ORIGINAL RESEARCH

A Genetic-LVQ neural networks approach for handwritten Arabic character recognition

Abir Alharbi*

Mathematics Department, King Saud University, Riyadh, Saudi Arabia

Received: September 26, 2018 **Accepted:** November 8, 2018 **Online Published:** November 26, 2018
DOI: 10.5430/air.v7n2p43 **URL:** https://doi.org/10.5430/air.v7n2p43

ABSTRACT

Handwritten recognition systems are a dynamic field of research in areas of artificial intelligence. Many smart devices available in the market such as pen-based computers, tablets, mobiles with handwritten recognition technology need to rely on efficient handwritten recognition systems. In this paper we present a novel Arabic character handwritten recognition system based on a hybrid method consisting of a genetic algorithm and a Learning vector quantization (LVQ) neural network. Sixty different handwritten Arabic character datasets are used for training the neural network. Each character dataset contains 28 letters written twice with 15 distinct shaped alphabets, and each handwritten Arabic letter is represented by a binary matrix that is used as an input to a genetic algorithm for feature selection and dimension reduction to include only the most effective features to be fed to the LVQ classifier. The recognition process in the system involves several essential steps such as: handwritten letter acquisition, dataset preparation, feature selection, training, and recognition. Comparing our results to those acquired by the whole feature dataset without selection, and to the results using other classification algorithms confirms the effectiveness of our proposed handwritten recognition system with an accuracy of 95.4%, hence, showing a promising potential for improving future handwritten Arabic recognition devices in the market.

Key Words: Handwritten character recognition system, Feature selection, Pattern recognition, Handwritten characters, Arabic alphabets, Neural network, Learning vector quantization network, Genetic algorithms

1. INTRODUCTION

The process of designing an efficient handwritten character recognition system (HCR) is a challenging research area in the field of artificial intelligence. It confronts a need for quick advancement in the automation process of smart devices and it aspires to improve the man and machine interface. Several research papers have been focusing on new techniques and algorithms that would provide a higher handwritten recognition accuracy while reducing the processing time in smart devices such as cell phones, tablets, and pen-based computers.[1] With handwritten recognition systems there is also the issue of facing a diverse dataset containing noise from different individual hand style writings and using different instruments (pens, pencils, ink-pens...etc.), also the existence of non-character objects such as dots and dashes.[2,3] Most of the reliable recognition systems used in many languages such as English, Chinese, Japanese, and Arabic are either based on hidden Markov model (HMM), fuzzy set recognition systems or on artificial neural networks (ANN).[4–10] ANN have been successfully applied to many areas of pattern recognition especially in the field of handwritten character recognition and they have shown significant improvement in results over

*Correspondence: Abir Alharbi; Email: abir@ksu.edu.sa; Address: Mathematics Department, King Saud University, P.O. Box 22435, Riyadh, 11419, Saudi Arabia.

the conventional methods.[13,14] Most of the attempts were on the English handwritten characters' recognition, but there is limited research done related to the Arabic handwritten character and text recognition.[11,12] Arabic character recognition confronts many difficulties such as the cursive nature of Arabic and that there are many external objects added to some letters such as dots, dashes and other diacritic objects (ء , ـک, ٠, ً, ٓ, أ).[15–18]

The most aspect affecting the accuracy of the handwritten recognition system is the quality of the feature space selected to represent the handwritten input samples.[19,20] The size of the feature subset is important too, since having too many features can affect the efficiency of the learning algorithm especially when irrelevant, weak or redundant features are present.[21–23] Also, the feature subset must include all the important information on the diversity in style, shapes, sizes, which can be found in handwritings of different writers.[24–27] Reducing the dimensionality of the dataset is also crucial to the accuracy of the classification method and to the selection of the feature set.[28–30] Many techniques exist for feature selection and reducing the dimensionality of the dataset and they all fall under two main categories: either a selection method which selects features from the existing original dataset without any kind of transformation, or an extraction method which transforms the original dataset into a smaller feature dimension space, via any linear or nonlinear transformation.[31–33] The selected feature subset must be the best according to a specified measure, because this feature subset will define the whole search space to be investigated during the learning phase.[33,34] Generally, the size of the selected feature subset affects the computational cost of the classification process, so if we start with a subset of size M from an original dataset of size Z then the number of computations depends on these dimensions and there will be 2^M possible subsets of Z, which adds a large computational burden on the procedure.[21,35]

There are many search algorithms that attempt to generate feature subsets by selecting the feature that produces the highest significant improvement in a cost function. Since these algorithms do not consider complex interaction among several features, in many cases they lead to unsatisfactory optimal solution[21] such as the Greedy selection, the branch and bound methods, and the floating search techniques.[27,36,37] An alternative way to cope with such a search need is to use Genetic Algorithms (GA), Which have demonstrated to be an effective search tool for finding near-optimal solution in complex and nonlinear search spaces.[28,38,39] For this reason, GA based search strategies have been widely used to solve feature selection problem and hence handwritten character recognition problems.[26,40,41] Most of the approaches proposed in the context of handwriting recognition apply methods which reduce the number of features, while guaranteeing the recognition accuracy to be high.[13,18,42,43] Character recognition can be used for many applications such as: in automatic car number plate recognition devices, in converting handwriting inputs to a computer, in touch screen computers, in a blind's reading aid, and as a general improvement of the performance of many smart devices.[9] From the character recognition point of view, the Arabic character recognition is a hard problem, and various classical methods have been used.[15,17,44–49] Some of them have used neural networks such as Sarfraz et al.,[12] Mezghani N.,[10] Abed & Alasad,[51] Altuwaijri & Bayoumi,[43] Sarhan et al.,[52] and Haraty & Hamid.[49]

In the literature there were many interesting research papers on the subject such as Breukelen et al.[13] who introduced a classifier for the recognition of off-line English handwritten digits based on images captured by a phone-camera. An adaptive thresholding method was used to get all needed information, and for the classification phase three classifiers were used: SVM, Naïve Bayes and a multilayer perceptron. The database was created containing 3,380 samples collected from different volunteering writers. The features were extracted to reach a total of 54 features by 9 $\times$ 6 pixels. The system is tested and gave accurate result on 570 different samples of 26 alphabets. While in Abed & Alasad[51] a high accuracy Arabic handwritten character recognition system was used with an ANN based on error back-propagation, with a segmentation stage, which divided letters into the categories (initial, medial, final and isolated). In Bouslama & Amin[18] a structural analysis method and fuzzy approach were combined to produce a hybrid algorithm that employed these techniques, where structural analysis classified letter classes and fuzzy logic allowed for handwriting variability within the same class samples. In Sarhan et al.[52] an Arabic character recognition system using ANN and statistical analysis was employed, which exhibited error minimizing in an on-line recognition system for isolated Arabic characters. In El-Wakil & Shoukry[17] an on-line handwritten isolated Arabic characters' recognition system was used, and the number of letter classes was based on a template matching. Recognition accuracy varied with the length of the initial string, but the optimal string length gave an accuracy of 84% by testing 7 writers on sets of 60 characters and weighting the features manually according to their relative importance, results gave a maximum accuracy of 93%. In Alimi[42] an evolutionary neuro-fuzzy approach was used for a recognition system that classified letters from different hand-writing styles. In Hussein et al.[53] an optical character recognition of Arabic Handwritten character using Hopfield neural networks, they designed the system and trained with 8 Arabic characters. Elleuch et al.[54] introduced an Arabic handwritten character recognition system using deep belief ANN it was tested on a large dataset and recorded a high 98% accuracy. In El-Salwy et al.[55] a convolutional neural network CNN is used for off line Arabic character recognition. CNN is a supervised deep learning feed forward neural network mostly used for image recognition with large sets of high dimensional possibly non-linear data. CNN is a multilayer perceptron with minimal pre-processing's, shared weights and usually three types of layers; a convolutional layer, a pool layer and multiple non-linear hidden layers.[55] It mimics animal visual cortex and the overlap between the neurons makes sure all the visual fields are covered. Unlike other learning machines that needs to combine a feature extractor and a learning classifier, CNN does these procedures through its unique neuron/layer architecture design and it automatically extracts the important features, but it needs a large amount of input data to prevent over fitting.[56]

