Resource-Efficient Hardware Implementation of a Neural-based Node for Automatic Fingerprint Classification

Vincenzo Conti[1], Leonardo Rundo[2,3], Carmelo Militello[3], Giancarlo Mauri[2], and Salvatore Vitabile[4]*
[1]*Facoltà di Ingegneria e Architettura*
Università degli Studi di Enna KORE, Enna, Italy
vincenzo.conti@unikore.it
[2]*Dipartimento di Informatica, Sistemistica e Comunicazione (DISCo)*
Università degli Studi di Milano-Bicocca, Milano, Italy
{leonardo.rundo, mauri}@disco.unimib.it
[3]*Istituto di Bioimmagini e Fisiologia Molecolare (IBFM)*
Consiglio Nazionale delle Ricerche (CNR), Cefalù (PA), Italy
carmelo.militello@ibfm.cnr.it
[4]*Dipartimento di Biopatologia e Biotecnologie Mediche (DIBIMED)*
Università degli Studi di Palermo, Palermo, Italy
salvatore.vitabile@unipa.it

Abstract

Modern mobile communication networks and Internet of Things are paving the way to ubiquitous and mobile computing. On the other hand, several new computing paradigms, such as edge computing, demand for high computational capabilities on specific network nodes. Ubiquitous environments require a large number of distributed user identification nodes enabling a secure platform for resources, services and information management. Biometric systems represent a useful option to the typical identification systems. An accurate automatic fingerprint classification module provides a valuable indexing scheme that allows for effective matching in large fingerprint databases. In this work, an efficient embedded fingerprint classification node based on the fusion of a Weightless Neural Network architecture and a technique, namely Virtual Neuron, which efficiently maps a neural network architecture into hardware resources, is presented. The key novelty of the proposed paper is a new neural-based classification methodology that can leverage devices and sensors with limited number of resources, allowing for resource-efficient hardware implementations. Furthermore, the classifier efficiency and the accuracy have been optimized to obtain high classification rate with the best trade-off between minimum area on chip and execution time. The proposed neural-based classifier analyzes a directional image, which is extracted from the original fingerprint image without any enhancement, and classifies the processed item into the five NIST NBIS classes. This approach has been designed for FPGA devices, by exploiting pipeline techniques for execution time reduction. Experimental results, based on a 10-fold cross-validation strategy, show an overall average classification rate of 90.08% on the whole official FVC2002DB2 database.

Keywords: Mobile and Ubiquitous Computing, Fingerprint Classification, Weightless Neural Networks, Virtual Neuron, Field Programmable Gate Array (FPGA).

1 Introduction

Modern mobile communication networks and Internet of Things (IoT) scenarios are paving the way to relevant advancements in ubiquitous and mobile computing. In addition, several new computing

Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA), 8:4 (December), pp. 19-36

*Corresponding author: Department of Biopathology and Medical Biotechnologies, University of Palermo, Via del Vespro 129, 90133 Palermo, Italy, Tel: +3909123865720 paradigms, such as edge computing, demand for high computational capabilities on specific network nodes [1]. In those scenarios, the implemented infrastructures must provide secure and trusted user information access and management in order to protect sensitive and confidential data resource and ensure the possibility of the user traceability inside a used platform [2, 3, 4, 5]. These environments require a large number of distributed user identification nodes enabling a secure platform for resources, services and information management. Such a kind of architecture can potentially meet the requirements regarding response times, battery lifetime, network bandwidth, also considering data confidentiality and privacy [6]. Especially, in the case of mobile devices, these constraints can be satisfied by offloading intensive computations to the edge of the network [7]. Therefore, in the last few years, conventional identification systems—mainly based on username/password pairs or Personal Identification Numbers (PINs)—are in crisis since they cannot always ensure high security degrees for critical applications. Biometric-based recognition solutions can offer a reliable e-infrastructure proposing an integrated approach that could increase the interoperability among different institutions. These systems have the main purpose to authenticate people by means of some personal physiological and/or behavioral characteristics [8, 9]. These systems represent a rapidly evolving technology that is well-suitable for real-life applications, such as in forensics, commerce and governance [10]. Nowadays, several embedded biometric identification solutions are available for mobile phones, personal digital assistants (PDAs), wireless processing nodes, and so on [11, 12, 13, 14]. Automatic recognition systems are usually composed of two modules for fingerprint classification and identification. The automatic fingerprint classification module enables a valuable indexing scheme that allows for effective matching in large fingerprint databases in software and hardware Automated Fingerprint Identification Systems (AFISes) [15][16][17]. Essentially, the AFIS module yields as binary response whether the recognition process is either successful or not, so allowing or denying access to the user.

In literature, many techniques have been used and a lot of prototypes have been proposed to implement fingerprint classification systems aiming at increasing the performance of the AFISes. They are often based on neural networks [18][19], hidden Markov models [20], support vector machines [21], directional image information and/or singularity features [22][23][24], and researches in this field are still active. To the best of our knowledge, all state-of-the-art classification systems are software-based, such as in [25], where the authors proposed a method based on topological and numerical considerations about the macro-characteristics (i.e., delta and core) extracted from the fingerprint directional image. This type of images was obtained from the original fingerprint images after a distortion and contrast reduction phase. Processing time is reduced since no thinning and image processing are performed. Resolution increases since single pixel direction and directional histograms for singular point detection are computed. In [26], the authors proposed a five-step structural approach based on relational graphs. The implemented steps are: (*i*) directional image calculation by minimizing the related region variance, (*ii*) directional image extraction, (*iii*) a linked fingerprint macro-structure relational graph generation, (*iv*) inaccurate graph matching, and (*v*) classification. In [27] a discrete wavelet transform based feature extraction strategy is presented. A preliminary evaluation of its utility is neural-based classification tasks is also reported. Their experimental results show that the wavelet transform is dependent on the quality of the images and their normalization. The authors of [28], again exploiting the directional image information, presented a comparison among three different software classification techniques: Fuzzy C-Means, Weightless Neural Network, and a Naïve Bayesian Network. A 100-image database and four fingerprint classes (Right Loop, Left Loop, Whorl and Tented Arch) are considered, achieving a 91% classification rate. In [29], some of the authors of this paper proposed a hardware classifier prototyped on a Celoxica RC2000 board equipped with a Xilinx Virtex-II 2V6000 FPGA. The classifier was based on a Weightless Neural Network architecture for fingerprint directional image classification. The NIST standard five classes (i.e., Left Loop, Right Loop, Whorl, Arch and Tented Arch) were used. The highest overall classification rate, achieved by the system, was of 86.54%. The goal of this work was to reduce the needed resources maintaining high level of accuracy and speed.

This paper follows and extends the above work [29]. The key novelty of this paper is a novel neural-based classification methodology that can leverage devices with limited number of resources, such as in the case of ubiquitous computing, allowing for resource-efficient hardware implementations. This efficient embedded fingerprint classification node is focused on the fusion of a Weightless Neural Network (WNN) architecture [30] and a technique, namely Virtual Neuron [31], which efficiently maps a neural network architecture into hardware resources. The use of a weightless neuron implies a remarkable simplicity on hardware design, reducing the computations required by a classic weighted neural network. The virtual neural-based classifier can be seen as a serial-parallel architecture representing the resulting trade-off between a fast classification rate and the memory structure efficient management in a neural based classifier. The resulting neural-based node aims at achieving high performance about execution time, needed hardware resources and classification rate. Our approach analyzes the directional image, previously extracted from the original fingerprint image without any pre-processing enhancement, and classifies the fingerprint into the five National Institute of Standards and Technology (NIST) Biometric Image Software (NBIS) standard classes: Arch, Tented Arch, Whorl, Left Loop and Right Loop [32]. The fingerprint classifier has been designed for Field Programmable Gate Array (FPGA) devices, exploiting pipeline techniques and parallelism for execution time reduction. So doing, we can identify the reliability and effectiveness of the proposed system as well as avoid unnecessary complications that may arise from choosing unsuitable technologies. The system has been prototyped on a Xilinx ML507 board (Xilinx Inc., San Jose, CA, USA) [33] and the experimental trials have been performed, following a 10-fold validation strategy [34], over the whole official FVC2002DB2 database composed of 880 fingerprint images [35].

