

Liquid State Machine Based Fingerprint Identification

¹Youssef Elmir, ²Zakaria Elberrichi and ²Reda Adjoudj

¹Colonel Ahmed Draia Africain University/Department of computer science, Adrar, Algeria.

²Djillali Liabes University/Department of computer science, Sidi Bel Abbes, Algeria.

Abstract: This paper deals with automatic fingerprint identification; this is performed by a supervised classification since it operates on a set of preset classes. The suggested approach is original since it is based on liquid state machine which gives a better simulation of biological neurons as spiking neurons than sigmoidal ones. Promising Results of first experiences using radial basis function neural network and support vector machine pushed us to continue the exploitation of this new version of neural networks and to develop codification and recognition algorithms which are specifically associated to this system. Tests were released on well known fingerprint databases such as: FVC2004 databases & Finger Cell database. Performance Evaluating has proved that liquid state machine achieved a good recognition rate closer to rates achieved by other methods but in a very short time and this make it more useful in online applications.

Key words: Biometrics, Gabor filters coding, Liquid State Machine, Spiking neurons, Support Vector Machine, Radial Basis Function Neural Network.

INTRODUCTION

Fingerprint -based identification is one of the most important biometric technologies which have drawn a substantial amount of attention recently (Umer, 2004).

Fingerprints are believed to be unique across individuals and across fingers of same individual. Even identical twins having similar DNA, are believed to have different fingerprints (Umer, 2004). A fingerprint is the pattern of ridges and valleys on the surface of a fingertip (Zhou, 2004).

Fingerprint recognition can be categorized into identification and verification. Fingerprint identification is the process of determining which registered individual provides a given fingerprint. Fingerprint verification, on the other hand, is the process of accepting and rejecting the identity claim of a person using his fingerprint. Fingerprint recognition can also be categorized into minutiae extraction based and spectral features of the image based.

All technologies of fingerprint recognition, identification and verification, minutiae extraction based and spectral features based, each has its own advantages and disadvantages and it may require different treatments and techniques. The choice of which technologies to use is application specific.

In the following sections, we present in detail our proposed algorithm of recognition using spiking neurons based liquid state machine which is a recently developed computational model (Maass, 2002). The principle of Fingerprint identification is presented in Section 2. Sections 3 and 4 present the problem statement and our proposed design for it, respectively. We give the algorithm used to codify fingerprint images using a bank of Gabor filters in section 5, the architecture of our liquid state machine in section 6, The experimental results on the FVC2004 and FINGERCELL fingerprint databases, are presented in Section 7. Finally, conclusion is given in Section 8.

State of the Art:

At the highest level, all fingerprint recognition systems contain two main modules feature extraction and feature matching. Feature extraction is the process that detects singular and all other minutiae points which are ridge ending and ridge bifurcation which differentiate one fingerprint from another which impart individuality to each fingerprint (See Fig. 1) from the original image that can later be used to represent each fingerprint. Feature matching involves the actual procedure to identify the unknown person by comparing extracted features from his/her fingerprint with the ones from a set of known persons. We will discuss each module in detail in next sections.

Corresponding Author: Youssef Elmir, Colonel Ahmed Draia Africain University/Department of computer science, Adrar, Algeria.
E-mail: ysfbr@yahoo.fr



Fig. 1: Ridge ending, core point and ridge bifurcation are shown (National Institute).

Although fingerprint authentication appears to be an easy authentication method in both how it is implemented and how it is used, there are some user influences that must be addressed.

Influence of several fingerprint image quality characteristics on the performance of the verification system (Simon-Zorita, *et al.*, 2003).

The displaced, false or missing singular points (Legenstein, *et al.*, 2007), and background noises will interfere with the user who is trying to authenticate to the device. The environment in which the user is authenticating to the device must be free of any major background noise (Adjoudj, 2005).