This paper presents a novel automated recognition system for isolated handwritten Arabic characters consisting of a combination between the learning vector quantization neural network (LVQ) and the genetic algorithm for feature selection. First, a genetic algorithm attribute selection algorithm is applied for the identification of the prominent feature, then the learning vector quantization network is used for recognition, as seen in Figure 1. The paper is organized based on the following section: Sections 2 and 3 describe the background of the proposed methods; genetic algorithms for feature selection and LVQ neural network for classification. Section 4 describes the preparation steps for the handwritten Arabic character dataset, and in section 5 the combined GA-LVQ method is presented. Finally, Sections 6 and 7 present our results and conclusions.

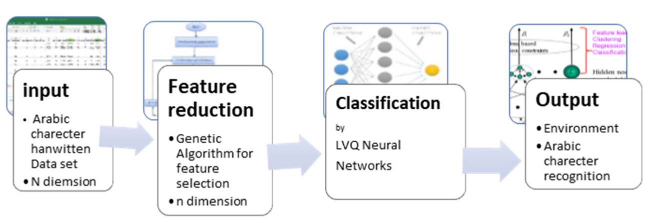

Figure 1. This Basic structure of a GA-LVQ approach

2. Genetic algorithm for feature selection

In pattern recognition the accuracy of the classification results depends greatly on the initial data feature set. Therefore, starting with an accurate extracted feature set is important in obtaining high performance. Defining appropriate features is a hard task since it involves work in data sets containing noise, redundancy and have high dimensionality.[31] Hence reaching an optimal feature set is not an easy task on the system. A good feature selection mechanism facilitates the classifier's job and eliminates noise, redundant and unwanted features.[24,41] It is preferred to reach a minimum sized set of features that still allows the classifier to choose pattern classes efficiently. One of these feature selection methods that processes the features according to their rank without overlooking the correlation between them is a heuristic search method and one of these heuristic approaches is the GA.[38] The genetic algorithm method is an iterative procedure that involves a population representing the search space for possible solutions to the given optimization problem.[57] An individual is a possible solution amongst these possible solutions in the population. Every individual has a chromosome, which is a bit encoding of the representative solution, generally, expressed in binary format.[57] Fitness is a characteristic measurement of how fit or good the individual is and is a sign whether the individual will be selected for the reproduction stage or not. The basic genetic algorithm proceeds as follows: an initial population of members is generated heuristically or at random.[36] Then, at every generation step, the members of the population are evaluated according to a fitness function that describes the given optimization problem in the search space. To form the new generation's population, members with high-fitness scores reproduce, while low-fitness members do not reproduce and hence gradually disappear.[26,58,59] This process simulates natural selection and evolution procedures, such as inheritance, selection, mutation, and crossover that influence the process of generating the best fit members throughout the evolving generations.[34,56,58,60] GA has been applied to feature selection problems in many research papers, such as by Siedlecki & Sklansky[38] where the GA was combined with a KNN classifier. The KNN classifier determined the classification performance of each feature set selected. Where a binary feature selection vector consists of a bit for each feature, with the bit = 1 indicating that the feature will participate in the KNN classification, and with a bit = 0 indicating that it will be omitted. The genetic algorithm searches for a feature selection vector with a minimal number of 1's, and a low error rate in the KNN classifier.[61] Later work by Punch et al.[20] expanded this approach to use the genetic algorithm for feature selection and instead of a selection vector consisting of only 0's and 1's, the genetic algorithm constructed also an associated weight vector, in which a real-valued weight is set to each selected feature. The classification accuracy obtained is then returned to the GA as a fitness score for this feature selection vector. Using this information, the GA searches a feature selection vector that maximizes classification accuracy while minimizing the dimensionality of the original dataset.[26,35] There are other feature extraction methods in the literature, two of the most popular are Linear discriminant analysis (LDA) and principle component analysis (PCA). PCA optimizes the transformation matrix by finding the largest variations in the original feature space. LDA pursues the largest ratio of between-classes variation and within classes variation when projecting the original feature space.[34] The advantage of these methods is their ability to reduce feature dimension however they have limitations that the decision boundaries generated are linear and cannot be relied on when nonlinearity is evident in the dataset, or when the classes have complex nonlinear decision boundaries.[32] Another major drawback in these independent feature extraction methods is the that the optimized criteria are different from the classifier minimum classification error criterion, which may cause inconsistency between feature extraction and the classification stages of a pattern recognizer. A direct way to overcome this is conduct feature extraction and classification jointly with consistent criteria, and that is how the GA for feature extraction is developed, because its feature extraction criteria depends on the classifier's score, and hence they are always aligned.[39] Another advantage with GA is its inherent nonlinear formulation which allows it to be well suited for higher dimension nonlinear complex feature spaces.[62]

Another important aspect in the feature selection procedure by genetic algorithms is the construction of the fitness function, which must measure the accuracy of the classification algorithm for the dataset.[8,27] The accuracy of the classification function can be estimated by calculating the percentage of patterns in a test set that are correctly classified. If Z is the number of original features in the problem, then every vector x in the population is of binary form, and a number 1 in the array indicates that the feature at this location is selected, and a number 0 indicates that the feature is rejected. Hence the proposed fitness function F(x) consists of two parts: the accuracy index indicated by the variable accuracy, which measures the number of correctly classified cases, and in the second part the cardinality of the subset. Hence, F(x) can be expressed as:

$$F(x) = \alpha \text{accuracy}(x) + \beta(M - m/M) \tag{1}$$

where M is the total number of features available, m is the cardinality of the feature subset represented by x (i.e. the number of bits equal to 1 in the genome), and the α and β parameters are weights to guide the minimization's priorities during the procedure. Our problem can be seen as a multi-objective optimization problem, where the fitness function guides the genetic algorithm to search for the optimum selected feature subset. The procedure runs until an optimum solution is reached, which will be the best selected feature vector subset satisfying the minimum of the fitness $F(x)$, measuring how well the patterns in the dataset are discriminated by this optimum selected feature subset.[21]

