Handwritten Digits Recognition Using an Ensemble Technique Based on the Firefly Algorithm

Hamed Agahi*
Department of Electrical Engineering, Shiraz Branch, Islamic Azad University, Shiraz, Iran
agahi@iaushiraz.ac.ir
Azar Mahmoodzadeh
Department of Electrical Engineering, Shiraz Branch, Islamic Azad University, Shiraz, Iran
mahmoodzadeh@iaushiraz.ac.ir
Marzieh Salehi
Department of Electrical Engineering, Shiraz Branch, Islamic Azad University, Shiraz, Iran
m.artista09@yahoo.com

Received: 24/Feb/2018 Revised: 22/Aug/2018 Accepted: 16/Sep/2018

Abstract

This paper develops a multi-step procedure for classifying Farsi handwritten digits using a combination of classifiers. Generally, the technique relies on extracting a set of characteristics from handwritten samples, training multiple classifiers to learn to discriminate between digits, and finally combining the classifiers to enhance the overall system performance. First, a pre-processing course is performed to prepare the images for the main steps. Then three structural and statistical characteristics are extracted which include several features, among which a multi-objective genetic algorithm selects those more effective ones in order to reduce the computational complexity of the classification step. For the base classification, a decision tree (DT), an artificial neural networks (ANN) and a k-nearest neighbor (KNN) models are employed. Finally, the outcomes of the classifiers are fed into a classifier ensemble system to make the final decision. This hybrid system assigns different weights for each class selected by each classifier. These voting weights are adjusted by a metaheuristic firefly algorithm which optimizes the accuracy of the overall system. The performance of the implemented approach on the standard HODA dataset is compared with the base classifiers and some state-of-the-art methods. Evaluation of the proposed technique demonstrates that the proposed hybrid system attains high performance indices including accuracy of 98.88% with only eleven features.

Keywords: Classifiers Ensemble; Feature Extraction; Feature Selection; Firefly Algorithm; Multi-objective Genetic Algorithm; Optical Character Recognition.

1. Introduction

Written pattern recognition is fast becoming a key instrument in document processing in various languages. Investigating this issue is a continuing concern for several researchers to obtain fast and reliable optical character recognition systems (OCRs). Such devices aim to import information in printed or scanned documents into computers [1]. Recognition of optical characters has many applications in the real world, including checking passport documents, processing bank checks, sorting mail letters and automatic identification of license plates [2-4]. Handwritten digit recognition in different languages is considered as one of the most significant current discussions in the issue of OCRs. Recent developments in human life and automated industry have heightened the need for this technology. Surveys such as that conducted by Savas and Eldén [5] reported that the main difficulty in allocating observations to ten different classes of Arabic digits is due to high inter-class similarity and intra-class variability in such problems. Several attempts have been made to automatically recognize Western Arabic digits (i.e., 0, 1, 2, ..., 9) and good results have been obtained [6-9]; however, far less research has been conducted on the recognition of Farsi digits (i.e., 0, 1, 2, …, 9) and the reported accuracies of the existing techniques are lower than those for Western Arabic methods [10-14]. In Farsi language, research has consistently shown that due to the large similarity of the digits and also the wide differences in their drawing methods, creating a recognition system with acceptable accuracy and reliability has a number of problems in practical use. Like in Western Arabic, we face ten digits in Farsi and Eastern Arabic. Meanwhile, despite in Eastern Arabic which digits are written in one type, six digits in Farsi (i.e., 0, 2, 3, 4, 5 and 6) are described in two shapes (e.g., 6: ‘6’ and ‘6’). Such a high diversity, as shown in Fig. 1, makes it more difficult to identify Farsi digits. A recognition system, in general, contains three important steps of pre-processing, feature extraction and classification. First, in the pre-processing step some operations are performed to improve the images quality and prepare them for the main steps. Noise elimination, smoothing and normalizing the input data are some examples of such operations. Extracting features is the second step in which some characteristics from the image are extracted to constitute a feature vector assigned

* Corresponding Author

to that image. Numerous methods have been proposed to extract features from handwritten digits in different languages, including pixels density functions, geometric momentums, wavelet coefficients, projections and profiles on multiple orientations, and digit contours; see [15] for a general review of such methods. In the classification step, plentiful techniques may be recruited for recognizing handwritten digits and texts including for example the k nearest neighbor (KNN), the artificial neural networks (ANN), and the decision trees (DT). The highest proportion of the research performed on this subject concentrates on adapting the features used for digit classification. The regular features testified in the literature are extracted from writing samples and a traditional classifier like ANN is used to learn to distinguish between the handwritten digits.

This paper develops a hybrid system (also called the classifiers ensemble model) which combines the classifiers to better recognize Farsi handwritten digits. The performance of the proposed technique on a standard dataset is evaluated and some comparisons with the existing methods are presented. The approach proposed in this study contains multiple steps. In the first step, the pre-processing operations are carried out which include (i) digit shape separation and frame enfolding, (ii) inversion, (iii) resizing and (iv) thinning and removing inner pixels. These operations make the images ready for the next steps. In the second step, some characteristics from the image are extracted containing the ‘branch points’, the ‘chain codes’, and the ‘crossing counts’, each of which contain several features. All of these features constitute a single feature vector allocated to that image. The classification is performed by discriminating the feature vectors. Using several features increases the computational complexity and the processing time of the OCR system. For this reason, selecting features with most discriminative properties is at the heart of every effective recognition system. Hence, in the third step we use a multi-objective version of the genetic algorithm called the ‘non-dominated sorting genetic algorithm II’ (NSGA-II) [16]. The feature selection task is typically considered as a single-objective optimization (SOO) problem. While SOO considers only one objective function to be improved, multi-objective optimization (MOO) tries to concurrently enhance multiple objective functions. In fact, MOO generates a set of trade-off answers, among which the designer may choose one answer depending on the desires of the problem. The literature demonstrates that for solving complicated problems, methods exploiting MOO commonly perform better than those make use of SOO methods [17]. For our problem, this search metaheuristic considers the cardinality (number of members) of the selected subset and the F-measure of the classification using the ANN, as the two objective functions. The goal is that the first index is minimized while the second one is concurrently maximized. To this end, features are encoded in the form of a chromosome and the NSGA-II is applied. The final outcome is a Pareto optimal front that consists of a set of answers, each of which characterizes a different set of selected features. Finally, the specific answer of the Pareto front that returns the best accuracy is chosen as the vector of ideal features subset. Once the features are selected, only these features are taken out from new images to organize the feature vector and be fed into the classifiers. In the fourth step, the classification of the images is performed solely using the ANN, the KNN and the DT classifiers. These classification models yield different performance rates.

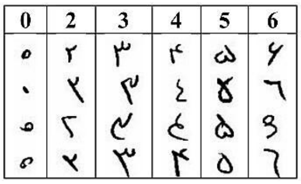


Fig. 1. Samples of Farsi handwritten digits with different shapes

Each classification system has pros and cons; thus appropriate combination of classifiers may strengthen the advantages and compensate for weaknesses of each classifier by others and provide a hybrid system with higher performance measures. The idea of the Multi-Classifier Systems (MCS), as a kind of Hybrid Intelligent Systems, was to take advantage of the individual classifiers to deliver classification systems that outperform these base classifiers. The idea of the MCS was first presented by Chow [18], who recommended conditions for ideal weighted mixture of binary classifiers. Dietterich [19] outlined the benefits of the MCS: (i) sweeping away the improper assumptions possibly caused due to small training dataset. (ii) Mixing classifiers that are trained by initiating from different starting values. This could help not to catch in local optimums. (iii) The correct decision making system may be unmanageable to be modelled by any single classifier, but mixture of classifiers may work. Consequently, we expect that a subtle mixture of the classifiers reaches higher performance measures. Subsequently, the outcomes of the mentioned base classifiers are mixed rather than considering the decision of the best classifier. Yet, the classifiers in the mixture should be selected as to be precise and diverse [20], meaning that their errors take place on different parts of the dataset. Due to its unsystematic nature and numerous neurons in hidden layers, MLP can be easily made diverse. As well, KNN yields the answers of the MCS for the patterns coming from the conflictive area of the search space. Using this classifier as a base contributor can considerably decrease the exploitation cost of the multi-classifier system [21]. The most important concern in the ensemble methodology is to find a correct method to mix the results of the classifiers. The majority voting and the weighted combination are the conventional procedures for mixing the classifiers [22]. While in the first technique, all the classifiers are mixed using the same weights; in the second technique, different weights are allocated to the classifiers. In the weighted method, the final conclusion and the overall recognition performance of the hybrid model severely depend on the weighting factors. In fact, all the donor classifiers may not be similarly capable of perceiving all the classes. For example, in a two-class problem (call classes C1 and C2), classifier A may be good at identifying class C1, while classifier B may be skilled at discovering class C2. Hence, the weights should be varied among the diverse classes for each classifier. Allocating different weights to the conclusion of a classifier about different classes increases the performance of the mixed classification coordination.

