International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:4, No:3, 2010

# Persian Printed Numerals Classification Using Extended Moment Invariants

Hamid Reza Boveiri

***Abstract*—**Classification of Persian printed numeral characters has been considered and a proposed system has been introduced. In representation stage, for the first time in Persian optical character recognition, extended moment invariants has been utilized as characters image descriptor. In classification stage, four different classifiers namely minimum mean distance, nearest neighbor rule, multi layer perceptron, and fuzzy min-max neural network has been used, which first and second are traditional nonparametric statistical classifier. Third is a well-known neural network and forth is a kind of fuzzy neural network that is based on utilizing hyperbox fuzzy sets. Set of different experiments has been done and variety of results has been presented. The results showed that extended moment invariants are qualified as features to classify Persian printed numeral characters.

***Keywords*—**Extended moment invariants, optical character recognition, Persian numerals classification.

## I. Introduction

PATTERN recognition is a scientific discipline whose goal is classification of objects into a number of categories or classes. Depending on application, these objects can be images, signal waveforms, or any type of measurements that need to be classified. The generic term *pattern* is used to refer these objects. Applications of pattern recognition systems and techniques are numerous and cover a broad scope of activities [1].

Optical character recognition (OCR) is one of the most important areas of pattern recognition to tackle different aspects of automatic recognition of written patterns. OCR system is an image processing system consists of several isolated processing stage like preprocessing, representation, classification, which gets a textural image as input and after processing it will produce corresponding editable text as output. Meanwhile, numeral character classification has a large number of applications in license plate recognition, automatic check ordering in banks, automatic postal code recognition of mails, processing of numeral forms etc.

Fig. 1 shows Persian numeral characters. Persian numeral characters classification has a lot of complexities and must be more sophisticated due to similarity of Persian "two", "four" and "six" and reversal shape of "seven" and "eight". With regard to importance and extensive application, many researches [2]-[14] have been reported in the literature and many data sets [15]-[17] have been introduced. Unfortunately, all of them are on handwritten digits and printed digits have not been attended a lot.

Ebrahimnezhad *et al*. [18] presented a fuzzy approach based on using entropy function to improve the fuzzifier function definition for Persian digit recognition irrespective of font and size. Box method was used to generate feature vector. The system tested with various fonts and finally achieved 97.5% correct recognition. In [19] they extended their system. They presented a synthesis approach using three fuzzifier functions simultaneously, and with post processing using another fuzzy system, they reported complete correct recognition. However, the character "zero" was neglected in their experiments and the utilized data set is unknown as well. The author in [20] proposed a system for Persian printed numeral characters recognition using geometrical central moments and fuzzy min-max neural network. The system tested on a data set encompassing various samples and finally 99.16% correct recognition was achieved. In addition, the author in [21] introduced a new proposed system for invariant Persian printed digits classification using seven moment invariants as characters image descriptor. The system tested on the same data set and finally 98.75% correct classification was reported.

Hu [22] first introduced the use of seven moment invariants which are defined on geometrical moments of the image as features for pattern classification. These moments are nonlinear and invariant under translation, rotation, scaling, and image reversal. Li [23] developed Hu's invariants and listed 52 extended moment invariants. Where lower orders of moment invariants are not enough to classify patterns, higher order of invariants will be used although the higher order invariants resulted in higher sensitivity. Hu invariants are frequently used in Persian optical character recognition [21, 24, 25, 26] while extended ones have not been attended a lot.

The rest of the paper is organized as follows. In the next section, proposed system and its structure are briefly described. Section III surveys preprocessing stage. Section IV gives detail description of extended moment invariants and feature vector enrichment in representation stage. Section V is devoted to classification stage and introduces the utilized classifiers. Section VI gives implementation remarks and reports the experimental results. Finally, the conclusions are outlined in section VII.

The author is with the Computer Department, Sama College, Islamic Azad University Shushtar Branch, Shushtar, Iran (phone and fax: +986126236015; e-mail: hr.boveiri@yahoo.com).

| Zero | One | Two | Three | Four |
|---|---|---|---|---|
| ۰ | ۱ | ۲ | ۳ | ۴ |
| Five | Six | Seven | Eight | Nine |
| ۵ | ۶ | ۷ | ۸ | ۹ |

Fig. 1 A typical Persian printed numeral characters

## II. The Proposed System and Its Structure

Fig. 2 shows the proposed system and its structure. In preprocessing stages, the operations like noise reduction and skew correction are done to enhance input images. In representation stage, set of twelve extended moment invariants is used as image descriptor. Also in this stage, feature vector is enriched using an extra statistical feature. In classification stage, four different classifiers namely minimum mean distance (MMD), nearest neighbor rule (NN), multi layer perceptron (MLP), and fuzzy min-max neural network (FMMNN) are used.

