Persian Printed Numeral Characters Recognition Using Geometrical Central Moments and Fuzzy Min-Max Neural Network

Hamid Reza Boveiri

Abstract—In this paper, a new proposed system for Persian printed numeral characters recognition with emphasis on representation and recognition stages is introduced. For the first time, in Persian optical character recognition, geometrical central moments as character image descriptor and fuzzy min-max neural network for Persian numeral character recognition has been used. Set of different experiments on binary images of regular, translated, rotated and scaled Persian numeral characters has been done and variety of results has been presented. The best result was 99.16% correct recognition demonstrating geometrical central moments and fuzzy min-max neural network are adequate for Persian printed numeral character recognition.

Keywords—Fuzzy min-max neural network, geometrical central moments, optical character recognition, Persian digits recognition, Persian printed numeral characters recognition.

I. INTRODUCTION

OPTICAL character recognition (OCR) is a subset of pattern recognition which involves different aspects of automatic recognition of written patterns and those techniques converting textural images to the editable text. Optical character recognition system is an image processing system which gets a textural image as input and after processing, it produces corresponding editable text as output.

With respect to how image is acquired, OCR systems are divided into the two different categories. In offline systems, first, a textural document is scanned completely and then its image is processed while in online systems, processing is done during the text is written by optical pen on a digitized tablet. Thus, not only the final image but also extra information like pen speed, acceleration, force, stroke and time-place information are available.

In addition to personal usages, OCR systems make it possible the wide use of computer in processing of huge volume of written data in different departments and institutes. Meanwhile numeral character recognition 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.

The author is with Department of Computer, Islamic Azad University Shushtar Unit, Shushtar, IR (e-mail: hr.boveiri@yahoo.com).

Fig. 1 shows Persian numeral characters. With respect to Persian "seven" and "eight" has reversal shape of each other and similarity of "two", "four" and "six"; Persian numeral characters recognition has special complexities. Regarding to importance and extensive application, many researches [1]-[13] have been done in the field of Persian numeral characters recognition and many data sets [14]-[16] have been introduced. Unfortunately, all of them are on handwritten digits and printed digits have not been attended a lot.

Ebrahimnezhad *et al.* [17] presented a fuzzy approach based on use of entropy function to improve fuzzifier function definition for Persian digits recognition irrespective of font and size. Box method was used for feature vector generation. The system was tested by various fonts and finally achieved 97.5% correct recognition. In [18] they extended their system, they presented a synthesis approach that uses three fuzzifier function simultaneously, and with post processing using another fuzzy system, they reported complete correct recognition. Although, "zero" has not been used in their experiments and utilized data set is unknown.

Geometrical moments are used a lot as region descriptor in image processing and pattern recognition. With image translation to coordinate origin while computing geometrical moments and then normalization, geometrical central moments are generated which are invariant under translation and scaling [19]. These moments are relatively sensitive to rotation which is necessary for recognition of some Persian numerals. They are also invariant under image reversal and can not discriminate some Persian numerals like "seven" and "eight". In these cases, usually feature vector enrichment is used.

Artificial neural network (ANN) has been inspired from biological neural structure of human brain. Although, ANN is very simple abstraction of its biological counterpart, but it has been interested a lot because of its extensive power in pattern classification and clustering in the resent years. Fuzzy logic is a consistent logic to encounter ambiguity and uncertainty. The world around us has a lot of these imprecisions, which arise from natural phenomena and human perception. In fuzzy logic, a predicate is not necessarily true or false and it can be relatively true and false with respecting to conditions. Aggregation of neural networks and fuzzy logic generates a new heuristic tool called fuzzy-neural networks (FNN), which

has a lot of potential power in solving problems related to human intelligence [20]. In an OCR system, FNN seems to be adequate; neural network recognizes characters while fuzzy logic can help it to encounter ambiguity. These ambiguities may arise from image noise, undesirable connections, character similarities etc.

Simpson first introduced fuzzy min-max neural network for pattern classification and clustering [21], [22]. Properties like no need to adjust fuzzy sets by domain expert, online adaptation, nonlinear separability, high ability to encounter overlapping classes, short training time, existence of parameters for better adaptation to input data and successful results in Persian printed letters recognition in [23] are good reasons for use this kind of FNN in this research.

The remainder of paper is organized as follow. In the next section, proposed system and its structure are described. Section III surveys preprocessing stage. Section IV gives detail description of geometrical central moments and feature vector enrichment in representation stage. In section V, fuzzy min-max neural network and its properties are introduced. Section VI reports the experimental results and finally, the conclusions are outlined in section VII.