The paper is structured as follows. In Section 2, the proposed neural-based node is detailed. In Section 3, the realized hardware prototype is described and the experimental results, in terms of classification rate, hardware resources and execution times, are shown. A comparison analysis, based on execution times and used resources, between the proposed hardware classifier and the literature hardware recognition systems is reported in Section 4. Finally, Section 5 provides some conclusive remarks.

2 The Proposed Neural-based Node

The main contribution of this work is represented by the fusion of a WNN architecture and a technique, namely Virtual Neuron, properly combined in a smart fashion to propose a novel embedded neural classification methodology for devices with limited number of resources. Several constraints, such as high modularity, high neuron density, high recognition rate, and low processing time were considered for the architectural design developing.

In the following subsections the fingerprint directional image extraction phase, the WNN architecture, Virtual Neuron technique, and the implemented fusion approach are detailed.

2.1 The Fingerprint Directional Image Extraction Phase

The proposed neural-based node classifies the fingerprints belonging to the whole official Fingerprint Verification Competition (FVC2002DB2) database [35] into the five NBIS standard classes: Right Loop (R), Left Loop (L), Whorl (W), Arch (A) and Tented Arch (T) (see Figure 1 for further details).

The proposed classification system processes the directional image, extracted from the original fingerprint image without any enhancement pre-processing phase [17][31]. The reduced size of the directional images decreases the design complexity, thus leading to less used resources and higher speed. As depicted in Figure 2, each element in the directional image represents the local orientation of the ridges in

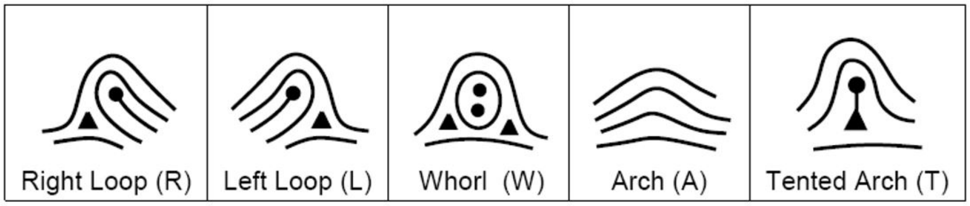


Figure 1: The used five fingerprint classes: Right Loop (R), Left Loop (L), Whorl (W), Arch (A) and Tented Arch (T).

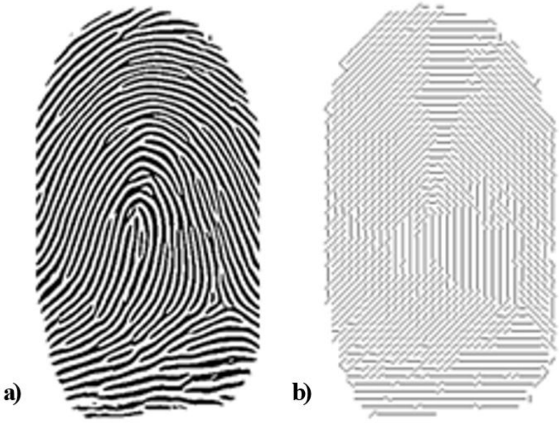


Figure 2: Example of the analyzed directional images: a) the original fingerprint image, and b) the extracted directional image.

the original gray-scale image.

The direction $K(i,j)$ of the point (i,j) is defined by the following Equation:

$$K(i,j) = \min\left\{\sum_{k=1}^{L}\left[C(i_k,j_k) - C(i,j)\right]\right\} \tag{1}$$

where $C(i_k,j_k)$ and $C(i,j)$ are the gray levels of the points (i_k,j_k) and (i,j), respectively, while L is the number of selected pixels in this computation along a given direction. In this work, 31 directions were chosen and $L = 16$ pixels along the direction were analyzed (from 0° to 180° degrees).

In the directional image extraction phase, firstly, the fingerprint image is divided into blocks of 8×8 pixels, then, for each foreground block, the directions of each pixel are calculated using Eq. 1, and finally the direction with the greatest frequency is attributed to the selected block. Starting from the fingerprint central position, an area consisting of 37×19 directions is selected. Therefore, each directional image contains 703 directions and each direction is encoded as a 5-bit word.

In some blocks, the possible presence of noise could generate directions that are considerably different with respect to their neighbor directions, thus a smoothing algorithm is applied on the directional image. This is achieved according to the directional histogram obtained from a direction comparison between blocks of 3×3 pixels. Finally, the central block direction of the noisy area is replaced by the majority of the neighboring blocks.

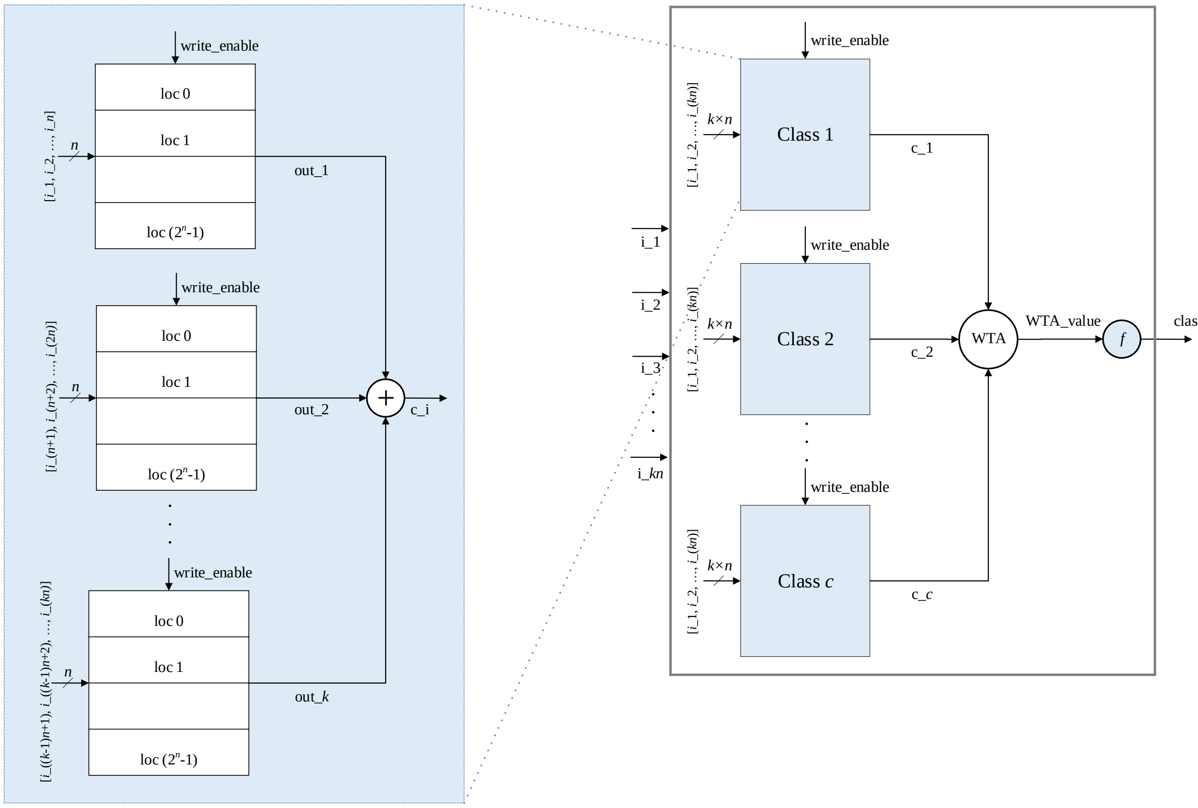


Figure 3: The WISARD-based WNN scheme: a) RAMs correspond to different patterns of the sampled input data; b) the discriminators correspond to different object classes.

