A Hand Geometry Biometric Identification System Utilizing Modular Neural Networks with Fuzzy Integration

José Luis Sánchez and Patricia Melin

Tijuana Institute of Technology, Tijuana, México
pmelin@tectijuana.mx

Abstract. The present work deals with the problem of identifying individuals from a database, and in so doing utilizing measurements taken from handpalm images. The techniques utilized for performing identifications are mainly those of artificial neural networks, which work upon the data through the use of two modular neural networks, one which is concerned solely with the handpalm image, another with the measurements taken thereof. Outputs from these two networks are integrated through a fuzzy inference system. Subsequent work will comprise improvement of the obtained results.

Keywords: Biometry, Fuzzy Logic, Modular Neural Networks.

1 Introduction

In recent decades an ever increasing number of automatic recognition methods have been developed, all of which are intended to help manage and control access to many different goods, places and services. Computational Intelligence paradigms such as Artificial neural networks (ANN) and Fuzzy systems (based on fuzzy logic) have proven invaluably useful when applied to pattern recognition problems, as well as being able to perform at very good levels of performance when dealing with such problems, these being the reason they are used in this work.

Features to be found in a human's hands are attractive as a means to build upon for the construction of methods for recognition; not the least of qualities associated with them are their permanence and uniqueness.

Among these methods only two will be mentioned: palmprint recognition and hand geometry recognition [3].

Palmprint recognition is an amplification of fingerprint recognition methods, and may or may not build upon the presence of three principal lines on most everyone's hands.

This kind of methods tend to be very precise, yet spend sizable amounts of computing power.

O. Castillo et al. (Eds.): Recent Advances on Hybrid Intelligent Systems, SCI 451, pp. 183–199.
springerlink.com

Hand geometry methods, in their original implementation, look to make several measurements from images of the outline of the hand (these images are taken while the hand is on a level surface and placed between unmovable pegs, to improve measurements).

The measurements commonly include: length of fingers, palm width, knuckle size.

Commonly, a very reduced amount of memory is needed for a single individual, but identification tasks aren't as accurately resolved through such methods, more so if the database is really large; for verification purposes, performance can be good [3].

The rest of the paper is organized as follows: Section 2 describes the tools used for this work and what they are about, section 3 describes the methodology applied for the resolution of the stated problem, section 4 describes the results obtained so far, section 5 gives the conclusions drawn so far, along with ideas for further work.

2 Computational Intelligence Tools

There is an enormous amount of Computational Intelligence tools that are useful for pattern recognition purposes. Mentioned below are those that are important to this work.

2.1 Fuzzy Logic

Fuzzy logic follows a path unlike that of traditional logic, allowing not only two possible states for a situation, but an infinite gamut between those two traditional states, which in a sense approaches more closely the way in which humans use language and what has traditionally been considered a byproduct of language: reasoning.

From this particular standpoint, fuzzy logic builds until it reshapes every tool of logic, including deductive inference [2].

2.2 Fuzzy Systems

A fuzzy inference system (FIS) is a system that takes advantage of fuzzy logic to create and control models of complex systems and situations without making use of ground-up specifications, which are many times unavailable and relying instead on descriptions more or less at a human language level.

A much used tool for working with fuzzy systems is that included within the Matlab programming language [2].

2.3 Artificial Neural Networks

Artificial neural networks are much simplified models of generic nervous systems, studied with the idea of applying them for the same tasks organisms excel at, including pattern recognition.

Nowadays there are many kinds of ANNs, but all build upon using lots of simple elements (neurons) capable of being interconnected and of maintaining weighted responses to “stimuli” between those connections; the similarities end there for many types of ANNs.

The afore mentioned networks are tipically divided into layers, and for those nets needing training to get the desired response to a given input, such training is accomplished through a training algorithm, like the much used backpropagation algorithm.

There would be a last stumbling stone to consider when in need to have a certain output derive from a given input: not all network architectures are equally successful at attaining this goal [5].

2.4 Modular Neural Networks (MNN)

It is possible to divide a problem so that several networks (modules or experts) work separately to solve it. Much needed improvements can be obtained with these MNN, such as a lower training time or an efficiency boost.