In the latest step, similar results are linearly mixed and then the maximization is accomplished to discover the final consequence. The weights of this recognition scheme are found using an optimization method. Random optimization algorithms are one of the approaches which are able to find the appropriate combinations of the classifiers, cf. [23-27]. In the paper the 'firefly algorithm' (FA) is used to mix the individual classifiers so that the recognition accuracy of the whole system is boosted. Firefly algorithm, proposed by Yang [24], is a metaheuristic search technique for the global optimization. This method finds the optimal solution with respect to the rules inspired by the movements of fireflies due to their attractiveness. This optimization process has become a progressively significant tool of swarm intelligence with numerous applications in almost all areas of optimization, as well as engineering practice. Various problems from many areas have been effectively solved using the firefly algorithm and its variants [28, 29]. Here, the mixing method based on the FA regulates the weights of voting for each class in any classifier by maximizing the accuracy of the final recognition outcomes as the objective function. Conclusions of classifiers about classes are encoded in the form of the locations of the fireflies. Accordingly, each entry in a location vector represents the voting weight allocated to the selection of every class by each classifier. The set of ideal voting weights is characterized by the location vector of the final answer found by the FA. This answer is related to the best accuracy attained by the fusion organization.

Briefly, the main novelties of this paper are: i) selecting the most effective features to reduce the dimensionality of the feature space by removing irrelevant, redundant or misleading features. This task also decreases the computational complexity and running time of the system while increases the classification accuracy. This operation is performed by a multi-objective GA. ii) Combining the outcomes of the base classifiers such that the hybrid system attains higher performance measures. For this purpose, a weighted combination approach is taken in which a specific voting weight is assigned to each classifier selecting each class. iii) For this purpose, the firefly algorithm, as a metaheuristic, finds the optimal weights such that the accuracy of the hybrid system is enhanced. The overall structure of the study takes the form of seven sections, including this introduction. Section 2 provides an overview on related work in the field of digit recognition. Section 3 begins by laying out the steps of the proposed recognition approach, including the pre-processing operations, feature extraction and selection, individual classification, and finally classifiers combination. The details and parameters settings for the proposed algorithm are described in Section 4. Then, Section 5 presents the results of applying the recognition system to the HODA dataset and measuring the performance indices. Section 6 performs discussion and comparison with some other techniques according to the standard performance criteria. Finally, Section 7 gives a brief summary and critique of the findings and includes a discussion of the implication of the results to future research into this area.

2. Related Works

In recent years, there has been an increasing interest in the recognition of handwritten digits. A considerable amount of literature has been published on this issue, some of which are referred to in this section. Soltanzadeh and Rahmati [30] found that utilizing outer profiles, crossing counts and projection histograms as features can result in acceptable accuracy values on the test samples provided in their research. Sadri et. al. [31] proposed a method for extracting a new attribute which considered four orientations for each digit and extracted sixteen features for each orientation. Finally a vector with 64 elements was given to the SVM to classify the image. The results of applying this method to the paper dataset showed the accuracy of 94.14%. Salimi and Giveki [32] suggested an ensemble of SVD classifiers to improve the system's performance. In their study, the combination of classifiers were performed using the PSO algorithm. In a paper by Ziaratban et al. [33], based on the template matching method a system for recognizing Farsi/Arabic handwritten digits was presented. In this approach, the patterns represented the pre-determined form of numbers. Khorashadizadeh and Latif [34] proposed a new method based on a feature set including directional chain code histogram and histogram of oriented gradient. This technique also utilized local features to improve the system performance by using 164 features. For this method, the SVM was used as the classifier.

Safdari and Moin [35] introduced a new method based on two-layer sparse auto-encoder and used the weights learned from the training phase for extracting the features. Hajizadeh et al. [36] proposed a new local manifold learning called FSLL, in which the locally linear embedding (LLE) and a Stochastic Laplacian Eigenmap (SLEM) are combined. This technique reduced the dimensionality of the feature space and represented the high-dimensional data manifold in low-dimensional space. Sadeghi and Testolin [37] presented a computational model of Persian character recognition based on deep belief networks. They emerged complex visual features in an unsupervised manner by fitting a hierarchical generative model to the sensory data. Zamani et al. [38] compared the performance of the random forest (RF) and convolutional neural network (CNN) for Persian handwritten digit recognition on the HODA dataset. They performed different experiments and finally showed that CNNs are the faster if appropriate hardware is available. It is worth mentioning that the techniques indicated above did not use the same dataset in their experiments. Although several techniques have been proposed on recognizing Farsi handwritten digits, results are not still as accurate as those achieved for Latin digits. Hence, finding a more accurate and reliable approach was the main motivation of this work.

3. The Proposed Approach

The hybrid system proposed in this paper aims to recognize Farsi handwritten digits in the HODA dataset [39]. Five main steps are carried out in our procedure consisting of pre-processing, feature extraction and selection, and finally individual and combined classification. The block diagram of the proposed system is shown in Fig. 2. In the following, the functioning of each block is described in details.

3.1 Pre-processing

Datasets used in the studies usually contain noises and incompatibilities due to their large size and also combination of several different resources. Using these data in the raw form leads to systems with unreliable results. Pre-processing is a step which plans to enhance the image quality and prepare it for the next actions. This phase significantly affects the performance of the main recognition steps. There are numerous pre-processing techniques for this purpose including blurring, histogram equalization and normalization [40]. This paper performs a particular course of operations to improve the efficiency of the recognition system including (i) segmentation and framing, (ii) image binary inversion, (iii) resizing, and finally (iv) thinning operation and inner pixels removal. In the following, the pre-processing stages are briefly outlined.

***i.* Image Segmentation and Framing:** The image of each digit in the HODA dataset is a binary image with no particular boundary who separates it from the rest of the images. Thus, first each digit image should be separated and then placed in a black frame [41]. An example of such image, which is manually performed, is shown in Fig. 3a which is surrounded by a rectangular black frame in Fig. 3b.

***ii.* Image Inversion**: In each binary image of the dataset, the digit shape is represented in white while the background is shown in black. An inversion operates by converting black to white and vice versa. This task is necessary for the next stages of the pre-processing [42]. The image inversion is shown in Fig 3c.

***iii.* Image Resizing**: Images in the HODA dataset have different sizes since they were taken from various resources. Having the same size is crucial when extracting the characteristics considered in this paper. For this purpose, the images of digits are resized according to a pre-specified size [42]. In this paper, the sizes of each image are changed to 46×46 as represented in Fig. 3d.

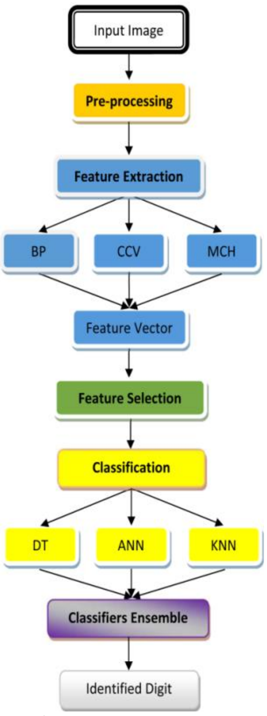


Fig. 2 Block diagram of the proposed hybrid system for recognizing Farsi handwritten digits.

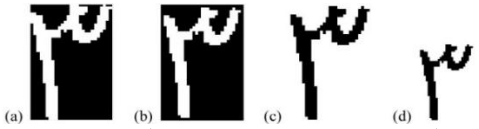


Fig 3 (a) A digit image from the HODA dataset [39], (b) framing, (c) inversion, (d) resizing

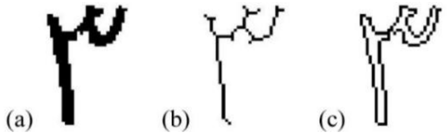


Fig. 4 (a) The original image, (b) the thinned image, and (c) the inner-pixel-removed image

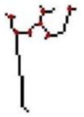


Fig. 5 Skeleton of a sample digit ‘3’ with its branch points represented by red spots; #BP = 7 [43].

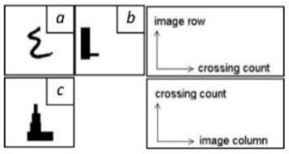


Fig. 6 (a) The original image, (b) VCCV=[1 1 1 1 1 1 2 1], and (c) HCCV=[1 2 3 4 3 2 1 1] [30].

