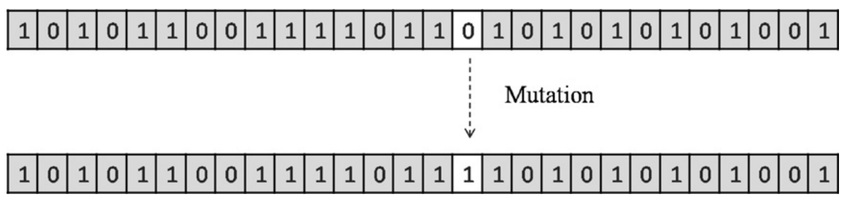


Fig. 12. Mutation operation for a chromosome.

containing discriminating features for *Bangla* digit patterns. SA is a probabilistic meta-heuristic search algorithm which uses the concept of metal annealing where it is possible to change the metal's internal structure at heated condition. The metal becomes more stable or creates resistance against change when it is getting cooled. Below a certain temperature it is not possible to modify further the metal's structure. For the present approach initial candidate solution is chosen randomly from the uniquely numbered local regions. The same fitness function as used in GA is used here to evaluate the performance of the candidate solutions. Each candidate solution is encoded with an n bit binary string, in which each bit corresponds to one of the generated local regions of a particular set on a digit image. In that string, a '1' in bit position k indicates that the region k is selected for feature extraction; otherwise the value is 0. Then, the neighborhoods of those local regions already selected in the candidate solution are repeatedly searched to get more improved fitness value. After completing a generation it creates a new generation with an acceptance probability value if the terminating condition is not achieved. The initial value of the acceptance probability and its decrement rate are so set that a maximum of 50 generations can be executed. The new generation's candidate string is also evaluated in the same way as discussed above.

HC technique is also implemented in a similar fashion. Here we used a randomly chosen n bit binary string as an initial candidate solution, where each bit represents a local region. The bit with value 1 is used to activate the corresponding region's features. The candidate string is evaluated in the same way as in GA. The next candidate solution is further used to achieve higher fitness value after mutating the initial n bit string.

3. Experimental results

3.1. Preparation of data set

To conduct experiments with the techniques described so far, a database of 6000 samples of handwritten *Bangla* digits is used. The database is formed by randomly selecting 600 samples for each of 10 digit classes from a larger database of 10,000 samples. The larger database is prepared in CVPR [23] unit, ISI, Kolkata and CMATER [24], Jadavpur University, Kolkata. Samples constituting the database of CVPR unit were collected from pin codes used on postal mail pieces. Samples prepared at CMATER Laboratory were collected on a pre-defined data sheet, from people of different age and sex groups as well as having different levels of education. Those data sheets were optically scanned with the resolution of 300 dpi using a HP F380 flatbed scanner to get their digital images. Each of the images were first preprocessed using basic operations of skew corrections, morphological filtering [25] and then binarized using an adaptive global threshold value computed as the average of minimum and maximum intensities in that image. Finally, the bounding rectangular box of each digit image was separately normalized to 32×32 pixels. After binarization, a '0' at a pixel position represents background and a '1' represents foreground. By dividing the database in 2:1 ratio, training and test sets are formed for evaluation of the present methodology. Thus, the training set and the test set, used for this work, consist of 4000 samples and 2000 samples, respectively. The database is freely downloadable from http://code.google.com/p/cmaterdb/as_CMATERdb_3.1.1. In order to evaluate the proposed method, we have used 7 different sets of variable sized local regions created in the experiments. We have already mentioned that variable sized local regions have some potential advantages over the fixed sized regions during classification of a pattern. But dependence on a single set of variable sized local regions may not help to draw a conclusion with reasonably high certainty. To overcome the problem, we have generated 7 different sets of variable sized regions using different region selection strategies described in Section 2.3. The sets are described below.

3.1.1. Set #1

The set consists of 200 randomly created regions without any overlap restriction.

3.1.2. Set #2

Here the whole image is divided into 4 equal parts and regions are created within each part maintaining the condition that the union of areas of all regions in a part must cover at least two-thirds of the area of that part. Total 26 regions are created in this way for each digit image. The major objective in this approach is to create regions from every zone of the digit. No overlap among the regions is allowed here.

3.1.3. Set #3

In this set, local regions are created randomly until the total area covered by the created regions is greater than two thirds of the area of the digit image. Here 25% overlap is allowed between two regions during their creation. In this way, 28 regions are created for each digit image.

3.1.4. Set #4

24 regions are created in this set following the same criteria of Set#3, excepting that no intersection is allowed between two regions.

3.1.5. Set #5

Here we have created 31 regions for each digit image considering the fact that total covered area of the regions is at least 80% of total area of the digit image, but there is no overlap between any two regions.

3.1.6. Set #6

Here, 28 local regions have been created for each digit image considering the fact that total area covered by the regions is at least 80% of the area of that digit image. Here 25% intersection is allowed.

3.1.7. Set #7

We have created here 16 fixed size overlapping regions covering the whole digit image, as used previously in [10].