2.2 The Weightless Neural Network Architecture

WNNs represent a class of methods for building pattern recognition systems [30]. They belong to the memory-based architectures and consist of a set of RAMs, called discriminators, which store the image numeric features (similarly to the conventional neural network weights). Each RAM memory samples and stores a certain amount of data from the input space.

WNNs based on the Wilkie, Stonham, Aleksander's Recognition Device (WISARD) approach [30] employ as many discriminators as the number of the final classes. Figure 3 shows the WISARD-based WNN scheme, where the 0 or 1 values are used as numeric features and stored in RAMs. Especially, a value of 1 corresponds to a specific feature of the training set for a certain class.

Unlike other neural networks, these architectures can be trained very rapidly and be implemented in simple and efficient hardware structures [25].

The RAM output value corresponds to a partial input data, while, the discriminator output value corresponds to the whole input data. With more details, the discriminator is composed of k unit n-RAM (addressable to n-bits) and it processes input of $k \times n$ bits. All RAM units belonging to a discriminator are linked to the same class. The input set is divided into n-bit sequences. Each RAM unit gives a result only on its own input sequence: if this sequence belongs to a RAM unit linked to the same class, then its output will be set to 1, otherwise to 0. Before performing the training phase, all RAM units are set to 0. The training phase is performed for each discriminator with input examples. Each n-bit input sequence belonging to each example will set to 1 the correct memory location. Finally, the Winner-Take-All (WTA) module performs the input pattern classification, according to the maximum among the

discriminators' output values. Therefore, a discriminator is a neuron specialized to classify the whole directional image: the higher is its output value, the more similar is the input with respect to the model learned from its training set.
The fingerprint images have to be classified into 5 classes (R, L, W, A, and T) using their directional images alone, wherein each directional image contains 703 directions and each direction is encoded as a 5-bit word. Thus, the proposed neural-based classifier should implement 5 discriminators and each discriminator should consist of 703 RAMs of 32 locations. However, each discriminator is composed of just 2 RAMs, in order to minimize the number of the required hardware resources. Thus, it is able to process only two directions in parallel for each iteration and, consequently, the input processing is performed in 352 iterations.

2.3 The Virtual Neuron Technique

Typical classification tasks in real-world applications, which exploit neural networks, need for a huge amount of neurons, forcing FPGA-based neural architectures to store the neuron weights on external memories. Neural networks are usually wide or deep and input data are fed to the network and then processed just a single time, giving generally rise to a memory-bound process. The Virtual Neuron technique exploits a hybrid serial-parallel architecture: input data are loaded in a serial way and these data are processed in parallel by the Virtual Neurons [31].
Let suppose that the WNN is composed of N neurons. The Virtual Neuron technique implements only h Virtual Neurons, where h is a sub-multiple of N, to implement the whole neural network.
Firstly, during the initialization of the network, the h Virtual Neurons, simulating the first h physical neurons, read the weights and input data from the memory. After data processing, the computed results are split into the h shift-register accumulators, one for each Virtual Neuron. Each shift-register accumulator consists of $\frac{N}{h}$ locations. Then, the physical neurons, with indices from $h+1$ to $2h$, are simulated by the same set of h Virtual Neurons. Accordingly, they take their weights from the memory and process the same input data again. They do not need to read the input data from the memory. The procedure ends when all the physical neurons are simulated by the virtual ones. Lastly, the WTA module considers all the yielded results and selects the highest one. The Virtual Neuron represents the fingerprint class discriminator. Only 2 virtual discriminators are implemented against 5 physical ones, so a total number of 6 physical discriminators can be simulated, but the latter will be never used. In more detail, the first Virtual Neuron simulates the discriminators for the fingerprint classes Arch, Whorl and Left Loop, while the second one simulates the discriminators for the fingerprint classes Tented Arch and Right Loop.

2.4 Overall Hardware Organization of the Classification Node

The proposed embedded classification node, which combines the WNN architecture and the Virtual Neuron technique, offers the following benefits. The use of a weightless neuron implies a remarkable simplicity on hardware design, reducing the computations required by a classic neural network. In more detail, the scalar products are replaced by a simple reading of memory locations, and the application of an activation function for each neuron is not required; just the calculation of the maximum between the responses of each discriminator is performed. In addition, the use of Virtual Neurons minimizes the number of required resources. This is much more evident by increasing the number of the network neurons and layers. Figure 4 shows the proposed neural-based classification scheme.

The PowerPC is used in both training and classification phases. In the training phase, it extracts the directional images from the training fingerprint images and stores them in the external memory. In the classification phase, it extracts the directional image from the fingerprint image to be classified, and sends to the FPGA two streams (i.e., according to the number of Virtual Neurons), composed of the input

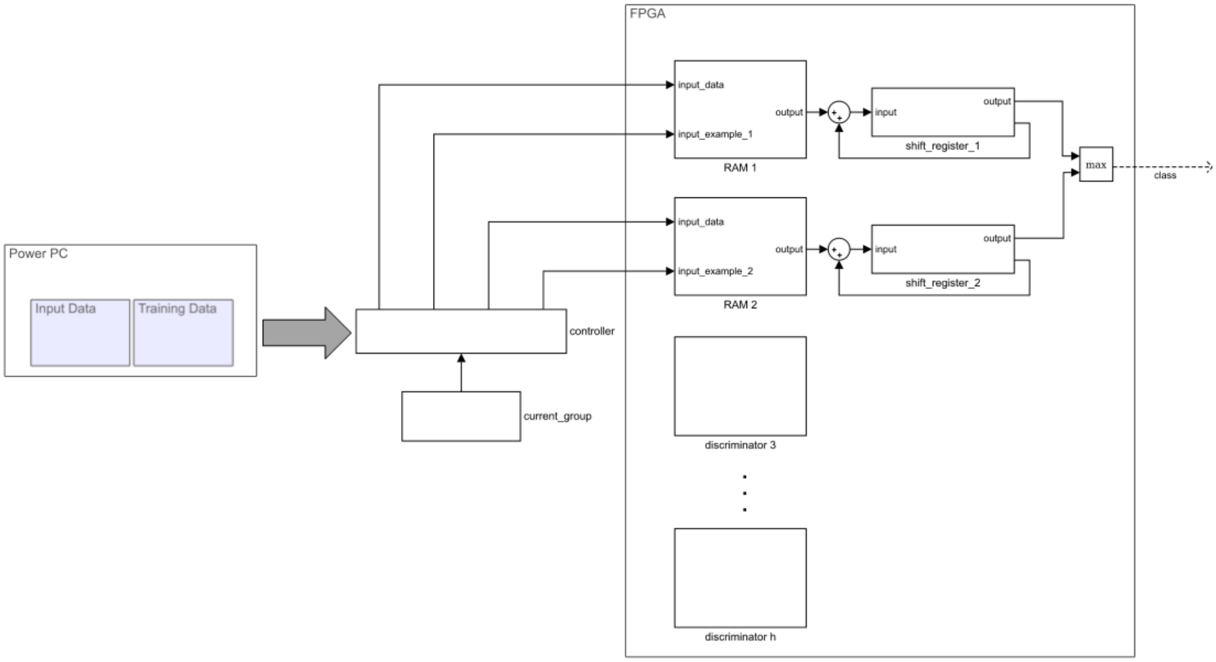


Figure 4: The proposed neural-based classification scheme, highlighting the hardware components.