3	2	1
4	**P**	0
5	6	7

Fig. 7. The mask centered at a pixel ‘P’ [41].

***iv.* Thinning Operation and the Inner pixels Removal:** To extract characteristics such as ‘branch points’ and ‘crossing counts’, the thinned image of the digits should be used. In the thinning operation, the skeleton of the digit image is extracted. To attenuate the noise effects, a median filter is applied and then the segments that are separate from the biggest connected segment are considered as disturbances and thus they are eliminated [30]. In this paper, the pixel-wise method is used for thinning the images. A sample image and its skeleton are illustrated in Fig. 4a,b.

Generating the 'chain codes' needs the digit shape boundaries to be found. Here, we use the technique of inner pixels removal to detect the edge points of the digit shape. For each pixel, four neighbor pixels are considered. If all four neighbors are black, the intended pixel is considered as an inner one and it is converted to white; otherwise, that pixel is an edge point and remains black [42]. This procedure for the image in Fig. 4a is shown in Fig. 4c.

3.2 Feature Extraction

In the second step, we investigate some characteristics of the images from which several features are extracted. These characteristics include 'the branch points' (BP), 'the crossing count vectors' (CCV) and 'the masked code histogram' (MCH). As the first feature, the 'number of the branch points characteristic' (#BP) in the skeleton of the digit image is found as a structural feature. A pixel is referred to as a *branch point* if at least three pixels in its 3×3 neighbor window (without considering the center pixel itself) are black. A digit image with its branch points are shown in Fig. 5 [43].

The next characteristics to be found are crossing counts which contain statistical features [30]. To find the horizontal crossing counts vector (HCCV), consider the first and the last non-empty columns of the image and the columns located within this interval. The HCCV is a vector whose length is equal to the number of columns in the mentioned interval and it is formed by setting any of its element as the number of the segments in the associated column of the digit image. In fact, each element represents the number of the connected segments in one column of the interval. The same approach is taken for the vertical crossing counts vector (VCCV) by considering rows instead of columns. The crossing counts vector in each orientation is normalized into a vector of size eight by carrying out the simple averaging or by up-sampling, when necessary. This normalization makes the features robust to the image stretch in the orientation of the crossing counts. Each element of the normalized HCCV and VCCV represents a single feature. The results for a sample digit is illustrated in Fig. 6.

Since we used only the skeleton characteristics, the outlier shape information may be lost. That is why we added a complementary characteristic; *i.e.*, the 'chain code' which consists of statistical features. This characteristic takes the boundary shape information into account. To do this, a 'mask' (see Fig. 7) is applied to each pixel on the boundary of the inner-pixels-removed image [41]. For any edge pixel, the black neighbor pixels in the 3×3 mask are weighted and summed up; the result is called a 'code'. When this computation is done for all the edge points, the histogram of the codes is determined. This feature vector is called the 'masked code histogram' (MCH). Finally, this vector is resampled into a vector of size 8, similarly to what happened to the crossing counts vectors.

Once the mentioned features are computed, the complete feature vector for each digit image will be generated by appending these feature vectors into one single vector containing #BP, HCCV, VCCV and MCH. Thus, a feature vector of size 25 will be obtained for each sample image (1 feature for the #BP, 8 features for the HCCV, 8 features for the VCCV, and 8 features for the MCH). It is important to note that the length of this vector is high. Accordingly dealing with such a vector is difficult and the computational burden will be high. Hence, in next step, by selecting more effective features, the feature vector length is reduced.

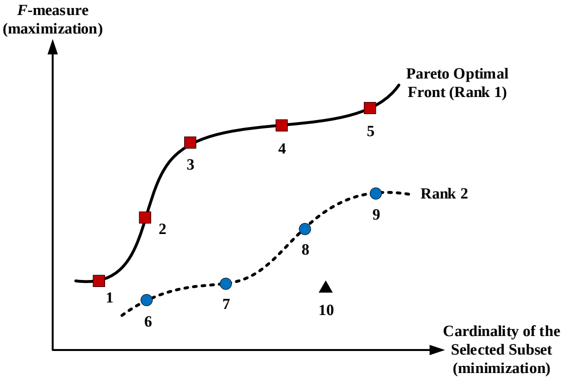
3.3 Feature Selection

Use of several features makes it more challenging to develop accurate classification models. From the practical viewpoint, using a large number of features leads to high computational complexity and large running time along with high risk of over-fitting and worsening the classification performance. Feature selection (FS) is a worthy approach to address these challenges. FS is the procedure of choosing a subset of significant features in order to make straightforward the model production and understanding and also to improve the model generality. Let m be the total number of features existing to pick from; then there exist 2^m possible feature subsets. Therefore, for large values of m (here, m=25), exhaustive search is impracticable. For the FS problem, many algorithms are proposed in the literature; see [44] for a review of the commonly used FS techniques. In this paper the genetic algorithm (GA) is used to find an optimal feature subset with large discrimination power. In fact, GA tries to remove redundant or irrelevant features. GAs are optimization methods based on the Darwin's principle inspired from the genetic reproductions. In this metaheuristic method, better populations among different species are developed during evolution. The GA presents an operative methodology for problems like FS, due to its capability of fast global search of large, non-linear and poorly understood spaces and also direct operations and low computational load [45].

In the process of selecting features by means of the GA, a population of chromosomes is considered each of which represents a candidate solution for this problem. Any chromosome is represented by a binary vector of length m. In the initial population, the genes of each chromosome (i.e., the vector elements) are randomly initialized to either 1 or 0. The value of '1' means that the corresponding feature is selected, while the value of '0' indicates that that feature is not chosen [46, 47]. The fitness of a chromosome is determined by evaluating some objective functions when an ANN classifier is applied to the test dataset. The input patterns of this classifier are represented by only the selected subset of features. If a chromosome has n ($n \leq m$) bits turned on, the associated ANN has n inputs. In this paper, a multi-objective genetic algorithm based on the ‘non-dominated sorting genetic algorithm II’ (NSGA-II) [16] is recruited for the FS purpose. The NSGA-II is a fast non-dominated sorting approach (NSA) with low computational complexity and an elitism approach. This technique is capable of dealing with multi-objective optimization problems [16]. For such tasks, the NSGA-II generates a set of sub-optimal solutions, among which one solution is nominated as the final chromosome depending on the desires of the problem. Accordingly, the designer has a large degree of freedom in selecting the final answer. For our FS problem, two objective functions (OF) are considered. OF1: The cardinality of the selected feature subset; OF2: The F-measure of the classification task using an ANN. The F-measure (also known as F_1 score) is a degree of a test’s accuracy, defined as the harmonic mean of the precision and recall indices of that test. Indeed, the precision and recall are united to form a single index by which the system performance could be generally evaluated [48]. The precise mathematical definitions of these measures are presented in Section 4. The NSGA-II plans to minimize OF1 while simultaneously OF2 is maximized. In each generation, numerous solutions are produced which NSA ranks them with respect to the concept of domination and non-domination relations in the objective function space, as shown in Fig. 8. Accordingly, a number of non-dominated solutions may be found on the final Pareto front due to multi-objective optimization. None of these results completely dominate the others. Some results have smaller subsets; while some are better in regard to the F-measure. To select the most suitable features for the FS problem, the system chooses, among the solutions in the final Pareto front, the solution with the best accuracy value assessed on the training set. The optimal subset found by the GA contains selected features that will be given to the classifiers of the next step. The procedure of the feature selection task by means of the GA is shown in Fig. 9.