A brief summary of these sets of local regions is given in Table 2.

4. Results and discussion

In the present work, we first designed a SVM based classifier for recognition of *Bangla* numerals using only the global feature set and obtained a recognition accuracy of 95.50% on test data set. For quantitatively analyzing the performance of a K-class classification problem, we have presented the classification results in the form of a confusion matrix. The confusion matrix C_K for a K-class pattern classification problem is presented as a $K \times K$ matrix which shows the classification results of the test pattern instances. The (i, j)th element of C_K shows the number of patterns belonging to ith class that are classified into jth class. The confusion matrix, corresponding to the recognition performance of the SVM based classifier on test data is shown in Table 3.

Next, we have applied GA, SA and HC based optimization techniques separately on each of the above mentioned local region sets (Set#1–Set#7) to obtain the optimal set of regions containing maximum discriminatory information for accurate recognition of *Bangla* digits. For GA, the initial population consists of 50 chromosomes. The lengths of the chromosomes are chosen as 200, 26, 28, 24, 31, 28 and 16 for the local regions Set#1–Set#7, respectively. The SVM classifier is used for evaluating the fitness values of the chromosomes. For this reason, one of the freely available open source SVM tools called LIBSVM [26], is used. LIBSVM uses Eq. (2) to design the SVM tool. Among different existing kernels in LIBSVM, we have used radial basis function (RBF) kernel, shown in Eq. (3),with gamma (γ) value 0.5. The rationale behind the choice of RBF kernel lies in its ability to perform better for handwritten digit recognition applications on the same dataset [27] compared to the kernel described in Eqs. (4) and (5). For HC, the initial population for Set#1 to Set#7 consist of a single candidate string generated randomly with maximum length of 200, 26, 28, 24, 31, 28 and 16, respectively. The activated bits of the string indicate the corresponding region's features are used for evaluation purpose with SVM based classifier. For SA, creation and evaluation of initial candidate solution is done in the same way as in HC. But SA does not stop after one generation. Rather it creates a new candidate solution for evaluation if the terminating condition is not reached.

With the seven different local region combinations (Set#1–Set#7), as mentioned above, we have run 50 variations of the GA, SA, HC. Table 4 shows the number of local regions selected to produce the optimum recognition accuracy for best 5 experimental runs (based on top recognition accuracies in

Table 2
Description of different region selection strategies.

Set #	Region selection strategy/set design strategy			Minimum Window size	Maximum window size
	Minimum image area covered	Maximum overlap between two regions	No of local regions created		
1	Not applicable	0	200	4	16
2	Image partitioned into four parts and 2/3 of each part is covered	0	26	4	16
3	2/3 area of total image	25%	28	4	16
4	2/3 area of total image	0	24	4	16
5	80% area of total image	0	31	4	16
6	80% area of total image	25%	28	4	16
7	100% of total image	50%	16	17	17

Table 3
Confusion matrix representing the recognition performance based on global feature set.

Class#	0	1	2	3	4	5	6	7	8	9
0	198	0	0	1	0	1	0	0	0	0
1	1	180	1	0	1	0	2	0	0	15
2	0	0	199	0	0	0	0	0	1	0
3	1	0	0	194	0	0	5	0	0	0
4	0	2	0	0	196	0	0	0	2	0
5	3	1	2	3	0	187	2	0	2	0
6	0	0	0	10	0	5	181	3	0	1
7	0	0	1	0	2	3	0	188	2	4
8	0	1	0	0	0	0	0	0	199	0
9	0	6	0	0	0	2	2	2	0	188

Table 4
Results of experiments with seven different local region combinations (Set #1–Set#7) for top five runs (among 50 runs) using GA, SA, and HC. Here Top#1 reflects the best performing experimental run in each set for each of the three techniques. Top#2–Top#5 show similar results in decreasing order of recognition accuracy.

	Set#	Total number of local regions	Top# 1		Top#2		Top# 3		Top# 4		Top#5	
			Optimal number of local regions selected for the run	Recognition accuracy (in percentage)	Optimal number of local regions selected for the run	Recognition accuracy (in percentage)	Optimal number of local regions selected for the run	Recognition accuracy (in percentage)	Optimal number of local regions selected for the run	Recognition accuracy (in percentage)	Optimal number of local regions selected for the run	Recognition accuracy (in percentage)
GA	1	200	14	97	2	96.95	5	96.9	22	96.85	10	96.85
	2	26	12	96.8	14	96.8	12	96.75	14	96.75	13	96.7
	3	28	18	97	17	97	22	97	23	97	15	96.95
	4	24	18	96.95	15	96.9	12	96.85	21	96.85	9	96.85
	5	31	16	97	18	97	19	96.95	19	96.95	15	96.95
	6	28	20	96.9	20	96.9	17	96.9	26	96.9	21	96.85
	7	16	11	96.7	8	96.7	11	96.7	9	96.65	11	96.65
SA	1	200	43	96.7	56	96.6	65	96.6	60	96.55	43	96.5
	2	26	25	96.45	23	96.45	13	96.4	15	96.34	22	96.25
	3	28	25	96.7	23	96.7	24	96.7	23	96.7	27	96.7
	4	24	16	96.65	18	96.6	17	96.55	14	96.45	22	96.45
	5	31	27	96.6	25	96.6	21	96.6	24	96.6	22	96.55
	6	28	21	96.65	23	96.65	27	96.6	26	96.6	22	96.55
	7	16	12	96.65	8	96.6	11	96.6	11	96.55	14	96.55
HC	1	200	67	96.5	80	96.5	45	96.5	80	96.45	70	96.4
	2	26	20	96.45	16	96.45	25	96.4	22	96.35	22	96.3
	3	28	27	96.7	17	96.5	21	96.4	22	96.4	21	96.4
	4	24	21	96.55	16	96.55	12	96.5	19	96.5	21	96.4
	5	31	26	96.65	17	96.5 5	12	96.5	30	96.5	13	96.45
	6	28	15	96.5	15	96.5	22	96.5	11	96.45	20	96.35
	7	16	14	96.6	13	96.55	12	96.55	10	96.5	15	96.5