The final step in getting an answer to a given problem with an MNN involves the integration of individual results, to mention just a few (not all applicable to every MNN): gating networks, voting, weighted average, winner-takes-all.

Winner takes all is possibly the simplest method and works by considering the highest ranking results in each module and comparing all such results, allowing only the very highest ranking to remain and be part of the final result [8].

It is also possible to have a fuzzy system integrate responses to several modules, according to the system’s rules and the compared modules’ results.

3 Methodology

The following items were necessary aspects for this work's problem solving methodology:

1. Having a database.
2. Having a general architecture for the system.
3. Having an architecture for each of the used modular neural networks.
4. Having an architecture for each of the modules within a modular neural network.
5. Having preprocessing modules for each of the MNNs.
6. Having an architecture for each of the used fuzzy integrators.

7. Having fuzzy rules for each of the fuzzy integrators.
8. Train all modules of every modular neural network and integrate results.

3.1 Utilized Database

The database used with this work is the Polytechnic University of Hong Kong's multispectral palmprint database (MS PolyU database) [7].

The images contained in the database are very nearly fingerless, and the middle and ring fingers are opened wide due to the presence of a central metallic stud; each image has a size of 352 by 288 pixels.

The database has four directories with a total of 6000 images each, taken from 250 volunteers.

The database gets its name from the volunteers being taken an image of every hand four times, under red, green, blue or infrared illumination.

The only directory used for this work is the "Red" one.

3.2 The General System's Architecture

The system comprises two modular neural networks, one dealing with the images and the other with the geometric measurements, with each of them having a different preprocessing for the whole dataset. At the end, the results of both MNNs are integrated by a FIS, as shown in Fig. 1:

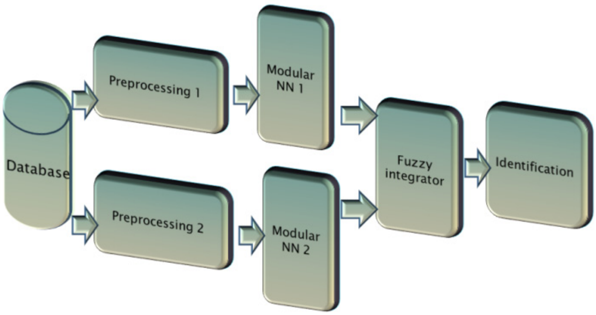

Fig. 1. Complete system's achitecture

3.3 Modular Neural Network 1's Architecture

This MNN is comprised of ten modules, which equally divide their dataset, along with an integrator (winner takes all).

The layout for this MNN is shown in Fig.2:

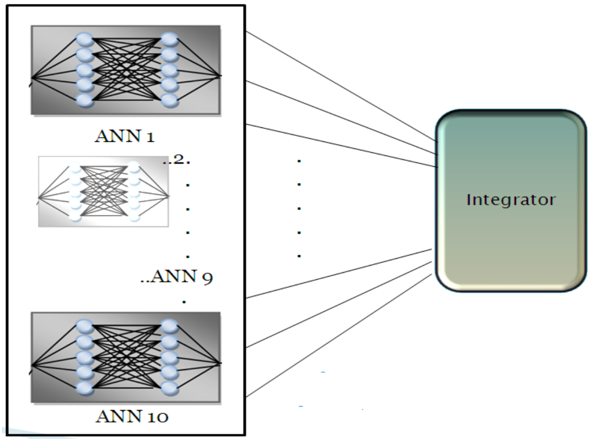

Fig. 2. MNN 1's architecture

3.3.1 MNN 1's Parameters by Module

The parameters for this MNN are as follows (all further networks share these parameters, except the number of neurons):

1. Number of layers: 3
2. Training function: scaled conjugate gradient (though in the first series of trainings for this MNN, it was Fletcher's conjugate gradient).
3. Transference function, layer 1: Tangent sigmoid
4. Transference function, layer 2: Logarithmic sigmoid
5. Transference function, layer 3: Linear
6. Number of neurons, layer 1: 350
7. Number of neurons, layer 2: 175
8. Number of neurons, layer 3: 50

As with all modules, the training set is 58% of the whole dataset.