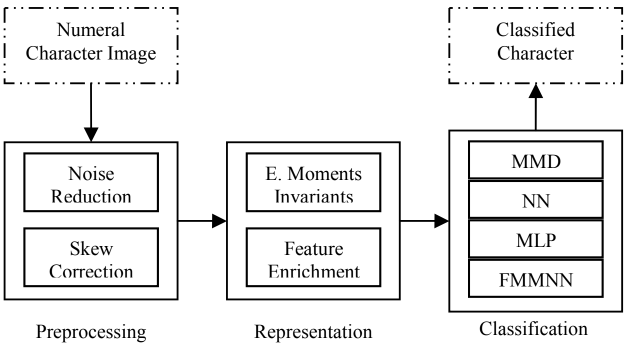


Fig. 2 The proposed system and its structure

## III. Preprocessing

Moment invariants and extra feature are sensitive to noise [27] and skew [20] respectively. Thus in this stage, noise reduction and skew correction are done to compensate these defects.

### A. Noise Reduction

A nonlinear median filter using a 3×3 mask has been used for image noise reduction. This filter has high ability to noise removal, retains sharp edges and fixes some undesired disconnections if possible [28].

### B. Skew Correction

Principle component analysis (PCA), using geometrical central moments, has been used to estimate skew angle [22]. Skew angle from principle axes of image is computed by geometrical central moments up to the order 2 by using (1).

$$\alpha = \frac{1}{2} Tan^{-1}\left(\frac{2\mu_{11}}{\mu_{20} - \mu_{02}}\right) \tag{1}$$

Where $\mu_{pq}$ is geometrical central moment from order $(p + q)$ which is computed by using (3). Image skew from its principle axes can be corrected by rotation of image with respect to $\alpha$ angle.

## IV. Representation

Extended moment invariants have been used as characters image descriptor. These moments are nonlinear, invariant under translation, rotation, scaling, and image reversal. Whereas Persian "seven" and "eight" have reversal shape of each other, moment invariants cannot discriminate them. Therefore, feature vector is enriched by an extra statistical feature, that is, count of upper half of image pixels divide by count of lower half of image pixels.

### A. Extended Moment Invariants

Geometrical moment of order $(p + q)$ for a *two*-dimensional discrete function like image is computed by using (2). If the image can have nonzero values only in the finite part of *xy* plane; then moments of all orders exist for it [22].

$$m_{pq} = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} x^p y^q f(x,y) \tag{2}$$

Where $f(x, y)$ is image function and $M$, $N$ are image dimensions. Then, geometrical central moments of order equal to $(p + q)$ can be computed using (3).

$$\mu_{pq} = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} (x-\bar{x})^p (y-\bar{y})^q f(x,y) \tag{3}$$

Where $\bar{x}$ and $\bar{y}$ are the gravity center of image and can be calculated by (4). Actually, by image translation to coordinate origin while computing central moments, they become translation invariant.

$$\bar{x} = \frac{m_{10}}{m_{00}}\ , \quad \bar{y} = \frac{m_{01}}{m_{00}} \tag{4}$$

Note that in a binary image, $m_{00} = \mu_{00}$ is count of foreground pixels and has direct relation to image scale, therefore central moments can become scale normalized using (5).

$$\eta_{pq} = \frac{\mu_{pq}}{m_{00}^{a}}\ , \quad a = \frac{p+q}{2} + 1 \tag{5}$$

Having normalized geometrical central moments up to the order 4, seven moment invariants ($\varphi_1$-$\varphi_7$) introduced by Hu [22] and then five extended ones ($\varphi_8$-$\varphi_{12}$) developed by Li [23], can be computed using (6) and (7) respectively.

$$
\begin{aligned}
\varphi_1 &= \eta_{20} + \eta_{02} \\
\varphi_2 &= (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \\
\varphi_3 &= (\eta_{30} - 3\eta_{12}) + (3\eta_{21} - \eta_{03})^2 \\
\varphi_4 &= (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \\
\varphi_5 &= (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\
&\quad + (3\eta_{12} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \\
\varphi_6 &= (\eta_{20} - \eta_{02})[(\eta_{30} + v_{12})^2 - (\eta_{21} + \eta_{03})^2] \\
&\quad + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) \\
\varphi_7 &= (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\
&\quad + 3(\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]
\end{aligned} \tag{6}
$$

$$
\begin{aligned}
\varphi_8 &= \eta_{40} + \eta_{22} + \eta_{02} \\
\varphi_9 &= (\eta_{40} - \eta_{04})^2 + 4(\eta_{31} - \eta_{13})^2 \\
\varphi_{10} &= (\eta_{40} - 6\eta_{22} + \eta_{04})^2 + 16(\eta_{31} - \eta_{13})^2 \\
\varphi_{11} &= (\eta_{40} - 6\eta_{22} + \eta_{04})^2[(\eta_{40} - \eta_{04})^2 + 4(\eta_{31} - \eta_{13})^2] \\
&\quad + 16(\eta_{40} - \eta_{04}) + (\eta_{31} + \eta_{13})(\eta_{31} - \eta_{13}) \\
\varphi_{12} &= (\eta_{40} - 6\eta_{22} + \eta_{04})^2[(\eta_{40} - \eta_{04})^2 + 4(\eta_{31} - \eta_{13})^2] \\
&\quad - 16(\eta_{40} - \eta_{04}) + (\eta_{31} + \eta_{13})(\eta_{31} - \eta_{13})
\end{aligned} \tag{7}
$$

Table I lists extended moment invariants ($\varphi_1$-$\varphi_{12}$) for different status of character in fig. 9 after preprocessing. Where $\mu$ and $\sigma$ are sample mean and sample standard deviation respectively and $\sigma/\mu$ % is percentage of spread of moment invariants values from their corresponding means. Because of wide range of the moments, logarithms of their magnitudes have been used.