Table 5
Averages and standard deviations of three different performance metrics, viz., recognition rate, number of selected local regions and number of fitness evaluations, are shown over fifty experimental runs for each of the seven sets corresponding to the techniques GA, SA, and HC. The highest average recognition accuracies, number of local regions and fitness evaluations for seven sets of the three techniques are highlighted in grey shades.

Performance metric	Technique	Set#1		Set#2		Set#3		Set#4		Set#5		Set#6		Set#7	
		Avg	Stddev	Avg	Stddev	Avg	Stddev	Avg	Stddev	Avg	Stddev	Avg	Stddev	Avg	Stddev
Recognition rate	GA	96.656	0.1672	96.498	0.1349	96.805	0.1282	96.601	0.1716	96.754	0.1417	96.659	0.1333	96.572	0.0642
	SA	96.3702	0.1235	96.1448	0.1037	96.592	0.0658	96.353	0.0839	96.416	0.1088	96.396	0.1256	96.375	0.1163
	HC	96.206	0.1475	96.123	0.1426	96.214	0.1549	96.132	0.1989	96.199	0.2014	96.136	0.1892	95.9958	0.219
Number of local regions selected	GA	30	14.8674	16.3	4.392	18.08	4.4038	15.84	3.546	18.34	3.9274	19.6	3.4814	10.74	1.6347
	SA	49.24	10.7844	20	3.677	21.78	2.6404	19.84	2.4929	21.96	2.3576	23.18	2.0069	13.4	1.3856
	HC	94.12	19.3408	20.02	2.6495	20.84	1.9012	20.14	2.1167	22.96	4.0986	20.22	3.6676	11.56	1.7794
Number of strings evaluated	GA	699	143.9941	628	97.5986	590.8	81.333	627	95.0799	643	103.8523	674	108.1212	583	51.0766
	SA	1069.48	224.0447	660.74	123.4013	639.58	142.5569	752.62	168.5595	745.68	176.4596	770.4	181.37	588.56	89.0173
	HC	135.2	19.949	20.52	1.9728	23.56	2.6654	19.94	1.4873	25.66	2.494	23.22	2.2254	12.44	1.1742

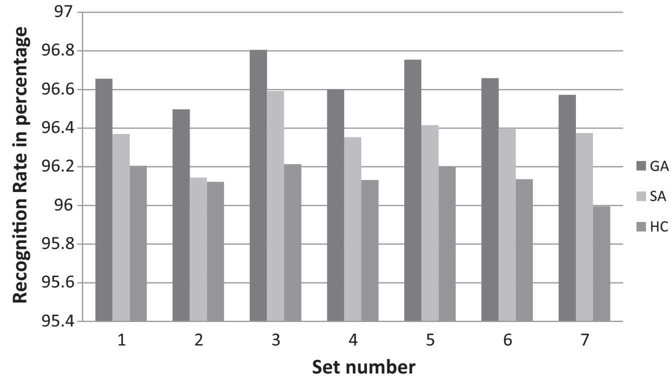


Fig. 13. Comparison of recognition accuracies achieved using GA, SA and HC.

descending order) for each of three methods. The average and standard deviations of the three performance metrics, viz., recognition accuracy, number of selected local regions and number of fitness evaluations, over 50 runs for each of the seven sets corresponding to the three optimization techniques are shown in Table 5.