Zero	One	Two	Three	Four
٠	١	٢	٣	۴
Five	Six	Seven	Eight	Nine
۵	۶	٧	٨	٩

Fig. 1. Persian numeral characters

II. PROPOSED SYSTEM

Fig. 2 shows the structure of proposed system, just like every classical image processing system [24], the proposed system is composed of several isolated processing stage. In preprocessing stages, the operations like noise reduction and skew correction are done for image enhancement. In representation stage, geometrical central moments are used as image descriptor. Also in this stage, feature vector is enriched using an extra statistical feature. In the last stage, extracted features are fed into a fuzzy min-max neural network and characters are recognized.
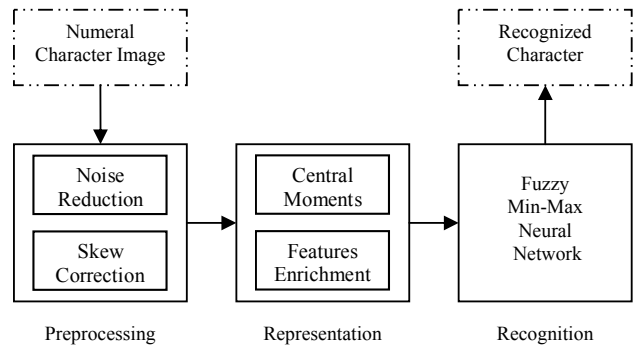

Fig. 2. Structure of proposed system

III. PREPROCESSING

Geometrical central moments are sensitive to noise [25] and skew [19]. Thus in this stage, noise reduction and skew correction are done to compensate these defects.

A. Noise Reduction

A nonlinear mean 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 [24].

B. Skew Correction

Principle component analysis (PCA), using geometrical central moments, has been used to estimate skew angle [19]. Skew angle from principle axes of image is computed by geometrical central moments from order 2 using (1).

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

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

IV. REPRESENTATION

Geometrical central moments have been used as character image descriptor. These moments are nonlinear, invariant under translation, scaling and image reversal. With regard to reversal shape of "seven" and "eight", central moments can not discriminate them. So 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. Geometrical Central Moments

Geometrical moment from order $(p + q)$ for a two dimensional discrete function like image is computed 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 [20].

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

Where f(x, y) is image function and M, N are image dimensions. With a little change in (2), geometrical central moments from order (p + q) can be computed using (3).

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

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

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

Note that in a binary image, M_{00} is count of foreground pixels and has direct relation to image scaling, so central moments can become scale normalized using (5). Table I lists geometrical central moments from different orders (less than 11) and their corresponding numbers.

$$M_{pq} = \frac{M_{pq}}{m_{00}^a} \quad , \quad a = \frac{p+q}{2} + 1 \qquad (5)$$

TABLE I
GEOMETRICAL CENTRAL MOMENTS FROM DIFFERENT ORDERS (LESS THAN 11) AND THEIR CORRESPONDING NUMBERS

Order	Central Moments	No. of Moments
0	M_{00}	1
1	M_{01}, M_{10}	2
2	M_{11}, M_{02}, M_{20}	3
3	M_{03}, M_{12}, M_{21}, M_{30}	4
4	M_{04}, M_{13}, M_{22}, M_{31}, M_{40}	5
5	M_{05}, M_{14}, M_{23}, M_{32}, M_{41}, M_{50}	6
6	M_{06}, M_{15}, M_{24}, M_{33}, M_{42}, M_{51}, M_{60}	7
7	M_{07}, M_{16}, M_{25}, M_{34}, M_{43}, M_{52}, M_{61}, M_{70}	8
8	M_{08}, M_{17}, M_{26}, M_{35}, M_{44}, M_{53}, M_{62}, M_{71}, M_{80}	9
9	M_{09}, M_{18}, M_{27}, M_{36}, M_{45}, M_{54}, M_{63}, M_{72}, M_{81}, M_{90}	10
10	$M_{0,10}$, M_{19}, M_{28}, M_{37}, M_{46}, M_{55}, M_{64}, M_{73}, M_{82}, M_{91}, $M_{10,0}$	11

Normalized central moments are relatively sensitive to rotation. This sensitivity is logical and can help system to separate numerals like "two", "four" and "six". Table II lists central moments from order less than three for different status of character in fig. 7 after preprocessing. Where μ and σ are sample mean and sample standard deviation respectively and σ/μ % is percentage of spread of central moment's values from their corresponding means. Because of wide range of central moments, logarithms of their magnitudes have been used.

