UNIVERSITÀ DEGLI STUDI DI PALERMO

Dottorato in Ingegneria Chimica, Gestionale, Informatica e Meccanica – Indirizzo Informatica
Dipartimento di Ingegneria Chimica, Gestionale, Informatica, Meccanica
Settore Scientifico Disciplinare ING-INF/05

# Embedded Biometric Sensor Devices: Design and Implementation on Field Programmable Gate Array

IL DOTTORE
**Ing. Giuseppe Vitello**

IL COORDINATORE
**Prof. Ing. Antonio Gaglio**

IL TUTOR
**Prof. Ing. Antonio Gentile**

CICLO XXV
ANNO ACCADEMICO 2015

# PREFACE

During the research activity in my Ph.D. course, I thoroughly studied the biometric systems and the relevant design and implementation techniques allowing the employment of such systems in embedded devices. I focused my attention on the fingerprint-based recognition and classification systems, and on their implementation on Field Programmable Gate Array (FPGA) devices. I was prompted to study biometric systems mainly because these systems may play a key role in the new emerging market of mobile devices (for example, they are recently available in the new generation of Apple and Samsung smart phones). Such market is rapidly growing and influencing the way people use network resources and functionalities (such as commercial, banking, and government services), requiring a security level higher than in the past. Consequently, novel design techniques and technologies for user recognition and are required to be investigated, in order to provide a secure services and resources access. The traditional authentication systems based on username and password are not able to guarantee a suitable protection level. Unlike password, instead, user biometric information is unique and unchangeable; therefore the biometric identity has the advantage to guarantee that only the authorized users have access to available resources and services. However, traditional biometric approaches involves interactions among a large number of entities: passive access points for user biometric trait acquisition, networked databases for user biometric identity storing, and trusted servers running the user recognition systems. So, traditional systems usually undergo several types of attacks, such as *Communication Attack* (attacking the channel between the server and the database), *Replay Attack* (replication of user biometric trait processed during the acquisition phase), and *Database Attack* (manipulation of the stored user biometric identity). Embedded architectures, instead, provide a more secure and flexible infrastructure, since all elaboration steps are performed on board, so biometric identities are securely managed and stored inside the system without any data leaking out.

The goal of this thesis is to illustrate the analysis and results of my research activity focused on the design and development of new fingerprint-based recognition systems for embedded devices. The study of the state-of-the-art about biometric systems led me to realize novel approaches to improve the performance of standard systems in order to enable their employment in embedded devices architectures. Most common literature approaches used to implement fingerprint-based recognition and classification systems are reported to provide a starting-point for understanding the contribution of this work. There are many literature approaches to deal with software systems, but few on design and implementation of embedded hardware prototypes. Referring to the developed and proposed fingerprint-based systems, this thesis represents an advancement of embedded biometrics respect to state-of-the-art. The step-over proposed in this work is focused on:

1. a heuristic fingerprint classification technique, requiring only a little set of images as training dataset;
2. an advanced matching technique for personal recognition based on partial fingerprint, able to enhance the system accuracy;
3. the design and implementation of an efficient fingerprint features extractor;
4. the design and implementation of a quality evaluator of raw fingerprint images (able to identify poor quality areas, such as dry and moist portions), allowing to define a novel flow of image processing steps for user recognition.

This thesis is divided into two parts, creating a path connecting the state-of-the-art about biometric systems and the novel implemented approaches. The knowledge of the state-of-the-art about biometrics is fundamental to understand the step over presented in this work. For this reason, in the first part, general characteristics of biometric systems are presented with particular reference to fingerprint-based approaches used in literature to realize embedded systems. The second part proposes the developed innovative sensor. A novel flow of image processing steps for user recognition is outlined. Successively, an efficient micro and macro fingerprint features extractor is illustrated. Then, an advanced matching technique for personal recognition using partial fingerprints is presented. Finally, an innovative fingerprint classification approach based on the fusion of Fuzzy C-Means and Naive-Bayes technique is detailed. Experimental results and comparisons with analogous literature systems show the effectiveness on the proposed sensor.

All the innovative approaches proposed in this thesis have been published in international conferences and journals.

# SUMMARY

# BIOMETRIC SYSTEMS: STATE-OF-THE-ART

Behavioural and physical human traits are unique and unchangeable, so they can be used to distinguish and to recognize a user from others.

Biometric systems can be classified as recognition systems or classification systems. Depending on the application context, a biometric recognition system may be used as authentication or identification system. An authentication system checks the person identity by comparing the captured biometric characteristic with his/her own biometric template enrolled in the system. It conducts a one-to-one comparison to determine whether the identity claimed by the individual is true. An identification system recognizes the subject by searching the entire template database for a match. It conducts one-to-many comparisons and establishes person identity or fails if he/she is not enrolled in the system database, without the subject having to claim an identity.

In real-time recognition systems it is required high response speed, so classification systems play a key role. One of the main goals of such systems is to reduce the item search time within the database without affecting the accuracy rate. The identification process performed in a database divided in classes is faster, since the number of necessary comparisons can be reduced by searching the fingerprint only in the same class of the database.

# Chapter 1: Biometric Systems

## 1.1 Introduction

Biometrics, from the Greek 'bios' (life) and 'metros' (measure), is the scientific analysis and study of human recognition techniques, referring to his/her physical and behavioral traits [1]. Biometric systems provide a more secure paradigm than traditional approaches for user recognition, since human biological traits are unique and can be used to distinguish a user from the others. The traditional authentication systems based on username and password are not able to guarantee a suitable protection level, since whoever can illegitimately know or reproduce the secret user password and ID.

## 1.2 Motivations

Secure systems allow user access only if the recognition phase of the user digital identity is correctly performed [2]. Such phase usually performs one of the following three approaches:

- ***it checks something that user knows***, for example if the user knows his/her information account, generally represented by a public ID and a secret password. Unfortunately, the probability that an impostor can know the password is high. This approach is called knowledge-based, because it uses information that only the user knows;
- ***it checks something that user has***, for example if the user possesses a predetermined token (generally a magnetic badge or a smartcard). The system allows user access without asking other additional information. Unfortunately, the probability that an impostor can steal the token is high. This approach is called token-based, because it uses information that the user has;
- ***it checks something that user is,*** if the system compares user biological traits with stored values, known as template. Unlike passwords or badge, user biometric information is unique; therefore the biometric identity has the advantage to guarantee that only the authorized users have access to the system.

Most common recognition systems use the first and the second approach. However, these two approaches are not able to guarantee a suitable protection level. In additions, they require the user remembers or carries with him "something" containing the necessary information for the recognition. In the third approach, instead, all information belong to the user: physical and behavioral traits (such as fingerprints, face, iris, retina and so on) are the core of biometric systems. Biometric characteristics cannot be stolen; therefore, biometric systems are not easily violable.

## 1.3 Biometric Traits

Figure 1.1 shows a classification of the most commonly used biometric traits.