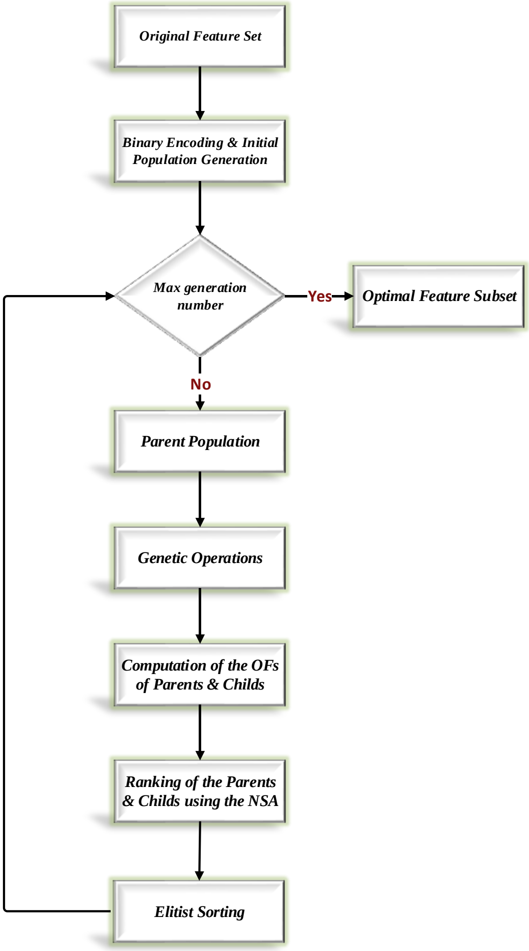
3.4 Classification Models

Classification is the main step in every recognition system. This task aims to determine each new pattern belongs to which of the known classes. Several algorithms are proposed for the classification of handwritten digits [15, 30, 31, 49]. In this paper, three classification models are individually used; *i.e.*, decision tree, k-nearest neighbor and the artificial neural network. Decision tree (DT) is a data mining technique which is widely used in classification and regression problems. As the name implies, this tree is composed of a number of nodes and branches. In a classification application, the leaf nodes represent the classes among which, one should be assigned to a query. To classify a query, the tree is traversed along a path from its root toward a leaf node. The path is decided by the subtrees chosen via the test answers in the internal nodes. No particular knowledge or parameter setting is needed to extract trees. Thus learning, deduction and decision making are straightforward and fast [50]. Depending on the application and the type of criteria, different decision trees may be utilized; the most famous ones to be noticed are ID3, C4.5, CART and CHAID [51]. This paper uses the CART decision tree as the first base classifier.



- Rank 1: Solutions 1,2,3,4 and 5 are non-dominating to each other.
- Rank 2: Solutions 6,7,8 and 9 are non-dominating but dominated by any one or more of Rank 1 solutions.
- Rank 3: Solution 10 is dominated by any one or more solutions from Rank 1 and Rank 2 Solutions.

Fig. 8. Representation of dominated and non-dominated solutions.



Original Feature Set
Binary Encoding & Initial Population Generation
Max generation number
Yes
Optimal Feature Subset
No
Parent Population
Genetic Operations
Computation of the OFs of Parents & Childs
Ranking of the Parents & Childs using the NSA
Elitist Sorting

Fig. 9. Block diagram of the optimal feature selection by GA

While most classification methods need precise models to recognize the input pattern, in some approaches it is possible to find the class of a query without creating models and just by comparing the query and samples in a dataset according to some similarity indices [52]. One of the most famous model-free methods is the '*k*-nearest neighbor algorithm' (KNN) which is frequently used for the pattern classification purposes. To assign a class label to a new input pattern, the algorithm first looks for a subset containing a *k* number of samples with the least distances to the query sample. Following this, the majority voting method determines the class with most number of samples in the nearest subset and allocates it to the input pattern. Commonly, the parameter *k* is adjusted after several experiments. The Euclidean and the Manhattan distances are mostly used as the similarity criteria in the nearest neighbor algorithms [52]. The KNN is the second base classifier used in this paper.

Artificial neural networks (ANNs) are computing systems vaguely inspired by the biological neural networks of human or animal brains. Such systems "learn" tasks from some examples, generally without any a priori knowledge about the patterns. The learning is realized only by evolving the set of characteristics of the ANNs from the learning material that they process. Some of the most appreciated neural networks are the multi-layer perceptron (MLP), the Hopfield network and the radial basis functions [53]. One of the simplest yet most efficient neural networks is the MLP network. MLP consists of an input layer, one or more hidden layers and one output layer. For the classification applications, the network should be "trained" using supervised methods. Backpropagation is a well-known algorithm for supervised learning of MLP networks. Given a MLP and an error criterion, this learning method computes the gradient of the error function backwards through the network [54]. This process continues for a certain number of iterations or it stops when an acceptable accuracy criterion is achieved. Once the MLP learned to identify the samples in the training set, it is ready to classify new patterns of the test dataset. The ANN (MLP) is the third classification model for the problem of recognizing digits of this paper.

It should be noticed that the main reasons for selecting these three classifiers is their high performance and simplicity when dealing with the classification problems. In addition, KNN is suitable for complex search spaces while DT has high speed in the classification problems. Moreover, ANN is capable of making diversity in the classifiers combination.

3.5 Classifiers Mixture

The idea of the classifiers mixture is to weigh several distinct classifiers, and mix them to acquire a fusion classification model that outperforms each of the base contributors. Given the potential advantages of the mixture methods, it is not unexpected that several methods are now accessible for theoretical researches and industrial applications [22]. Multi-Classifier Systems (MCS) try to mix some distinct classifiers and generate ensemble systems that give the final results. The following benefits of the MCS are commonly approved: (*i*) MCS act well both in the cases the data samples for learning are very limited and when an enormous number of them exist. (*ii*) The classifier mixture may outdo the best distinct base classifier. (*iii*) Several classifiers act on the basis of the heuristic search algorithms. Such procedures are not assured to find optimal answers. Accordingly, the mixture method, possibly initialing from different start points in the search space, might be considered as a multi-start local random search. This method may boost the probability of determining the optimal answer. (*iv*) MCS can be easily realized in parallel computer architectures or on distributed computing systems (*e.g,* Cloud computing). When a dataset is partitioned and the partial answer is determined on each partition, the final conclusion is made by mixing the networked consequences. (*v*) As stated by Wolpert [55], any classifier has its own competence domain; on which it exceeds other competing classifiers. As a result, a single classifier cannot be found that it beats every other one for all recognition problems. MCS attempt to generate an optimal hybrid model from the trained classifiers. The main concerns in the MCS design are: (*a*) The topology of interconnecting distinct classifiers; (*b*) Choosing valuable classifiers; and (*c*) Constructing the suitable decision fusion model (fuser) [22]. In our paper, we chose the evolutionary firefly algorithm as the fuser.

3.5.1 The Firefly Algorithm

Firefly Algorithm (FA), proposed by Yang [24], is a metaheuristic for finding the global solution in an optimization problem. This algorithm is inspired by flashing behavior of firefly insects for attracting other fireflies. Attractiveness of a firefly is relative to the brightness of the light it emits; the brighter one will be more attractive towards which the less bright ones are moved. Motivated by this process of bioluminescence, the FA updates the attractiveness and movement of any firefly in the population on every iteration. After running per-determined iterations, the firefly in the last population with the best fitness function yields the optimal solution. Decentralized decision-making structures in fireflies behavior and other natural species (Like ants, bees and birds), as examples of natural swarm intelligence, were inspiring to design of plentiful algorithms for solving complex issues such as optimization, multi-agent decision-making and robotics. The randomness characteristics of these algorithms avoid getting stuck in local optimums and helps to find the global solution for the problem [56]. Applying to several standard optimization problems validated that FA is more efficient than other meta-heuristic algorithms such as genetic algorithms (GA), simulated annealing (SA), particle swarm optimization (PSO) and differential evolution (DE) [57]. For this reason, we choose the FA for the problem of optimizing the voting weights of the MCS of this paper.

The FA is a parallel direct search technique which explores complex spaces to determine optimal answers for an optimization task. In this technique, a number of d parameters, whose optimal values are requested to be found, are encoded using some vectors called the 'locations of fireflies'. Each location is a nominee for the optimization problem, which is represented by a *d*-dimensional vector $\boldsymbol{x} = [x_1, x_2, \cdots, x_d]$. The collection of such vectors is called a *population*. The initial population of fireflies is generated at random locations in the search space. In the FA, a cost function $f(\boldsymbol{x})$ should be enhanced; the fitness of each firefly is characterized by this index. Three main assumptions are made in this algorithm: (*i*) all fireflies have the same sex; (*ii*) the attractiveness of each firefly is proportional to its brightness; and (*iii*) the brightness of each firefly at every location $I(\boldsymbol{x})$ is determined by the objective function of the problem at that location; *i.e.*, $I(\boldsymbol{x}) \propto f(\boldsymbol{x})$. Yet, the attractiveness β is relative and should be judged by other fireflies. Here β is adjusted as a proportion to the Euclidean distance between the i^{th} and the j^{th} fireflies (represented by r_{ij}). In the simplest form, the light intensity $I(r)$ is approximated using the following Gaussian form of the distance [58].