Proposed Design:

The problem of fingerprint recognition is one of much broader topics in scientific and engineering so called pattern recognition. The goal of pattern recognition is to classify objects of interest into one of a number of categories or classes (Adjoudj, 2005). The objects of interest are generically called patterns and in our case are images of fingerprints matrix called vectors codes or fingercodes that are extracted from an input image using the techniques described in the later section. The classes here refer to individual person. Since the classification process in our case is applied on extracted features, it can be also referred to as feature matching.

This article demonstrates how fingerprint identification can be released with a spike neural network as matching process, but before the minutiae image of the fingerprint was converted into a vector code, also called fingercode, by using Gabor filter bank (Batra, 2004).

Problem Statement:

A spike neurons based liquid state machine is to be designed and trained to recognize the fingercode of the databases that are actually used. An imaging system that converts each minutiae image obtained from a fingerprint image in fingercode or minutiae matrix code by using a bank of Gabor filters. The result is that each fingerprint image is represented as a vector of 512 real values. Finally, results obtained by spike neural network will be compared to those obtained by SVM and RBF as is shown in Fig 2.

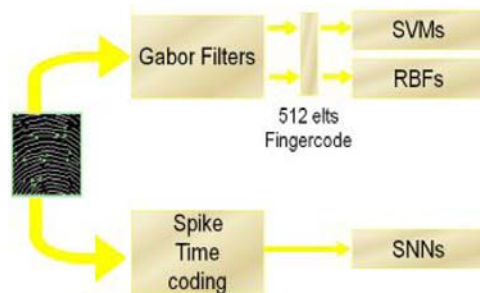


Fig 2: General Architecture of fingerprint identification system (Elmir Youssef, 2007).

Creating Fingercodes:

The following steps are observed to create the fingercodes:

1. Preprocessing of the image (to remove noise) by window wise normalization, Histogram Equalization, low pass and median filtering (Greenberg, *et al.*).
2. Core point location using max concavity estimation (Mansfield, 2002).
3. Tessellation of circular region around the reference point.
4. Sector wise normalization followed by application of bank of Gabor filters which has following general form in the spatial domain (Anil, *et al.*, 1999).

$$G(x, y; f, \theta) = \exp \left\{ \frac{-1}{2} \left[\frac{x'^2}{\delta_x^2} + \frac{y'^2}{\delta_y^2} \right] \right\} \cos(2\pi f x')$$

$$x' = x \sin \theta + y \cos \theta. \text{ (Elmir, 2009)}$$

$$y' = x \cos \theta - y \sin \theta. \text{ (Elmir, 2009)}$$

Where f is the frequency of the sine plane wave along the direction θ (0, 45, 90 and 135 degrees) from the x-axis, δ_x and δ_y are the space constants of the Gaussian envelope along X' and Y' axes, respectively.

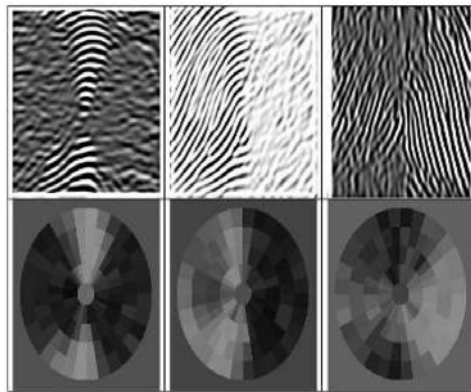


Fig. 3: Filtered images and their corresponding feature vectors for the orientations 0°, 5°, 22.5° and 45° are shown (Umer, 2004).

Finally feature code generation by obtaining standard deviation values of all the sectors, (Anil, *et al.*, 1999).