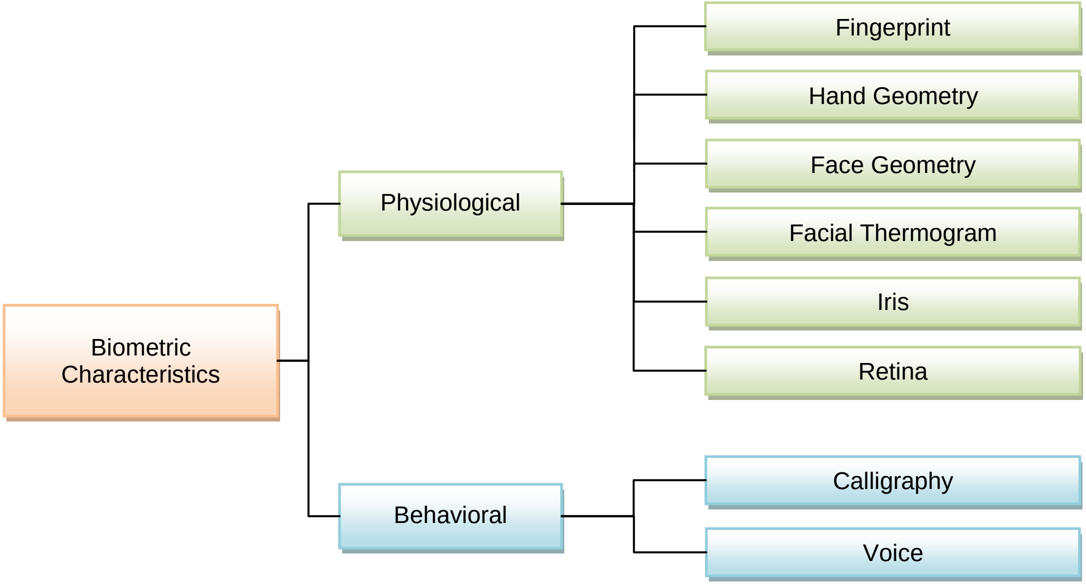


**Figure 1.1.** Classification of the most commonly used biometrics traits.

To be used in biometric systems, such traits must have the following properties [3]:

- ***Universality***: they must be present in each person;
- ***Distinctiveness***: they must be distinguishable;
- ***Permanence***: they must not change during the life;
- ***Acceptability***: the technology used to acquire them must be user-friendly and not intrusive.

The following Table 1.1 shows the properties of the most commonly used biometric traits.

**Table 1.1.** Properties of the most commonly used biometric traits.

| Biometric Technology | Universality | Distinctiveness | Permanence | Acceptability |
|---|---|---|---|---|
| Fingerprint | Medium | High | High | Medium |
| Hand Geometry | Medium | Medium | Medium | Medium |
| Face Geometry | High | Low | Medium | High |
| Facial Thermogram | High | Low | Medium | High |
| Iris | High | High | High | Low |
| Retina | High | High | Medium | Low |
| Calligraphy | Low | Low | Low | High |
| Voice | Medium | Low | Low | High |

## 1.3.1 Physiological Traits

### 1.3.1.1 Fingerprint

Fingerprints (Figure 1.2) are unique (also in identical twins) and they not change during the life. However, the technology using these traits has some limits: wet and dry skin, cuts, scars, image quality and so on can compromise the systems performances. Such technology is the most commonly used but its implementation requires high computational costs and a lot of processing resources [4, 5].

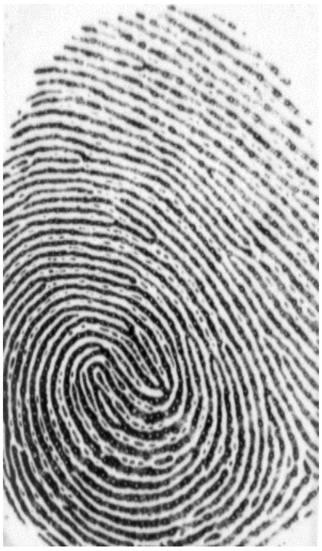


**Figure 1.2.** Fingerprint image.

### 1.3.1.2 Hand Geometry

Hand geometry recognition systems [6] use the hand form and dimensions as distinctive traits (Figure 1.3). They have many advantages than fingerprints systems (they require less space to store the template, are less expensive, meet a less psychological resistance, and so on) and few disadvantages (usually the user do not want to put its palm where many other have put theirs, the performances depend on the weather conditions and on the hand cleanness, they have a big dimension (so they can’t be implemented in portable devices), and so on).

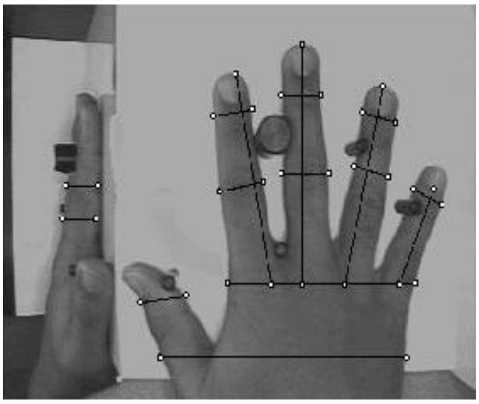


**Figure 1.3.** Hand geometry.

### 1.3.1.3 Face Geometry

Face geometry recognition systems [7] use the distance among facial attributes and the face shape as distinctive traits (Figure 1.4). This technology is not expensive and is less intrusive having a good impact on the user. However, they are very susceptible to illumination variations, face position and expressions; their performances decrease when the dimension of the database increases; and finally twins are hardly distinguishable.

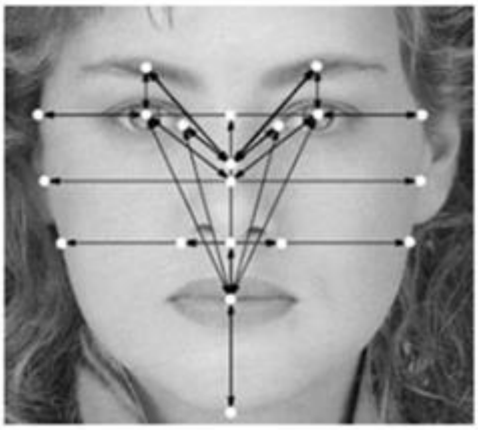


**Figure 1.4.** Face geometry.

#### 1.3.1.4 Facial Thermogram

Facial thermogram recognition systems [8] use the blood vases and the temperature of many face points as distinctive traits (Figure 1.5). They are susceptible to the user health and emotional state.

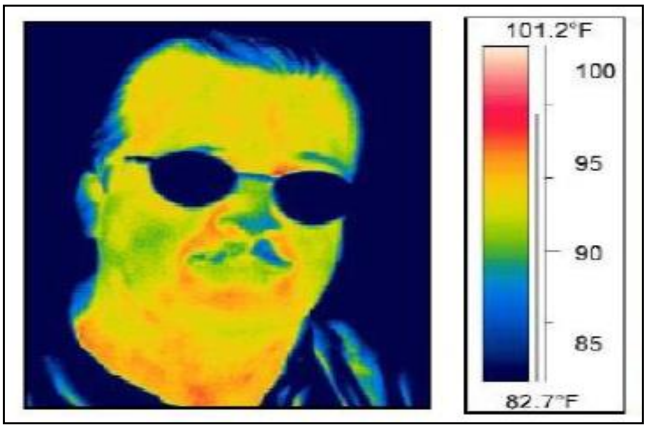


**Figure 1.5.** Facial thermogram.

#### 1.3.1.5 Iris

Iris is one of the most discriminating biometric traits (also identical twins have different iris). It is less susceptible to damage than other parts of the body, and the relevant template requires only few bytes [9]. Iris recognition systems use special cameras not needing the contact with the user eye. They work fine even if the user is carrying glasses, and utilize the following features as distinctive traits: nucleus, collarets, valleys and strike off channels (Figure 1.6).