3. LVQ NEURAL NETWORKS

The learning vector quantization neural network algorithm is a nearest neighbor pattern classifier based on a competitive learning algorithm, which is mostly used for data classification with nonlinear separation. LVQ is made of two layers, the first is the clustering phase of the input vectors into clusters created during the training, and in the second layer the clusters are combined into classes defined by the target data.[62,64,66] The number of clusters is set according to the number of hidden neurons; the more hidden neurons exist in the network the more clusters the first layer can find. During initialization according to the distribution of target classes the number of first layer clusters are assigned to each target class. The architecture of the LVQ is shown in Figure 2, where R is the number of elements in the input vector, C is the number of competitive neurons and finally T is the number of linear neurons.[62] The linear layer is where the competitive layer's classes are transformed to target classification classes defined initially by the user. The classes created in the linear layer are called target classes while the classes in the competitive layer are called subclasses. The architecture of the LVQ is similar to that of the Kohonen self-organizing map, with a difference in the output unit's topological structure[32,64] As can be seen in Figure 2, when an input vector is given to the competitive layer it is first classified into a class as in the competitive layer of the self-organizing map NN.[65] The linear layer then transforms these classes of the competitive layer to target classes. The layers in the LVQ have one competitive neuron per class, and the linear layers have one neuron for each class. Hence, the total C subclasses learn in the competitive layer, and T target classes formed by the linear layer, must always satisfy T $\leq$ C.[66] In the competitive layer, each neuron is assigned to a class, and the number of classes in the competitive layer is determined by the number of neurons in the hidden layer, where the neuron learns a sample vector which allows it to recognize a region of the input space. Euclidean distance is the method used to measure the similarity between the input vector and any of the weight vectors around it.[66] The linear layer is then used to combine subclasses formed in the competitive layer into a single class, by a weight matrix created in the training process. Therefore, the main goal of LVQ is to find the output unit that has a matching pattern with the given input vector, similar to the self-organizing maps, where the winner unit is located. The weights are adjusted, such that if two input vectors belong to the same class then the weights move towards the new inputs vector, and if they do not belong to the same class then the weight is moved away from the input vector. The winner unit index is compared to that of the target vector and the weights are adjusted accordingly.[66,67]

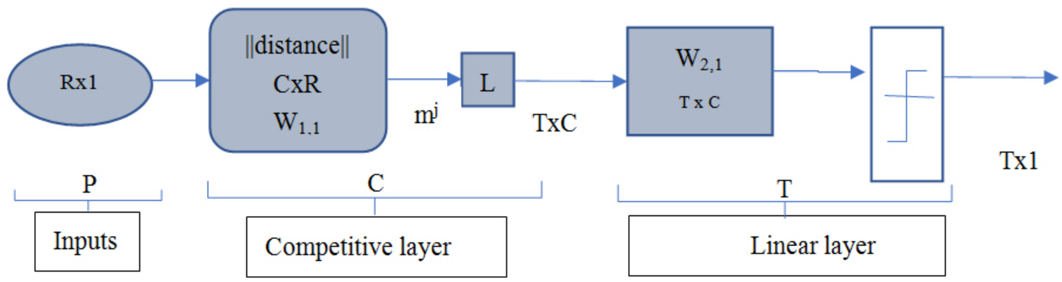

Figure 2. LVQ Neural Network structure

LVQ may be considered as neural network that allows the user to choose the number of training instances to keep and tries to learn their features.[67] LVQ acts like a collection codebook of vectors, or a list of members that have the same input and output features. So that each codebook is considered as a neuron and each feature is as a weight and this collection of codebook vectors makeup the architecture of the neural net. There are two reasons why LVQ is favored by researchers; for its heuristic simplicity, and for its direct adaptivity to the given classification problem.[69] LVQ is a competitive neural network which permits clustering in either a negative since by penalization or positive sense as a reward, and each input layer unit is connected to every output layer unit by feedforward connections. There are also inter-connections between all the output layer units by lateral inhibitory relations.[69,70] For each training vector the output layer unit will compete to look for the champion, such that only champion units can adjust its weight factor and using the learning rule of LVQ. The learning rate $\rho(t)$ used in the learning rule, satisfies: $0 < \rho(t) < 1$, and is decreasing with time to regulate the speed of the adjustments in the weight factor.[66] The main steps for the standard LVQ algorithm are as stated as:[68,69]

Step 1: Initialize the parameters of LVQ, $W_{i,j}$ (codebook) and ρ.

Step 2: Select an input vector x randomly.

Step 3: Find the champion which is nearest to the input layer vector:

$$\min \|x - W_{c,d}\| \tag{2}$$

Step 4: Modify the champion's unit weight factor using the learning equation.

$$W_{c,d}(t+1) = W_{c,d}(t) + \rho(t)[x(t) - W_{c,d}(t)] \tag{3}$$

Step 5: Reduce the learning rate ρ.

Step 6: Repeat steps 2 to 5 until the termination criterion is met.

The main advantage of LVQ neural net over traditional training methods like gradient-based algorithms or error back-propagation neural nets, is that it is much faster, has less parameters to initialize, and all hidden neurons are determined independent of the training data.[70] Moreover, LVQ can work with any bounded non-constant continuous piecewise activation function and does not have to deal with the possibility of overfitting the dataset, or being trapped in some local minima, or having to determine suitable learning rate.[71] In general, it is a much simpler algorithm with a higher generalization performance rate.[67,70]

4. The Arabic letter alphabet dataset

The Arabic language contains 28 letters and these letters are based on 15 distinct shapes that vary according to the either the following or preceding letters connecting to them. All 28 letters of the Arabic alphabet are constructed by adding some dots and symbols above and below these 15 shapes.[10,15,21] Our proposed GA-LVQ recognition system recognizes these 15 shapes or classes (see Figure 3). Arabic is written cursively, but some letters cannot connect to all the following letters, also there are no capital letters, and Arabic letters can have more than one shape when handwritten.[12,43] Other problems facing handwritten Arabic letter recognition systems include:

(1) A few letters in some fonts can be misread as different letters in another font. Therefore, in our dataset each letter has up to four different forms depending on its position in the text.
(2) Arabic letters can be handwritten with different heights and has different styles for connecting them with other letters, and that fact adds an extra burden on the sensitivity of the algorithm used.[5]