TABLE II
CENTRAL MOMENTS FROM ORDER LESS THAN THREE FOR DIFFERENT STATUS OF CHARACTER IN FIG. 7 AFTER PREPROCESSING

	M_{00}	M_{01}	M_{02}	M_{10}	M_{11}	M_{20}
Fig. 7a	0	-37.477	-1.447	-34.547	-6.3452	-2.4484
Fig. 7b	0	-36.791	-1.477	-34.974	-6.9273	-2.4167
Fig. 7c	0	-36.272	-1.590	-34.348	-6.6289	-2.5054
Fig. 7d	0	-36.747	-1.463	-34.907	-5.6857	-2.5463
μ	0	-36.822	-1.494	-34.694	-6.3968	-2.4792
σ	0	0.4959	0.065	0.2972	0.5303	0.0579
σ/μ %		1.3467	4.3487	0.8567	8.29	2.3339

B. Feature Vector Enrichment

Persian "seven" and "eight" are vertically reverse of each other (see fig. 1). With respect to invariance of central moments under image reversal, they can not discriminate these characters. Thus, feature vector is enriched using 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 vertically reverse of each other, so relation of their upper half of image pixels to lower half of image pixels is completely different which has also been shown in fig. 3.
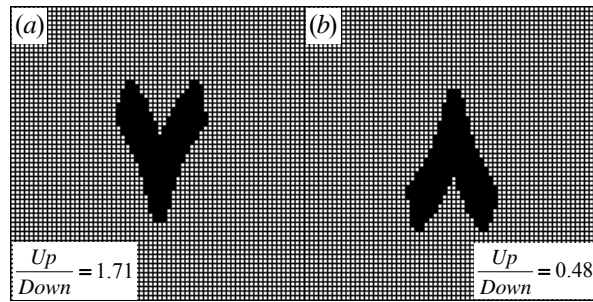

Fig. 3. (a) Persian "seven". (b) Persian "eight". With computed extra feature for them

V. RECOGNITION USING FUZZY MIN-MAX NEURAL NETWORK

In this paper, fuzzy min-max neural network (FMMNN) [21] has been used for recognition of Persian numeral characters. FMMNN 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 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.
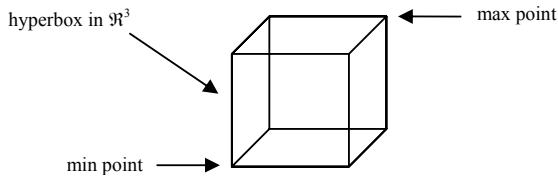

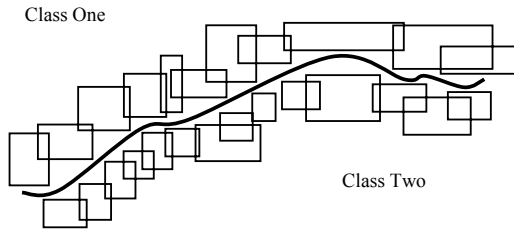
Fig. 4. The min-max three-dimensional hyperbox [21]

Fig. 5. Separation of two different classes by two-dimensional hyperboxes [21]

A. FMMNN Properties

Some properties of FMMNN, which are good reasons for selecting this kind of FMM in this paper, 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. θ parameter is a user defined value that bounds the maximum size of hyperboxes during the FMMNN training. Lower values of θ 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. γ 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. In all experiments, it is assumed that $\theta=0.1$ and $\gamma=1$ unless where it is expressed explicitly.

VI. IMPLEMENTATION AND EXPERIMENTAL RESULTS

The proposed system was implemented on a Penitum4 (2.6GHz) desktop computer with Windows XP (SP2) platform using version 6.0 of Microsoft Visual Basic programming language. Fig. 6 depicts time diagram needed for extracting different orders of central moments from a typical 64×64 character image. Mean of 10 times of algorithm execution has been used.
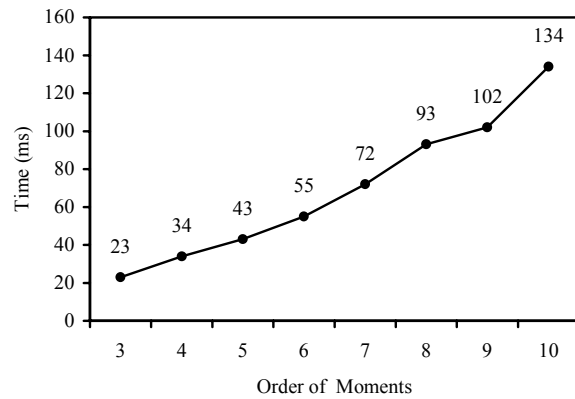