$$I(r) = I_0 . \mathrm{e}^{-\gamma r^2} \tag{1}$$

where $I(r)$ is the light intensity emitted by a firefly received to another firefly at the distance r, I_0 is the original light intensity and γ is the absorption coefficient. Since the attractiveness of a firefly is relative to the light intensity seen by adjacent fireflies, it can be defined as follows with similar definition in (1):

$$\beta(r) = \beta_0 . \mathrm{e}^{-\gamma r^2} \tag{2}$$

A firefly *i* located at $\boldsymbol{x}_i$ moves toward a more attractive firefly *j* at $\boldsymbol{x}_j$ ($I_j > I_i$) as described below:

$$\boldsymbol{x}_i^{new} = \boldsymbol{x}_i + \beta_0 . e^{-\gamma . r_{ij}^2} (\boldsymbol{x}_j - \boldsymbol{x}_i) + \alpha . \boldsymbol{\varepsilon}_i \tag{3}$$

Where the second term shows the attraction of the i^{th} firefly to the j^{th} firefly and the third one is a random term with a constant parameter α and a random vector $\boldsymbol{\varepsilon}_i$ with Gaussian or uniform distribution. The Pseudo code of the FA [58, 59] is brought here in order to the paper be self-contained:

```
Start
Determine the fitness function $f(\boldsymbol{x})$.
Generate the initial population of fireflies $\boldsymbol{x}_i$ , $i = 1,\ldots,n$
Define the brightness intensity $I_i$ at $\boldsymbol{x}_i$ by $f(\boldsymbol{x}_i)$
Determine the absorption coefficient $\gamma$.
Iterate the following steps to exceed the termination criteria:
        For all fireflies: (i= 1 to n)
            For all fireflies: (j= 1 to n)
                If $I_j > I_i$ :
                    Move firefly i toward firefly j using (3).
                End If.
                Evaluate new solutions and update the
                brightness using (1).
            End For j.
        End For i.
        Rank fireflies to find the current best one.
End Iteration.
Return the best result
End.
```

3.5.2 The Classifiers Ensemble Using the Firefly Algorithm

Although the three classification models mentioned in Section 3.4 are solely able to perform the classification task, combination of their decisions may improve the overall performance of the recognition system. The decision making based on the classifiers mixture is executed according to some weights allocated to the classifiers. In fact, for each classifier picking each class, a specific voting weight is assigned. A large weight shows that the classifier choice about that class is more assured and reliable. These weights are taken to mix the outcomes of classifiers to attain the final judgement.

Assume that the number of classifiers and classes are N and M respectively. The aim is to find the voting weights so that an objective function –*i.e.*, the accuracy of the overall system- is maximized. The weights are enclosed in a real matrix V of size $N \times M$; in which *V*(*n,m*) is the weight of the n^{th} classifier for the m^{th} class. In this paper, the FA considers vectors of length $D = N.M$ as the locations of the fireflies. The entries of each location in the population are randomly initialized to cover the search space. Here, we selected the *F-measure* as the index of the reliability of each classifier; the higher the *F*-measure, the more reliable the outcome of that classifier. Symbolize the *F*-measure of the classifiers for the training set by F_n, n=1,…,N. Consider each sample in the training set. The mixture result about the class of this sample is found using the weights of the classes of the classifiers. The weight for the n^{th} classifier is equal to F_n. The score of a specific class for a sample '*s*' is:

$$g(c_m) \ = \ \sum_{n=1}^{N} F_n * Q(n,m), \ s.t.\, op(s,n) = c_m, \ m = 1,\cdots,M \tag{4}$$

Here, $Q(n,m)$ symbolizes an entry of the firefly location associated with the voting weight of the n^{th} classifier for the m^{th} class. Furthermore, $op(s,n)$ characterizes the output class allocated by the n^{th} classifier to the sample *s*. In fact, only those classifiers that pick the m^{th} class are integrated in calculating $g(c_m)$. Finding m^* as $m^* = \arg Max_m \ g(c_m)$ yields the final conclusion for the class allocated to that sample in the training dataset. Then the accuracy of the classifier mixture for the training dataset is computed to be used as the cost function. FA attempts to determine an optimal location vector consisting of all entries $Q(n,m)$ that maximizes this cost function. Final results on the test dataset are reported using the classifier mixture associated with this best answer of the voting weights.

For each query pattern, its feature vector is entered to the three base classifiers to determine the class label. Then, the results are given to the ensemble system to make the final decision by combining outcomes according to the optimal voting weights. To clarify the ensemble approach, a simple artificial example is brought here with three base classifiers and two classes; thus the length of the location vector is 3*2=6. It is assumed that the optimal voting weights are found by the FA (shown in the third row of Table 1) along with the *F*-measure of each classifier on the training dataset. Now consider a query whose class label is needed to be determined. Suppose that the results for the first, second and third classifiers are the class labels A, B and B respectively. Then the score of class A is computed via multiplying the *F*-measure of the first classifier (0.98) in the weight of the first classifier/class_A (0.9). Similarly the score of class B is found by summing up two values: *i*) the second classifier *F*-measure (0.96) in the weight of the second classifier/class_B (0.2); and *ii*) the *F*-measure of the third classifier (0.9) in the weight of the third classifier/class_ B (0.3). Finally, the score of class B (0.466) is compared to that of class A (0.882). The class with the higher score (hear, the class A) wins, see Table 2.

Table 1. Voting weights and F-measure for an illustrative example

Classifiers	**First classifier**		**Second classifier**		**Third classifier**	
Classes	Class A	Class B	Class A	Class B	Class A	Class B
Class/classifier voting weights	0.9	0.3	0.7	0.2	0.8	0.3
F-measure of classifiers	0.98		0.96		0.90	

Table 2. Computation of the scores of the classes

First Classifier	Decision: Class A	Classifier score: 0.98*0.9=0.882
Second Classifier	Decision: Class B	Classifier score: 0.96*0.2=0.196
Third Classifier	Decision: Class B	Classifier score: 0.90*0.3=0.270
Score of the classes	Class A: **0.882**	Class B: 0.270+0.196=0.466

4. Simulation Details

Appropriate tests must be performed to answer to this question that “whether using the mentioned feature selection and classifiers combination methods can improve the efficiency of handwritten digits recognition system, or not?”. In this paper, the HODA dataset [39] is used to evaluate the proposed system. This dataset contains 80000 Farsi handwritten digit images with the resolution of around 200 dpi (dots per inch). Fig. 10 shows some examples of the HODA dataset. The proposed algorithm is run under the MATLAB R2014a programming environment on a PC equipped with 3.2 GHZ CPU and 8 GB RAM memory.

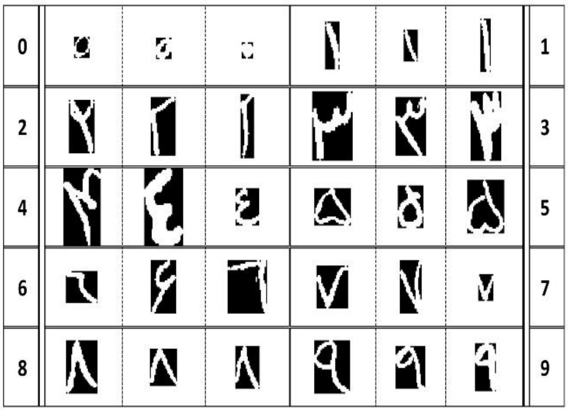


Fig. 10. Examples of the HODA dataset. Three example for each digit [39].

To assess the performance of our proposed method we outline the following experiments:

- Experiment 1: all features being fed into the distinct classifiers (no feature selection, no classifier mixture);
- Experiment 2: selected features entered into the individual classifiers (i.e., feature selection, yet no classifier mixture);
- Experiment 3: the classifiers mixture is applied to patterns with all features (no feature selection, but classifiers mixture).
- Experiment 4: The classifiers trained by the chosen features are mixed with the help of a voting weight methodology. These weights are determined by solving an optimization problem using the FA. The final conclusion is made based on the maximum score. (i.e., feature selection and classifier mixture). This experiment characterizes our proposed method.

The GA is employed to select the most discriminative features. The selection is carried out by the Roulette wheel method, and single-point crossover with the probability of 0.7, mutation rate of 0.2 and penalty coefficient of 0.5 are used in this paper. The population size is 30 and the number of generations is 50. The base classifiers in this paper are (1) the MLP with 20 neurons in one hidden layer, (2) the CART decision tree, and (3) the *k*- nearest neighbor by setting $k = 3$. In addition, the values of the parameters of the FA algorithm for finding optimal weights of classifiers combination are $\alpha = 0.02$, $\beta_0 = 2$, $\gamma = 1$; the values of these parameters are generally taken from [57]. Moreover, the population size is 20 and the number of iterations is 50.