	Class_L	Class_W	input	Class_A
Stream_1	32 bit	32 bit	10 bit	32 bit

	Class_6	Class_R	input	Class_T
Stream_2	32 bit	32 bit	10 bit	32 bit

Figure 5: Structure of the two streams used to drive the two Virtual Neurons.

data to be classified and the RAMs configuration data (see Figure 5).

The two streams are used by the controller module to drive the two Virtual Neurons concerning the fingerprint classes Arch, Whorl and Left Loop, and the classes Tented Arch and Right Loop, respectively. With more details, each stream consists of 4 fields: the 32-bit fields represent the content of the RAMs referred to the current input and to a given discriminator group, while the 10-bit field refers to the input portion to which that group is associated with. So, each RAM conveys all the necessary information so that each discriminator can perform the classification of a portion of the input.
The input is divided into 10-bit words so, for each iteration, only a 10-bit input is classified by each discriminator. Especially, for each step, the controller configures the Virtual Neurons with the appropriate 32-bit *inputs_example_1* and *inputs_example_2*, by providing to them the 10-bit *input_data_1* and *input_data_2*, respectively (see Figures 4 and 5). Then, the two RAMs for each discriminator will yield the contents of the locations addressed by the 5 bits. These output values are summed with the contents of the shift-register and then the register is shifted. In each step, the two Virtual Neurons execute these operations in parallel. At the end of the 352 iterations, each shift-register location will contain the response of each virtual discriminator. Finally, a WTA module classifies the input according to the maximum value of the two registers. Figure 6 shows the proposed neural-based classification node designed

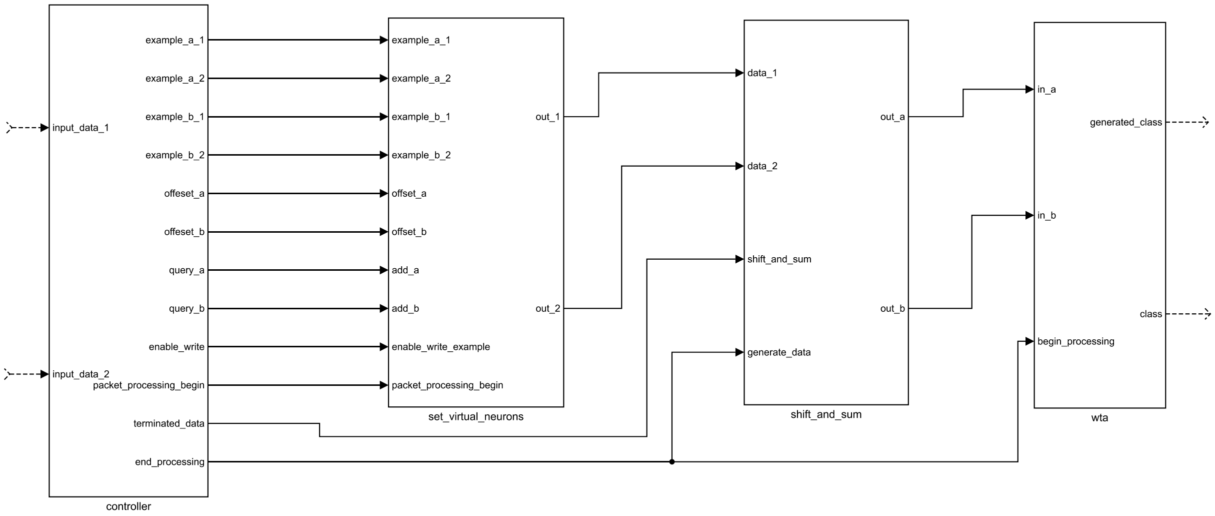


Figure 6: Schematics of the proposed neural-based fingerprint classification node designed using the Simulink®/MATLAB® software package.

using the Simulink tool integrated in the MATLAB environment (The MathWorks Inc., Natick, MA, USA) [36]. The designed modules are synchronized and implement a three-stage pipeline. While the "Controller Module" sends the class L configuration data, the "Set Virtual Neurons Module" configures the RAMs with the class W and class R configuration data and then interrogates each RAM by using the input data. Finally, the "Shift and Sum Module" updates the shift-registers with the RAMs output value for the classes A and T.

3 Experimental Results

The proposed node has been developed and deployed on the Xilinx ML507 board [33], equipped with a Virtex-5 FPGA, and tested on the official FVC2002DB2 database containing a total of 880 fingerprint images, belonging to the five classes defined by NIST NBIS [33].
A comparison against two different benchmark architectures, a Multi-Layer Perceptron (MLP) with 30 hidden neurons and a WNN with 350 RAMs each of which consists of 10 locations, was performed to test the neural-based classification performance. So doing, also the improvements introduced by the developed fusion scheme were evaluated.
This section firstly describes the components of the final hardware-based prototype. Afterwards, the experimental results—in terms of hardware resources, classification rate, and execution times—are shown and analyzed, highlighting the effectiveness of the proposed architecture.

3.1 Prototype Implementation

The realized prototype (see Figure 7 for a graphical representation) uses two Intellectual Properties (IPs) [37] corresponding to the training module and the recognition module implemented on the XC5VFX70T-1FFG1136 FPGA. These two IP cores are connected to the hard-core PowerPC 440 through the Processor Local Bus (PLB) [38]. In what follows, each component is described:

- PLB, which connects all peripherals to hard-core PowerPC 440 for the arbitrage system;
- *classify_mod* is the classification module;