3.4 Preprocessing for MNN 1

The preprocessing consists (as shown in Fig. 3) of:

1. Equalization of histogram.
2. Edge detection.
3. Reduction of images.
4. Binarizing and standardization of images.

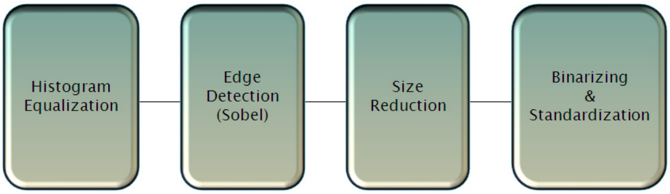

Fig. 3. Block diagram for MNN 1's preprocessing

3.4 Modular Neural Network 2's Architecture

This MNN is comprised of four modules, which equally divide their dataset, along with an integrator (winner takes all).

The layout for this MNN is shown in Fig. 4.

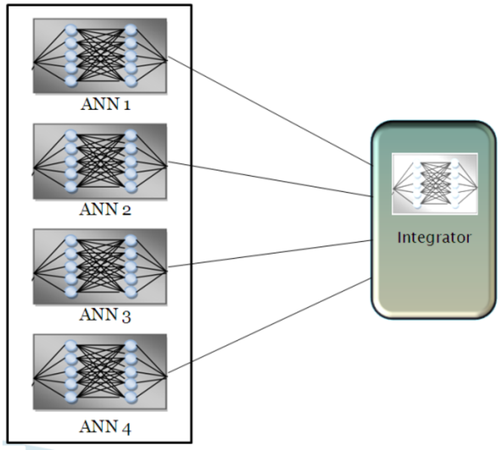

Fig. 4. MNN 2's architecture

3.4.1 MNN 2's Parameters by Module

1. Number of neurons, layer 1: 875
2. Number of neurons, layer 2: for succesive series, 60, 654, 1200
3. Number of neurons, layer 3: 125

(At the beginning, the input to these modules were formatted to Gray code, but it proved unfruitful).

3.4.2 MNN 2's Integrator's Network

When it was evident that results of modules in this MNN were ill-scaled for integration, a NN was developed that could tell which module should be carrying an individual's identity, by first placing each one in the appropriate fourth of the dataset, and therefore the right module.

This network has the following parameters:

1. Number of neurons, layer 1: 3500
2. Number of neurons, layer 2: 34
3. Number of neurons, layer 3: 4

Success rate for such a net is around 95% and it took some 20 hours of training in a 2.7 MHz dual core machine with 2 Gb of RAM.

3.4.3 Preprocessing for MNN 2

Preprocessing for this modular neural network consists of:

1. Custom filtering image to enhance principal lines.
2. Getting edges of above mentioned image.
3. Getting outline of the whole palm.
4. Finding valleys near index and small fingers, with a process shown in Fig. 5:

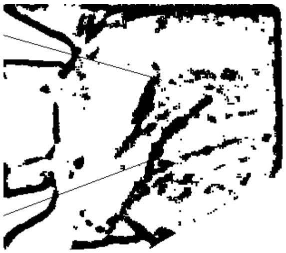

Fig. 5. Finding outer valleys

5. Getting the finger baseline.
6. Getting finger width aided by prior steps plus location of central reference (stud).
7. Starting from fixed points unique to all images, three for each principal line, the nearest points to them, presumably belonging to the principal lines are found and three splines are traced through them. This is necessary because edge detection leaves segments too fragmented.
8. Center of each principal line is located with aid from its spline.
9. Total length of each spline is found.
10. Distance from each "center" to central reference is found.
11. Wrist width is found.
12. Distances from each splines extremes to central reference is found.

Figs. 6 and 7 show an image prior to spline tracing and other, with its spline overlayed (notice the central reference):

Fig. 6. Image of origin

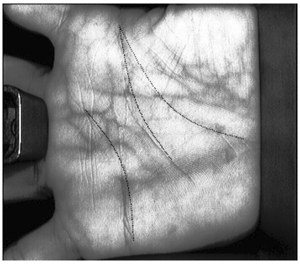

Fig. 7. Overlayed image

Hands that were left were reflected to be right. Preprocessing of the whole dataset took 1.91 hours.