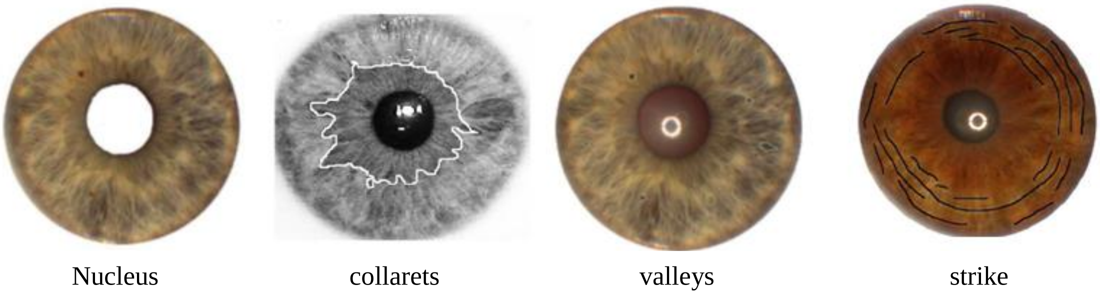


**Figure 1.6.** Iris features.

#### 1.3.1.6 Retina

Retina recognition systems [10] uses the veins conformation under the retina surface of the eye as distinctive traits (Figure 1.7). In the acquisition task, they send a beam of low intensity light inside the user ocular bulb. This approach is intrusive since the user has to be near the scanner and has to focus a specific point. In addition, the retina veins distribution may change during the life.

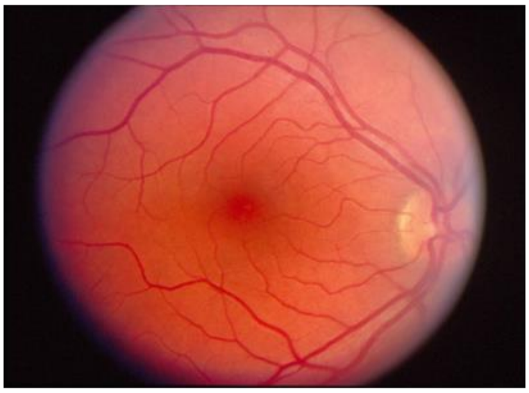


**Figure 1.7.** Human eye retinal blood vessel distribution.

## 1.3.2 Behavioral Traits

### 1.3.2.1 Calligraphy

Calligraphy can be used for personal recognition since every user has a distinctive style to write. However, two writings of the same user are never perfectly identical. Therefore, this technology has a medium reliability and can be used only on small target of population.

Two different approaches for calligraphy recognition are static and dynamic [11]. The latter uses also acceleration, speed and pressure of writing for improving the accuracy.

### 1.3.2.2 Voice

The voice recognition (Figure 1.8) is the preferred method by the users since it is not intrusive and can be used through the telephone lines. However, it is least accurate technology [12]: environmental noises can affect the recognition, twins and the brothers are hardly distinguishable, etc. It can be text-dependent (user says a predetermined sentence) or text-independent (user simply says something). The second case is less accurate.

**Figure 1.8.** Voice signal representation.

## 1.4 Recognition Systems

### 1.4.1 Operation Modes

Depending on the application context, a biometric recognition system may be used as authentication or identification system. An authentication system checks the person identity by comparing the captured biometric characteristic with his/her biometric template enrolled in the system. It conducts a one-to-one comparison to determine whether the identity claimed by the individual is true. An identification system recognizes the subject by searching the entire template database for a match. It conducts one-to-many comparisons and establishes person identity or fails if he/she is not enrolled in the system database, without the subject having to claim an identity. Two different phases are performed:

- ***Enrolment phase***: the system stores the user biometric identity (and the user ID only in authentication systems) in the database. Three specific tasks are performed: biometric trait acquisition, digital biometric features extraction and template storing in the database (Figure 1.9);
- ***Verification phase***: the system compares the acquired biometric trait with the template previously stored in the enrolment phase.

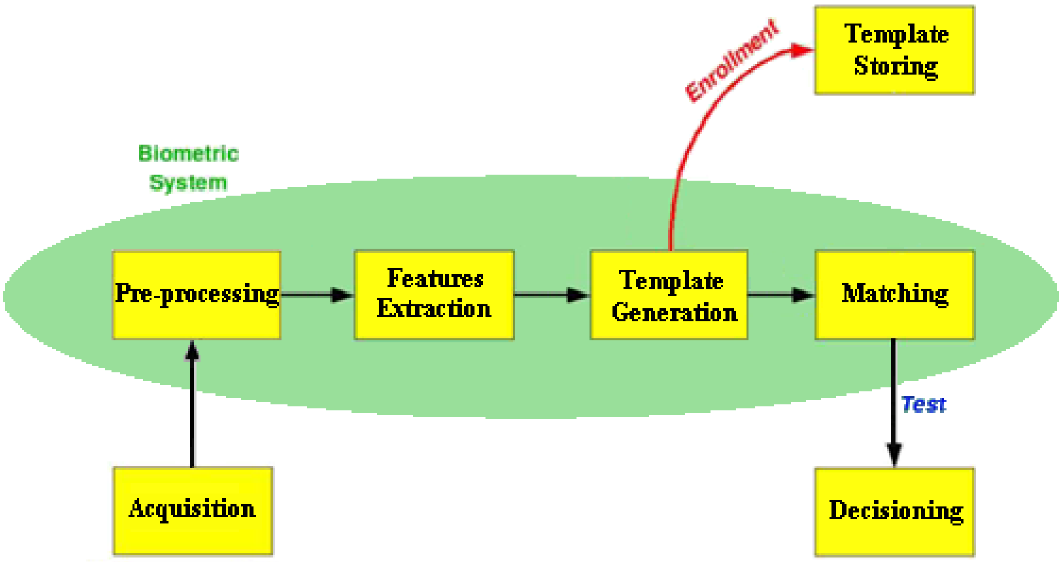


**Figure 1.9.** Biometric system processing tasks.

### 1.4.2 Performance Indexes

Biometric systems can be evaluated considering two aspects: the database dimension and the performance (speed and accuracy about user recognition). The answer time is fundamental in real-time applications; while the recognition accuracy determines the system security.

The recognition performance is evaluated using the following two correlated indexes (Figure 1.9):

- FAR (False Acceptance Rate), percentage of approved impostors;
- FRR (False Rejection Rate), percentage of refused registered user.

Besides, to evaluate the global percentage of error is used the EER (Equal Error Rate) index, defined as the percentage when the FAR and FRR are equal (Figure 1.10), or the ROC (Receiver Operating Characteristic) index, defined as FAR versus FRR.

To establish the closeness of a comparison the system uses a metric: only if the value associated to the measured biometric trait overcomes a pre-defined threshold value, then the system will recognize the user identity. The threshold value is usually a trade-off between the probability of false acceptances (FAR) and the probability of false rejections (FRR).