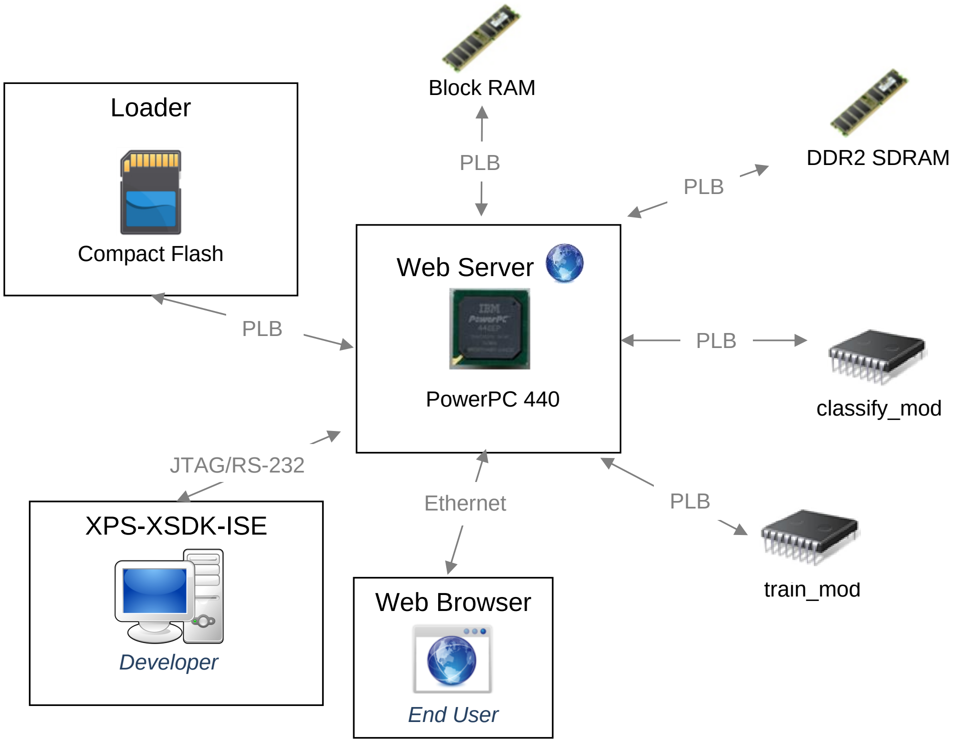
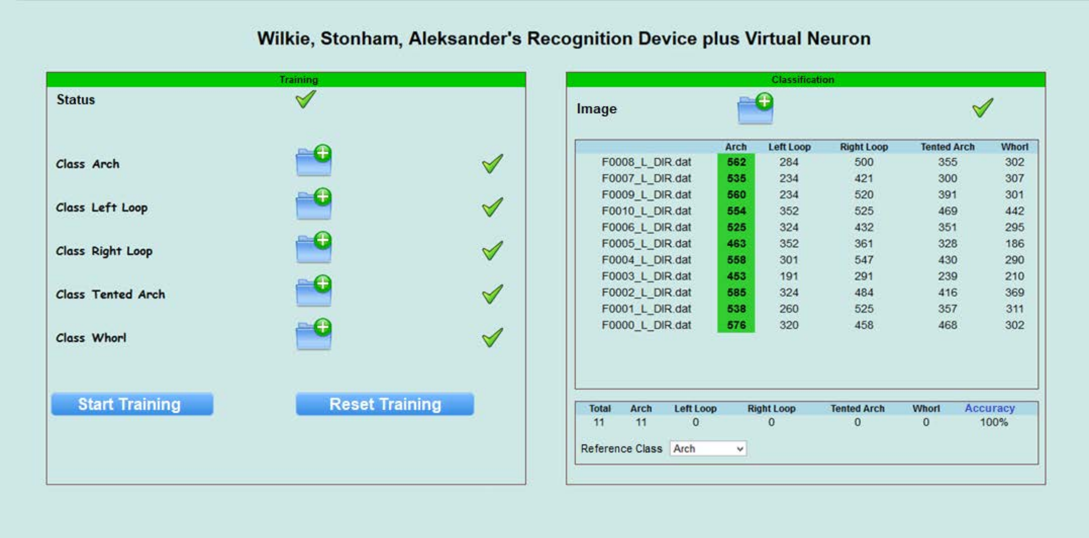


Figure 7: Implementation of the hardware-based prototype for automatic fingerprint classification.

- *train_mod* is the training module;
- System Ace Controller allows to load, in reading phase, a compact flash where a website MFS image is stored as a front-end for the test environment and the configuration stream generated by the training system;
- DDR2 SDRAM is a buffer of both data transmission sent to classification and training modules as well as reception for the results acquisition obtained by the two implemented modules. Moreover, it also maintains the front-end website;
- Block RAM stores the program startup procedure to run on the PowerPC;
- Ethernet allows to send the user data from the Web Server to the compact flash, allowing the user to communicate with the card and then with the two implemented modules;
- JTAG and RS-232 ports support hardware designers and developers during programming and debugging phases (by exploiting the Xilinx ISE Design Suite tools), while these communication interfaces are not used by the end user.

The Web Server, running on the PowerPC 440 processor, enables the communication of the user with the system by means of a simple Web-based interface, displayed in Figure 8. This Web-based interface allows to operate in two modes: training and classification. The training mode requires the user to select a number of images per class. Once the "Start Training" command is launched, the Web server encodes consistently the "examples" with the input format of the training system and provides the data to the training module. After that the training phase is over, the server will overwrite the training



	Arch	Left Loop	Right Loop	Tented Arch	Whorl
F0008_L_DIR.dat	**562**	284	500	355	302
F0007_L_DIR.dat	**535**	234	421	300	307
F0009_L_DIR.dat	**560**	234	520	391	301
F0010_L_DIR.dat	**554**	352	525	469	442
F0006_L_DIR.dat	**525**	324	432	351	295
F0005_L_DIR.dat	**463**	352	361	328	186
F0004_L_DIR.dat	**558**	301	547	430	290
F0003_L_DIR.dat	**453**	191	291	239	210
F0002_L_DIR.dat	**585**	324	484	416	369
F0001_L_DIR.dat	**538**	260	525	357	311
F0000_L_DIR.dat	**576**	320	458	468	302

Total	Arch	Left Loop	Right Loop	Tented Arch	Whorl	Accuracy
11	11	0	0	0	0	100%

Reference Class: Arch

Figure 8: Screenshot of the Web-based interface developed to enable the interaction between the end user and the system.

Table 1: FPGA used resources by the selected architectures.

Resource type	WNN	MLP	The proposed method
Slices	5317	1041	358
IOBs	120	190	18
Block RAMs	14	9	0

stream on the memory card and load into the RAM memory the same stream that will be ready for a subsequent classification phase. The classification mode can be used to both classify fingerprints on-the-fly and evaluate the system. The fingerprint classification phase requires the selection of the files containing the directional information, once the recognition is completed, for each image the assigned class as well as the number of scores received for each class will be yielded. The system evaluation uses a graphical interface where the user can specify the selected image membership class; after file selections, the system will query the Web Server and produce the correct class index as response. The accuracy metrics represents the percentage of fingerprints that have been assigned to the class specified by the user (i.e., a test set of input samples whose classes are known is provided for validation).

3.2 Used Resources Analysis

Table 1 reports the FPGA used resources by the selected architectures. In the proposed fusion architecture, no Block RAMs have been used since the whole architecture has been implemented using Slices and IOBs. This table shows the high resources reduction when the proposed solution is adopted. Furthermore, the solution based on the MLP architecture consists of the test set alone, while the WNN and the proposed fusion solutions involve the whole classification process (i.e., training and testing phases).

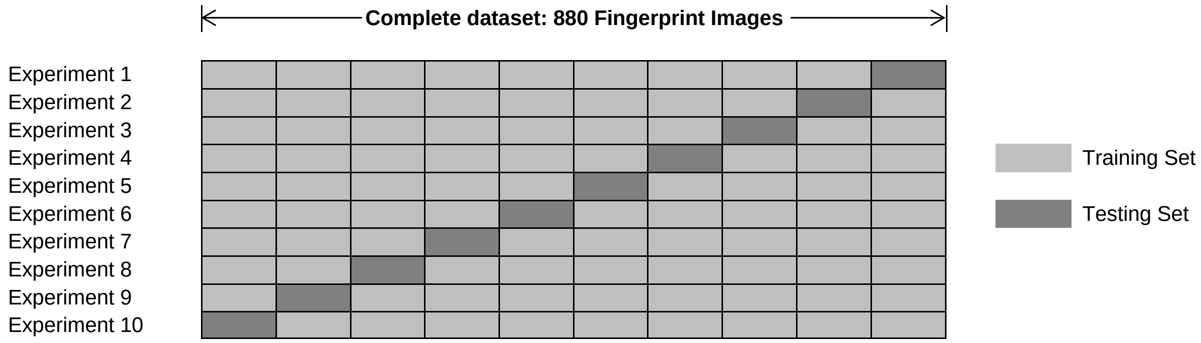


Figure 9: The 10-fold cross-validation strategy used to evaluate the proposed methodology.

Table 2: The average classification rates achieved for each fingerprint class by the tested architectures: WNN, MLP, and the proposed fusion method.