Figure 3. The 28 Arabic isolated letters, and the 15 classification classes

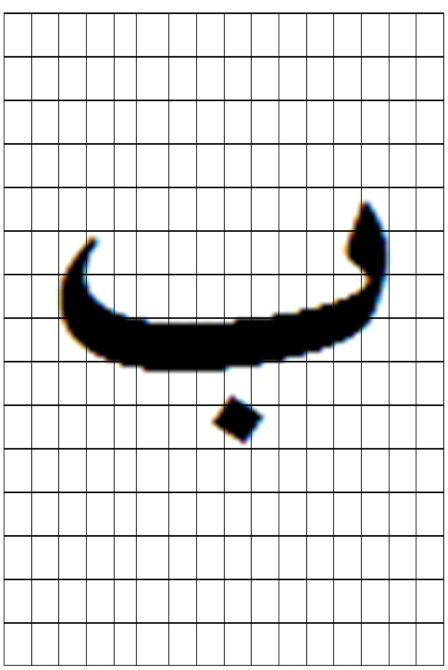

Figure 4. The handwritten Arabic letter “Baa” as a 16×16 matrix

In our dataset there are 1,680 inputs (rows) and 256 features (columns). Each isolated Arabic handwritten character is presented as a 16 by 16 binary pixels, making up an input vector of 256 elements. Figure 4 shows an example of the second letter in the Arabic alphabet “Baa”, which enters the feature selection stage as a binary vector of 256 elements. Hence, our dataset consists of 1,680 off-line isolated handwritten letters (using a regular pen on paper) from 30 persons who were asked to write each Arabic letter in the alphabet from the first letter أ to the last letter ي twice, and they were advised to write normally the first time so that it could be considered with accuracy, and the second time in a quick mode so that it would be considered without accuracy. Each image was scanned with a resolution of 256 gray scale after being stretched in a rectangle box of size 16 by 16, such that each pixel of each image was scaled into a Boolean (1/0) value using a fixed threshold, setting 0 to a pixel of value in the gray scale under 125, and setting 1 to a pixel whose value in the gray scale was over 125.

5. THE GENETIC-LVQ ALGORITHM

Here we give an overview of our hybrid method proposed for the isolated offline handwritten Arabic character recognition system. The system consists of two phases first a feature selection method, genetic algorithm, which selects the feature vectors that has the most important features for the recognition, and second the LVQ neural network learning algorithm for the classification phase and give a score to this selected feature vector subset. The selected features given by the genetic algorithm are fed to the LVQ neural network classifier, where the performance of the system is evaluated using the measure of accuracy and the number of features in the system. In our GA design the population size is fixed to 50 individuals, with the length of each genome set to 256 which is the number of features given in our handwritten Arabic character dataset. Hence, the experiment starts by finding from a population of random generated 50 genomes of length 256 bits where the xi entries are binary, and where a value 1 indicates the features to be kept and a value 0 for the features to be discarded. Hence, we encode the solution in the form of a binary bit string, each bit of the chromosome represents whether the feature represented by that bit is used or not. For example, if in a dataset with 6 features, there is a bit string denoted [111111], it would mean that all the features are used, and if the bit looked like [000001] it would mean only the last feature is used. The elite candidates for reproduction called parents are chosen according to their fitness. We use simple stochastic uniform selection method to select an individual from a population, based on their fitness, to participate in the reproduction phase (crossover). The crossover between the two chosen parents' genome is done at a single point randomly chosen with probability 0.8 to produce the new generation offspring. All data beyond that point in either genome string is swapped between the two-parent genome. Followed by an application of a probabilistic mutation step in which a 0.015% of bits of the solution strings are randomly flipped from zero to one or vice versa. The GA runs throughout the generations to find the best genome in this population which represents the best feature subset selection.[40] In other words, the best genome is the one, which classifies correctly the largest number of the 1,680 cases in the 15 Arabic character classes in the dataset and has the least number of features. Our fitness function used is F given in Eq.(1), which is computed from the percentage of cases incorrectly classified, and the cardinality of the subset features selected. Then our GA seeks the minimum of F. The stopping condition is a simple maximum number of generations run, post which the evolutionary process stops or when the increase in fitness of the best individual over five successive generations falls below a certain threshold, set at 5×10^{-7}.

The LVQ neural net is trained on the selected feature subset training dataset and then it is tested on the feature subset testing dataset. The LVQ neural net, has 20 hidden neurons in the competitive layer and 15 neurons in the linear layer, and with the learning rate starting with 0.01. The GA-LVQ algorithm follows the following steps:

(1) Initialize random population of individuals in the search space (binary strings of length 256 bits) representing all the possible solutions.

(2) For 300 generations or while the stopping condition is satisfied repeat the following steps:

   A. For each individual x in the population calculate the fitness F(x) by evaluating of the dataset with the respective feature as those represented by the chromosome x using the LVQ classifier.

   B. Using the stochastic uniform selection method select the most fit individuals for reproduction stage.

   C. Perform crossover on the selected elite individuals, then induce mutation in the offspring's produced.

When all generation have evolved and a continuous improvement in the fitness of the generations is observed, an optimum solution is reached which is a subset of selected features that produce the best classification of the given Arabic handwritten character dataset. The evolutionary experiments performed fall into the four learning categories, in accordance with the data partitioning into two sets; training set and testing set.[40] The four experimental categories are:

- Randomly assign data points to two sets d_1 and d_2, so that both sets are of equal size.
- Training set contains all 1,680 cases of the database, and the testing is done on the same training set.
- Training set contains 75% of the data cases, and the testing set contains the remaining 25% of the cases.
- Training set contains 50% of the database cases and the testing set contain the remaining 50% of the cases.

In the first category, we use a simple variation of k-fold cross-validation. For each fold, we randomly assign data point to two sets denoted d_1 and d_2, so that both sets have the same number of data (this is done by shuffling the data array before dividing it in two equal parts). We then train on d_1 and test on d_2, and vice versa. This has the advantage that our training and testing sets are both large enough, and each data point is used for both training and validation on each fold at the same time.[26] In the second category. We trained the system with all available data and then tested it with the same set of data. In the categories III and IV, the choice of training-set cases is done randomly and is performed at the outset of every evolutionary run. MATLAB was used to implement the GA-LVQ algorithm, and to generate the graphs of the results. A separate code for the fitness function is used during the GA procedure according to the fitness function given in Eq. (1) through the LVQ code (MATLAB lvq code).[63] Moreover, every testing process of a system is repeated 50 times and the mean and standard deviations are calculated for each experiment. A typical run for 300 generation using all the dataset takes on average of 5.4 minutes to execute. For each time, the performance of the GA-LVQ system in recognition of Arabic handwritten isolation letters dataset was measured and averaged over the 50 times repetition of the data diving (random sub-sampling method).[40] Figure 5 shows that the best number of features in feature subset is 151 with the average accuracy of 95.4% from 50 runs with the same number of features. In our case, after around 300 generations (repeated 50 times) the genetic algorithm found this optimum genome; hence, it found the best feature combination vector consisting of 151 feature that best classifies the given handwritten Arabic character dataset.