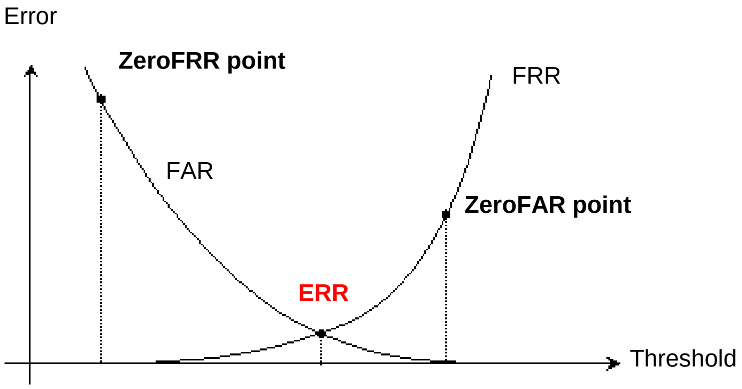


**Figure 1.10.** Typical course of FAR and FRR of a biometric system.

To choice the right biometric system usually the administrator considers many factors. Some of these factors are reported in Table 1.2.

**Table 1.2.** Comparison between the biometric recognition systems previously described.

| Biometric Systems | FRR range (%) | FAR range (%) | Cost | Template dimension (Bytes) |
|---|---|---|---|---|
| Fingerprint | 3÷7 | 0.0001÷0.001 | Medium | 300÷1200 |
| Hand Geometry | 1÷10 | 1 | Medium | <10 |
| Face Geometry | 10÷20 | 0.001÷1 | Medium | Few bytes |
| Iris | 1÷10 | ~0 | High | 512 |
| Retina | 1 | 0.01 | Very High | <1000 |
| Calligraphy | 3÷10 | 1 | Medium | 1500 |
| Voice | 10÷20 | 2÷5 | Low | 1500 |

### 1.4.3 Security Issues

The traditional authentication systems based on username and password are not able to guarantee a suitable protection level. Unlike passwords, instead, user biometric information is unique and unchangeable; therefore the biometric identity has the advantage to guarantee that only the authorized users have access to the system available resources and services. However, traditional biometric approaches involves interactions among a large number of entities: passive access points for user biometric trait acquisition, networked databases for user biometric identity storing, and trusted servers running the user recognition systems. So, such traditional systems usually undergo several types of attacks as [13,14]:

- *Communication Attack* (attacking the channel between the server and the database);
- *Replay Attack* (resubmitting the user biometric trait, or overriding the extracted features and/or the final decision);
- *Database Attack* (tampering with the stored user biometric identity);
- *Software Attack* (corrupting the matcher).

Embedded architectures, instead, provide a more secure and flexible infrastructure, since all elaboration steps are performed on board, so biometric identities are securely managed and stored inside the system without any data leaking out. In addition, their hardware implementation overcomes the limits of performance and response time.

# Chapter 2: Fingerprint-Based Systems

## 2.1 Introduction

Fingerprint is composed of ridges and valleys which form unique geometric patterns in the skin [15]. With more details, ridge lines are characterized by minutiae (Figure 2.1 a)) such as end-points (the point where the ridge line terminates) and bifurcations (point where the ridge line intersects another ridge line). Usually, fingerprint area contains about 30 to 60 minutiae points depending on the finger size and the sensor area dimension. Fingerprints are also characterized by singularity points, called Delta and Core, in which the ridge line flow is irregular. With more details, the Core point is the center of a circular edge pattern, and the Delta point is the center of a triangular edge pattern (Figure 2.1 b)).

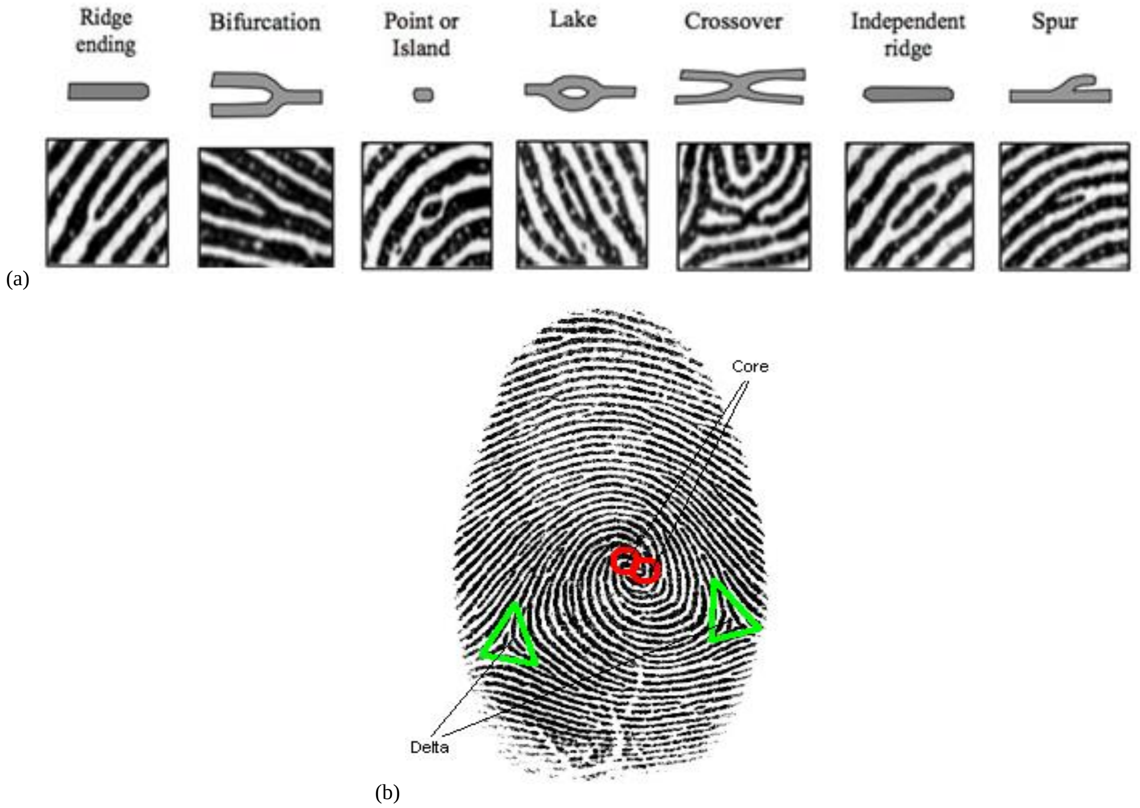


**Figure 2.1. a)** fingerprint minutiae; **b)** fingerprint singularity points.

## 2.2 Fingerprint Recognition

Fingerprint recognition is a well-researched problem, and automatic fingerprint authentication /verification techniques have been successfully adapted to both civilian and forensic applications for many years. Although, automated systems, software and hardware implementation, are usually adopted in commercial and security applications for access and denial operations, the relative technology can also be used in other emerging areas of interest. One of the most important areas of considerable utility to law enforcement agencies is concerning to partial fingerprints identification. Since the fingerprint templates constitute the largest data in the biometric field, considering partial fingerprint samples, the amount of data in the database shall be decreased and this will consequently lead to a faster processing for fingerprint identification [16]. Matching the small parts (partial) fingerprint to the stored images in database usually has different problems. The partial fingerprints (for example, obtained from a crime scene) are normally small, noisy and have the following characteristics:

- a less minutiae number with respect to complete fingerprint image;
- high probability of loss of singularity points;
- unspecified roto-translation problems due to uncontrolled acquisition environments;
- distortions introduced by human skin such as elasticity;
- difficult to determine correspondence of the obtained partial fingerprint to one of the fingers.