3.5 *Fuzzy Integrator*

The output from both MNNs is used as input to a fuzzy system to give the final system's response. The integrator is of Mamdani type (Fig. 8):

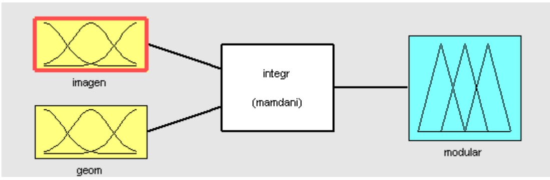

Fig. 8. Fuzzy integrator's architecture

Input variables are "imagen" and "geom", with three membership functions (MF's) each, as shown plotted in Figs. 9 and 10:

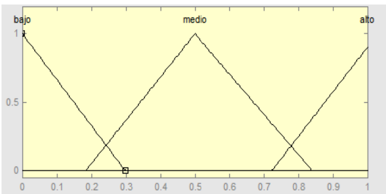

Fig. 9. "imagen" variable

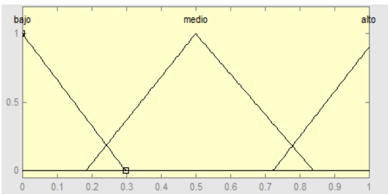

Fig. 10. "geom" variable

Where the MF's "bajo", "medio", "alto" mean "low", "medium", "high". The output variable is "modular" (Fig 11):

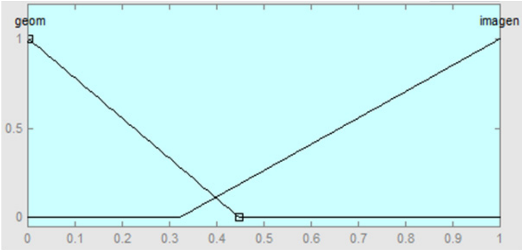

Fig. 11. "modular" variable

3.5.1 Fuzzy Rules

There's a total of nine fuzzy rules:

1. If (imagen is bajo) and (geom is bajo) then (modular is imagen)
2. If (imagen is bajo) and (geom is medio) then (modular is imagen)
3. If (imagen is bajo) and (geom is alto) then (modular is geom)
4. If (imagen is medio) and (geom is bajo) then (modular is imagen)
5. If (imagen is medio) and (geom is medio) then (modular is imagen)
6. If (imagen is medio) and (geom is alto) then (modular is geom)
7. If (imagen is alto) and (geom is bajo) then (modular is imagen)
8. If (imagen is alto) and (geom is medio) then (modular is imagen)
9. If (imagen is alto) and (geom is alto) then (modular is imagen)

3.5.2 Fuzzy System # 2

Another fuzzy system was created starting from the first one, with the same rules, variables and membership functions, the difference being that instead of triangular membership functions, generalized bells were used.

Figs. 12 and 13 graphically show the input variables:

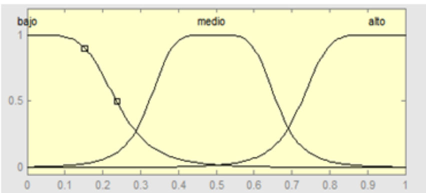

Fig. 12. "imagen" variable

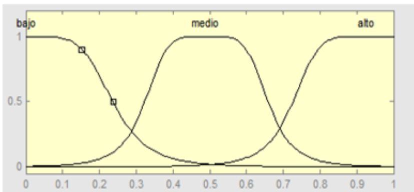

Fig 13. "geom" variable

Fig. 14 shows a plotting of the output variable:

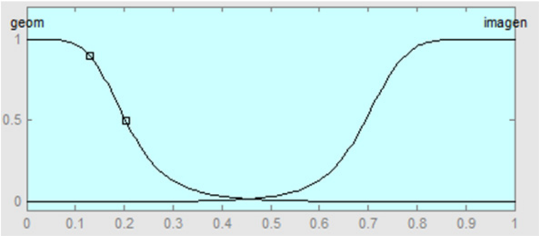

Fig. 14. "modular" variable

4 Results

The following are several tables for series of ten trainings each with the best identification performer marked in blue.