TABLE I
EXTENDED MOMENT INVARIANTS FOR DIFFERENT STATUS OF CHARACTER IN FIG. 9 AFTER PREPROCESSING

| | Fig. 9a | Fig. 9b | Fig. 9c | Fig. 9d | μ | σ | σ/μ % |
|---|---|---|---|---|---|---|---|
| $\varphi_1$ | -1.240 | -1.312 | -1.240 | -1.247 | -1.260 | 0.035 | 2.782 |
| $\varphi_2$ | -4.039 | -4.426 | -4.039 | -3.883 | -4.09 | 0.231 | 5.658 |
| $\varphi_3$ | -3.761 | -3.992 | -3.761 | -3.886 | -3.850 | 0.111 | 2.888 |
| $\varphi_4$ | -5.59 | -5.884 | -5.59 | -5.748 | -5.704 | 0.140 | 2.469 |
| $\varphi_5$ | -10.7 | -11.05 | -10.7 | -10.95 | -10.86 | 0.161 | 1.485 |
| $\varphi_6$ | -8.00 | -8.515 | -8.00 | -7.947 | -8.116 | 0.267 | 3.294 |
| $\varphi_7$ | -9.47 | -11.28 | -9.47 | -9.798 | -10.00 | 0.862 | 8.618 |
| $\varphi_8$ | -1.917 | -2.101 | -1.917 | -1.983 | -1.97 | 0.087 | 4.397 |
| $\varphi_9$ | -4.703 | -4.969 | -4.703 | -4.785 | -4.78 | 0.122 | 2.568 |
| $\varphi_{10}$ | -5.379 | -5.687 | -5.379 | -5.521 | -5.49 | 0.146 | 2.665 |
| $\varphi_{11}$ | -7.096 | -7.872 | -7.096 | -7.266 | -7.33 | 0.368 | 5.027 |
| $\varphi_{12}$ | -9.17 | -9.016 | -9.17 | -9.354 | -9.18 | 0.138 | 1.506 |

### B. *Feature Vector Enrichment*

Persian "seven" and "eight" are vertically reverse of each other. With respect to invariance of moment invariants under image reversal, they can not discriminate these characters. Thus, feature vector is enriched with an extra statistical feature, that is, count of upper half of image pixels divide by count of lower half of image pixels. Persian "seven" and "eight" both have triangle shape and are reverse of each other, so relation of their upper half of image pixels to lower half of image pixels is completely different [20]. It has been also shown in Fig. 3.

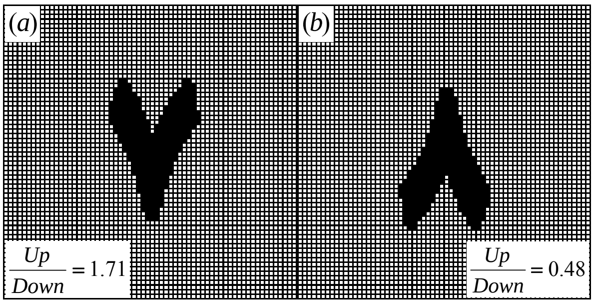


Fig. 3 (a) Persian "seven" digit (b) Persian "eight" digit. With computed extra feature for them [20]

## V. CLASSIFICATION

Four different classifiers namely minimum mean distance, nearest neighbor rule, multilayer perceptron, and fuzzy min-max neural network has been used. MMD and NN are traditional nonparametric statistical classifier. MLP is a well-known neural network classifier and FMMNN is a kind of fuzzy neural network based on utilizing hyperbox fuzzy sets.

### A. *Minimum Mean Distance*

Minimum mean distance is a conventional nonparametric statistical classifier. At first, it makes mean feature vector of each class using training samples. Then in the testing stage, it assigns an unknown input pattern to the class which has minimum distance with corresponding mean feature vector among the all classes. Different definitions of distance can be used practically e.g. Euclidian distance, city-block distance, and so on.

### B. *Nearest Neighbor Rule*

Nearest neighbor rule is also a conventional nonparametric statistical classifier. In training stage, it stores all training samples in a table. Then in testing stage, it assigns an unknown input pattern to which class has minimum distance to a training sample of that class. Just such as minimum mean distance classifier, different definitions of distance can also be used here.

### C. *Multi Layer Perceptron*

Artificial neural network (ANN) has been inspired from biological neural structure of human brain. Although ANN is a very simple abstraction of its biological counterpart, it has been interested a lot because of its extensive power in pattern classification and clustering in the resent years [29]. Multi layer perceptron is a feed-forward neural network with one or more layers of nodes between the input and output layers. These in-between layers are called hidden layers. Each node in a layer is connected to the all nodes in the next layer. Using

MLP in the context of a classifier requires all output nodes to be set to 0 expect for the node that is marked to correspond to the class the input is from. That desired output is 1. MLP training is done using an iterative gradient descent procedure known as back-propagation algorithm [27].