From Table 4, it may be observed that recognition accuracies of all the three optimization methods are comparable across different sets. For GA, highest recognition accuracy of 97% is achieved for different combination of local regions in sets 1, 3 and 5. Again, from Table 5, it may be observed that Set#3 gives the best average performance across 50 runs of experiment, for all the three optimization techniques under consideration. The table also highlights the fact that the local region selection strategy of Set#3 is superior to the rest. The local-region selection strategy is stable in the sense that the average recognition accuracy over 50 runs (96.805%) is close to the maximum (97%) over all runs. From Table 5 and Fig. 13, it can also be observed that the average recognition accuracy achieved by GA is always better than the SA and HC based optimization techniques. Moreover, the number of local regions selected for each set is lesser in case of GA over other two methods. For example, Table 5 shows that in case of Set#3, GA requires an average of 18.08 selected regions in comparison to 21.78 regions for SA and 20.84 regions for HC, computed over 50 experimental runs. This is shown in Fig. 14. It may also be observed that GA always takes lesser number of fitness evaluations than SA. However, the comparison of the three techniques on the basis of this metric is not straight forward and needs additional insight. It is worthy to mention that GA is intrinsically parallel. During implementation of GA, we evaluated the fitness functions parallelly for all the 50 chromosomes of a single population. But SA and HC methods are purely sequential in nature. Again due to having *crossover* and *mutation* operations, GA is never trapped at local minima. On the other hand, SA and HC

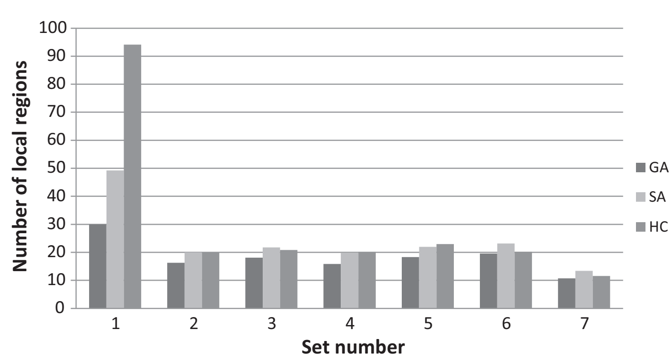


Fig. 14. Comparisons of the optimal number of local regions obtained with GA, SA and HC.

Table 6
Comparison of execution times for three different optimization techniques.

	Time (in s) required to execute the run with maximum success rate						
	Set#1	Set#2	Set#3	Set#4	Set#5	Set#6	Set#7
HC	205,114	13,180	5392	7475	6414	5308	1388
SA	313,858	256,598	251,206	244,861	267,974	246,725	233,031
GA	96,300	3912	3874	4047	3897	3782	3332

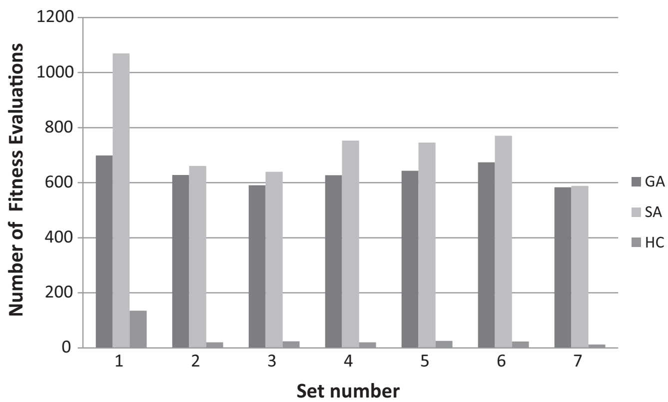


Fig. 15. Comparisons of the number of fitness evaluations for GA, SA and HC.

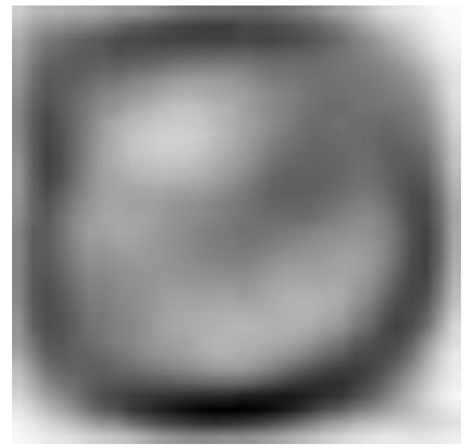


Fig. 16. Grey level template of Bangla digits.

methods may get stuck at local minima. Due to parallelism in GA it is possible to explore more search space easily and number of selected local regions may decrease here in different generations due to *crossover* and *mutation* operations. But for SA it is not possible to decrease the number of local regions from the initial ones. A comparison of execution times of the three different methodologies to obtain best result is shown in Table 6. From that table, we observe that the time required to execute GA is much lower than SA but little bit higher than HC for the Set#7, where the number of local regions is only 16. But from Fig. 15 it is observed that average number of chromosomes/strings evaluated for HC using 50 runs is much lower than that of SA and GA. It is because HC performs a single fitness evaluation for each run. Therefore the top recognition rates as well as average recognition rates for each of the seven sets using HC is lower than the recognition rate achieved using GA. On the other hand, SA and GA both are set for execution of 50 generations initially. Although we have continued GA up to 50 generations, stopping criteria is reached within 20 generations. SA has to be continued for 50 iterations sequentially, with a different string of local regions for each generation. Fig. 15 also demonstrates that the average number of chromosomes evaluated to reach optimal recognition rate using GA is lower than that of SA for each set. The maximum difference is obtained for Set#1 where SA evaluated the chromosome consisting of 200 genes. Thus GA provides better results in terms of success rate, number of selected local regions and execution time. Therefore, we have selected GA as the final optimization technique for our case.