4.1 Results for the MNN 1

Results for the first series are shown in Table 1; the best success rate (SR) was 93.33%.

Table 1. Results for MNN 1, series 1

Training	Σ Epochs	training time	ident. time	identified	success rate
1	4316	41.649 min	18.2500 s	5526	92.10
2	4124	40.190 min	13.8187 s	5527	92.12
3	3887	37.867 min	14.0418 s	5552	92.53
4	3711	35.797 min	14.9982 s	5600	93.33
5	4198	40.662 min	15.1318 s	5527	92.12
6	4087	39.447 min	17.9906 s	5526	92.10
7	4254	41.458 min	14.5147 s	5503	91.71
8	4205	40.583 min	14.6658 s	5517	91.95
9	4076	39.707 min	14.0422 s	5526	92.10
10	4069	39.273 min	14.1105 s	5587	93.11

Results for the second series are shown in Table 2; the best SR was 93.74%.

Table 2. Results for MNN 1, series 2

Training	Σ Epochs	training time	ident. time	identified	success rate
1	4220	40.919 min	20.465 s	5558	92.63
2	4005	39.028 min	13.930 s	5528	92.13
3	3798	36.832 min	14.520 s	5556	92.60
4	3954	38.229 min	15.065 s	5589	93.15
5	4143	40.055 min	16.562 s	5624	93.74
6	4170	38.452 min	19.728 s	5511	91.85
7	4229	41.021 min	14.591 s	5545	92.42
8	4143	40.145 min	14.978 s	5524	92.07
9	4072	39.490 min	14.076 s	5526	92.09
10	4193	40.461 min	15.180 s	5562	92.70

Results for the third series are shown in Table 3; the best SR was 97.73%.

Table 3. Results for MNN 1, series 3

Training	Epochs	training time	ident. time	identified	success rate
1	2151	29.1508 min	4.5072 s	5860	97.67
2	2177	29.1985 min	4.3679 s	5864	97.73
3	2169	28.8461 min	4.6182 s	5861	97.68
4	2181	28.8524 min	4.8882 s	5861	97.68
5	2151	28.5450 min	5.1730 s	5864	97.73
6	2178	28.8729 min	4.3149 s	5856	97.60
7	2180	28.9235 min	4.4670 s	5858	97.63
8	2185	28.9521 min	4.3540 s	5860	97.67
9	2174	28.8521 min	4.3386 s	5860	97.67
10	2113	28.0448 min	4.4143 s	5851	97.52

4.2 Results for the MNN 2

Results for the first series are shown in Table 4; the best SR was 71.74%.

Table 4. Results for MNN 2, series 1

Training	Σ Epochs	training time	ident. time	identified	success rate
1	3422	74.684 min	31.571 s	4280	71.33
2	7897	173.03 min	31.555 s	4057	67.62
3	4079	88.96 min	31.861 s	4304	71.74
4	4955	108.97 min	32.033 s	4071	67.85
5	6017	131.71 min	31.83 s	4087	68.12
6	5762	126.30 min	32.2514 s	4091	68.18
7	4051	88.757 min	32.1057 s	4053	67.55
8	3386	74.188 min	31.729 s	4063	67.72
9	3407	74.687 min	31.4619 s	4078	67.97
10	3846	83.492 min	31.903 s	4029	67.15

Results for the second series are shown in Table 5; the best SR was 77.95%.

Table 5. Results for MNN 2, series 2

Training	Σ Epochs	training time	ident. time	identified	success rate
1	5874	86.820 min	31.779 s	4653	77.550
2	6289	87.635 min	31.785 s	4670	77.833
3	4811	85.611 min	31.924 s	4655	77.583
4	5487	90.340 min	31.932 s	4668	77.800
5	5879	89.005 min	32.041 s	4661	77.683
6	4906	102.528 min	32.178 s	4677	77.950
7	3818	71.473 min	31.917 s	4662	77.699
8	3494	57.437 min	31.595 s	4673	77.883
9	3627	79.091 min	31.682 s	4669	77.817
10	3851	92.051 min	31.974 s	4675	77.917

Results for the third series are shown in Table 6; the best SR was 80.45%.