Fig. 6. Time diagram for extracting different orders of central moments from a typical 64×64 character image. The numbers on the top of the diagram represent the precious time values in millisecond (ms)

A. The Utilized Data Set

The data set consists of 64×64 binary images of all 10 Persian numeral characters in four groups. The first group was regular characters with the 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 which they were totally 400 samples (40 samples per character). Fig. 7 shows a sample image of each group of Persian "three" in the data set. In the all experiments, half of samples have been used for training and the remainders as test data unless where it is expressed explicitly.
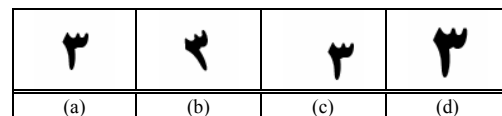

Fig. 7. A sample image of each group of Persian "three" in the data set. (a) Regular (b) Rotated (c) Translated (d) Scaled.

B. Effect of Size of Feature Vector

Fig. 8 illustrates correct recognition diagram with respect to using different orders of central moments to generate the feature vector. In each experiment, lesser orders of central moments have also been added to the feature vector. M_{00} has been ignored because of its constant value after normalization. In addition, no feature vector enrichments has been used. The numbers under diagram are size of utilized feature vector and the above ones present count of the generated hyperboxes

during FMMNN training. The best result (91%) was achieved using feature vector consists of central moments from order four and the lesser ones. Table III shows the corresponding confusion matrix. Where rows and columns are input character and recognized character respectively. It is clear that source of the most errors is similarity of "four" and "six" and reversal shape of "seven" and "eight".
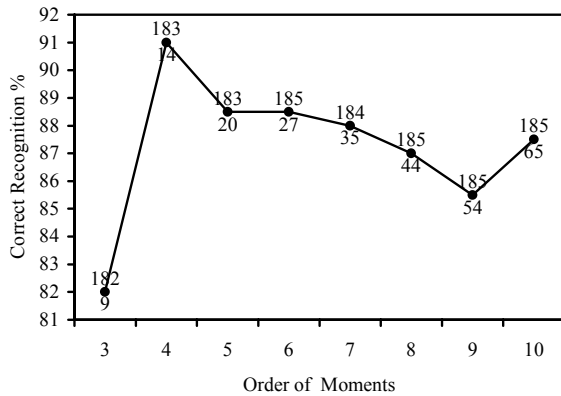

Fig. 8. Correct recognition diagram with respect to using different orders of central moments to generate the feature vector. The numbers under diagram are size of utilized feature vector and the above ones present count of the generated hyperboxes during FMMNN training

TABLE III
CONFUSION MATRIX OF THE EXPERIMENT USING THE FEATURE VECTOR
CONSISTS OF CENTRAL MOMENTS FROM ORDER FOUR AND THE LESSER ONES.

	٠	١	٢	٣	۴	۵	۶	٧	٨	٩
٠	20									
١		20								
٢			20							
٣				20						
۴					20					
۵			1			19				
۶					2	1	16			1
٧								15	5	
٨								7	13	
٩							1			19

C. Effect of Enrichment of Feature Vector

Fig. 9 illustrates correct recognition diagram after feature vector enrichment with respect to using different orders of central moments to generate the feature vector. These results, in comparison with diagram of fig. 8, demonstrate improvement of correct recognition after the feature vector enrichment in the all orders. It these experiments, the best result (93.5%) was also achieved using the feature vector consists of central moments from order four and the lesser ones. Therefore, this feature vector seems to be adequate and it is used in the next experiments.
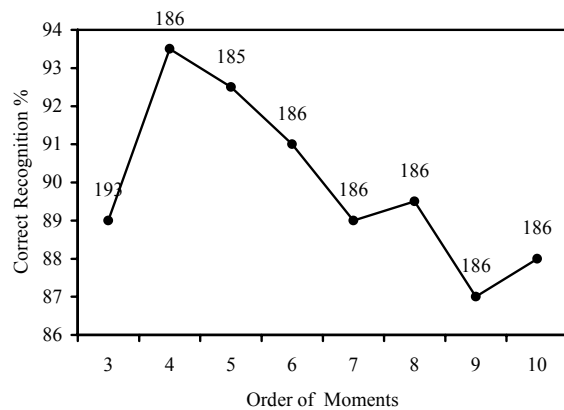