All the methods used here do not assure optimality even when it has been reached due to their probabilistic nature. It is therefore very much difficult to say that the obtained regions for a set are truly optimum. Moreover, as the local regions selected by different runs on the same set are not spatially identical, it is difficult to say which combination has highest discriminating power. Moreover from Table 4 it may also be observed that for the same set, different local region combinations are selected for the same optimal recognition accuracy. To resolve these ambiguities, up to 5 quality consensuses, based on the results of top five runs, shown in Table 4 are employed for GA. The corresponding results are shown in Table 7.

From Table 7 it may be observed that maximum recognition accuracy of 97.70% with 14 local regions has been achieved for the Set#3 in the quality 4 consensus. The result shows the significance of the local regions during calculation of optimal recognition performance. From Table 8 it can also be observed that for every set the number of local regions, required to obtain maximum recognition accuracies is lower than the number of local regions initially created for the set. The table also demonstrates that the recognition accuracy obtained with the optimized set of local regions is

Table 7
5 quality consensus result based on the optimal set of local regions selected by GA.

Set#	Total number of local regions	Quality 1		Quality 2		Quality 3		Quality 4		Quality 5	
		Number of local regions	Recognition accuracy (in percentage)	Number of local regions	Recognition accuracy (in percentage)	Number of local regions	Recognition accuracy (in percentage)	Number of local regions	Recognition accuracy (in percentage)	Number of local regions	Recognition accuracy (in percentage)
1	200	44	97.05	9	96.95	–	–	–	–	–	–
2	26	24	96.65	21	96.75	13	97	6	96.95	1	96.35
3	28	28	96.55	26	97	20	97.4	14	97.7	6	97.25
4	24	24	96.45	21	97	17	97.25	10	97	4	96.4
5	31	29	96.6	24	96.8	21	97	11	97.05	3	97.05
6	28	28	96.49	27	96.7	25	96.85	16	97	8	96.8
7	16	16	96.25	13	96.7	10	96.75	9	96.7	4	96.25

Table 8
Improvement in recognition accuracy and number of regions using GA.

		Set#1	Set#2	Set#3	Set#4	Set#5	Set#6	Set#7
Without GA	*Total number of windows*	200	26	28	24	31	28	16
	Recognition accuracy in percent	96.5	95.94	96.35	96.15	96.3	96.19	96.19
After GA	*Number of windows in the optimal set*	44	13	14	17	11	16	10
	Recognition accuracy in percent	97.05	97	97.7	97.25	97.05	97	96.75
Improvement using GA	*Reduction in number of windows*	156	13	14	7	20	12	6
	Enhancement in recognition accuracy	0.55	1.06	1.35	1.1	0.75	0.81	0.56

Table 9
Confusion matrix representing the highest recognition performance of the present method.

Class#	0	1	2	3	4	5	6	7	8	9
0	197	0	0	1	0	2	0	0	0	0
1	0	192	1	0	2	0	0	0	0	5
2	0	0	200	0	0	0	0	0	0	0
3	1	0	0	196	0	0	2	0	0	1
4	0	1	0	0	199	0	0	0	0	0
5	2	0	0	2	0	192	2	0	1	1
6	0	1	0	4	0	1	192	2	0	0
7	0	0	1	0	0	1	0	193	2	3
8	0	0	0	0	0	0	0	0	200	0
9	0	3	0	0	0	1	1	2	0	193

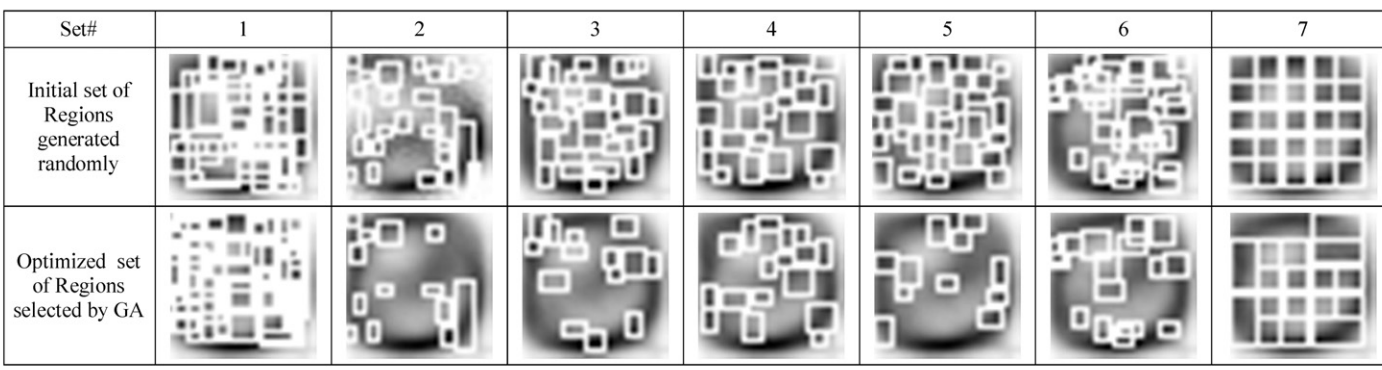


Fig. 17. Grey level templates with initial sets of local regions and optimal sets of local regions selected by GA.