In literature many fingerprint-based recognition systems have been proposed and implemented [17, 18]. According to the used technique, they can be divided into systems using correlation-based approaches, and systems using minutiae-based approaches. Since the minutia-based fingerprint representation is an ANSI-NIST standard [19], all approaches and techniques implemented following these recognition guidelines have the advantage of being directly applicable to existing systems and databases.

### 2.2.1 Correlation-Based Approaches

These approaches compute the correlation between pixels of two fingerprint images (previously superimposed) for different alignments (e.g., various displacements and rotations), considering the global pattern of ridges and valleys. However, they require several points to register the fingerprint image with roto-translation operations [20].

Let f and g be the two fingerprint images (the template and the input fingerprint, respectively) [21]. The Sum of Squared Differences (SSD) measures their diversity and is calculated by the Equation (2.1):

$$\mathrm{SSD}(\mathrm{f},\mathrm{g}) = \|\mathrm{f}-\mathrm{g}\|^2 = (\mathrm{f}-\mathrm{g})^{\mathrm{T}}(\mathrm{f}-\mathrm{g}) = \|\mathrm{f}\|^2 + \|\mathrm{g}\|^2 - 2\mathrm{f}^{\mathrm{T}}\mathrm{g} = \|\mathrm{f}\|^2 + \|\mathrm{g}\|^2 - 2\mathrm{CC} \qquad (2.1)$$

If $||\mathrm{f}||^2$ and $||\mathrm{g}||^2$ are constant and the CC (Cross-Correlation) between f and g is maximized, then the diversity between the two images is minimized. Therefore, CC can be considered as a measure of the images similarity, however in Equation (2.1) also displacement and rotation are considered. If $\mathrm{g}^{(\Delta\mathrm{x},\Delta\mathrm{y},\theta)}$ is the roto-translation of the input image g by an angle $\theta$ around the origin (usually the image center) and by $(\Delta\mathrm{x},\Delta\mathrm{y})$ pixels; then the similarity between f and g can be measured through the Equation (2.2):

$$\mathrm{S}(\mathrm{f},\mathrm{g}) = \max_{\Delta\mathrm{x},\Delta\mathrm{y},\theta} \mathrm{CC}(\mathrm{f},\mathrm{g}^{(\Delta\mathrm{x},\Delta\mathrm{y},\theta)}) \qquad (2.2)$$

However, it is computationally very expensive and rarely leads to acceptable results mainly due to skin condition, impressing pressure and non-linear distortions. In these scenarios, it is useful adopt more sophisticated correlation measures, such as the normalized cross-correlation or the zero-mean normalized cross-correlation.

#### 2.2.1.1 Phase Only Correlation (POC) Approach

This approach considers only the correlation among phase components [22]. It is immune to image displacements and rotations, to changes in light exposure, and to noisy acquisitions.

Let $f(n_1,n_2)$ and $g(n_1,n_2)$ be two $N_1 \times N_2$ images, where:

$$n_1 = -M_1,\ldots,M_1 \quad (M_1 > 0)$$

$$n_2 = -M_2,\ldots,M_2 \quad (M_2 > 0)$$

$$N_1 = 2M_1 + 1$$

$$N_2 = 2M_2 + 1$$

The relevant Discrete Fourier Transforms are defined by the following equations, respectively:

$$F(k_1,k_2) = \sum f(n_1,n_2) W_{N_1}^{k_1 n_1} W_{N_2}^{k_2 n_2} = A_F(k_1,k_2) e^{j\theta_F(k_1,k_2)} \qquad (2.3)$$

$$G(k_1,k_2)=\sum g(n_1,n_2)W_{N_1}^{k_1n_1}W_{N_2}^{k_2n_2}=A_G(k_1,k_2)e^{j\theta_G(k_1,k_2)} \tag{2.4}$$

where:

$A_F(k_1,k_2)$ and $A_G(k_1,k_2)$ are the module components

$e^{j\theta_F(k_1,k_2)}$ and $e^{j\theta_G(k_1,k_2)}$ are the phase components

$$k_1=-M_1,...,M_1$$

$$k_2=-M_2,...,M_2$$

$$W_{N_1}=e^{-j\frac{2\pi}{N_1}}$$

$$W_{N_2}=e^{-j\frac{2\pi}{N_2}}$$

The Cross-Spectrum $R_{FG}(k_1,k_2)$ between $F(k_1,k_2)$ and $G(k_1,k_2)$ is calculated by the Equation (2.5):

$$R_{FG}(k_1,k_2)=F(k_1,k_2)\overline{G(k_1,k_2)}=A_F(k_1,k_2)A_G(k_1,k_2)e^{j\theta(k_1,k_2)} \tag{2.5}$$

where:

$\overline{G(k_1,k_2)}$ is the complex conjugate of $G(k_1,k_2)$

$\theta(k_1,k_2)\ =\ \theta_F(k_1,k_2)-\theta_G(k_1,k_2)$ is the phase difference

The ordinary correlation function $r_{fg}(n_1,n_2)$ is calculated by the Inverse Discrete Fourier Transform (IDFT) of the Equation (2.5).

The normalized cross-spectrum is calculated by the Equation (2.6):

$$\hat{R}_{FG}(k_1,k_2)=\frac{F(k_1,k_2)\overline{G(k_1,k_2)}}{\left|F(k_1,k_2)\overline{G(k_1,k_2)}\right|}=e^{j\theta(k_1,k_2)} \tag{2.6}$$

The POC function $\hat{r}_{fg}(n_1,n_2)$ is the normalized spectrum IDFT, and it is calculated by the following equation:

$$\hat{r}_{fg}(n_1,n_2)=\frac{1}{N_1N_2}\sum\hat{R}_{FG}(k_1,k_2)W_{N_1}^{-k_1n_1}W_{N_2}^{-k_2n_2} \tag{2.7}$$

It has the following characteristics:

- for two identical aligned images, it shows only one peak (Figure 2.2);
- for two identical translated images, it shows only one peak and its position has the same images translation (Figure 2.3);
- for different images, it shows many peaks (Figure 2.4);
- for noisy acquisitions, the peak value reducing is not considerably high.