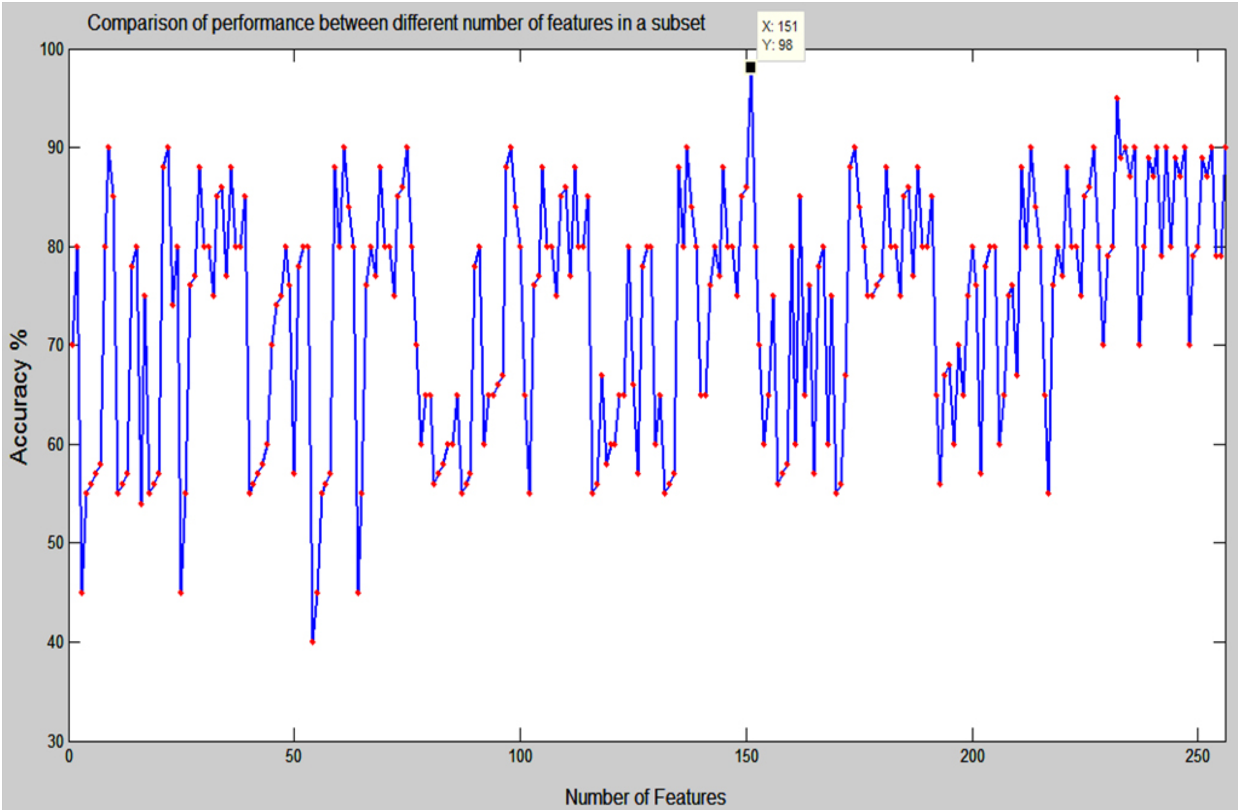

Figure 5. The relationship between the number of features selected and the accuracy of performance, showing the best performance is with 151 features

6. RESULTS

The result of our work shows that the best classification performance for our GA-LVQ system for recognition of Arabic handwritten characters is obtained when 151 features out of the original 1,680 are used. Table 1 presents the average performance obtained by the GA-LVQ algorithm with this optimal system over all 50 evolutionary runs, divided according to the four experimental categories described in Section 5. The performance value denotes the percentage of handwritten characters correctly classified. Three such performances' values are shown; the performance over the test set, the performance over the training set: and the entire database repeated 50 times, with the mean and standard deviation in each category calculated over these 50 runs. To test the effectiveness of the proposed system, our results have been compared with those obtained by other approaches in the literature on similar handwritten Arabic character dataset. Our results are compared to the results obtained from the whole 1,680 feature dataset (without feature selection), and to a re-enacted work done with the backpropagation classifier in Abed & Alasad,[50] to the LMS neural networks in Sarhan et al., to the Neuro-fuzzy network in Alimi, and to the CNN given in El-Sway et al. In the first approach the solutions were obtained by using backpropagation ANN with our dataset but with the same parameters settings given in the original papers: 12 classes, 0.1 learning rate, weights initialized at [-0.1,0.1], and using a 3 hidden layers backpropagation. Backpropagation approach scored an accuracy of 93.61%. The second approach used an LMS neural network, 29 classes with standard deviation between letters considered as a class, two layers, with 10 neurons on the first layer and 28 neurons on the second, and log-sigmoid activation function. The results gave an accuracy of 75% on the same dataset and recorded the higest performance time. The third approach used neuro-fuzzy strategy with a fuzzy beta radial basis neural network for classification and a simple GA with 50 initial population single point cross over with mutation rate 0.01 for tuning the results. An accuracy of 89% on our dataset was reported with this approach. For the last approach a CNN algorithm was used, where our input image is $16 \times 16 \times 1$. In the convolution layer there are 15 filters with filter size of 8x8, and the number of weights per filter is $8 \times 8 \times 3 = 192$, hence the total number of parameters in that layer is $192 + 1 \times 15 = 2,895$ (1 for bias). For each input vector the filter moves along the input both horizontally and vertically and re-computes outputs for each region, in another since filters are convolving each input vector. The number of feature maps is determined by the number of filters. The convolutional layer applies a convolution operation to the input passing the results to the next layer where the response of an individual neuron is emulated.[55] Each convolutional neuron process data only for its receptive fields. The pooling layer combines the output of neurons clusters at one layer into a single neuron in the next layer. Then the fully connected layer connects all neurons to other from all layers.[55] CNN was applied to our relatively small collected dataset, but since it is not a very large dataset deep learning could not give good results as expected. The CNN has its own built in feature extraction technique and classification at the same time and the results on our dataset is good, but it could be improved if we enlarge our dataset and collect more samples from new writers. To accurately compare these results coming from different approaches we applied the algorithms to our collected handwritten Arabic characters' dataset, and used all algorithm available on MATLAB R2016a, such as codes available in the different Neural Network toolboxes and optimization toolboxes, such as: convoltion2dLayer, Block LMS Filter, ga(fitnessfcn, nvars) lvqnet, pattrennet, netTrainfcn, traingd, fuzzyLogicDesigner. To ensure efficient comparisons we also used the standard test, made of a 10-fold-cross validation approach, which were re-enacted for these evaluations on our collected handwritten Arabic characters' dataset. Table 2 shows that the feature subsets selected by our method gave better results in terms of accuracy and time than those obtained by using the whole (1,680) set of features. Hence, the trend of the performance obtained by using of the selected feature subsets is similar to that of the whole feature set, conforming the generality of the proposed feature selection method. Our results show that the GA-LVQ system is a classifier that can successfully distinguish between different handwritten Arabic characters with a 95.4% accuracy. In the future, this accuracy can be further improved with a larger Arabic handwritten dataset from more writers and with more writing trial for each letter. Figure 6 shows the mean squared error decreasing throughout the generations of the procedure compared in training, testing validation and best sets.