Table 6. Results for MNN 2, series 3

Training	Σ Epochs	training time	ident. time	identified	success rate
1	4990	250.0487 min	14.6676 s	4813	80.22
2	4857	232.5927 min	14.5411 s	4812	80.20
3	4978	236.7514 min	14.5536 s	4817	80.28
4	4926	240.5805 min	14.6846 s	4809	80.15
5	4970	238.6703 min	14.8786 s	4807	80.12
6	5049	218.5406 min	14.6734 s	4813	80.22
7	4992	222.8399 min	14.6320 s	4810	80.17
8	4979	227.9744 min	14.6869 s	4816	80.27
9	5010	223.3908 min	14.6582 s	4827	80.45
10	4991	224.7490 min	14.9844 s	4814	80.23

4.3 Complete System Results

Results for the first series are shown in Table 7; the best SR was 95.25%.

Table 7. Complete system results, series 1

Training	Σ Epochs	training time	ident. time	identified	success rate
1	7738	116.33 min	45.324 s	5712	95.20
2	12021	213.22 min	44.519 s	5690	94.83
3	7966	126.80 min	45.693 s	5715	95.25
4	8665	144.77 min	45.227 s	5617	93.62
5	10215	172.37 min	47.631 s	5671	94.51
6	9849	165.75 min	45.086 s	5652	94.20
7	8305	130.22 min	44.479 s	5608	93.47
8	7591	114.77 min	47.208 s	5514	91.90

Table 7. (*continued*)

9	7483	114.39 min	46.881 s	5661	94.35
10	7915	122.76 min	44.671 s	5579	92.98

Results for the second series are shown in Table 8; the best SR was 93.949%.

Table 8. Complete system results, series 2

Training	**Σ Epochs**	**training time**	**ident. time**	**identified**	**success rate**
1	10094	127.74 min	45.726 s	5561	92.683
2	10294	126.67 min	45.108 s	5528	92.133
3	8609	122.44 min	45.926 s	5553	92.550
4	9441	128.57 min	46.429 s	5588	93.133
5	10022	129.06 min	44.904 s	5637	93.949
6	9076	140.98 min	45.389 s	5516	91.933
7	8047	112.49 min	45.843 s	5543	92.383
8	7637	97.59 min	47.371 s	5539	92.317
9	7699	118.58 min	47.986 s	5527	92.117
10	8044	132.52 min	44.997 s	5562	92.699

Results for the third series are shown in Table 9; the best SR was 97.75%.

Table 9. Complete system results, series 3

Training	**Σ Epochs**	**training time**	**ident. time**	**identified**	**success rate**
1	7141	279.1995 min	23.4003 s	5859	97.65
2	7034	261.7912 min	23.1624 s	5865	97.75
3	7147	265.5975 min	23.2537 s	5860	97.67
4	7107	269.4329 min	23.2042 s	5861	97.68
5	7121	267.2153 min	23.1404 s	5864	97.73

Table 9. (*continued*)

6	7227	247.4135 min	22.9807 s	5856	97.60
7	7172	251.7634 min	22.9578 s	5858	97.63
8	7164	256.9265 min	23.9409 s	5860	97.67
9	7184	252.2429 min	23.1345 s	5860	97.67
10	7104	252.7938 min	23.1556 s	5851	97.52

4.4 Complete System Results (2)

Results for the third series are shown in Table 10; the best SR was 97.77%.

Table 10. Complete system(2) results, series 3

Training	Σ Epochs	training time	ident. time	identified	success rate
1	7141	279.1995 min	23.5901 s	5860	97.67
2	7034	261.7912 min	23.3187 s	5866	97.77
3	7147	265.5975 min	23.5999 s	5861	97.68
4	7107	269.4329 min	24.0719 s	5860	97.67
5	7121	267.2153 min	24.4985 s	5864	97.73
6	7227	247.4135 min	23.3928 s	5858	97.63
7	7172	251.7634 min	23.5359 s	5858	97.63
8	7164	256.9265 min	23.4708 s	5860	97.67
9	7184	252.2429 min	23.3946 s	5860	97.67
10	7104	252.7938 min	23.9661 s	5851	97.52

In all previously shown tables, identification time leaves out preprocessing time.