### *D. Fuzzy Min-Max Neural Networks*

Simpson first introduced fuzzy min-max neural network for pattern classification and clustering [30], [31]. Fuzzy min-max neural network is a kind of fuzzy neural networks (aggregation of fuzzy logic and neural networks) which works based on generating and utilizing hyperbox fuzzy sets. A hyperbox defines a region of the *n*-dimensional pattern space that has patterns with full class membership. A hyperbox is completely defined by its min point and its max point, and a membership function is defined with respect to these hyperbox min-max points. An illustration of the min and max points in a *three*-dimensional hyperbox is shown in Fig. 4. Membership function of each hyperbox gives membership value of the input patterns relative to that hyperbox. Patterns that are near from the hyperbox get high membership values and the others, which are far from the hyperbox, get the lower ones.

FMMNN with cooperation of membership functions of all hyperboxes, determine membership value of each input pattern related to each class and it classifies them. Training in FMMNN is done by proper adjusting of size and location of each hyperbox in the pattern space. Fig. 5 illustrates separation of two classes by *two*-dimensional hyperboxes.

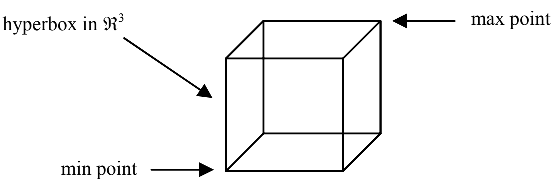


Fig. 4 The min-max *three*-dimensional hyperbox [30]

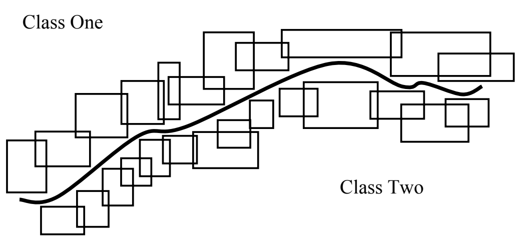


Fig. 5 Separation of two different classes by *two*-dimensional hyperboxes [30]

Some properties of fuzzy min-max neural network are presented as follow.

*Online Adaptation:* FMMNN is able to learn new classes and refine existing classes quickly and without destroying old class information or need to retrain them.

*Nonlinear Separability:* FMMNN is able to make decision regions that separate classes of any shape and size.

*Overlapping Classes:* It has ability to form a decision boundary that minimizes the amount of misclassification for all of the overlapping classes.

*Training Time:* one of the excellent properties of FMMNN is the short training time as in most cases it needs only one pass for training.

*Tuning Parameters:* FMMNN has parameters for better adaptation with respect to the input patterns. $\theta$ parameter is a user defined value that bounds the maximum size of hyperboxes during the FMMNN training. Lower values of $\theta$ make more (and smaller) hyperboxes that seem to be proper for separating nonlinear, ambiguity and overlapping classes. Although some of these hyperboxes are not necessary increasing training and classification time. $\gamma$ parameter is also a user defined value so called sensitivity parameter and regulates how fast the membership values decrease as the distance between input pattern and hyperbox increases.

## VI. Implementation and Experimental Results

The proposed system was implemented on a Penitum4 (2.6GHz) desktop computer with Microsoft Windows XP (SP2) platform using Microsoft Visual Basic 6.0 programming language. Required time to compute twelve moment invariants and extra feature from a typical 64×64 character image was 13 and 19 millisecond (ms) respectively (Mean of 10 times of algorithm execution has been used).

A *three*-layer MLP has been utilized. In hidden layer and output layer, tangent sigmoid function and logarithm sigmoid function are used respectively. Fig. 6 demonstrates the results of a set of experiments using feature vector consists of extended moment invariants and extra feature with respect to different numbers of nodes in the MLP hidden layer. Results show 22 nodes seem to be adequate as the number of hidden layer nodes. In all experiments, 0.05 and 1000 are used as learning rate and number of training pass respectively. In addition, the features are normalized to have zero mean and unit variance before being input to the MLP. This is necessary to ensure that a subgroup of features do not dominate the weight assignment process during the training.

A set of experiments was done to choose proper values as FMMNN parameters. Fig. 7 illustrates correct classification diagram with respect to different values of $\theta$ parameter (maximum size of the hyperboxes). The numbers above the diagram present count of generated hyperboxes during FMMNN training which increase as the value of $\theta$ decreases. The results (94.5%) were not improved using values less than "0.09" which seems to be the best value for this parameter. Fig. 8 illustrates correct classification diagram with respect to different values of $\gamma$ parameter (sensitivity parameter). The results (94.5%) was not improved using values less than "1" which is the best value for this parameter (decreasing this parameter increases the hyperboxes sensitivity).