To evaluate the system performance, the accuracy, precision, recall and the *F*-measure are used. These indices are defined according to the *TP*, *TN*, *FP* and *FN* values [60], as follows:

$$\mathrm{Pr}ecision(\mathrm{Pr}e.) = TP/(TP+FP)*100 \tag{5}$$

$$\mathrm{Re}call(\mathrm{Re}c.) = TP/(TP+FN)*100 \tag{6}$$

$$F-measure = (2*\mathrm{Pr}e.*\mathrm{Re}c.)/(\mathrm{Pr}e.+\mathrm{Re}c.)*100 \tag{7}$$

$$Accuracy(Acc.) = (TP+TN)/(TP+TN+FP+FN)*100 \tag{8}$$

Where *TN* is the number of negative truly recognized as negative; *TP* is the number of positive truly recognized as positive; *FN*, positive falsely recognized as negative; and *FP*, negative falsely recognized as positive. *F*-measure, defined in the interval [0,1], is the harmonic mean of the precision and recall and considers both rates in a single index. Values close to 1 is desired for the *F*-measure of a classification system. Furthermore, accuracy is the proportion of correctly classified samples from the total number of samples. To evaluate the results, the *k*-fold cross validation [61] is carried out. In this scheme, the total number of data is divided into *k* subsets. In each round, one subset is left out for the test and the classifier is trained using the rest. This process is repeated so that each subset is left out once. Lastly, the average of the cost function of all rounds is calculated to develop a more accurate estimate of the system prediction performance. In this paper *k* is set to 4 and the dataset is divided into four subsets, each containing 20000 samples. Accordingly in each round, 60000 samples are devoted to the training and the rest are left for the testing.

5. Simulation Results

Table 3 shows the performance measures for four mentioned experiments. The GA selected eleven most dominant features among the 25 features extracted from each image. The optimal feature subset includes 1 feature for #BP, 4 features from HCCV, 2 features from VCCV and 4 features from 8DCC. As shown in Table 3, almost all performance measures of the proposed method are higher than those of other experiments which demonstrate the advantage of selecting the most dominant features and also combining the classifiers results. Fig. 11 shows the confusion matrix for 10 digits when the proposed approach is applied to the HODA dataset. It is obvious from this figure that there are some digits more frequently misclassified. The major mistakes occurred for discriminating the digits ‘2’, ‘3’, ‘4’ and for discriminating the digits ‘0’ and ‘5’. This is caused by the fact that they are fairly similar in shape.

Table 4 shows some of the misclassifications of the proposed system, which are mainly due to poor quality of images or bad handwriting. Table 5 compares the proposed system with some other methods applied to the Farsi handwritten digit recognition problem with respect to the accuracy measure. The high performance of our method is due to selecting the most dominant features and utilizing diverse classifiers in the combination, and also because of applying the FA for finding the best voting weights in the classifiers ensemble.

6. Discussion

The method proposed in this paper presents a hybrid multi-procedure system for recognizing Farsi handwritten digits. Once the image pre-processing is performed, 25 features are extracted among which eleven ones selected by the two-objective GA are given to three base classifiers (*i.e.*, DT, ANN and KNN). These classifiers are widely used in recognition applications. Nevertheless, the existing literature shows that each classification model may outperform the others in different situations; which points to this fact that each technique has its shortcomings. This is the main motivation of this paper to integrate different classifiers in order to improve the accuracy and other performance indices. This combination is accomplished by assigning some voting weights to the contributor classifiers. The appropriate selection of these weights is critical to attain a more accurate recognition organization. The mixture with different weight values might return very different consequences. An unsuitable setting may lead to poor and erroneous classification algorithm, even worse than the distinct classifiers. The FA approach finds the weights of the classifiers according to their effectiveness so that a classification model with higher performance indices has a larger weight and thus a greater role to play and more discriminative information. Therefore, the inadequacy of each classifier is compensated by adequacy of other classifiers to obtain better classification measures. Our results show that the combination of classifiers through the FA ensemble technique is accurate and satisfactory and yields the classification accuracy of 98.88%. This rate is higher that the base classifiers acting solo.

From the results of Experiments 1 and 2 in Table 3, in which each classifier individually works, it can be seen that the ANN, outperforms the other two classifiers on both experiments with a best classification accuracy of 97.88%. The accuracy results of the KNN, as a lazy learner, are slightly smaller than those of the ANN. However, this classifier has an advantage of being computationally less expensive than ANN since it has no training phase. Although the DT performed worse than the other two classifiers, as shown in Table 3, still it is used to help other contributors in the hybrid system to increase the overall accuracy. It should be noticed that the accuracy values in Experiments 2 and 4 are greater than those of Experiments 1 and 3. The reason is that in the former experiments the most discriminative features are selected; while in the latter experiments all features are used which may reduce the generalization characteristics of the classification system or some features might be misleading. The feature selection technique reduces the computational cost and concurrently increase the classification accuracy. It is noteworthy to point out that simultaneously considering two performance objective functions -i.e., the *F*-measure and the cardinality of the selected features subset- for the feature selection problem is of great use to benefit from informative data.

Table 3. Comparison of performance measures for different experiments using 4-fold cross validation. The average values are shown in the table. Here, the following abbreviations are used: ‘Acc.’: Accuracy, ‘Pre.’: Precision, ‘Rec.’: Recall, ‘Fmea.’: F-measure. ‘TrT’: training time and ‘TsT’: testing time (Second). Also, ‘PrM’: The proposed method.

Experiment\Measure		Acc.	Pre.	Rec.	*F*mea.	TRT	TST
Experiment 1	*ANN*	97.63	96.61	89.15	92.73	0.136	0.0035
	DT	92.55	90.38	88.31	89.33	0.117	0.0031
	KNN	93.91	91.77	90.36	91.06	0.058	0.0010
Experiment 2	*ANN*	97.88	96.13	92.15	94.10	0.063	0.0023
	DT	93.18	91.18	89.36	90.26	0.049	0.0018
	KNN	94.78	90.78	91.19	90.98	0.021	0.0016
Experiment 3		98.02	97.11	94.23	95.65	2.835	0.0367
Experiment 4 (PrM)		**98.88**	97.52	93.75	95.60	0.424	0.0218

	1	2	3	4	5	6	7	8	9	0
1	98.43	0.64	0	0.07	0	0.16	0.29	0.18	0.04	0.3
2	0.42	98.48	0.66	0.29	0	0	0.05	0.06	0.09	0.28
3	0.06	0.52	98.62	1.21	0.13	0	0.05	0.04	0	0.06
4	0	0.36	0.47	98.14	0.16	0	0	0.05	0	0
5	0	0	0.16	0.17	98.59	0.04	0.05	0	0	0.57
6	0.21	0	0	0.08	0	99.57	0.1	0	0.29	0.09
7	0.13	0	0	0	0	0.16	99.31	0	0	0.22
8	0.16	0	0.03	0.04	0	0	0	99.67	0	0
9	0	0	0	0	0	0.07	0	0	99.51	0
0	0.59	0	0.06	0	1.12	0	0.15	0	0.07	98.48

Fig. 11 Confusion matrix for the 10-class problem of the proposed method on the HODA dataset (%). Columns show the input digits, while rows present the recognition results.

Table 4. Some examples of misclassifications of the proposed system

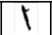
Handwritten digits										
True digit	0	1	2	3	4	5	6	7	8	9
Recognized digit	1	0	1	2	3	0	7	1	1	6

Table 5. Comparison of the proposed approach with some related methods based on the accuracy measure of the test data. Here, the following abbreviations are used: ‘Mtd.’: Method, ‘Acc.’: Accuracy, ‘TrD’: training data, ‘TsD’: testing data. Also, ‘PrM’: The proposed method.

Mtd.	TrD	TsD	Acc.	Mtd.	TrD	TsD	Acc.
[31]	7390	3035	94.14	[12]	6000	2000	97.10
[30]	4979	3939	99.57	[14]	60000	20000	98.84
[33]	6000	4000	97.01	[32]	1000	5000	97.02
[10]	60000	20000	98.71	[34]	60000	20000	99.31
[13]	6000	2000	95.30	**PrM**	60000	20000	**98.88**

In Experiment 1, ANN has the highest accuracy while its recall is less than that of the KNN. Similar conditions exist in some other experiments and models in Table 3. The main performance measure for evaluating the recognition systems of the paper is the accuracy. Handwritten digit recognition is a nonlinear and complicated problem. Thus, it should not expect that the behaviors and rates have a constant harmony. Another example contains the recall and *F*-measure of Experiment 4 which are less than those of Experiment 3, while converse condition holds for their accuracies. This is not unconnected to the fact that the mathematical relation of the *F*-measure contains the recall index. Hence when recall is small, the *F*-measure will also be small. Nonetheless, the system in Experiment 4 achieves greater accuracy with smaller number of features (eleven) while the system of Experiment 3 uses 25 features to obtain the performance indices mentioned in Table 3.