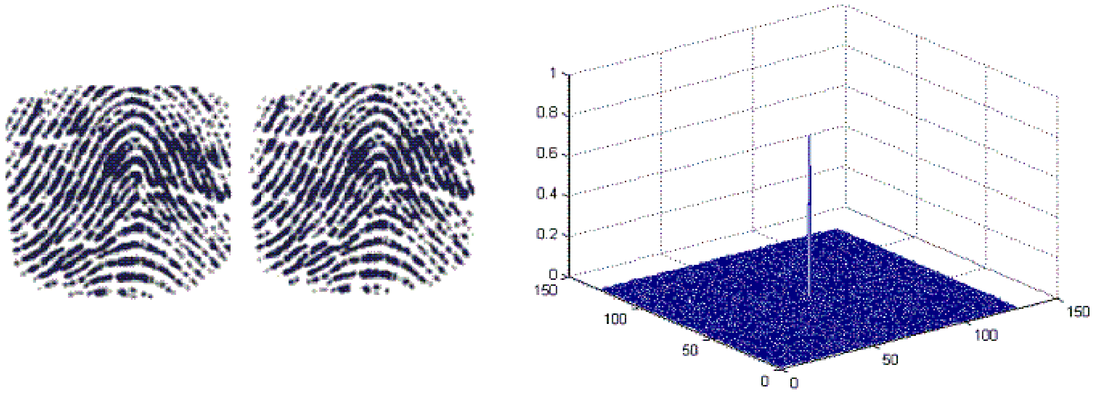


**Figure 2.2.** Matching of two identical aligned images: only one peak with an high value is present.

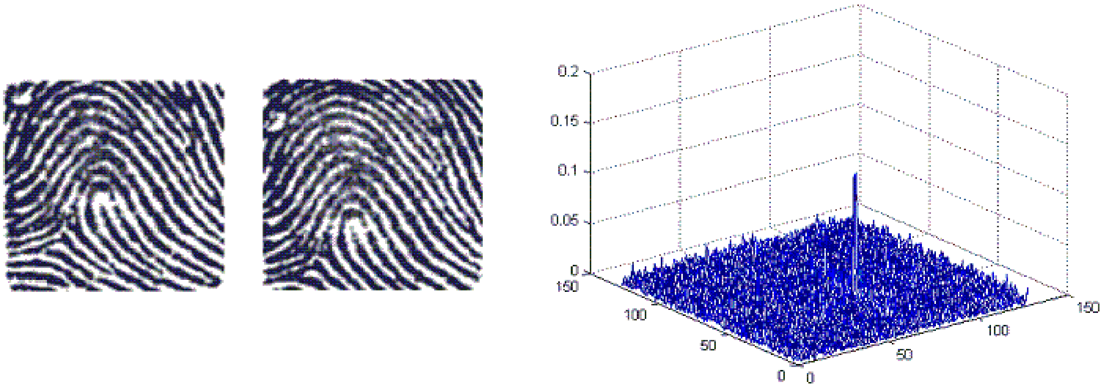


**Figure 2.3.** Matching of two identical translated images: the peak position has the same images translation.

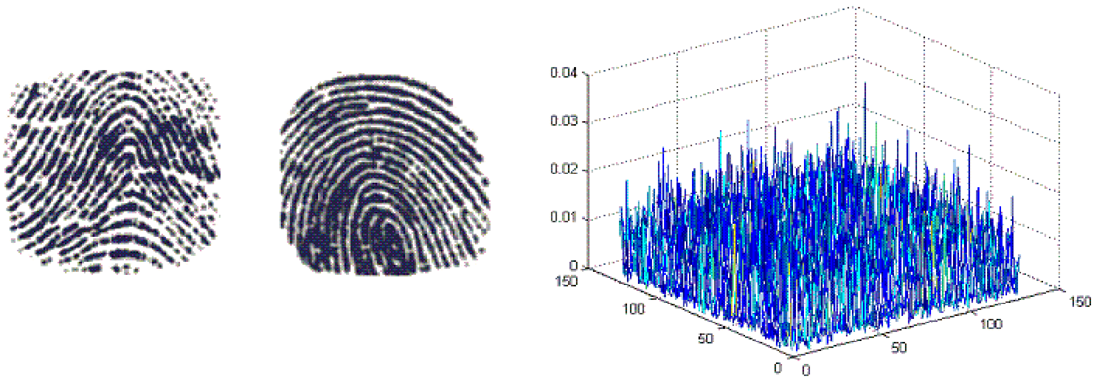


**Figure 2.4.** Matching of two different images: many peaks are present.

### 3.1.1 Minutiae-Based Approaches

#### 2.2.1.2 Theoretical Remarks

Let $T = \{m_1, m_2, ...m_m\}$ and $I = \{m_1^{'}, m_2^{'}, ..., m_n^{'}\}$ be the biometric signature of the template and of the input fingerprint, respectively. Their elements are the fingerprint minutiae, each described by several attributes. However, the most commonly used minutiae matching/extraction algorithms consider each minutia as a quadruplet including the spatial coordinates, the orientations (Figure 2.5) and the type (i.e., termination or bifurcation).

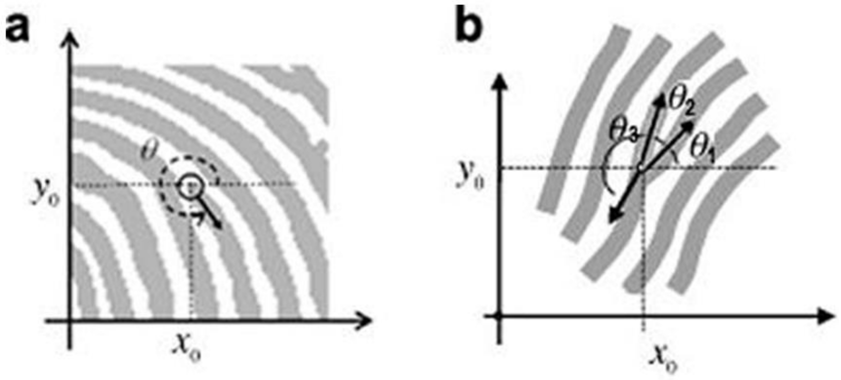


**Figure 2.5. a)** minutia spatial coordinates; **b)** minutia orientations.

The type attribute does not play a key role for the algorithm description, so for T and I, respectively, each minutia is described as follow:

$$m_i = \{x_i, y_i, \theta_i\}, i = 1...m$$

$$m_j^{'} = \{x_j^{'}, y_j^{'}, \theta_j^{'}\}, j = 1...n$$

Usually, a matching algorithm calculate a spatial distance, $sd(.)$, and a direction difference, $dd(.)$, between two minutiae of two different biometric signatures. Then, it check if $sd(.)$ is smaller than a given tolerance $r_0$ and if $dd(.)$ between them is smaller than an angular tolerance $\theta_0$:

$$sd(m_j^{'}, m_i) = \sqrt{(x_j^{'} - x_i)^2 + (y_j^{'} - y_i)^2} \leq r_0 \qquad (2.8)$$

$$dd(m_j^{'}, m_i) = \min(\left|\theta_j^{'} - \theta_i\right|, 2\pi - \left|\theta_j^{'} - \theta_i\right|) \leq \theta_0 \qquad (2.9)$$

The $r_0$ and $\theta_0$ tolerances compensate the inevitable errors in fingerprint acquisition and features extraction phases.

Before the matching task, an aligning task is performed. It requires the following geometrical transformations:

- displacement (in x and y);
- rotation (θ);
- scale, if the two fingerprints have different resolutions;
- other distortion-tolerant geometrical transformations could be useful to match minutiae in case of one or both of fingerprints are affected by severe distortions.

According to a given geometrical transformation, $map(.)$ is the function that maps a minutia $m_j^{'}$ into $m^{''}$. For example:

$$map_{\Delta x,\Delta y,\theta}(m_j^{'} = \{x_j^{'}, y_j^{'}, \theta_j^{'}\}) = m_j^{''} = \{x_j^{''}, y_j^{''}, \theta_j^{'} + \theta\}$$

where

$$\begin{bmatrix} x_j^{''} \\ y_j^{''} \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x_j^{'} \\ y_j^{'} \end{bmatrix} + \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \tag{2.10}$$

is a roto-translation considering a displacement of $(\Delta x, \Delta y)$ and a counter clockwise rotation $\theta$ around the origin.