Fingerprint Recognition:

In less than few seconds, even on a database of hundreds of records, the matrix of pixels generated from an image of fingerprint is compared to previously enrolled ones to see if it matches any of them. The decision threshold is automatically adjusted for the size of the search database to ensure that no false matches occur even when huge numbers of matrixes of fingerprints are being compared with the live one. Some of the bits in a matrix signify if some data is corrupted (e.g. the image of fingerprint has degraded by noise), so that it does not influence the process, and only valid data is compared. Decision thresholds take account of the amount of data, and the matching operation compensates for any tilt of the image of fingerprint. A key advantage of fingerprint recognition is its ability to perform identification using a one-to-all search of a database, with no limitation on the number of fingerprint records and no requirement for a user first to claim an identity, for example with a card. For our method we use and experiment a recent neural network (Delorme, 2001), called a spike neural network. The neural network is used for matching and performs recognition using a one-to-all search of a database.

Training with Spiking Neurons:

In tasks like pattern recognition, the performances of artificial networks are still poor when compared to humans. Using a go-no-go categorization task in which human subjects had to release a button when they detected an animal in a briefly flashed (20 ms) natural photograph (Greenberg, *et al.*) showed that the processing required in such a task could be performed in less than 150 ms. In order to reach high order cortical

areas, the retinal information must go through at least 10 processing stages, and knowing the constraint of real neurons, (maximum firing rate usually under 100 spikes per seconds), they proposed that this processing was essentially based on a unidirectional flow of information in which, in any given layer, each neuron would rarely generate more than one spike. With one spike per neuron, codes based on neuron discharge rate are questioned. As an alternative to rate based neural codes, temporal coding is based on the relative order in which the spikes are produced is consistent with lateral inhibition in the first areas of the visual cortex. Spike neural network can use the latency or the relative latency of neuronal discharges means their order of arrival and still attain high performances for image processing with only one spike per neuron. Furthermore, order coding is contrast independent and can be analyzed from a mathematical point of view. Spike neural network is designed for simulating very large networks of asynchronous spiking neurons. Neurons are simulated with a limited number of parameters that includes classic properties like the post-synaptic potential and threshold, but also more novel features like dendritic sensitivity. Spike neural network can be used to simulate networks with millions of neurons and hundreds of millions of synaptic weights. Optimization of computation time and the aim of a real time computation has been one of the driving forces behind the development of spike neural network (Delorme, 2001). There are some characteristics of spike neural network (Delorme, 2001) (Van, *et al.*, 1998):

1. Perform image processing using biologically plausible network of neurons.
2. Simulate millions of integrate-and-fire neurons organized in retinotopic maps.
3. Connect these neuronal maps using projection files, and regroup common synaptic weights to save memory in order to be able to declare several hundreds of billions of synaptic connections.
4. Convert gray level images into lists of spikes (also spike neural network can perform a variety of preprocessing on the input images).
5. Implement a complex mechanism for projection between neuronal maps of different sizes.
6. Implement supervised learning.

For these characteristics, we create a new fingerprint recognition system which trained with spike neural network. When we tested rigorously the performances of spike neural network, it could compute roughly 20 million connections per second on a standard desktop computer (PC with speed of 3.0 GHz). This is sufficient to model a network of 400 000 neurons in real time, using a time step of 1 ms (assuming 40 connections per neuron, and an average firing rate of 1 spike per second, a value which is a reasonable estimate for the average firing rate of cortical neurons). Note that time resolution increases has virtually no effect on the computation time, and that adding a decay parameter to neurons adds roughly 30-40% to the computation time. Note that Number of synapses updated per second by spike neural network as a function of the number of neurons and the number of input synapses per neuron. Shared and individual weights indicate whether some neurons share same sets of synaptic weights.