Class	WNN Classification Rate	MLP Classification Rate	The proposed method Classification Rate
Right Loop	95.88%	86.47%	89.41%
Left Loop	94.68%	93.62%	91.49%
Whorl	75.29%	83.53%	92.94%
Arch	62.50%	70.83%	87.50%
Tented Arch	100.00%	90.00%	70.00%
Total Average	89.03%	86.68%	90.08%

3.3 Classification Rate Analysis

Table 2 shows the average classification rates achieved for each fingerprint class by the selected architectures. Aiming at calculating the best performance among the three implemented architectures and eliminate possible image selection problems between training and test sets, $k = 10$ experiments have been executed using a 10-fold cross-validation strategy [34]: the whole database has been divided into 10 different disjoint sets of 88 fingerprint images. In each round, 9 partitions have been used for training phase and the remaining set for the testing phase, to reduce problems related to over-fitting and evaluate generalization capabilities of the model (see Figure 7).

The achieved results highlight that the MLP overall average classification rate is the lowest value (i.e., this type of neural architecture needs for a bigger and opportunely selected training set) and that the proposed fusion approach achieves the best overall performance.

3.4 Clock Cycles and Execution Time Analysis

Table 3 depicts the number of clock cycles and the execution time for each of the selected architectures running at 60 MHz. The low clock cycles number of the MLP architecture is due to the fact that only the testing phase can be implemented on a real board.

More interestingly, it is worth noting the significant difference between the proposed solution and the WNN architecture. The Virtual Neuron technique fused with the WNN architecture reduces execution time, obtaining a speed-up of $2.9\times$ with respect to the traditional WNN architecture (more than 65%), considering both training and testing phases. As a matter of fact, the MLP characteristics do not allow for the whole implementation on a board, so only the testing phase has to be taken into account.

Table 3: Clock cycles and Execution Time of the implemented architectures running at 60 MHz.

Architecture	Clock Cycles	Execution Time (ms)
WNN	51,072	0.85
MLP	15,384	0.20
The proposed	17,225	0.29

4 Discussion

Automatic fingerprint recognition systems may be composed of two modules: (*i*) classification, and (*ii*) identification. Basically, the AFIS module returns a binary response denoting if the recognition process is either successful or not, so allowing or denying a user access. On the other hand, the automatic fingerprint classification module enables a valuable indexing scheme to promote an effective retrieval from large fingerprint databases in software and hardware authentication systems. Several techniques and prototypes have been proposed for fingerprint classification system implementations aiming at increasing AFIS performance. However, as said before, no hardware-based fingerprint classification system has yet been proposed. Therefore, the following quantitative analysis is focused on the improvements, in terms of execution times and used resources, introduced by our classification node with respect to the literature hardware recognition systems. The main idea of this paper is to integrate the proposed hardware classifier, which employs few resources as well as achieves low execution time, in the full recognition system pipeline to improve recognition performance by remarkably reducing total processing time on devices with limited resources.

Table 4 compares the execution times achieved by the proposed fingerprint classification node with respect to the literature hardware recognition systems (considering both complete and partial recognition pipelines). It is worth noting that all these systems implemented an identification module, without proposing a classification scheme. The achieved execution times show that our hardware-based classifier is considerably faster than the state-of-the-art hardware recognition systems, also with reference to the several different hardware platforms (running at different working frequencies).

In addition, Table 5 shows the hardware resources used by the state-of-the-art fingerprint identification prototypes. As a result of this analysis, it can be argued that the proposed classification module can be efficiently implemented and integrated into a complete hardware recognition system, since the total amount of needed resources is certainly available on the current boards.

On the other hand, the above results make feasible the efficient implementation of the proposed neural-based automatic classifier either as a whole standard module or as a component of a complex fingerprint recognition system on devices with limited amount of resources.

5 Conclusion

This paper presented an efficient embedded neural-based node for fingerprint classification exploiting the fusion of a Weightless Neural Network architecture and a technique, namely Virtual Neuron, implementing an efficient mapping of a neural network architecture for hardware devices with limited number

Table 4: Execution times of the proposed fingerprint classification node compared against the state-of-the-art hardware identification prototypes. The working frequency of the hardware platform is also reported.

System	Hardware platform	Working frequency (MHz)	Processing time (ms)
Bonato *et al.* [39]	Altera FLEX10KE	N/A	306
Schaumont *et al.* [40]	Xilinx Virtex-II XC2V1000	N/A	4000–6000
Fons *et al.* [41]	Xilinx Virtex-4	100	205.025
Lopez and Cantò [42]	Xilinx Spartan 3	40	261.9
Fons *et al.* [43]	Atmel AT40K	25	7239040
Garcia and Cantò [44]	Xilinx Spartan 3	50	261.9
Vitabile *et al.* [31]	Xilinx Spartan II 200K	22.5	541
Militello *et al.* [45]	Xilinx Virtex-II	25.175	34.8
The proposed classification node	Xilinx ML507 board equipped with Virtex-5	60	0.29

Table 5: Hardware resources required by the state-of-the-art hardware identification prototypes compared against the proposed classifier.

System	Gates	RAMs	Flip Flops	Slices	IOBs	LUTs
Fons *et al.* [41]	N/A	1.77 MB	31,948	N/A	N/A	43,879
Fons *et al.* [43]	1612	N/A	462	N/A	N/A	N/A
Militello *et al.* [45]	N/A	1.81 MB	5,138	12,863	212	20,825
The proposed classifier	N/A	0	377	358	18	1,941

of resources. In other words, an embedded and resource-efficient hardware classification architecture for ubiquitous and mobile computing was proposed [5].

The proposed neural-based node classifies an acquired fingerprint into the five NIST NBIS classes using only its directional image. These images are extracted without any enhancement phase. The fingerprint classifier was implemented on a Xilinx ML507 board equipped with a Virtex-5 FPGA and it is characterized by hardware resources, high accuracy, and low execution time (also with respect to the literature hardware recognition systems). The achieved average classification rate on the official FVC2002DB2 database is 90.08%, with a significant resources and execution time reduction, and a speed-up of 2.9× (higher than 65%), considering both training and testing phases, compared against the traditional WNN architecture.

This is the first automatic fingerprint classifier proposing a resource-efficient hardware implementation. Hence, the developed neural-based node could be specifically exploited as a feasible biometric-based authentication solution by mobile devices in edge computing contexts with a huge number of end users [46, 9]. As a matter of fact, the proposed solution could have a valuable impact in practical applications, such as mobile health, ambient intelligence, and smart transportation, since authentication is a critical issue in ubiquitous, mobile, and edge computing [6].

References

[1] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016.

[2] H. Liu and C.-M. Zhang, "Research on use of distributed authentication in pervasive computing," in *Proc. of the 1st International Symposium on Pervasive Computing and Applications (SPCA'06), Urumqi, China*. IEEE, August 2006, pp. 571–574.

[3] S. Yin, I. Ray, and I. Ray, "A trust model for pervasive computing environments," in *Proc. of the 2006 International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom'06), Atlanta, Georgia, USA*. IEEE, November 2006. [Online]. Available: https://doi.org/10.1109/COLCOM.2006.361880

[4] R. Sailer and J. Giles, "Pervasive authentication domains for automatic pervasive device authorization," in *Proc. of the 2nd IEEE Conference on Pervasive Computing and Communications Workshops Pervasive Computing and Communications Workshops (PerCom'04), Orlando, Florida, USA*. IEEE, March 2004, pp. 144–148.

[5] V. Desnitsky, D. Levshun, A. Chechulin, and I. Kotenko, "Design technique for secure embedded devices: application for creation of integrated cyber-physical security system," *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, vol. 7, no. 2, pp. 60–80, June 2016.