Let $mm(.)$ be the function described in Equation (2.11):

$$mm(m_j^{''}, m_i) = \begin{cases} 1 & sd(m_j^{''}, m_i) \le r_0 \; and \; dd(m_j^{''}, m_i) \le \theta_0 \\ 0 & otherwise \end{cases} \tag{2.11}$$

that returns 1 only when the minutiae $m^{''}$ and $m'$ match according to Equations (2.8) and (2.9). Therefore, the matching problem can be formulated as in Equation (2.12):

$$\max_{\Delta x,\Delta y,\theta,P} imize \sum_{i=1}^{m} mm(map_{\Delta x,\Delta y,\theta}(m_{P(i)}^{'}), m_i) \tag{2.12}$$

where $P(i)$ is an unknown function that determines the pairing between minutiae of I and T. With more details, each minutia can have either exactly one mate in the other fingerprint or have no mate at all, therefore:

1. $P(i) = j$ if the mate of the $m_i$ in T is the minutia $m'_j$ in I;
2. $P(i) = null$ if the minutia $m_i$ in T has no mate in I;
3. $\forall i = 1...m, P(i) \neq j$ if a minutia $m'_j$ in I has no mate in T;
4. $\forall i = 1..m, k = 1..m, i \neq k \Rightarrow P(i) = P(k) = null$ (if each minutia in I is associated with only one minutia in T).

Note that $P(i) = j$ is not necessarily linked to the $m'_j$ and $m_i$ match using the Equations (2.11) and (2.12), but it only means that they are the most likely pair under the current transformation.

#### 2.2.1.3 Processing Steps

The minutiae-based approach is the most commonly used for fingerprint matching [21]. However it is a complex and intensive task. Usually, to increase the system speed and accuracy a fingerprint image pre-processing task, a dedicated extraction algorithm [15, 17], and a post processing task are performed.

The most commonly used pre-processing task steps are:

- **Normalization**, used to force the fingerprint gray levels to an average value within a desired variance;
- **Segmentation** [23, 24], used to erase the fingerprint background;
- **Directional image extraction**. Every element in the directional image represents the local orientation of the fingerprint ridges in the original gray-scale image. Usually the directional image is extracted in three steps [25]: extraction of the direction for each pixel; processing of the previous step output assembling the pixels in blocks; and computing of the predominant direction for each block;
- **Enhancement**, used to improve the quality image. Many fingerprint image enhancement filters have been proposed in literature, such as Gabor filter [26], Median filter [27], Morphological filter [28], and so on. The most used filter in fingerprint recognition systems is the Gabor filter; however, its implementation requires a lot of computational resources and a not negligible execution time;

- **Binarization**, used to obtain an image where pixels assume a binary value: white as background and black as foreground. Generally, it uses static thresholds to determine the binary pixels;
- **Thinning** [29], used to reduce the ridge thickness to the unitary value.

The minutiae extraction is performed after the pre-processing task: for each minutia, it determines the type (i.e. ending or bifurcation point), the spatial coordinates and the orientations. Then, a post processing task is performed in order to reduce the potential false minutiae number [72]. It is usually based on the Euclidean distance between minutiae pair. Finally, the biometric signature is created and the matching is performed (Figure 2.6). It calculates the correspondence between the obtained biometric signature and one or more stored templates, assigning a matching score to each template pair comparison.

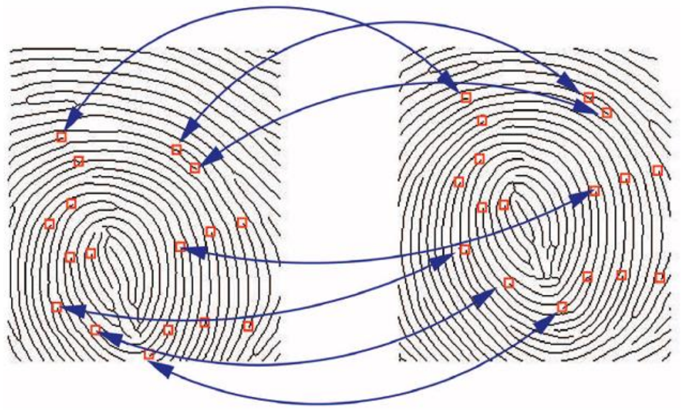


**Figure 2.6.** Example of a minutiae-based matching: each arrow is linked to a pair of matched minutiae.

### 3.1.2 Singularity Points Approaches

Typical fingerprint recognition systems use singularity points for classification tasks. On the other hand, the common and available optical and photoelectric sensors give high quality fingerprint images with well-defined Core and Delta points, if they are present in the complete fingerprint.

In [75] authors propose a fingerprint matching based on singularity points position, orientation, and relative distance detection. As result, fingerprint matching involves the comparison between few features leading to a very fast system with recognition rates comparable to the standard minutiae based recognition systems. Their approach can be divided in two main steps: singular points extraction and matching phase. The singularity points extraction is performed using three sequential steps: directional image extraction, Poincarè indexes computation and core and delta extraction. The matching phase is performed using singularity regions analysis and topological region analysis. The most interesting technique of this approach is related with the two analysis in the matching phase. With more details, the singularity regions, concerned on core and delta points, have the following characteristic: the matching algorithm can be performed only if there is almost one singularity point for region of the same type. In the follows, a brief description of this techniques outlined.

In the singularity regions analysis, the algorithm receives as input two fingerprint images and the type of extracted singularity points. The comparison between the same type of singularity points (Core/Core and Delta/Delta) is performed analyzing the directional image. The considered part of image is a round neighbor centered on the singularity point with a dimension of 45x45 pixel. To perform the comparison between two singularity regions, the sum of the modules of difference among corresponding angles, using kernel of 5x5 pixel, is computed. The computed value represents the error distance (Equation (2.13)) This distance is used as input in an exponential function, (Equation (2.14)), to calculate the similarity index (Equation (2.15)). In the following equations the values K1, K2 and K3 are constant experimentally fixed.

$$m = \sum_{i=1}^{5} \sum_{j=1}^{5} \left| d_{Test}(i,j) - d_{Template}(i,j) \right| \tag{2.13}$$

$$error = K_1 \times (e^{(K_2 \times m)} - 1) \tag{2.14}$$

$$similarity = K_3 - error \tag{2.15}$$

In the topologic relations analysis, if the two fingerprint images have at least two singularity points then for each image the pair with smaller distance is chosen to perform the next comparison based on global characteristics (Figure 2.7). The smaller pairs is chosen to decrease the distortion problem and to increase the probability to extract real singularity points. After pairs individualization step, the type of these points is checked. For each pair of same points the Euclidean distance is calculated and this value is used as parameter. This parameter is the input of the exponential function, used for the singularity regions analysis (Equation (2.14)). If the distances are very near, the directional fields, of the four singularity points, will be checked:

- if the two extracted pairs are composed by one Core and one Delta then a comparison between the two Core and the two Delta is performed respectively;
- if the two extracted pairs are composed only by one singularity point (Core or Delta) then four comparison will be executed: the 1st point of 1st pair with the 1st point of 2nd pair; the 2nd point of 1st pair with 2nd point of 2nd pair; and vice versa.