Fig. 9. Correct recognition diagram after feature vector enrichment with respect to using different orders of central moments to generate the feature vector. The numbers above the diagram present count of the generated hyperboxes during FMMNN training

Whereas size of feature vector generated by central moments is relatively large, so adding the extra feature only one time to the feature vector has a little effect. Therefore, in the next experiments, the extra feature has been augmented to the feature vector several times to increase its effect. Fig. 10 illustrates the results of these experiments using central moments from order four and the lesser ones. The best result (97%) was achieved by adding the extra feature two times to the feature vector. Table IV shows the corresponding confusion matrix. It is clear that feature vector enrichment has an excellent effect in correct recognition of "seven" and "eight".
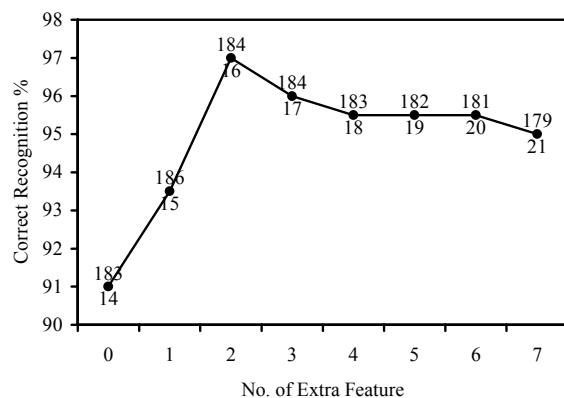

Fig. 10. Correct recognition diagram using feature vector consists of central moments from order four and the lesser ones with respect to adding the extra feature several times to the feature vector. The numbers under diagram are size of utilized feature vector and the above ones present count of the generated hyperboxes during FMMNN training

TABLE IV
CONFUSION MATRIX OF THE EXPERIMENT USING THE FEATURE VECTOR CONSISTS OF CENTRAL MOMENTS FROM ORDER FOUR AND THE LESSER ONES AFTER ENRICHMENT

	٠	١	٢	٣	٤	٥	٦	٧	٨	٩
٠	20									
١		20								
٢			20							
٣				20						
٤			1	19						
٥						20				
٦				2		17				1
٧							19	1		
٨								20		
٩					1					19

D. Fuzzy Min-Max Neural Network Parameters

Fig. 11 illustrates correct recognition 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. The results (97%) were not improved using values less than "0.1" which seems to be the best value for this parameter.

Fig. 12 illustrates correct recognition diagram with respect to different values of γ parameter (sensitivity parameter). The results (97%) was not improved using values less than "1" which is the best value for this parameter (decreasing this parameter increases the hyperboxes sensitivity).
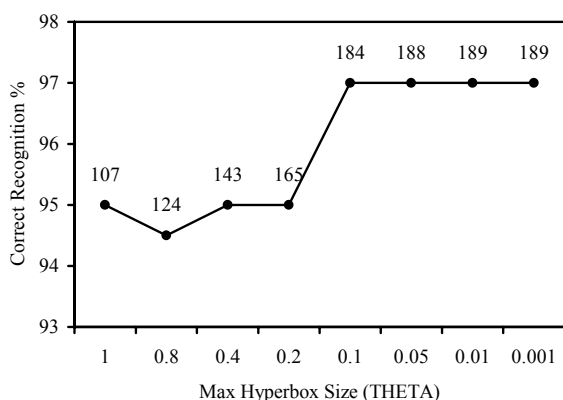

Fig. 11. Correct recognition diagram with respect to different values of θ parameter (maximum size of hyperboxes). The numbers above the diagram present count of the generated hyperboxes during FMMNN training which increase as the value of θ decreases
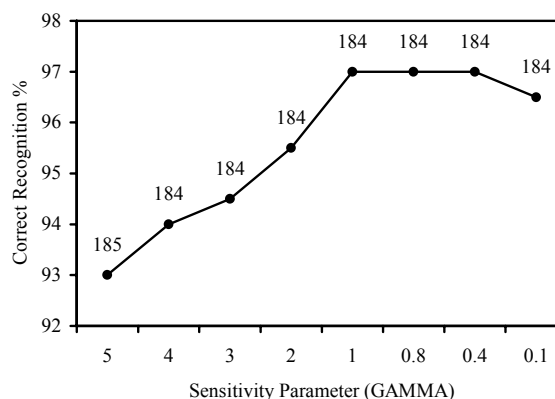