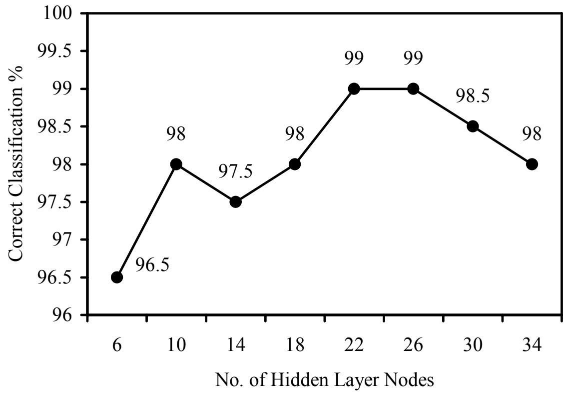


Fig. 6 The results of a set of experiments using feature vector consists of extended moment invariants and extra feature with respect to different numbers of nodes in the MLP hidden layer

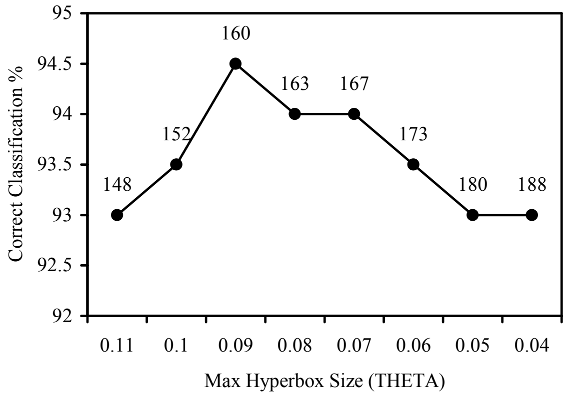


Fig. 7 Correct classification diagram with respect to different values of θ parameter (maximum size of hyperboxes). The numbers above the diagram present count of the generated hyperboxes during FMMNN training which increase as the value of θ decreases

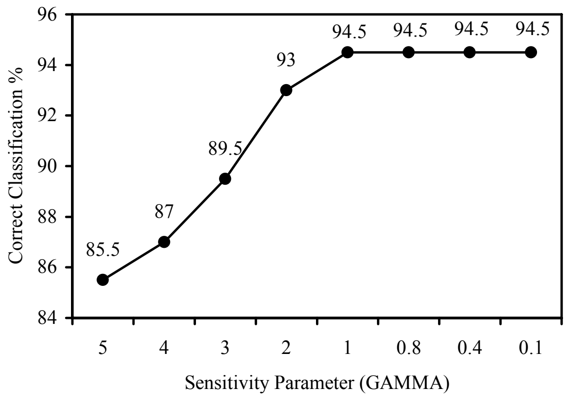


Fig. 8 Correct classification diagram with respect to different values of γ parameter (sensitivity parameter)

### A. *The Utilized Data Set*

In order to evaluate the system, the data set introduced in [20] has been used. This data set consists of 64×64 binary images of all 10 Persian numeral characters in four groups. The first group was regular characters with same size and without any translations and rotations. The second group was rotated characters in the range of (-45, 45) degree. The third group was randomly translated characters and characters of the fourth group had various sizes. There were 10 samples for each character in each group so that those were totally 400 samples (40 samples per character). Fig. 9 shows a sample image of each group of Persian "three" in the data set. In all experiments, half of samples have been used for training and the remainders as test data unless where it will be expressed explicitly.

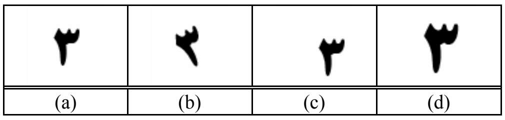


Fig. 9 A sample image of each group of Persian "three" in the data set. (a) Regular (b) Rotated (c) Translated (d) Scaled [20]

### B. *Experimental Results*

The results of correct classification of each classifier using feature vector consists of twelve moment invariant with and without enrichment is demonstrated in Table II. It is clear that feature vector enrichment has improved the result of all classifiers significantly as we expected.

TABLE II
THE RESULTS OF THE CORRECT CLASSIFICATION OF EACH CLASSIFIER USING FEATURE VECTOR CONSISTS OF TWELVE MOMENT INVARIANT WITH AND WITHOUT ENRICHMENT

| Utilized Classifier | Correct Classification Without Enrichment | Correct Classification With Enrichment |
|---|---|---|
| Minimum Mean Distance | 70.5% | 73.5% |
| Nearest Neighbor Rule | 88.5% | 96.5% |
| Multi Layer Perceptron | 88.5% | 99% |
| Fuzzy Min-Max Neural Network | 88.5% | 94.5% |

In the next experiments, the extra feature has several times been added to the feature vector to increase its effect. Fig. 10 illustrates the results of these experiments of each classifier. The best results were achieved by adding the extra feature 7, 3, 1, and 3 times to the feature vector for MMD, NN, MLP, and FMMNN respectively.

Another set of experiments was done to see the effect of size of training samples. Fig. 11 illustrates correct classification diagram with respect to different numbers of training samples for each classifier. With more training sample, better results were usually achieved. In each experiment, reminder samples were used as test samples. The best results were achieved using 80% of samples to train and the remainder 20% as testing. Note that feature vector for each classifier was generated separately in that the extra feature