The running time of the experiments is also stated in Table 3. Experiments were executed on a PC (3.2 GHZ CPU, 8 GB RAM memory). The running time rests on the size of the training dataset, number of features to be given to the classifiers, number of the classes, etc. The testing time is very less in comparison to the training time. It can be seen from Table 3 that the systems based on the feature selection (Experiments 2 & 4) are faster than those use all features (Experiment 1 & 3), when examined in similar circumstances. It should be noted that when several classifiers are mixed using any weighted mixture procedure, the training time complication rises because of several runs wanted for discovering ideal voting weights. However, when these weights are established, the testing time comprises the time needed for any base classifier to deliver its outcome accompanied by the time for a simple decision making based on the weighted results. This testing time is trivial as shown in the last column of Table 3. Hence, using this collaborative attitude does not have much time complexity. On the other hand, the performance measures of the mixture methods are higher. The results in Table 3 validate the advantage of selecting the most dominant features in conjunction with reasonable mixing the classifiers results.

A number of researches have been reported on the recognition of Farsi handwritten digits, some of them are reported in Table 5. It deserves to be noted that the methods in Table 5 were assessed on different datasets with different image sizes and resolutions. The method of Soltanzadeh and Rahmati [30] is evaluated on their own dataset including 8918 high resolution (300 dpi) samples with a feature vector of length 257. They removed the incorrectly or unusually written digits to obtain a dataset with well-written numerals. In this paper, the HODA dataset is used which contains 80,000 samples with the resolution of 200 dpi. In work of Alaei et al. [10], Rashnodi et al. [14] and Khorashadizadeh and Latif [34] with the dataset same as that of this paper respectively 196, 154 and 164 features are used for the classification. Although our method achieved the accuracy of 98.88% which is a little smaller than some of the results mentioned in Table 5, it uses only eleven features while others utilize many more ones.

7. Conclusion

This paper proposed a hybrid multi-step procedure for the recognition of Farsi handwritten digits. First a set of pre-processing operations were performed on each digit image to make it prepared for the next steps. Following this, multiple structural and statistical features were extracted leading to a feature space with large dimensionality. For this reason, the multi-objective GA selected the most discriminative features to being fed to a decision tree, an artificial neural network and a *k*-nearest neighbor classifier. At the last step, the final decision about the digit class label was made by a classifiers ensemble system whose voting weights were found by the firefly evolutionary algorithm. The performance of the individual and combined classifiers were evaluated on the standard HODA dataset and compared with other existing methods from the literature. The proposed approach achieved Farsi handwritten digit classification with acceptable accuracy. The results of this research support the idea that the best classification performance could be obtained when the results of individual classifiers are combined into a single decision made by an ensemble classifier. Considering that different approaches suggested for this pattern recognition problem did not use the similar dataset, the precise comparison of the presented method with others is not possible. Though, due to the high recognition rate that is touched by the technique proposed in this paper, we can say that, to best of our knowledge, this system is at least one of the best procedures proposed until now for the recognition of Farsi handwritten digits.

For future work, the proposed method can be applied to the recognition of handwritten digits and characters in different languages and styles. In addition, using other features and different feature selection techniques (e.g., student’s t-test or PCA) coupled with other base classifiers (such as SVM) can be considered for generating and selecting dominant features and classifying the handwritten digits. More broadly, research is also needed to determine the effectiveness of other ensemble techniques (for example multi-objective PSO or Cuckoo Search) when dealing with different digit images. The main imperfection of this technique is that the computational complexity of the ensemble technique is naturally higher than each sole classifier since it needs all classifiers to be run and give their results to the ensemble classifier to make the final decision. Due to the complexity and high applicability of handwritten recognition, the truthful classification of digit patterns is crucial and of great importance in several technical and non-technical tasks.

References

[1] C.-L. Liu, K. Nakashima, H. Sako, and H. Fujisawa, "Handwritten digit recognition: benchmarking of state-of-the-art techniques," Pattern recognition, vol. 36, pp. 2271-2285, 2003.

[2] R. Al-Jawfi, "Handwriting Arabic character recognition LeNet using neural network," Int. Arab J. Inf. Technol., vol. 6, pp. 304-309, 2009.

[3] M. N. Ayyaz, I. Javed, and W. Mahmood, "Handwritten character recognition using multiclass svm classification with hybrid feature extraction," Pakistan Journal of Engineering and Applied Sciences, 2016.

[4] A. K. A. Hassan, "Arabic (Indian) Handwritten Digits Recognition Using Multi feature and KNN Classifier," Journal of University of Babylon, vol. 26, pp. 10-17, 2018.

[5] B. Savas and L. Eldén, "Handwritten digit classification using higher order singular value decomposition," Pattern recognition, vol. 40, pp. 993-1003, 2007.

[6] Y. Chen, Z. Xu, S. Cai, Y. Lang, and C.-C. J. Kuo, "A Saak Transform Approach to Efficient, Scalable and Robust Handwritten Digits Recognition," in 2018 Picture Coding Symposium (PCS), 2018, pp. 174-178.

[7] W.-S. Lu, "Handwritten digits recognition using PCA of histogram of oriented gradient," in Communications, Computers and Signal Processing (PACRIM), 2017 IEEE Pacific Rim Conference on, 2017, pp. 1-5.

[8] A. Boukharouba and A. Bennia, "Novel feature extraction technique for the recognition of handwritten digits," Applied Computing and Informatics, vol. 13, pp. 19-26, 2017.

[9] J. Qiao, G. Wang, W. Li, and M. Chen, "An adaptive deep Q-learning strategy for handwritten digit recognition," Neural Networks, 2018.

[10] A. Alaei, U. Pal, and P. Nagabhushan, "Using modified contour features and SVM based classifier for the recognition of Persian/Arabic handwritten numerals," in Advances in Pattern Recognition, 2009. ICAPR'09. Seventh International Conference on, 2009, pp. 391-394.

[11] M. Nahvi, K. Kiaee, and R. Ebrahimpour, "improvement the feature extraction method of Gradient based on the discrete cosine transform for recognizing Farsi handwritten digits," presented at the 18th Iranian Conference on Electrical Engineering, Isfahan, Iran, 2010.

[12] M. J. Abdi and H. Salimi, "Farsi handwriting recognition with mixture of RBF experts based on particle swarm optimization," International Journal of Information Science and Computer Mathematics, vol. 2, pp. 129-136, 2010.

[13] R. Ebrahimpour, A. Esmkhani, and S. Faridi, "Farsi handwritten digit recognition based on mixture of RBF experts," IEICE Electronics Express, vol. 7, pp. 1014-1019, 2010.

[14] O. Rashnodi, H. Sajedi, M. Abadeh, A. Elci, M. Munot, M. Joshi, et al., "Persian Handwritten Digit Recognition Using Support Vector Machines," International Journal of Computer Applications, vol. 29, pp. 1-6, 2011.

[15] D. Ghosh, T. Dube, and A. Shivaprasad, "Script recognition—a review," IEEE Transactions on pattern analysis and machine intelligence, vol. 32, pp. 2142-2161, 2010.

[16] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," IEEE transactions on evolutionary computation, vol. 6, pp. 182-197, 2002.

[17] U. K. Sikdar, A. Ekbal, and S. Saha, "MODE: multiobjective differential evolution for feature selection and classifier ensemble," Soft Computing, vol. 19, pp. 3529-3549, 2015.

[18] C. Chow, "Statistical independence and threshold functions," IEEE Transactions on Electronic Computers, pp. 66-68, 1965.

[19] T. G. Dietterich, "Ensemble methods in machine learning," in International workshop on multiple classifier systems, 2000, pp. 1-15.

[20] L. I. Kuncheva and C. J. Whitaker, "Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy," Machine learning, vol. 51, pp. 181-207, 2003.

[21] B. V. Dasarathy and B. V. Sheela, "A composite classifier system design: Concepts and methodology," Proceedings of the IEEE, vol. 67, pp. 708-713, 1979.

[22] M. Woźniak, M. Graña, and E. Corchado, "A survey of multiple classifier systems as hybrid systems," Information Fusion, vol. 16, pp. 3-17, 2014.

[23] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in Micro Machine and Human Science, 1995. MHS'95., Proceedings of the Sixth International Symposium on, 1995, pp. 39-43.