Fig. 12. Correct recognition diagram with respect to different values of γ parameter (sensitivity parameter). The numbers above the diagram present count of the generated hyperboxes during FMMNN training

E. Effect of Size of Training Samples

Fig. 13 illustrates correct recognition diagram with respect to different values of training samples. With comparing count of generated hyperboxes in each experiment (the above numbers of diagram) we find that, using more training sample, more hyperboxes are generated which usually can help system to better recognition. In each experiment, reminder samples were used as test samples. The best result (99.16%) was achieved using 70% of samples for training and the remainder 30% for testing.
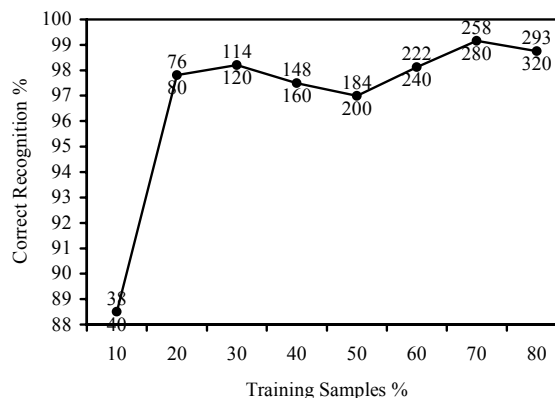

Fig. 13. Correct recognition diagram with respect to different values of training samples. The above numbers of diagram present count of the generated hyperboxes during FMMNN training and the under ones are count of training samples

VII. CONCLUSIONS AND FUTURE WORK

A new proposed system for Persian printed numeral characters recognition using geometrical central moments and fuzzy min-max neural network was introduced. In the preprocessing stage, a nonlinear mean filter was used for noise reduction. In addition, with principle component analysis

using geometrical central moments, image skew was corrected. In the representation stage, geometrical central moments were used as image descriptor. Whereas these moments are invariant under image reversal, thus feature vector was enriched using an extra statistical feature to discriminate the characters like "seven" and "eight". In the recognition stage, a fuzzy min-max neural network was used for characters images classification. Set of different experiments was done and various results were obtained can be summarized as follow.

1) Feature vector consists of geometrical moments from order four and the lesser ones had better results.
2) Most errors occurred because of similarity of "four" and "six" plus reversal shape of "seven" and "eight".
3) Feature vector enrichment is necessary to discriminate reversal characters.
4) The extra feature should be added several times to the feature vector for increasing its effect.
5) In the proposed system, "0.1" and "1" seem to be adequate as θ and γ parameters respectively.
6) Usually the better results are achieved by increasing of training samples.

Finally, the best result was 99.16% correct recognition which shows geometrical central moments and fuzzy min-max neural network are adequate for Persian printed numeral character recognition.

Suggested future work may include testing of the system with Persian handwritten numerals and introducing a standard data set of Persian multi-font printed digits and sharing it for the future researches.

REFERENCES

[1] 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.
[2] H. Ketabdar, "Persian Handwritten Digits Recognition with Structural Approach," in Proc. 6th Iranian Conf. on Electrical Engineering, Tehran, 1998, pp. 4.19-4.24.
[3] 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.
[4] 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.
[5] 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.
[6] 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.
[7] 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.
[8] 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.
[9] 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.
[10] A. Harifi and A. Aghagolzadeh, "A New Pattern for Handwritten Persian/Arabic digit Recognition," Proc. of World Academy of Sci., Eng. and Technology, vol. 3, pp. 130-133, Jan. 2005.
[11] 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.
[12] 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.
[13] 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.
[14] 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.
[15] 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.
[16] 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.
[17] 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.
[18] 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.
[19] M. K. Hu, "Visual Pattern Recognition by Moment Invariant," IRE Trans. Info. Theory, vol. IT – 8, pp. 179–187, Feb 1962.
[20] 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.
[21] P. K. Simpson, "Fuzzy Min-Max Neural Networks- Part1: Classification," IEEE Trans. Neural Netw., vol. 3, no. 5, pp. 776-786, Sep. 1992.
[22] P. K. Simpson, "Fuzzy Min-Max Neural Networks- Part2: Clustering," IEEE Trans. Fuzzy Syst., vol. 1, no. 1, pp. 32-45, Feb. 1993.
[23] 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.
[24] C. R. Gonzalez and E. R. Woods, Digital Image Processing, 2nd Ed., New Jersey: Prentice-Hall Inc., 2002.
[25] 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.