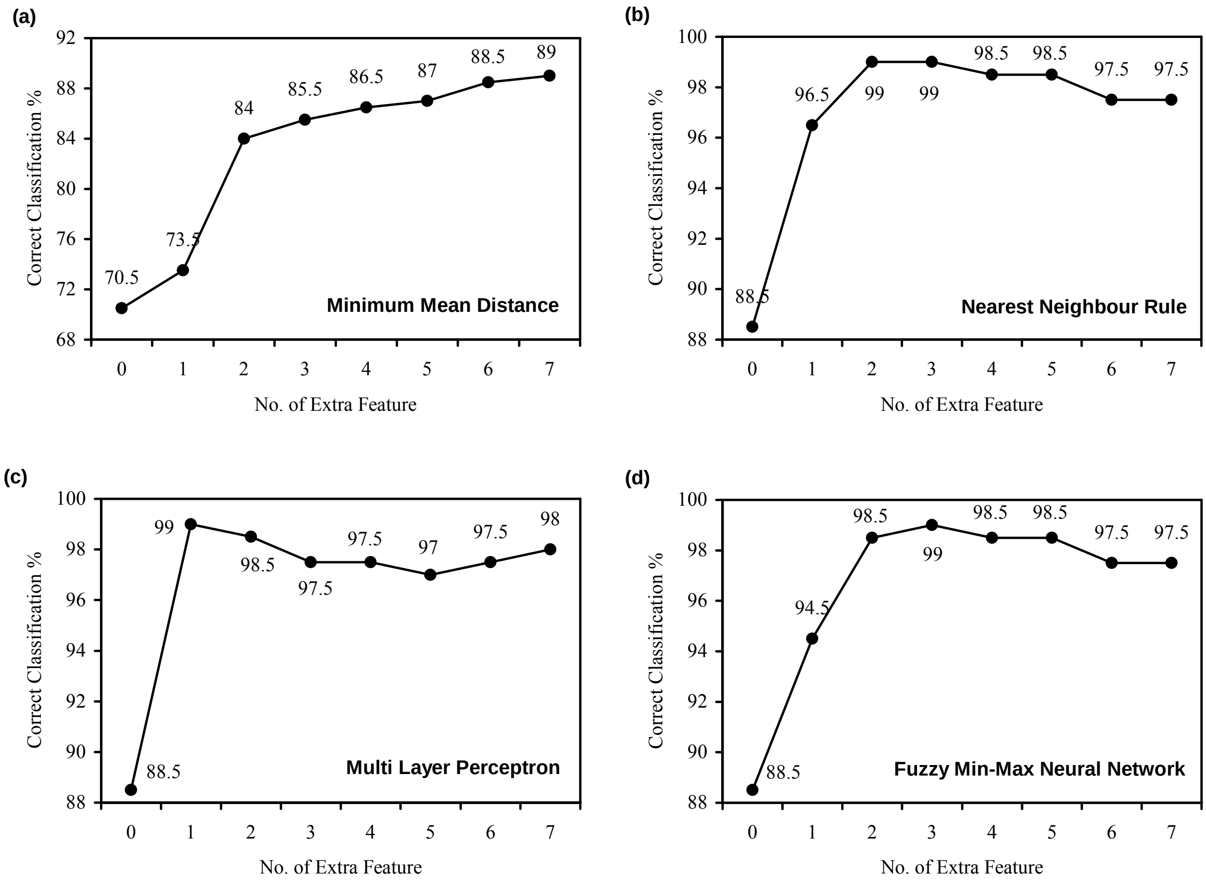


Fig. 10 Correct classification diagram for each classifier with respect to adding the extra feature several times to the feature vector. (a) Minimum mean distance (b) Nearest neighbor rule (c) Multi layer Perceptron (d) Fuzzy min-max neural network

was added 7, 3, 1, and 3 times to the feature vector for MMD, NN, MLP, and FMMNN respectively.

## VII. Conclusion

Application of extended moment invariants in classification of Persian printed numeral characters was investigated and a proposed system was introduced. In representation stage, twelve extended moment invariants developed by Li, which are extension of Hu invariants, were used as image features. Whereas these moment invariants are invariant under image reversal, thus feature vector was enriched with an extra statistical feature to discriminate the characters like "seven" and "eight". In classification stage, four different classifiers namely minimum mean distance, nearest neighbor rule, multilayer perceptron, and fuzzy min-max neural network are used. MMD and NN are traditional nonparametric statistical classifier. MLP is a well-known neural network classifier and FMMNN is a kind of fuzzy neural network. Set of different experiments was done to select proper classifiers parameters (MLP and FMMNN) and to see effect of enrichment and number of training samples. In all experiments, NN, MLP, and FMMNN had good results instead of MMD. It is due to some overlapping of moment invariants while MMD is a linear classifier. Finally, results showed extended moments invariant are adequate for classification of Persian printed numeral characters.

Future work may include use of combination of multiple classifiers with multiple feature sets to improve the accuracy of classification.

## References

[1] S. Teodoridis and K. Koutroumbas, *Pattern Recognition*, 2nd Ed., CA: Academic Press, 2003.

[2] A. Darvish, E. Kabir, and H. Khosravi, "Application of Shape Matching in Persian Handwritten Digits Recognition," *Modares Technology and Eng.*, vol. 22, pp. 37-47, Dec 2005.

[3] H. Ketabdar, "Persian Handwritten Digits Recognition with Structural Approach," in *Proc. 6th Iranian Conf. on Electrical Engineering*, Tehran, 1998, pp. 4.19-4.24.

[4] V. Johari and M. Razavi, "Fuzzy Recognition of Persian Handwritten Digits," in *Proc. 1st Iranian Conf. on Machine Vision and Image Processing*, Birjand, 2001, pp. 144-151.