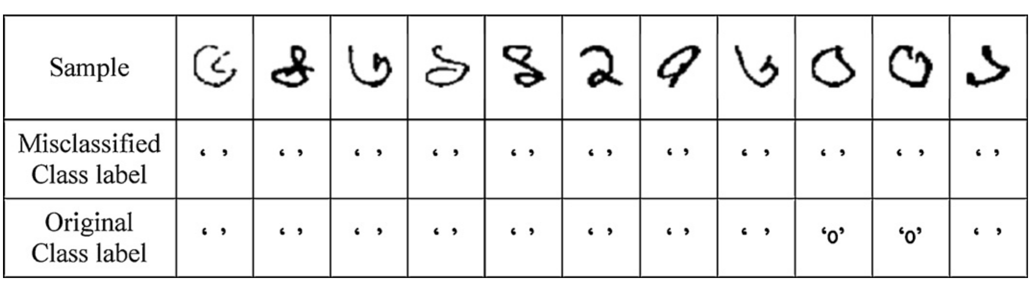


Fig. 18. Some examples of misclassified samples along with their misclassified and original class labels.

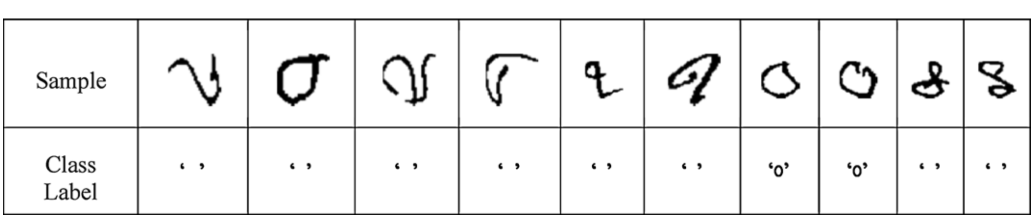


Fig. 19. Examples of bad data with their corresponding class.

Table 10
Comparative overview of different zone based methods on the present database.

Work reference	Number of samples taken for forming the database		Classifier/classification scheme	Recognition accuracy on the test set
	Training	Testing		
Basu et al. [3]	4000	2000	Multi-layer perceptron	96.65%
Basu et al. [7]	3600	400	Support vector machine	97.15%
Basu et al. [8]	4000	2000	Multi-layer Perceptron	96.45%
Das et al. [10]	2000	1000	Multi-layer perceptron	93.4%
Present work	4000	2000	Support vector machine	97.70%

higher than that obtained with all the local regions initially created for the respective set. The fact highlights the usefulness of region sampling.

To justify the result of our experiments, we have generated a grey level template T shown in Fig. 16, using the formula described in Eq. (1), by considering 200 samples each of the ten *Bangla* digit images from the training set.

From Eq. (1), it can be observed that if $\mu(x,y)$ is closed to 255, it signifies that the pixel (x,y) may be considered as a background pixel in the digit image. On the other hand, regions having $\mu(x,y)$ value close to 0 signify dark regions which indicate more commonality among the digit pattern shapes. The regions having intermediate $\mu(x,y)$ values are the regions where the digit patterns differ among each other and they generally contain discriminating information among the digit patterns. Fig. 17 shows the optimized regions selected over the template by the GA for different sets together with the corresponding initial set of randomly created local regions. It is to be noted that, for Set#1 only first 30 out of 200 initial local regions are shown for simplicity.

From Fig. 17, it may be noted that all the initial sets of variable sized local regions generated randomly do not cover all the regions of dissimilarities due to the stochastic nature of the region generation methodologies. Therefore, the optimal recognition performance achieved after execution of the present region sampling techniques using genetic algorithm varies with the different sets of local regions. From Fig. 17, it is also evident that for Set#3 the local regions initially created cover almost all the regions containing dissimilarity information. Therefore GA produces maximum recognition accuracy consistently for most of the runs for the Set#3; the result obtained by 5 quality consensus generates maximum recognition rate of 97.70% for the Set#3 only. The selected regions shown on the grey level template indicates that the regions containing discriminating information among the digit patterns are identified properly. Confusion matrix corresponding maximum recognition performance of 97.70% for Set#3 is shown in Table 9.

To compare the utility of considering features extracted from local regions besides those from global region, we have evaluated the recognition performance of SVM classifier on the test data using only the global feature set and corresponding confusion matrix of the process is shown in Table 3. Here, the maximum success rate obtained is 95.50%. It may be noted that the recognition accuracy obtained by using the global features only is lower than the highest recognition accuracy obtained by inclusion of features from optimal set of regions along with the global features.