[6] W. Shi and S. Dustdar, "The promise of edge computing," *Computer*, vol. 49, no. 5, pp. 78–81, 2016.

[7] Y. Ai, M. Peng, and K. Zhang, "Edge cloud computing technologies for Internet of Things: A primer," *Digital Communications and Networks*, July 2017.

[8] S. Yi, C. Li, and Q. Li, "A survey of fog computing: concepts, applications and issues," in *Proc. of the 2015 International Workshop on Mobile Big Data (MoBiData'15), Hangzhou, China*. ACM, June 2015, pp. 37–42.

[9] I. Stojmenovic and S. Wen, "The fog computing paradigm: scenarios and security issues," in *Proc. of the 2014 Federated Conference on Computer Science and Information Systems (FedCSIS'14), Warsaw, Poland*. IEEE, September 2014, pp. 1–8.

[10] D. Maltoni, D. Maio, A. Jain, and S. Prabhakar, *Handbook of Fingerprint Recognition 2nd Edition*. Springer Publishing Company Inc., 2009.

[11] C. Militello, V. Conti, S. Vitabile, and F. Sorbello, "An embedded iris recognizer for portable and mobile devices," *International Journal of Computer Systems Science & Engineering*, vol. 25, no. 2, pp. 119–131, 2010.

[12] V. Conti, C. Militello, S. Vitabile, and F. Sorbello, "A multimodal technique for an embedded fingerprint recognizer in mobile payment systems," *International Journal on Mobile Information Systems*, vol. 5, no. 2, pp. 105–124, 2009.

[13] V. Conti, S. Vitabile, G. Vitello, and F. Sorbello, "An embedded biometric sensor for ubiquitous authentication," in *Proc. of the AEIT Annual Conference (AEIT'13), Mondello (PA), Italy*. IEEE, October 2013. [Online]. Available: https://doi.org/10.1109/AEIT.2013.6666815

[14] H. Sbeyti, M. Malli, K. Al-Tahat, A. Fadlallah, and M. Youssef, "Scalable extensible middleware framework for context-aware mobile applications (SCAMMP)," *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, vol. 7, no. 3, pp. 77–98, 2016.

[15] M. Kamijo, "Classifying fingerprint images using neural network: deriving the classification state," in *Proc. of the 1993 IEEE International Conference on Neural Networks (ICNN'93), San Francisco, CA, USA*. IEEE, March 1993, pp. 1932–1937.

[16] V. Conti, C. Militello, F. Sorbello, and S. Vitabile, "Biometric sensors rapid prototyping on FPGA," *The Knowledge Engineering Review*, vol. 30, no. 2, pp. 201–219, 2015.

[17] J. Hong, L. Xueyan, G. Shuxu, and F. Hua, "Classification of the incomplete fingerprint image," in *Proc. of the 2010 International Conference on Computer, Mechatronics, Control and Electronic Engineering (CMCE'10), Changchun, China*. IEEE, August 2010, pp. 281–284.

[18] T. Kristensen, J. Borthen, and K. Fyllingsnes, "Comparison of neural network based fingerprint classification techniques," in *Proc. of the 2007 International Joint Conference on Neural Networks (IJCNN'07), Orlando, Florida, USA*. IEEE, August 2007, pp. 1043–1048.

[19] R. Wang, C. Han, and T. Guo, "A novel fingerprint classification method based on deep learning," in *Proc. of the 23rd International Conference on Pattern Recognition (ICPR'16), Cancun, Mexico*, December 2006, pp. 931–936.

[20] H. Guo, Z. Ou, and Y. He, "Automatic fingerprint classification based on embedded hidden Markov models," in *Proc. of the 2003 International Conference on Machine Learning and Cybernetics (ICMLC'03), Xi'an, China*, vol. 5. IEEE, November 2003, pp. 3033–3038.

[21] N. Alias and N. Radzi, "Fingerprint classification using support vector machine," in *Proc. of the 5th ICT International Student Project Conference (ICT-ISPC'16), Nakhon Pathom, Thailand*. IEEE, May 2016, pp. 105–108.

[22] K. Dorasamy, L. Webb, J. Tapamo, and N. Khanyile, "Fingerprint classification using a simplified rule-set based on directional patterns and singularity features," in *Proc. of the 2015 International Conference on Biometrics (ICB'15), Phuket, Thailand*. IEEE, May 2015, pp. 400–407.

[23] V. Conti, C. Militello, S. Vitabile, and F. Sorbello, "Introducing pseudo-singularity points for efficient fingerprints classification and recognition," in *Proc. of the 4th IEEE International Conference on Complex, Intelligent and Software Intensive Systems (CISIS'10), Krakow, Poland*. IEEE, February 2010, pp. 368–375.

[24] N. Chauhan, M. Soni, V. Anand, and V. Kanhangad, "Fingerprint classification using crease features," in *Proc. of the 2016 IEEE Students' Technology Symposium (TechSym'16), Kharagpur, India*. IEEE, September 2016, pp. 56–60.

[25] M. Ballan, F. Sakarya, and B. Evans, "A fingerprint classification technique using directional images," in *Proc. of the 31st Asilomar Conference on Signals, System and Computers (ACSSC'97), Pacific Grove, California, USA*. IEEE, November 1997, pp. 101–104.

[26] D. Maltoni and D. Maio, "A structural approach to fingerprint classification," in *Proc. of the 13th International Conference on Pattern Recognition (ICPR'96), Vienna, Austria*. IEEE, August 1996, pp. 578–585.

[27] H. Neto and D. Borges, "Fingerprint classification with neural networks," in *Proc. of the 4th IEEE Brazilian Symposium on Neural Networks (SBRN'97), Goiania, Brazil*. IEEE, December 1997, pp. 66–72.

[28] V. Conti, C. Militello, S. Vitabile, and F. Sorbello, "An embedded fingerprints classification system based on weightless neural networks," *Frontiers in Artificial Intelligence and Applications: New Directions in Neural Networks*, vol. 193, pp. 67–95, 2009.

[29] G. Vitello, F. Sorbello, G. Migliore, V. Conti, and S. Vitabile, "A novel technique for fingerprint classification based on fuzzy c-means and naive Bayes classifier," in *Proc. of the 8th IEEE International Conference on*

Complex, Intelligent and Software Intensive Systems (CISIS'14), Birmingham, UK. IEEE, July 2014, pp. 155–161.

[30] I. Aleksander, "Learning and distributed memory: Getting close to neurons," in *Machine Learning: Principles and Techniques*, R. Forsyth, Ed. Chapman & Hall Ltd., 1988, pp. 104–123.

[31] S. Vitabile, V. Conti, F. Gennaro, and F. Sorbello, "Efficient MLP digital implementation on FPGA," in *Proc. of the 8th IEEE Euromicro Conference on Digital System Design: Architectures, Methods and Tools (DSD'05), Porto, Portugal*. IEEE, August 2005, pp. 218–222.

[32] K. Ko and W. J. Salamon, "NIST Biometric Image Software," http://www.nist.gov/itl/iad/ig/nbis.cfm [Online; Accessed on December 17, 2017], April 2015.

[33] Xilinx Inc., "ML505/ML506/ML507 Evaluation Platform: User Guide," https://www.xilinx.com/support/documentation/boards_and_kits/ug347.pdf [Online; Accessed on December 17, 2017], May 2011.

[34] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Proc. of the 14th International Joint Conference on Artificial intelligence (IJCAI'95), Montreal, Quebec, Canada*, vol. 2. Morgan Kaufmann Publishers Inc., August 1995, pp. 1137–1143.