Table 1. Results divided according to the experimental categories for the best 151 feature GA-LVQ recognition system

	Training set	**Test set**	**Overall results for all 50 runs**		
	Percentage			**Mean**	**Standard deviation**
Training/test 100/100	96.9%	96.9%	96.23%	1,628.15	4.1
Training/test 75/25	96.30%	96.21%	96.5%	1,624.37	4.4
Training/test 50/50	94.50%	94.61%	94.5%	1,593.59	4.4
Training/test d_2/d_1	95.7%	95.1%	95.4%	1,610.5	4.3

Table 2. Comparing overall results for the 151 feature subset with our GA-LVQ recognition system to other approaches on the same dataset

Research Method	**This work GA-LVQ extracted feature subset 151 attributes**	**This work LVQ on whole feature set 1680 attributes**	**Backpropagation classifier**	**LMS neural network**	**Neuro-fuzzy neural network**	**CNN Convolutional neural network**
Features	* LVQ classifier * GA feature extraction	*LVQ classifier without feature extraction	*ANN classifier without feature extraction *2 classes *learning rate 0.01 *weights initialized at [-0.1,0.1] *3 hidden layers	*Feature extraction with statistical analysis *LMS for classification*29 classes * two layers *10 neurons on the first layer *28 neurons on the second *log-sigmoid activation function	*GA feature extraction *Neural-Fuzzy classifier *beta radial basis *50 initial population *single point cross over *mutation rate 0.01	*CNN Classifier *Automatic feature extraction *15 filters with filter size 8×8 *192 weights *Total number of convolutional layer parameters is 2,895 *fully connected neurons and multi layers
Average Time	5.2 minutes	12.4 minutes	11.4 minutes	14.1 minutes	10.4 minutes	10 minutes

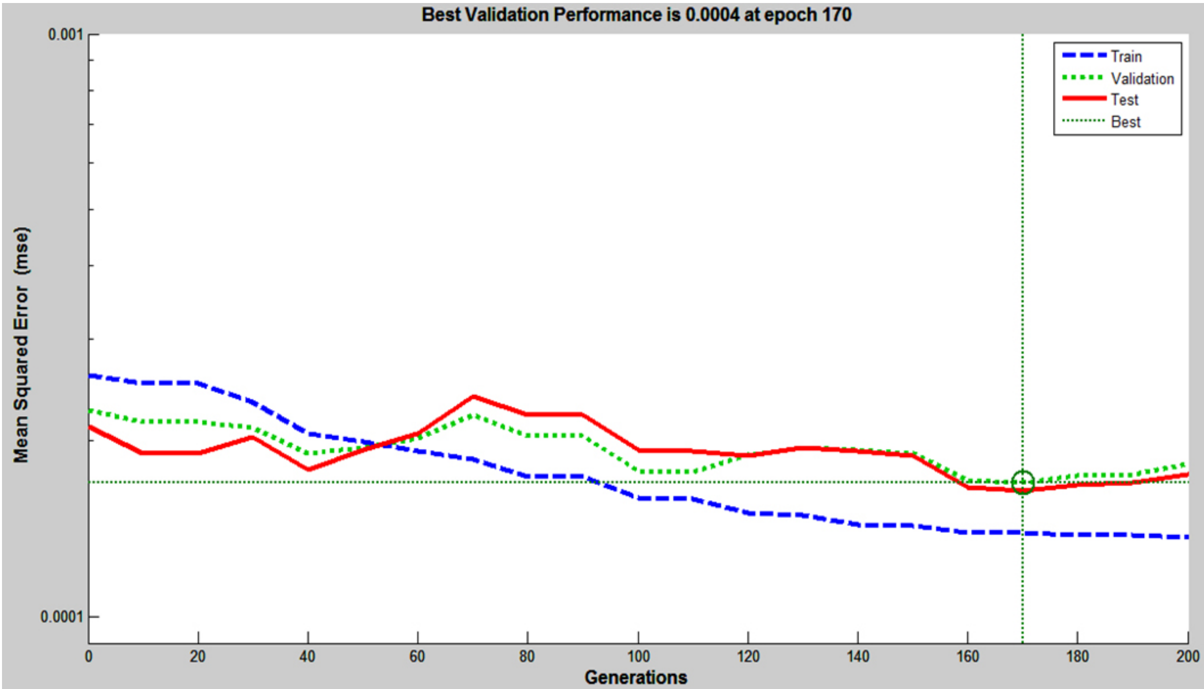

Figure 6. Plots of the mean squared error in the training, testing and validation sets in the GA-LVQ recognition system

7. CONCLUSIONS

In this study, we introduced a novel hybrid automated recognition system to distinguish the handwritten Arabic off line isolated characters. The proposed method is based on a feature selection part done using a genetic algorithm which discovers those features that are more relevant for the classification of the dataset. The selected features are then fed to classifier part using LVQ neural network which finds out the number of successfully recognized cases in the dataset with only the specifies selected features from the GA. The recognition process involves several preliminary steps including handwritten character samples acquisition, dataset preprocessing and preparation steps, feature selection, training of the neural net, and finally classification. The investigation of the results showed that the proposed GA-LVQ method is efficient for the interpretations of Arabic handwritten isolated characters and can correctly identify more than 95.4% of the handwritten letters and with a reduced number of features 151 out of the original 1,680. These results were compared to results obtained from the full feature dataset without selection and with the backpropagation classifier in (Abed & Alasad), and the LMS neural network classifier in (Sarhan et al.), to the neuro-fuzzy network in (Alimi), and finally to the CNN in (El-Sawy et al.) which did not perform as well as it should because it needs a larger data set for efficient deep learning. Our results indicated that our system has a high success rate and performed better in terms of time as well than these approaches. Future work will concentrate on developing our algorithm to recognize full handwritten words and phrases. In hope that our work can positively contribute to the ongoing research for improving algorithms for handwritten Arabic recognition applications and devices used in the market.

ACKNOWLEDGEMENTS

This research project was supported by a grant from the “Research Centre of the Female Scientific and Medical Colleges”, Deanship of Scientific Research, King Saud University.

REFERENCES

[1] Duda RO, Hart PE, Stork DG. Pattern Classification, 2nd ed., New York: Wiley; 2001.

[2] Oliveira L, Sabourin R, Bortolozzi F, et al. Automatic recognition of handwritten numerical strings: A recognition and verification strategy. IEEE Trans. Pattern Analysis and Machine Intelligence. 2002; 24(11): 1438-1454. https://doi.org/10.1109/TPAMI.2002.1046154

[3] Oliveira LS, Sabourin R, Bortolozzi F, et al. A methodology for feature selection using multi-objective genetic algorithms for handwritten digit string recognition. Int. Journal of Pattern Recognition and Artificial Intelligence (IJPRAI). 2003; 17: 5-15.