[24] X.-S. Yang, "Firefly algorithm, stochastic test functions and design optimisation," arXiv preprint arXiv:1003.1409, 2010.

[25] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, "GSA: a gravitational search algorithm," Information sciences, vol. 179, pp. 2232-2248, 2009.

[26] H. Duan and P. Qiao, "Pigeon-inspired optimization: a new swarm intelligence optimizer for air robot path planning," International Journal of Intelligent Computing and Cybernetics, vol. 7, pp. 24-37, 2014.

[27] S. Mirjalili, S. M. Mirjalili, and A. Hatamlou, "Multi-verse optimizer: a nature-inspired algorithm for global optimization," Neural Computing and Applications, vol. 27, pp. 495-513, 2016.

[28] I. Fister, I. Fister Jr, X.-S. Yang, and J. Brest, "A comprehensive review of firefly algorithms," Swarm and Evolutionary Computation, vol. 13, pp. 34-46, 2013.

[29] X.-S. Yang and X. He, "Firefly algorithm: recent advances and applications," arXiv preprint arXiv:1308.3898, 2013.

[30] H. Soltanzadeh and M. Rahmati, "Recognition of Persian handwritten digits using image profiles of multiple orientations," Pattern Recognition Letters, vol. 25, pp. 1569-1576, 2004.

[31] J. Sadri, C. Y. Suen, and T. D. Bui, "Application of support vector machines for recognition of handwritten Arabic/Persian digits," in Proceedings of Second Iranian Conference on Machine Vision and Image Processing, 2003, pp. 300-307.

[32] H. Salimi and D. Giveki, "Farsi/Arabic handwritten digit recognition based on ensemble of SVD classifiers and reliable multi-phase PSO combination rule," International Journal on Document Analysis and Recognition (IJDAR), vol. 16, pp. 371-386, 2013.

[33] M. Ziaratban, K. Faez, and F. Faradji, "Language-based feature extraction using template-matching in Farsi/Arabic

handwritten numeral recognition," in Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on, 2007, pp. 297-301.

[34] S. Khorashadizadeh and A. Latif, "Arabic/Farsi Handwritten Digit Recognition usin Histogra of Oriented Gradient and Chain Code Histogram," International Arab Journal of Information Technology (IAJIT), vol. 13, 2016.

[35] R. Safdari and M.-S. Moin, "A hierarchical feature learning for isolated Farsi handwritten digit recognition using sparse autoencoder," in Artificial Intelligence and Robotics (IRANOPEN), 2016, 2016, pp. 67-71.

[36] R. Hajizadeh, A. Aghagolzadeh, and M. Ezoji, "Fusion of LLE and stochastic LEM for Persian handwritten digits recognition," International Journal on Document Analysis and Recognition (IJDAR), vol. 21, pp. 109-122, 2018.

[37] Z. Sadeghi and A. Testolin, "Learning representation hierarchies by sharing visual features: a computational investigation of Persian character recognition with unsupervised deep learning," Cognitive processing, vol. 18, pp. 273-284, 2017.

[38] Y. Zamani, Y. Souri, H. Rashidi, and S. Kasaei, "Persian handwritten digit recognition by random forest and convolutional neural networks," in Machine Vision and Image Processing (MVIP), 2015 9th Iranian Conference on, 2015, pp. 37-40.

[39] H. Khosravi and E. Kabir, "Introducing a very large dataset of handwritten Farsi digits and a study on their varieties," Pattern recognition letters, vol. 28, pp. 1133-1141, 2007.

[40] R. O. Duda, P. E. Hart, and D. G. Stork, Pattern classification: John Wiley & Sons, 2012.

[41] S. Askari, M. KHarashadizadeh, and J. Sadri, "A new method for Recognizing Farsi handwritten numbers based on Pre-Classification," presented at the First Conference on Pattern Recognition and Image Analysis, Birjand, Iran, 2012.

[42] D. Deodhare, N. R. Suri, and R. Amit, "Preprocessing and Image Enhancement Algorithms for a Form-based Intelligent Character Recognition System," IJCSA, vol. 2, pp. 131-144, 2005.

[43] F. K. Zeyaratban M, Mozzafari S, Azvaji M. , "Presenting a New Structural Method Based on Partitioning Thinned Image for Recognition of Handwritten Farsi-Arabic Digits," in Third Conference on Machine Vision, Image Processing and Applications, 2005, pp. 76-82.

[44] A. Jović, K. Brkić, and N. Bogunović, "A review of feature selection methods with applications," in Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2015 38th International Convention on, 2015, pp. 1200-1205.

[45] C. Gunavathi and K. Premalatha, "Performance analysis of genetic algorithm with kNN and SVM for feature selection in tumor classification," Int J Comput Electr Autom Control Inf Eng, vol. 8, pp. 1490-7, 2014.

[46] A. Padma and R. Sukanesh, "A wavelet based automatic segmentation of brain tumor in CT images using optimal statistical texture features," International Journal of Image Processing, vol. 5, pp. 552-563, 2011.

[47] Y. Marinakis, G. Dounias, and J. Jantzen, "Pap smear diagnosis using a hybrid intelligent scheme focusing on genetic algorithm based feature selection and nearest neighbor classification," Computers in Biology and Medicine, vol. 39, pp. 69-78, 2009.

[48] O. Vechtomova, "Introduction to Information Retrieval Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze (Stanford University, Yahoo! Research, and University of Stuttgart) Cambridge: Cambridge University Press, 2008, xxi+ 482 pp; hardbound, ISBN 978-0-521-86571-5, $60.00," ed: MIT Press, 2009.

[49] M. Razavi and E. Kabir, "On-line Recognition of Farsi separate letters using the neural network. ," presented at the Third conference on machine vision and image processing, Tehran, Iran, 2004.

[50] L. A. Breslow and D. W. Aha, "Simplifying decision trees: A survey," The Knowledge Engineering Review, vol. 12, pp. 1-40, 1997.

[51] S. Singh and P. Gupta, "Comparative study ID3, cart and C4. 5 decision tree algorithm: a survey," International Journal of Advanced Information Science and Technology (IJAIST), vol. 27, pp. 97-103, 2014.

[52] C. C. Aggarwal and S. Y. Philip, "A general survey of privacy-preserving data mining models and algorithms," in Privacy-preserving data mining, ed: Springer, 2008, pp. 11-52.

[53] M. van Gerven and S. Bohte, Artificial neural networks as models of neural information processing: Frontiers Media SA, 2018.

[54] J. W. Shavlik, R. J. Mooney, and G. G. Towell, "Symbolic and neural learning algorithms: An experimental comparison," Machine learning, vol. 6, pp. 111-143, 1991.

[55] D. H. Wolpert, "The supervised learning no-free-lunch theorems," in Soft computing and industry, ed: Springer, 2002, pp. 25-42.

[56] X. Zhang, Y. Tian, and Y. Jin, "A knee point-driven evolutionary algorithm for many-objective optimization," IEEE Transactions on Evolutionary Computation, vol. 19, pp. 761-776, 2015.

[57] A. H. Gandomi, X.-S. Yang, and A. H. Alavi, "Mixed variable structural optimization using firefly algorithm," Computers & Structures, vol. 89, pp. 2325-2336, 2011.

[58] O. Bozorg-Haddad, M. Solgi, and H. A. Loáiciga, Meta-heuristic and evolutionary algorithms for engineering optimization vol. 294: John Wiley & Sons, 2017.

[59] X.-S. Yang, "Firefly algorithms for multimodal optimization," in International symposium on stochastic algorithms, 2009, pp. 169-178.

[60] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, Data Mining: Practical machine learning tools and techniques: Morgan Kaufmann, 2016.

[61] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in Ijcai, 1995, pp. 1137-1145.

Hamed Agahi received B.Sc. M.Sc. and Ph.D. degrees in Electrical Engineering from University of Shiraz, Amirkabir University of Technology and University of Tehran, Iran, in 2005, 2008 and 2013, respectively. From 2009, he was with the Islamic Azad University, Shiraz Branch, Shiraz, Iran. His research interests include pattern recognition, image and signal processing, and fault detection and diagnosis applications.

Azar Mahmoodzadeh received B.Sc., M.Sc. and Ph.D. degrees in Electrical Engineering from University of Shiraz, University of Shahed and University of Yazd, Iran, in 2005, 2008 and 2013, respectively. From 2009, she was with the Islamic Azad University, Shiraz Branch, Shiraz, Iran. Her research interests include pattern recognition and image and signal processing.

Marzieh Salehi received her MSc in telecommunications engineering from Islamic Azad University, Shiraz Branch, Shiraz, Iran. Her research interests are pattern recognition and Image processing for practical applications.