Architecture of Liquid State Machine:

This architecture of a LSM, consisting of Liquid and readouts, makes sense; because it turns out that in many contexts there exist common computational preprocessing needs for many different readouts with different computational goals. This can already be seen from the trivial fact that computing all pairwise products of a set of input numbers (say: all components of a multi-dimensional input $u(t)$ for a fixed time point t) gives any subsequent linear readout the virtual expressive power of any quadratic computation on the original input numbers $u(t)$. A pre-processor for a linear readout is even more useful if it maps different input streams $u(t')$ onto linearly independent liquid states $\sim x(t)$ (Legenstein, 2007), similarly as an RBF-kernel for Support Vector Machines. A remarkable aspect of this more general characterization of the preprocessing task for a Liquid is that it does not require that it computes precise products, or any other concrete nonlinear mathematical operation. Any "found" analog computing device (it could even be very imprecise, with mismatched transistors or other more easily found nonlinear operations in physical objects) consisting of sufficiently diverse local processes, tends to approximate this requirement quite well. A closer look shows that the actual requirement on a Liquid is a bit more subtle, since one typically only wants that the maps Liquid "saliently" different input streams $u()$ onto linearly independent liquid states $\sim x(t)$, whereas noisy variations of the input stream should rather be mapped onto a lower dimensional manifold of liquid states, see (Legenstein, *et al.*, 2007) and (Legenstein, 2007) for details.

An at least equally important computational preprocessing task of a Liquid is to provide all temporal integration of information that is needed by the readouts. If the target value $y(t)$ of a readout at time t depends not only on the values of the input streams at the same time point t , but on a range of input values $u(s)$ for many different time points s (say, if $y(t)$ is the integral over one component of $u(s)$ for a certain interval $[t - 1, t]$), then the Liquid has to collect all required information from inputs at preceding time points $u(s)$, and present all this information simultaneously in the liquid state $\sim x(t)$ at time point t . This is necessary, because the readout stage has by assumption no temporal integration capability of its own, i.e. it can only learn to carry

out "static" computations that $\sim x(t)$ and here just numbers, no functions. map $\sim x(t)$ onto $y(t)$ (where $y(t)$ are of time) A readout does not even know at any time what the current time t is. It just learns a map f from input numbers to output numbers. Hence it just learns a fixed recoding (or projection) f from liquid states into output values. This severe computational limitation of the readout of a LSM is motivated by the fact, that learning a static map f is so much simpler than learning a map from input streams to output streams. And a primary goal of the LSM is to make the learning as fast and robust as possible.

The advantage of choosing for a LSM the simplest possible learning device is twofold: Firstly learning, for a single readout neuron is fast, and cannot get stuck in local minima (like backprop or EM). Secondly, the simplicity of this learning device entails a superior-in fact: arguably optimal-generalization capability of learned computational operations to new inputs streams. This is due to the fact that its VC -dimension (see (Bartlett, *et al.*, 2003) for a review) is equal to the dimensionality of its input plus 1, I. e., the smallest possible value of any nontrivial learning device with the same input dimension.

It is a priori not clear that a Liquid can carry the highly nontrivial computational burden of not only providing all desired nonlinear preprocessing for linear readouts, but simultaneously also all temporal integration that they might need in order to implement a particular mapping from input streams $u(\cdot)$ onto output streams $y(\cdot)$. But there exist two basic mathematical results (Maass, 2009) which show that this goal can in principle be achieved, or rather approximated, by a concrete physical implementation of a Liquid which satisfies some rather general property. A remarkable discovery, which had been achieved independently and virtually simultaneously around 2001 by Herbert Jaeger (Jäger and Haas, 2004), is that there are surprisingly simple Liquids, i.e. generic preprocessors for a subsequent linear learning device, that work well independently of the concrete computational tasks that are subsequently learned by the learning device. In fact, naturally found materials and randomly connected circuits tend to perform well as Liquids, which partially motivate the interest of the LSM model both in the context of computations in the brain, and in novel computing technologies. Herbert Jaeger (Jäger and Haas, 2004) had introduced the name Echo State Networks (ESNs) for the largely equivalent version of the LSM that he had independently discovered. He explored applications of randomly connected recurrent networks of sigmoidal neurons without noise as Liquids (in contrast to the biologically oriented LSM -studies, that as some significant internal noise in the Liquid) to complex time series prediction tasks, and showed that they provide superior performance on common benchmark tasks. The group of Benjamin Schrauwen (see (Verstraeten, *et al.*, 2005) (Schrauwen, *et al.*, 2008) (Vandoorne, *et al.*, 2008) (Schrauwen, *et al.*, 2009) introduced the term Reservoir Computing for the investigation of LSMs, ESNs and variations of these models. A variety of applications of these models can be found in a special issue of Neural Networks 2007 see (Jäger, 2007). All these groups are currently collaborating in the integrated EU-project ORGANIC (= Self -organized recurrent neural learning for language processing) that investigates applications of these new models to speech understanding and reading of handwritten text (see <http://reservoir-computing.org/>). An industrial partner in this project, the company PLANET (<http://english.planet.de/>) had already good success in applications of Reservoir Computing to automated high -speed reading of hand-written postal addresses.