It may be observed from Table 3 that digit pattern '১' (class 1) has been highly misclassified as digit pattern '৯' (class 9) due to having shape similarities among them in some regions. Similar explanation is also being applicable for classes 7 (digit pattern '৭') and 9 (digit pattern '৯') and classes 3 (digit pattern '৩') and 6 (digit pattern '৬'). If we go through the table showing the confusion matrix after implementation of region sampling technique, we notice that digit pattern '১' and '৯' are misclassified there also; but the number of mutual misclassification is 8 which is much lower than that in Table 3 where the number of mutual misclassification among '১' and '৯'are 21, i.e. 13 numbers of samples are properly classified due to inclusion of local regions' information. Misclassifications between digit patterns '৩' and '৬' are also decreased after inclusion of the local features. But it may be observed from Table 9 that one extra samples of class 0 (digit pattern '0') is misclassified as class 5 (digit pattern '৫') after inclusion of the optimal local regions' features. This happens due to having similarities in the selected local regions for digit patterns '0' and '৫'. A few misclassified test samples from the consensus 4 recognition result on Set#3 are shown in Fig. 18. Though, it is difficult to make any sort of conclusion from the recognition accuracies obtained by other zone based methods due to differences in databases, scripts and constraints on sample space, still we have presented a comparative overview of some of those methods using the same database as the present one in Table 10.

5. Conclusion

An effective scheme has been designed here to create variable sized local regions which may be used by appropriate region sampling techniques for pattern recognition problems. It has already been mentioned that fixed sized local regions may contain some ambiguous information besides the discriminating ones. But creation of proper sized local regions which contain exact discriminating information is a difficult task. Therefore one single set of randomly generated variable sized local regions is not good enough to make any conclusion. To overcome the problem, 7 different sets of variable sized local regions are generated using different region selection strategies in this work. To select an optimal subset of local regions containing high discriminating information about the pattern shapes from the above mentioned set, a GA based region sampling strategy has been employed. We have evaluated our methodology on handwritten *Bangla* digit samples from a database containing 6000 samples. By using GA, it has been possible to eliminate the regions of the digit patterns having no significant contribution on the recognition performance. With different variations of GAs and 5 quality consensuses it is possible to find out the local regions which are selected by most of the variations of GA. Result obtained by the present method outperforms those obtained by the previous methodologies [3,7,8,27,28] using the same database. We have used other region sampling techniques employing SA and HC techniques. It has been found that GA is most efficient among them (lower execution time with higher recognition rate). However, the recognition accuracy obtained here is lower than that in [29]. This may be attributed to the presence of a large number of poor quality handwritten digit samples in the test set. Some of the poor quality digit samples are shown in Fig. 19. However, inclusion of rejection criteria may improve the recognition performance.

The work may be extended to digits as well as characters of other scripts such as *Roman*, *Devanagari*, *Arabic*, etc.

Acknowledgements

Authors are thankful to the "Center for Microprocessor Application for Training Education and Research", "Project on Storage Retrieval and Understanding of Video for Multimedia" of Computer Science & Engineering Department, Jadavpur University, for providing infrastructural facilities during progress of the work. The work reported here has been partially funded by DST, Govt. of India, PURSE (Promotion of University Research and Scientific Excellence) Program. The author, Dipak Kumar Basu would also like to thank the AICTE, New Delhi for providing him the Emeritus Fellowship (F. No: 1-51/RID/EF(13)/2007-2008, dated 28-02-2008).