The tables in section 4 show that the greatest boost in performance when integrating both of the MNNs occurred in series 1, with an advantage of some 2.7 % over MNN 1, which, as readily seen always carries most of the weight in finding the correct identifications, and it happens to be the non-geometric MNN.

What happens in later series is easy to explain, even though both MNNs get to perform better: since MNN 2 doesn't improve as much as MNN 1, there is close to no net gain in using MNN 2, and in fact, some times there is a loss; this is only marginally better with FIS #2, which gave the best overall performance, at 97.77%.

A few comparisons with prior works in the same (or close) area of research can be very illustrative:

Zhenhua Guo uses the same database we do, but using all spectra at the same time; he uses a fusion algorithm based on Haar wavelets and PCA, and on one of many configurations tried, obtained 97.877 as a success rate [1].

Kumar and Zhang use entropy based discretization for a database smaller than ours and use Support vector machines and neural networks as classifiers for a respective accuracy of 95 and 94% identifying, for a hand geometry method [4].

Cenker Öden's group claims that their method, using implicit polynomials, is capable of achieving 95% accuracy in identification tasks [6].

It is noteworthy that, when dealing with hand geometry systems in verification mode, Ingersoll Rand's ID3D system is reported by Sandia Labs as performing as low as 0.2% of total error, this as far back as 1991 [9].

5 Conclusions and Future Work

As seen in the last section, the best overall performance so far obtained with our method is 97.77%.

Performance, compared to prior works does not seem too bad, but as a whole, comparisons show that there is still much to be improved and open a few lines of future work.

First, trying to elevate performance of the net within MNN 2's integrator; that would gain no more than a few tens of well identified individuals over MNN 1 alone.

Second, the problem might reside in the arrangement of modules (their number) in MNN 2.

Third, a true verification comparison should take into account that MNN 1's modules on their own have a success rate at least two percentage points higher than MNN 1 itself.

Fourth, comparison with other systems, including verification, would be aided by performing complete statistical evaluations of the system.

References

[1] Guo, Z.: Online Multispectral Palmprint Recognition. Thesis, Hong Kong Polytechnic University (2009)

[2] Jang, R., Tsai-Sun, C., Mizutani, E.: Neuro-Fuzzy and Soft Computing, 1st edn. Prentice Hall, Upper Saddle River (1996)

[3] Kong, A., Zhang, D., Kamel, M.: A survey of palmprint recognition. In: Pattern Recognition, vol. 42, pp. 1408–1418. Elsevier (2009)

[4] Kumar, A., Zhang, D.: Biometric recognition using entropy-based discretization. In: Proceedings of 2007 IEEE International Conference on Acoustics Speech and Signal Processing, pp. 125–128 (2007)
[5] Luger, G., Stubblefield, W.: Artificial Intelligence, 3rd edn. Structures and strategies for complex problem solving. Addison Wesley, Reading (1998)
[6] Öden, C., Erçil, A., Yldz, V.T., Krmzta, H., Büke, B.: Hand Recognition Using Implicit Polynomials and Geometric Features. In: Bigun, J., Smeraldi, F. (eds.) AVBPA 2001. LNCS, vol. 2091, pp. 336–341. Springer, Heidelberg (2001)
[7] PolyU Multispectral Palmprint Database, `http://www4.comp.polyu.edu.hk/Ebiometrics/MultispectralPalmprint/MSP_files` (accessed October 2011)
[8] Rojas, R.: Neural Networks - A Systematic Introduction. Springer, New York (1996)
[9] Zunkel, R.: Hand geometry based authentication. In: Jain, A.K., Bolle, R., Pankanti, S. (eds.) Biometrics: Personal Identification in Networked Society. Kluwer Academic Publishers, Boston (1998)