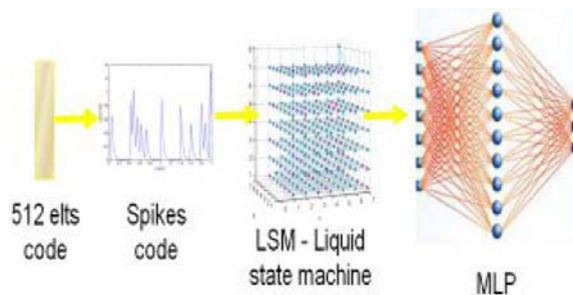


Fig. 4: Architecture of liquid state machine based identification system (Elmir Youssef, 2007).

Experiment and Tests:

A. Databases and Datasets:

To evaluate the performance of the proposed methods, it is necessary to dispose a database of finger prints representing the purchasing system. Unfortunately these databases are not always available immediately.

It is available from the National Institute of Standards and Technology (National Institute,) bases of several thousand prints. These images were scanned from prints identity papers obtained by the traditional method of ink, they are very poor quality. To compare different methods of treating them, a competition was held to assess the performance of algorithms from the same images (Fingerprint Verification, 2004). The databases are used openly and from three different sensors trade.

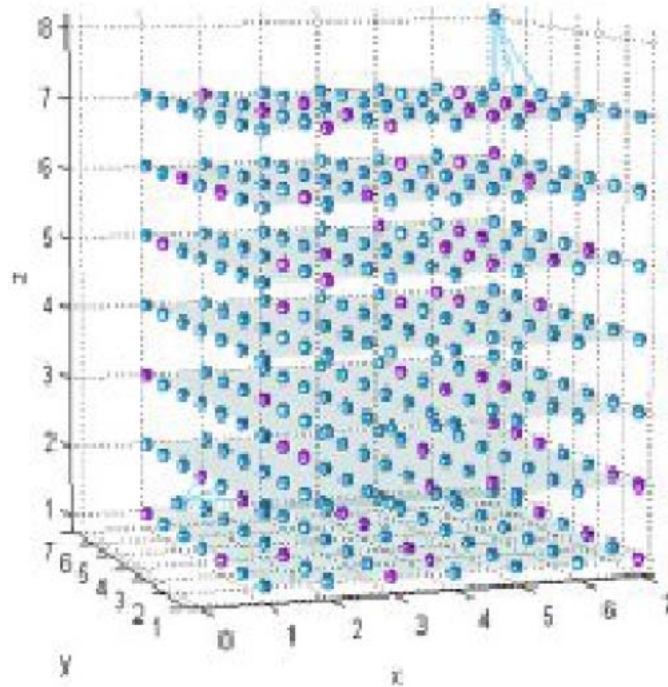


Fig. 5: Architecture of the Liquid state machine (Elmir Youssef, 2007).

These databases are used in our case despite the acquisition systems used can rotate finger any angle. This is not the case in our system (assumption the existence of a guidance system).

Our system was tested with four databases of FVC2004 and Finger Cell database.

Table 1: FVC2004 DATABASES (Fingerprint Verification Competition 2004).