[5] H. Mozafari, F. Rahgozar, and Sh. Sharif, "Online Recognition of Persian Handwritten Digits," in *Proc. 2nd Computer Society of Iran Annu. Conf.*, Tehran, 1996, pp. 196-204.

[6] N. Sadati, E. Sheikhi, and B. Nazari, "Utilizing Fuzzy Logic in Recognition of Persian Handwritten Digits," in *Proc. 3rd Iranian Conf. on Electrical Engineering*, Tehran, 1995, pp. 241-248.

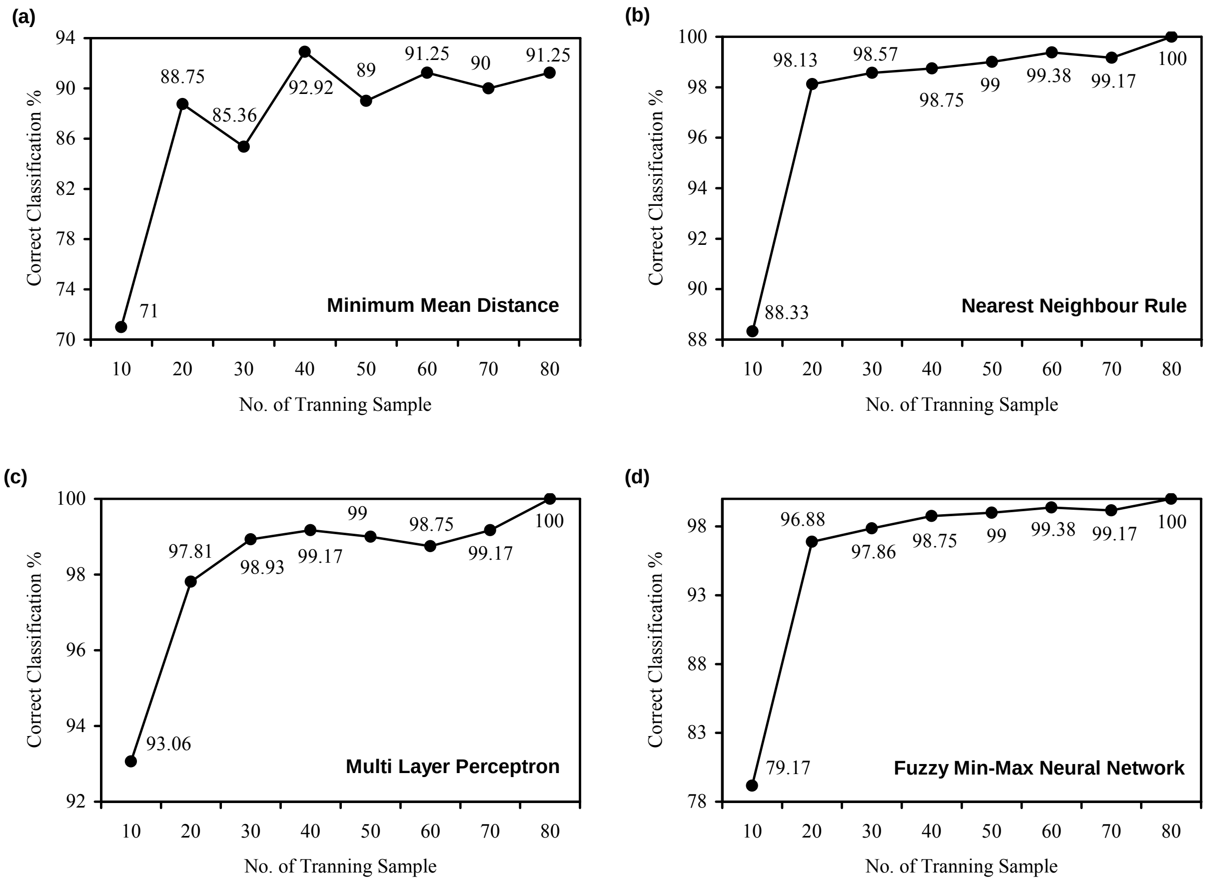


Fig. 11 Correct classification diagram with respect to different numbers of training samples for each classifier. (a) Minimum mean distance (b) Nearest neighbor rule (c) Multi layer Perceptron (d) Fuzzy min-max neural network

[7] K. Faez, E. Khotanzad, and M. H. Shiralishahreza, "Recognition of Persian Handwritten Letters and Digits Using Pseudo-Zernike Moments and Neural Networks," in *Proc. 3rd Iranian Conf. on Electrical Engineering*, Tehran, 1995, pp. 231-240.

[8] H. Nabavi, H. Sadoughi, E. Kabir, and Karlos, "Improvement of Persian Handwritten Digits Recognition Using Composition of Classifiers with ANFIS," in *Proc. 5th Iranian Conf. on Fuzzy Systems*, Tehran, 2004, pp. 337-345.

[9] K. Masrori and V. Pourmohseni, "Recognition of Persian Handwritten Digits Using DWT Algorithm," in *Proc. 4th Nat. Computer Society of Iran Annu. Conf.*, Tehran, 1998, pp. 1-7.

[10] L. C. Liu and C. Y. Suen, "A new benchmark on the recognition of handwritten Bangla and Farsi numeral characters," *Pattern Recognition*, to be published.