[4] Gayathri P, Ayyappan S. Off-line handwritten character recognition using Hidden Markov Model. International Conference on Advances in Computing, Communications and Informatics (ICACCI). 2014.

[5] Ahmed P, Al-Ohali Y. Arabic Character Recognition: Progress and Challenges. Journal of King Saud University - Computer and Information Sciences. 2000; 12: 85-116. https://doi.org/10.1016/S1319-1578(00)80004-X

[6] Haraty R, Ghaddar C, Hammami L, et al. Arabic Text Recognition system for printed multi-font and multi-size Arabic characters. The Arabian Journal for Science and Engineering. 2009; 27(1B): 57-72.

[7] Khatatneh K. Probabilistic Artificial Neural Network for Recognizing the Arabic Hand Written Characters. Journal of Computer Science. 2006; 3(12): 881-886.

[8] Shi D, Shu W, Liu H. Feature selection for handwritten Chinese character recognition based on genetic algorithms. 1998 IEEE Int. Conference on Systems, Man, and Cybernetics. 1998; 5: 4201-4206.

[9] Pradeep J, Srinivasan E, Himavathi S. Neural network based handwritten character recognition system without feature extraction. International Conference on Computer Communication and Electrical Technology (ICCCET). 2011.

[10] Mezghani N. A new representation of shape and its use for high performance in online Arabic character recognition by an associative memory. International Journal of Document Analysis and Recognition (IJDAR). 2005; 7(4): 201-210. https://doi.org/10.1007/s10032-005-0145-8

[11] Boufenar C, Kerboua A, Batouche M. Investigation on deep learning for off-line handwritten Arabic character recognition. Cognitive Systems Research. 2018; 50: 180-195. https://doi.org/10.1016/j.cogsys.2017.11.002

[12] Sarfraz M, Nawaz SN, Al-Khuraidly A. Offline Arabic text recognition system. 2003 International Conference on Geometric Modeling and Graphics 2003 Proc. 2003.

[13] Breukelen MV, Duin R, Tax D, et al. Combining classifiers for the recognition of handwritten digits, in: 1st international workshop on statistical techniques in pattern recognition. 1997: 13-18.

[14] Chung K, Yoon J. Performance comparison of several feature selection methods based on node pruning in handwritten character recognition, in: Proceedings of the 4th Int. Conf. on Document Analysis and Recognition (ICDAR 97), IEEE Computer Society; 1997. p. 11-15.

[15] Klasse T. Towards Neural Network Recognition of Handwritten Arabic Letters, Master's Thesis, Dalhousie University Halifax, Nova Scotia; 2001.

[16] Al-Badr A, Mahmoud A. Survey and Bibliography of Arabic Optical Text Recognition. Signal Processing. 1995; 41: 49-77. https://doi.org/10.1016/0165-1684(94)00090-M

[17] El-Wakil M, Shoukry A. On-Line Recognition of Handwritten Isolated Arabic Characters. Pattern Recognition. 1989; 22(2): 97-105. https://doi.org/10.1016/0031-3203(89)90058-7

[18] Bouslama F, Amin A. Pen-based Recognition System of Arabic Character Utilizing Structural and Fuzzy Techniques, Second International Conference on Knowledge-Based Intelligent Electronic Systems, 1998. 21-23, April Adelaide, Australia. Editors, L.C. Jain and R.K. Jain, pp. 76-85.

[19] Han X, Chang X, Quan L, et al. Feature subset selection by gravitational search algorithm optimization. Information Sciences. 2014; 281: 128-146. https://doi.org/10.1016/j.ins.2014.05.030

[20] Punch W, Goodman E, Pei M, et al. Further Research on Feature Selection and Classification Using Genetic Algorithms. Proceedings of International Conference on Genetic Algorithms. Springer; 1993. p.557-564.

[21] De Stefano C, Fontanella F, Marroccoc, Scotto di Freca A. A GA-based feature selection approach with an application to handwritten character recognition. Pattern Recognition Letters, Elsevier; 2014. p.1-14.

[22] Cordella L, De Stefano C, Fontanella F, et al. A feature selection algorithm for handwritten character recognition, in: 19th International Conference on Pattern Recognition (ICPR 2008). 2008. p.1–4.

[23] Oh IS, Lee JS, Moon BR. Hybrid genetic algorithms for feature selection. IEEE Trans. Pattern Anal. Mach. Intell. 2004; 26(11): 1424-1437. PMid:15521491. https://doi.org/10.1109/TPAMI.2004.105

[24] Ho TK, Basu M. Complexity measures of supervised classification problems. IEEE Trans. Pattern Anal. Mach. Intell. 2002; 24(3): 289-300. https://doi.org/10.1109/34.990132

[25] Khorsheed M, Clocksin W. Structural Features of Cursive Arabic Script (BMVC 1999). 1999; p. 422-431.

[26] Daliri MR. A Hybrid Automatic System for the Diagnosis of Lung Cancer Based on Genetic Algorithm and Fuzzy Extreme Learning Machines images. Journal of Medical systems. Springer US. 2012; 36(2): 1001-1005.

[27] Kim G, Kim S. Feature selection using genetic algorithms for handwritten character recognition, in: Proc. of the 7th Int. Workshop on Frontiers in Handwriting Recognition, IWFHR Amsterdam. 2000. p. 103-112.

[28] Yang J, Honavar V. Feature subset selection using a genetic algorithm. IEEE Intelligent Systems. 1998; 13: 44-49. https://doi.org/10.1109/5254.671091

[29] Sasikala M, Kumaravel N. A wavelet-based optimal texture feature set for classification of brain tumors. Journal of Medical Engineering & Technology. 2008; 32(3).

[30] Bassat MB. Pattern recognition and reduction of dimensionality, Handbook of Statistics-II, North Holland; 1982. p.773-791.

[31] Fukunaga K. Introduction to statistical pattern recognition, (2nd ed.), Academic Press Professional, Inc., San Diego, CA, USA; 1990.

[32] Ripley B. Pattern Recognition and Neural Networks. New York: Cambridge University Press; 1996. https://doi.org/10.1017/CBO9780511812651

[33] Liu H, Motoda H. Feature Extraction, Construction and Selection: A Data Mining Perspective, Springer Nature. Kluwer Academic Publishers Norwell, MA, USA; 1998.

[34] Liu H, Setiono R. Feature Selection and Classification- A Probabilistic Wrapper Approach. In: Proceedings of the Ninth International Conference on Industrial and Engineering Applications of AI and ES; 1996.

[35] Somol P, Pudil P, Kittler J. Fast branch and bound algorithms for optimal feature selection. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2004; (7): 900-912. PMid:18579948.

[36] Michalewicz Z. Genetic Algorithms+Data Structures=Evolution Programs, 3rd ed., Springer-Verlag; 1996.