	Device type	Image size	Set A (wxd)	Set B (wxd)	Resolution
DB1	Optical Captor	640x48 (307 Kpixels)	100x8	10x8	500 dpi
DB2	Optical Captor	328x364 (119 Kpixels)	100x8	10x8	500 dpi
DB3	Thermal Captor	300x480 (144 Kpixels)	100x8	10x8	512 dpi
DB4	SfinGe v 3.0	288x384 (108 Kpixels)	100x8	10x8	About 500 dpi
FC	Digital PersonaU.are.U4000	128x128 (16 Kpixels)	19x10	/	500 dpi

To effectively represent the performance of the dataset's size should be large enough (Mansfield, 2002) but the number of images was large and the duration of tests was very long. For practical reasons we are constrained to a bank of 152 images with 19 different sets of fingers, each dataset consisting of 8 images in the case of FVC2004 corresponding to various acquisitions of the same finger and in the case of FingerCell, a bank 190 images with the same number of different sets of fingers, but a different number of pictures related to various acquisitions of the same finger (10 images).



Fig. 6: Examples of fingerprints from FVC2004 databases (Elmir Youssef, 2007).



Fig. 7: Examples of fingerprints from FingerCell database (Elmir Youssef, 2007).

B. Results:

Table 1 shows the results of recognition rate and performance of the proposed system.

Table 2: Recognition Results on the Person's Fingerprint Databases (Of Five Data -Sets) on a Pc with Speed of 3.0 Ghz (Elmir, 2009).

Datasets	Time for Train (s)			Time forTest (s)			RecognitionRate(%)		
	SNN	RBF	SVM	SNN	RBF	SVM	SNN	RBF	SVM
FVC2004 DB2	19,21	2,26	32,09	0,09	71	1,37	73.6	63.1	68.4
FVC2004 DB2	12,32	1,04	17,46	64	29	1,12	68.4	63.1	63.1
FVC2004 DB3	14,10	2,89	18,29	71	107	0,90	84.2	68.4	73.6
FVC2004 DB4	12,56	1,04	17,81	60	31	1,05	73.6	68.4	78.9
Finger Cell	3,40	23,6	4,50	47	47	0,92	84.2	89.4	94.7

C. Resistance Against Noise:

The major obstacle in the fingerprint identification systems is the presence of noise in captured images of fingers during the training or test phase, which pushed us to study the reaction and performance of the system made against different noise levels applied to analyze its performance and see the error rates obtained for each noise level.

The noise used for the test system is a white noise, which is a matrix of random integer numbers distributed following a normal law of mean 0 and variance 1.

$$Noise = Randn \times Noise_Level .$$

Where Randn is a function that gives the random integer number and Noise_Level is in the interval [0 0.5] with a step of 0.1. ...

The performances of proposed systems are illustrated by the following graphs:

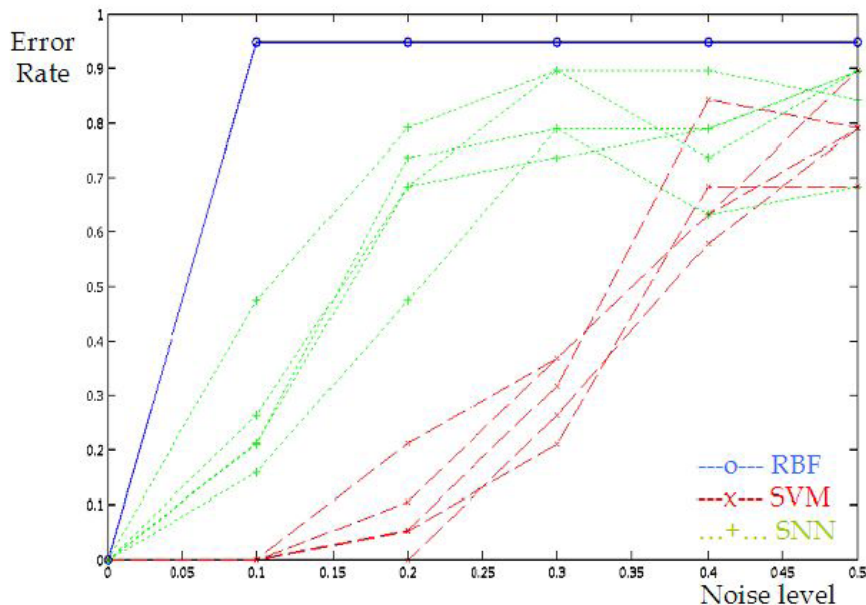


Fig 8: Resistance of proposed systems against different level of noise (Elmir, et al., 2009).

In Fig. 8 we can see that SVM resists against noise applied better than neuronal systems and that the graph of the spike neural network is very close to the graph of SVM unlike RBF that produces a significant number of errors from the beginning of test.

The error rates of the three methods are calculated at each noise level for the entire base. Performance graphs show that the SVM and the spike neural network methods are more resistant to noise than RBF which gives an outstanding error rates around noise level 0.1, while both spike neural network and SVM make low rate of error at this level, SVM maintain these rates until degree of noise (0.2), for against, spike neural network starts to produce a significant error rate in this noise level and after this value of noise level, rates of error produced by all methods begin to increase in a natural way with a significant degree of noise which can disrupt the even a human being recognition system.

Conclusion:

Fingerprint recognition is a challenging problem and there is still a lot of work that needs to be done in this area. Over the past ten years, fingerprint recognition has received substantial attention from researchers in biometrics, pattern recognition, image processing, and cognitive psychology communities. This common interest in fingerprint recognition technology among researchers working in diverse fields is motivated both by the remarkable ability to recognize people and by the increased attention being devoted to security applications.

Applications of fingerprint recognition can be found in security, tracking, multimedia, and entertainment domains. We have demonstrated how a fingerprint recognition system can be designed by a spike neural network, to capture the minutiae characteristics of fingerprint, to simulate the human visual system, although a spike neural network designed as a tool for modeling biological neural networks, the level of performance obtained with spike neural network is such that in a variety of tasks, processing architectures developed using spike neural network can perform at least as well and in many cases substantially better than more conventional image processing and pattern recognition techniques. The levels of performance achieved by the human visual system are orders of magnitude better than even the most sophisticated artificial vision systems (Delorme, 2001) (Van, *et al.*, 1998). By elucidating the computational principles which make this level of performance possible, it may well be possible not only to demonstrate the power of computational neuroscience as a paradigm for understanding biology, but may reveal the potential of the discipline in areas as diverse as machine vision and artificial intelligence.