[11] A. Harifi and A. Aghagolzadeh, "A New Pattern for Handwritten Persian/Arabic digit Recognition," in *Proc. of World Academy of Sci., Eng. and Technology*, vol. 3, pp. 130-133, Jan. 2005.

[12] S. Mozaffari, K. Faez, and M. Ziaratban, "A Hybrid Structural/Statistical Classifier for Handwritten Farsi/Arabic Numeral Recognition," in *IAPR Conf. on Machine Vision Application*, Tsukuba, 2005, pp. 218-221.

[13] A. Mowlaei and K. Faez, "Recognition of Isolated Handwritten Persian/Arabic Characters and Numerals Using Support Vector Machine," *IEEE XIII Workshop on Neural Networks for Signal Processing*, 2003, pp. 547-554.

[14] J. Sadri, C. Y. Suen, and T. D. Bui, "Application of Support Vector Machines for Recognition of Handwritten Arabic/Persian Digits," in *Proc. 2nd Conf. on Machine Vision and Image Processing*, Tehran, 2003, pp. 300-307.

[15] H. Khosravi and E. Kabir, "Introducing a very large dataset of handwritten Farsi digits and a study on their varieties," *Pattern Recognition Lett.*, vol. 28, pp. 1133-1141, Dec. 2007.

[16] F. Soleimanipour, J. Sadri, and C. Y. Suen, "Standard Databases for Recognition of Handwritten Digits, Numerical Strings, Legal Amounts, Letters and Dates in Farsi Language," in *Proc. 10th Int. Workshop on Frontiers of Handwritten Recognition*, La baule, 2006, pp. 3-7.

[17] S. Mozaffari, K. Faez, F. Faradji, M. Ziaratban, and S. M. Golzan, "A Comprehensive Isolated Farsi/Arabic Character Database for Handwritten OCR Research," in *Proc. 10th Int. Workshop on Frontiers of Handwritten Recognition*, La baule, 2006, pp. 379-384.

[18] H. Ebrahimnezhad and Gh. Montazer, "Presenting Improved Fuzzy Approach for Persian Numeral Fonts Using Entropy Function," in *Proc. 11th Nat. Computer Society of Iran Conf.*, Tehran, 2006, pp. 883-889.

[19] H. Ebrahimnezhad, Gh. A. Montazer, and N. Jafari, "Recognition of Persian Numeral Fonts by Combining the Entropy Minimized Fuzzifier and Fuzzy Grammar," in *Proc. 6th PWASET Int. Conf. on Artificial Intelligence, Knowledge Engineering and Data Bases*, Corfu Island, 2007, pp. 22-27.

[20] H. R. Boveiri, "Persian Printed Numeral Characters Recognition Using Geometrical Central Moments and Fuzzy Min-Max Neural Network," IEEE Trans. Neural Netw., submitted for publication.

[21] H. R. Boveiri, "Classification of Persian Printed digits Using Moment Invariants and Fuzzy Min-Max Neural Network," *14th International CSI Conf.*, Tehran, 2009, submitted for publication.

[22] M. K. Hu, "Visual Pattern Recognition by Moment Invariant," *IRE Trans. Info. Theory*, vol. IT – 8, pp. 179–187, Feb 1962.

[23] Y. Li, "Reforming the Theory of Invariant Moments for Pattern Recognition," *Pattern Recognition Lett.*, vol. 25, pp. 723-730, Jul 1992.

[24] H. R. Boveiri, "Scanned Persian Printed Text Characters Recognition Using Fuzzy-Neural Networks," M. S. thesis, Islamic Azad Univ. Science and Research Branch, Ahvaz, 2009.
[25] S. Alirezaee, H. Aghaeinia, M. Ahmadi, and K. Faez, "Recognition of Middle Age Persian characters Using a Set of Invariant Moments," in *Proc. 33rd Applied Imagery Pattern Recognition Workshop*, Washington DC, 2004, pp. 196-201.
[26] B. Attaran and M. Jahed, "Extraction and Recognition of letters and digits in Vehicle License Plate Using Sobel Filter, Neural Networks, and Hu Moments," in *Proc. 1st Iranian Conf. on Machine Vision and Image Processing*, Birjand, 2001, pp. 287-299.
[27] A. Khotanzad and J-H. Lu, "Classification of Invariant Image Representations using a Neural Networks," *IEEE Trans. on Acoust., Speech and Signal Process*., vol. 38, no. 6, pp. 1028-1038, Jun. 1990.
[28] C. R. Gonzalez and E. R. Woods, *Digital Image Processing*, 2nd Ed., New Jersey: Prentice-Hall Inc., 2002.
[29] M. M. Gupta, L. Jin, and N. Homma, *Static and Dynamic Neural Networks: From Fundamentals to Advance theory*, New Jersey: John Wiley & Sons Inc., 2003.
[30] P. K. Simpson, "Fuzzy Min-Max Neural Networks- Part1: Classification," *IEEE Trans. Neural Netw*., vol. 3, no. 5, pp. 776-786, Sep. 1992.
[31] P. K. Simpson, "Fuzzy Min-Max Neural Networks- Part2: Clustering," *IEEE Trans. Fuzzy Syst*., vol. 1, no. 1, pp. 32-45, Feb. 1993.