[37] Nunes CM, Britto Jr. A. d. S., Kaestner CAA, et al. An optimized hill climbing algorithm for feature subset selection: Evaluation on handwritten character recognition, in: Proceedings of the 9-th Int. Workshop on Frontiers in Handwriting Recognition (IWFHR'04), IEEE Computer Society, Washington, DC, USA; 2004. p.365-370. https://doi.org/10.1109/IWFHR.2004.18

[38] Siedlecki W, Sklansky J. A note on genetic algorithms for largescale feature selection. Pattern Recognition Letters. 1989; 10(5): 335-347. https://doi.org/10.1016/0167-8655(89)90037-8

[39] Guyon I, Elisseeff A. An introduction to variable and feature selection. Journal of Mach. Learn. Res. 2003; 3: 1157-1182.

[40] Alharbi A, Tchier F. Using a Genetic-Fuzzy Algorithm as a Computer Aided Diagnosis Tool on Saudi Arabian Breast Cancer Database. Mathematical Biosciences. 2017; 286: 39-48. PMid:28185926. https://doi.org/10.1016/j.mbs.2017.02.002

[41] Chouaib H, Terrades OR, Tabbone S, et al. Feature selection combining genetic algorithm and ad boost classifiers, in: Proc. of International Conference on Pattern Recognition ICPR. 2008.

[42] Alimi A. An Evolutionary Neuro-Fuzzy Approach to Recognize On-Line Arabic Handwriting. Proceedings of the 4th International Conference Document Analysis and Recognition (ICDAR 1997). 1997; 382-386.

[43] Altuwaijri M, Bayoumi MA. Arabic Text Recognition Using Neural Network, ISCAS 94. IEEE International Symposium on Circuits and systems. 1994; 6.

[44] Hany A, Azeem S. Off-line Arabic handwriting recognition system based on concavity features and HMM classifier, Conference: Proceedings of the 21th International Conference on Pattern Recognition (ICPR'12), Tsukuba Science City, Japan; 2012. p.705.

[45] Abed MA, Alasad HA. High Accuracy Arabic Handwritten Characters Recognition Using Error Back Propagation Artificial Neural Networks. International Journal of Advanced Computer Science and Applications. 2015; 6(2): 145.

[46] Al-Yousefi H. Recognition of Arabic Characters. IEEE Pami. 1992; 14(8): 853-857. https://doi.org/10.1109/34.149585

[47] Amor B, Messoukri N, Amara B. A hybrid approach for Multifront Arabic Characters Recognition, 5th WSEAS Int. Conf. On Artificial Intelligence, Knowledge Engineering and Data Bases (AIKED'06) Madrid, Spain; 2006 February 15-17.

[48] Haraty R, Zabadani H. An Off Line Arabic Handwritten Recognition System, in Proceedings of the 2002 International Arab Conference on Information Technology (ACIT'2002), Doha, Qatar; 2002.

[49] Haraty R, Hamid A. Segmenting Handwritten Arabic Text. ACIS International Journal of Computer and Information Science. 2002; 3(4).

[50] Lamghari N, Charaf MEH, Raghay S. Hybrid feature vector for the recognition of Arabic handwritten characters using feed-forward neural network. Arabian Journal for Science and Engineering. 2017; 1-9.

[51] Abed MA, Alasad HA. High Accuracy Arabic Handwritten Characters Recognition Using Error Back Propagation Artificial Neural Networks. International Journal of Advanced Computer Science and Applications. 2015; 6(2): 145.

[52] Sarhan AM, Al Helalat OI, Alimi TW, et al. Arabic Character Recognition using Artificial Neural Networks and Statistical Analysis, The Analysis of Error in an On-Line Recognition System of Arabic Handwritten Characters, Proceedings of ICDAR 1995, 14-16 August Montreal, Canada. pp. 890-893.

[53] Hussein S, Elkhidir A, Elnourani MG. Optical character recognition of Arabic handwritten character using neural network, Proc. Computing, Control, Networking, Electronic, and Embedded system engineering, ICCNEEE. International Conference; 2015, pp. 456-461.

[54] Elleuch M, Tagougui N, Kherallah M. Arabic handwritten character recognition using deep belief neural network. Proc. Systems signal & devices (SSD) 12th International multi-conference. 2015; p. 1-5.

[55] El-Sawy A, Loey M, El-Bakry H. Arabic handwritten character recognition using convolutional neural networks. WEAS Transactions on Computer research. 2017; 5: 11-19.

[56] Shin HC, Roth HR, Gao M, et al. Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning. IEEE Transactions on Medical Imaging. 2016; 35: 1285-1298. PMid:26886976. https://doi.org/10.1109/TMI.2016.2528162

[57] Alharbi A. An Automated Computer System Based on Genetic Algorithm and Fuzzy Systems for Lung Cancer Diagnosis. International Journal of Nonlinear Sciences and Numerical Simulation. 2018; 19(6): 583-594. https://doi.org/10.1515/ijnsns-2017-0048

[58] Goldberg D. Genetic Algorithms in Search, Optimization, and Machine Learning. New York, Addison-Wesley; 1989.

[59] Alharbi A, Tchier F. A Fuzzy-Genetic Algorithm Method for the Breast Cancer Diagnosis Problem. The Ninth International Conference on Advanced Engineering Computing and Applications in Sciences, ADVCOMP, Proc. 2015. p.122-127.

[60] Alharbi A, AlQahtani K. The A Genetic Algorithm solution for the Doctor Scheduling Problem. International Journal On Advances in Software. 2017; 10: 80-90.

[61] Mitchell M. An Introduction to Genetic algorithms. Cambridge, MIT Press; 1996.

[62] Hart PE, Duda RO, Stork DG. Pattern Classification, John Wiley & sons, Inc; 2001.

[63] MATLAB lvq code retrieved 2017. Available from: http://uk.mathworks.com/help/deeplearning/ref/lvqnet.htmle

[64] Kohonen T. Learning vector quantization for pattern recognition, Technical Report TKKF-A601, Helsinki University of Technology, Espoo, Finland; 1986.

[65] Hassoun M. Fundamentals of Artificial Neural Networks. Boston, MIT Press; 1995.

[66] Kohonen T. Self-Organizing Maps, second edition, Springer, Berlin; 1997.

[67] Sonawane AS, Patil DR. Prediction of heart disease using learning vector quantization algorithm, Conference on IT in Business, Industry and Government (CSIBIG 2014). 2014.

[68] Sakr GE, Elhajj I, Huijer H. Support vector machines to define and detect agitation transition. IEEE Transactions On Affective Computing. 2010; I: 98-108.

[69] Biehl M, Ghosh A, Hammer B. Dynamics and generalization ability of LVQ algorithms. Journal of Machine Learning Research. 2007; 8: 323–360.

[70] Ouyang A, Li K, Zhou X, et al. Improved LDA and LVQ for Face Recognition. Appl. Math. Inf. Sci. 2014; 8(1): 301-309.

[71] Makino S, Endo M, Sone T, et al. Recognition of phonemes in continuous speech using a modified LVQ2 method. J. Acoustical Society of Japan. 1992; 13(6): 351-360.