References

[1] M. Cheriet, H. Bunke, J. Hu, F. Kimura, C.Y. Suen, New frontiers in handwriting recognition, Pattern Recognition 42 (2009) 3129–3130.
[2] S.V. Rajashekararadhya, P.V. Ranjan, Efficient zone based feature extraction algorithm for handwritten numeral recognition of four popular South Indian scripts, Journal of Theoretical and Applied Information Technology 4 (2008) 1171–1181.
[3] S. Basu, N. Das, R. Sarkar, M. Kundu, M. Nasipuri, D. Basu, An MLP based approach for recognition of handwritten 'Bangla' numerals, in: B. Prasad (Ed.), 2nd Indian International Conference on Artificial Intelligence, Pune, India, 2005, pp. 407–417.
[4] J. Cao, M. Ahmadi, M. Shridhar, Recognition of handwritten numerals with multiple feature and multistage classifier, Pattern Recognition 28 (1995) 153–160.
[5] J. Park, V. Govindaraju, S.N. Srihari, OCR in a hierarchical feature space, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (2000) 400–407.
[6] N. Das, S. Basu, R. Sarkar, M. Kundu, M. Nasipuri, D.K. Basu, An improved feature descriptor for recognition of handwritten Bangla alphabet, in: D.S. Guru, T. Vasudev (Eds.), International Conference on Signal and Image Processing, Excel India Publishers, Mysore, India, 2009, pp. 451–454.
[7] S. Basu, N. Das, R. Sarkar, M. Kundu, M. Nasipuri, D. Kumar Basu, A novel framework for automatic sorting of postal documents with multi-script address blocks, Pattern Recognition 43 (2010) 3507–3521.
[8] S. Basu, N. Das, R. Sarkar, M. Kundu, M. Nasipuri, D. Basu, Recognition of numeric postal codes from multi-script postal address blocks, in: S. Chaudhury, S. Mitra, C. Murthy, P. Sastry, S. Pal (Eds.), Pattern Recognition and Machine Intelligence, Springer, Berlin/Heidelberg, 2009, pp. 381–386.
[9] S. Basu, M. Kundu, M. Nasipuri, D.K. Basu, A two-pass fuzzy-geno approach to pattern classification, in: International Conference on Computer Processing of Bangla, Dhaka, Bangladesh, 2006, pp. 130–134.
[10] N. Das, S. Basu, R. Sarkar, M. Kundu, M. Nasipuri, D.K. Basu, A soft computing paradigm for handwritten digit recognition with application to Bangla digits, in: International Conference on Modeling and Simulation, MS-07, Kolkata, India, 2007, pp. 771–774.
[11] S. Basu, N. Das, R. Sarkar, M. Kundu, M. Nasipuri, D.K. Basu, A hierarchical approach to recognition of handwritten Bangla characters, Pattern Recognition 42 (2009) 1467–1484.
[12] J.-x. Dong, A. Krzyzak, C.Y. Suen, An improved handwritten Chinese character recognition system using support vector machine, Pattern Recognition Letters 26 (2005) 1849–1856.
[13] V.N. Vapnik, The Nature of Statistical Learning Theory, 1st ed., Springer, 1995.
[14] B. Scholkopf, S. Kah-Kay, C.J.C. Burges, F. Girosi, P. Niyogi, T. Poggio, V. Vapnik, Comparing support vector machines with Gaussian kernels to radial basis function classifiers, IEEE Transactions on Signal Processing 45 (1997) 2758–2765.
[15] J.D. Powell, Radial basis function approximations to polynomials, in: D.F. Griffiths, G.A. Watson (Eds.), Numerical Analysis 1987, Longman Publishing Group, 1988, pp. 223–241.
[16] M. Srinivas, L.M. Patnaik, Genetic algorithms: a survey, Computer 27 (1994) 17–26.
[17] S. Shrivastava, M.P. Singh, Performance evaluation of feed-forward neural network with soft computing techniques for hand written English alphabets, Applied Soft Computing 11 (2011) 1156–1182.
[18] Y.-L. Wu, C.-Y. Tang, M.-K. Hor, P.-F. Wu, Feature selection using genetic algorithm and cluster validation, Expert Systems with Applications 38 (2011) 2727–2732.
[19] M. Soryani, N. Rafat, Application of genetic algorithms to feature subset selection in a Farsi OCR, World Academy of Science, Engineering and Technology (2006) 113–116.
[20] S. Sural, P.K. Das, A genetic algorithm for feature selection in a neuro-fuzzy OCR system, in: Document Analysis and Recognition, Proceedings, Sixth International Conference on, 2001, pp. 987–991.
[21] R. Meiri, J. Zahavi, Using simulated annealing to optimize the feature selection problem in marketing applications, European Journal of Operational Research 171 (2006) 842–858.
[22] B. Xi, Z. Liu, M. Raghavachari, C.H. Xia, L. Zhang, A smart hill-climbing algorithm for application server configuration, in: Proceedings of the 13th International Conference on World Wide Web, ACM, New York, NY, USA, 2004, pp. 287–296.
[23] Off-Line Handwritten Bangla Numeral Database, http://www.isical.ac.in/~ujjwal/download/database.html (accessed July 22, 2011).
[24] CMATERdb 3.1.1: Handwritten Bangla Numeral Database, http://code.google.com/p/cmaterdb/ (accessed July 22, 2011).
[25] D. Wang, V. Haese-Coat, J. Ronsin, Shape decomposition and representation using a recursive morphological operation, Pattern Recognition 28 (1995) 1783–1792.
[26] C.-C. Chang, C.-J. Lin, LIBSVM: a library for support vector machines, ACM Transactions on Intelligent Systems and Technology 2 (2011) 1–27.
[27] N. Das, B. Mandal, S. Basu, R. Sarkar, M. Kundu, M. Nasipuri, An SVM-MLP classifier combination scheme for recognition of handwritten Bangla digits, in: K.V. Kale, S.C. Malhrota, R.R. Manza (Eds.), 2nd International Conference on Advances in Computer Vision and Information Technology, vol. I, K. International Publishing House Pvt. Ltd., Aurangabad, India, 2009, pp. 615–623.
[28] S. Basu, R. Sarkar, N. Das, M. Kundu, M. Nasipuri, D. Basu, Handwritten Bangla digit recognition using classifier combination through DS technique, in: S. Pal, S. Bandyopadhyay, S. Biswas (Eds.), Pattern Recognition and Machine Intelligence, Springer, Berlin/Heidelberg, 2005, pp. 236–241.
[29] C.-L. Liu, C.Y. Suen, A new benchmark on the recognition of handwritten Bangla and Farsi numeral characters, Pattern Recognition 42 (2009) 3287–3295.