REFERENCES

- Anil, K., *et al.*, 1999. "A Multichannel Approach to Fingerprint Classification," in IEEE Transactions on Pattern Analysis and Machine Intelligence, 21(4): 348-359.
- Adjoudj, R. and A. Boukelif, 2005. New MFCC Speaker Recognition, In the Proceedings of the 2005 Optimization and Information Systems Symposium, Béjaïa, Algeria.
- Batra, D., G. Singhal and S. Chaudhury, 2004. Gabor Filter based Fingerprint Classification using Support Vector Machines, In the Proceedings of the 2004 IEEE India Annual Conference.
- Bartlett, P.L., *et al.*, 2003. The Handbook of Brain Theory and Neural Networks, MIT Press (Cambridge), 2nd edition.
- Buonomano, D. and W. Maass, 2009. State-dependent computations: Spatiotemporal processing in cortical networks. Nature Reviews in Neuroscience, 10(2): 113-125.
- Delorme, A., Van. Rullen R and S. Thorpe, 2001. SpikeNet: a simulator for modeling large networks of integrate and fire neurons, Neurocomputing Journal, Elsevier, 38-40(1-4): 539-545.
- Elmir, Youssef., 2007. Persons identification by their fingerprints, Magister thesis, Mohamed Boudiaf University of Sciences and Technology, Oran, Algeria.
- Elmir, Y., *et al.*, 2009, Spike Neural Network based Fingerprint Identification, in the proceeding of 2009 Image and Signal Processing and their Applications, Mostaghanem, Algeria.
- Elmir, Y. and M. Benyettou, 2009. Gabor Filters Based Fingerprint Identification Using Spike Neural Networks, In the proceeding of 2009 Sciences of Electronic, Technologies of Information and Telecommunication conference, Hammamet, Tunisia.
- Elmir, Y., *et al.*, 2009. Personal Identification by Fingerprints based on Gabor Filters, In the proceedings of 2009 Conférence sur l'Informatique et ses Applications, Saida, Algeria, May 2009.
- Fingerprint, Verification Competition, 2004. (FVC2004), website: <http://bias.csr.unibo.it/fvc2004>
- Greenberg, S., *et al.*, Fingerprint Image Enhancement using Filtering Techniques, Ben-Gurion University of the Negev, Beer-Sheva, Israel.
- Jäger, H. and H. Haas, 2004. harnessing nonlinearity: predicting chaotic systems and saving energy in wireless communication. Science, 304:78-80.

- Jäger, H., W. Maass and J. Principe, 2007. Introduction to the special issue on echo state networks and liquid state machines. *Neural Networks*, 20(3): 287-289.
- Jain, K., *et al.*, 2003. *Handbook of Fingerprint Recognition*, Springer Professional, Computing.
- Legenstein, R. and W. Maass, 2007. Edge of chaos and prediction of computational performance for neural microcircuit models. *Neural Networks*, 20(3): 323-334.
- Legenstein, R. *et al.*, 2007. *New Directions in Statistical Signal Processing: From Systems to Brains*, MIT Press.
- Maass, W., 2009. *Motivation, Theory, and Applications of Liquid State Machines*.
- Maass, W. T., N. Ager and H. Markram, 2002. Real-time computing without stable states: A new frame-work for neural computation based on perturbations, *Neural Computation*, 14(11): 2531-2560.
- Mansfield, A.J. and J.L. Wayman, 2002. *Best Practices in Testing and Reporting Performance of Biometric Devices, Version 2.01*, Communications Electronics Security Group, (CESG).
- National Institute of Standards and Technology (NIST), website: www.nist.gov/data/biomet.htm
- Schrauwen, B., *et al.*, 2008. Compact hardware liquid state machines on FPGA for real-time speech recognition. *Neural Networks*, 21: 511-523.
- Schrauwen, B., *et al.*, 2009. On computational power and the order-chaos phase transition in reservoir computing. *Neural Information Processing Systems*. MIT Press, 2009.
- Simon-Zorita, D., *et al.*, 2003. *IEE Proc.-Vis. Image Signal Process*, 150(6).
- Thorpe, S., D. Fize, C. Marlot, 1996. Speed of processing in the human visual system, *Nature journal*, 381: 520-522.
- Umer, Munir M. and M. Younas Javed, 2004. *College of Electrical and Mechanical Engineering*, National University of Sciences and Technology Rawalpindi, Pakistan.
- Verstraeten, D., *et al.*, 2005. Isolated word recognition with the liquid state machine: a case study. *Information Processing Letters*, 95(6): 521-528.
- Van Rullen, R., *et al.*, 1998. Face detection using one spike per neuron, *Biosystems journal*, 22.
- Vandoorne, K., *et al.*, 2008. Toward optical signal processing using photonic reservoir computing. *Optics Express*, 16(15): 11182-11192.
- Zhou, J. and J. Gu, 2004. A Model-Based Method for the Computation of Fingerprints' Orientation Field, *IEEE transactions on image processing*, 13(6).
- Zhaoping, L., 1998. A neural model of contour integration in the primary visual cortex, *Neural Computation journal*, Elsevier, 10: 903-40.