[35] BIOLAB University of Bologna, "Second International Competition for Fingerprint Verification Algorithms, FCV2002 database," http://bias.csr.unibo.it/fvc2002/ [Online; Accessed on December 17, 2017], April 2002.

[36] The MathWorks Inc., "MathWorks Simulink," https://it.mathworks.com/products/simulink.html [Online; Accessed on December 17, 2017], December 2017.

[37] Xilinx Inc., "Intellectual Property," http://www.origin.xilinx.com/products/intellectual-property.html [Online; Accessed on December 17, 2017], November 2017.

[38] Xilinx Inc., "LogiCORE IP Processor Local Bus (PLB) v4.6 (v1.05a)," http://www.xilinx.com/support/documentation/ip_documentation/plb_v46.pdf [Online; Accessed on December 17, 2017], September 2010.

[39] L. V. Bonato, R. F. Molz, J. C. Furtado, M. F. Ferrão, and F. G. Moraes, "Design of a fingerprint system using a hardware/software environment," in *Proc. of the 11th International Symposium on Field Programmable Gate Arrays (ACM/SIGDA'03), Monterey, California, USA*. ACM, February 2003, pp. 240–240.

[40] P. Schaumont, K. Sakiyama, Y. Fan, D. Hwang, S. Yang, A. Hodjat, B. Lai, and Verbauwhede, "Testing ThumbPod: softcore bugs are hard to find," in *Proc. of the 8th IEEE International High-Level Design Validation and Test Workshop (HLDVT'03), San Francisco, California, USA*. IEEE, November 2003, pp. 77–82.

[41] M. Fons, F. Fons, and E. F. Cantò Navarro, "Fingerprint image processing acceleration through run-time reconfigurable hardware," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 57, no. 12, pp. 991–995, 2010.

[42] M. Lopez and E. F. Cantò Navarro, "FPGA implementation of a minutiae extraction fingerprint algorithm," in *Proc. of the 2008 IEEE International Symposium on Industrial Electronics (ISIE'08), Cambridge, UK*. IEEE, June 2008, pp. 1920–1925.

[43] M. Fons, F. Fons, and E. Cantò Navarro, "Design of FPGA-based hardware accelerators for on-line fingerprint matcher systems," in *Proc. of the 2006 Ph.D. Research in Microelectronics and Electronics (RME'06), Otranto, Italy*. IEEE, June 2006, pp. 333–336.

[44] M. L. Garcia and E. F. Cantò Navarro, "FPGA implementation of a ridge extraction fingerprint algorithm based on microblaze and hardware coprocessor," in *Proc. of the 2006 International IEEE Conference on Field Programmable Logic and Applications (FPL'06), Madrid, Spain*. IEEE, August 2006. [Online]. Available: https://doi.org/10.1109/FPL.2006.311198

[45] C. Militello, V. Conti, S. Vitabile, and Sorbello, "A novel embedded fingerprints authentication system based on singularity points," in *Proc. of the 2nd IEEE International Conference on Complex, Intelligent and Software Intensive Systems (CISIS'08), Barcelona, Spain*. IEEE, March 2008, pp. 72–78.

[46] S. Yi, Z. Qin, and Q. Li, "Security and privacy issues of fog computing: a survey," in *Proc. of the 10th the International Conference on Wireless Algorithms, Systems, and Applications (WASA'15), Qufu, China*, ser. LNCS, vol. 9204. Springer, August 2015, pp. 685–695.

Author Biography

Vincenzo Conti is an Assistant Professor with the Faculty of Engineering and Architecture at the Kore University of Enna, Italy. He received his Laurea cum Laude degree and his Ph.D. in Computer Science from the University of Palermo in 2000 and 2005, respectively. His research interests include biometric recognition systems, programmable architectures, bio-inspired processing system, and user ownership in multi-agent systems. He has joined the Editorial Board of the Neurocomputing Journal, of the International Journal on Security and Communication Networks, of the International Journal of Biosensors and Bioelectronics, of the Computational Biology and Bioinformatics Journal and of the Complex and Intelligent Systems Journal, has chaired and participated at several national and international conferences, and he has co-authored over 60 scientific publications, journals and conferences. Moreover, he has participated to several research projects funded by industries and research institutes in his research areas. Currently, he collaborates with the Department of Industrial and Digital Innovation and with the Department of Biopathology and Medical Biotechnologies of the University of Palermo, and with Italian National Research Council (CNR - Palermo).

Leonardo Rundo received his Bachelor and Master Degrees (summa cum laude) in Computer Science Engineering from the University of Palermo, Palermo, Italy, in 2010 and 2013, respectively. Since December 2013, he has been Research Fellow at the Institute of Molecular Bioimaging and Physiology, National Research Council of Italy (IBFM-CNR), Cefalù (PA), Italy. He is currently a Ph.D. Student in Computer Science at the University of Milano-Bicocca, Milan, Italy. He was also Visiting Ph.D. Student at the Vanderbilt University, Nashville, TN, USA. He has also ongoing international collaborations with the Department of Creative Informatics, The University of Tokyo, Tokyo, Japan, and the Computer Laboratory, University of Cambridge, Cambridge, UK. His main scientific interests include Digital Image Processing, Biomedical Image Analysis, Machine Learning, Computational Intelligence, Natural Computing, Systems Biology, and High-Performance Computing.

Carmelo Militello received the Laurea (summa cum laude) and the Ph.D. degrees in computer engineering from the University of Palermo, Italy, in 2006 and 2010, respectively. During his Ph.D. program, he was involved in biomedical and biometrics imaging and digital hardware architectures development and implementation. Since 2011 he is researcher at the Institute of Molecular Physiology and Bioimaging, National Research Council of Italy (IBFM-CNR). His activity is also conducted in collaboration with the Department of Biopathology and Medical Biotechnology (DIBIMED), University of Palermo, the Department of Informatics, Systems and Communication (DISCo), University of Milano-Bicocca, and the Kore University of Enna, Italy. His main research fields are Digital Image Analysis and Processing, with particular interest to Biomedical and Biometric ones, Machine Learning and Clustering, Digital Architectures, and Hardware Programmable Devices.

Giancarlo Mauri is full professor of Computer Science at the University of Milano-Bicocca. His research interests include: natural computing and unconventional computing models; bioinformatics; computational systems biology, in particular stochastic modeling and simulation of biological systems and processes with High Performance Computing approaches, biomedical data analysis. On these subjects, he published about 350 scientific papers in international journals, contributed volumes and conference proceedings. He is currently member of the steering committees of the International Conference on Membrane Computing, of the International Conference on Unconventional Computing and Natural Computing and of the International workshop on Cellular Automata.

Salvatore Vitabile received the laurea degree in electronic engineering and the doctoral degree in computer science from the University of Palermo in 1994 and 1999, respectively. He is currently an associate professor with the Department of Biopathology and Medical Biotechnologies, University of Palermo, Italy. In 2007, he was a visiting professor in the Department of Radiology, Ohio State University, Columbus, USA. Prof. Vitabile joined the Editorial Board of the Journal of Ambient Intelligence and Humanized Computing (Springer), the International Journal of Information Technology, Communications and Convergence (Inderscience Publishers), and the International Journal of Space-Based and Situated Computing (Inderscience Publishers). He has co-authored more than 200 scientific papers in referred journals and conferences. He has chaired, organized, and served as member of the organizing committee of several international conferences and workshops. Dr. Vitabile is currently member of the Board of Directors of SIREN (Italian Society of Neural Networks). His research interests include specialized architecture design and prototyping, biometric authentication systems, driver assistance systems, and medical data processing and analysis.