The pair with the lower error is chosen. The three error measures obtained are weighted and added to individualize the similarity index: the error measure related to the Core, the error measure related to the Delta and the error measure related to the topological relations.

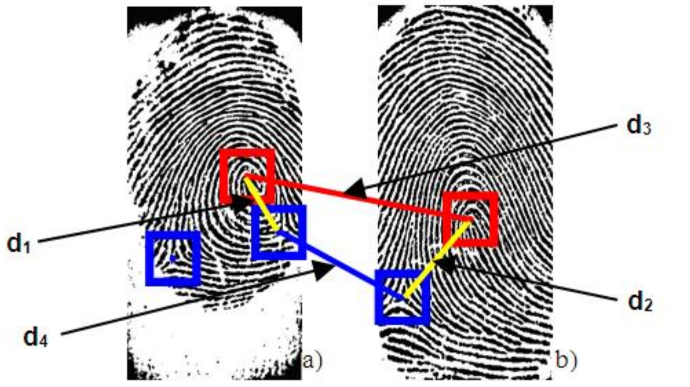


**Figure 2.7. a)** fingerprint template image**, b)** test fingerprint image.

## 2.3 Fingerprint Classification

User identification requires a comparison of his/her template with all the ones stored in the database. When the database is very large, as in forensic scenarios, the identification time could be not acceptable. This time can be strongly reduced by simply dividing the fingerprint database into several classes, since the matching is performed only with fingerprints of the same class.

Fingerprint classification aims to assign a fingerprint to a class in a consistent and reliable way. It is generally based on global features, structural information, neural networks, fuzzy-neural networks, probabilistic model, artificial intelligence techniques and so on. Unfortunately, singularity points are not always present in fingerprint images, for example due to image acquisitions not correctly performed such as in partial fingerprint; in this case, may be useful the approach proposed in [30] using pseudo-singularity points.

### 2.3.1 Probabilistic Approaches

Jung and Lee in [31] use a probabilistic approach (Markov model) based on the ridge characteristics of fingerprint classes on FVC2000 DB1 and FVC2002 DB1 databases.

Senior in [32] describes a novel method of classification based on hidden Markov models and decision trees to recognize the ridge structure on NIST-4 database.

### 2.3.2 Singularity Points Approaches

This approach is commonly used by both automatic devices and human experts for manual classification [21].

The National Institute Standard Technology (NIST) has classified human fingerprints in five classes (Figure 2.6 a)), each characterized by *n* Core and *m* Delta, where $n = 0,..,2$ and $m = 0,..,2$ [19]. The Whorl class can be further divided into 4 sub-classes (Figure 2.6 b)).

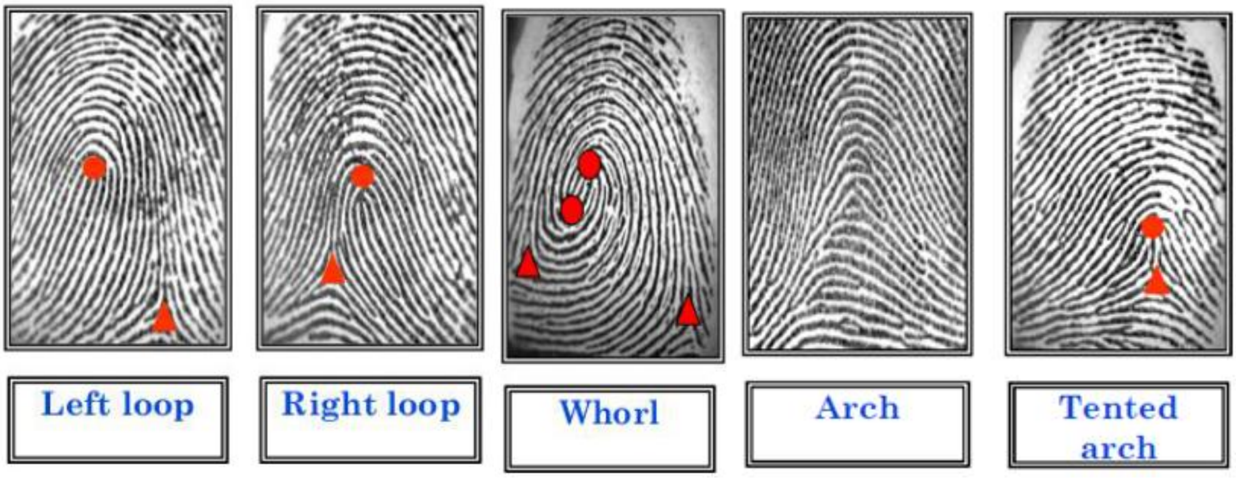


**Figure 2.6. a)** The five NIST fingerprint standard classes.

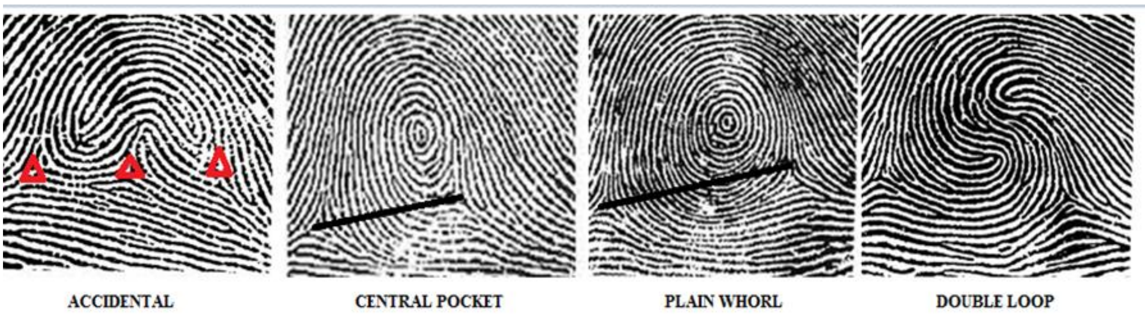


**Figure 2.6. b)** Whorl classification in 4 sub-classes.

Generally, before the singularity points extraction the following steps are performed: directional map extraction, and computation of the Poincarè index [33].

Authors in [34] reduce the image distortion and contrast, before computing the fingerprint directional image on NIST Database. From this image, successively, they extract singular points and classify the fingerprint using topological and numerical considerations about these points

Mohamed and Nyongesa in [35] present a classification scheme based on the encoding of singular points (Core and Delta) together with their relative positions and directions. The image analysis is carried out in four stages: segmentation, directional image estimation, singular point extraction and feature encoding. Successively, a fuzzy-neural network is used to implement the classification of input feature codes obtaining an average classification accuracy of 98.5% on NIST-4 database.

### 2.3.3 Syntactic Approaches

These approaches describe fingerprint patterns using terminal symbols and production rules. Then, they define a grammar or a string-matching technique for each class. Finally, by a parsing process they classify each new pattern [21]. Due to the great diversity of fingerprint patterns, syntactic approaches generally require very complex grammars. Their inference requires complicated and unstable techniques, therefore, the use of syntactic methods for fingerprint classification is not commonly used.

In [36], authors propose an approach based on the analysis of ridge line flow represented by a set of connected lines (Figure 2.7). It labels the lines according to the direction changes, thus obtaining a set of strings. Ad hoc grammars is used for the fingerprint classification.

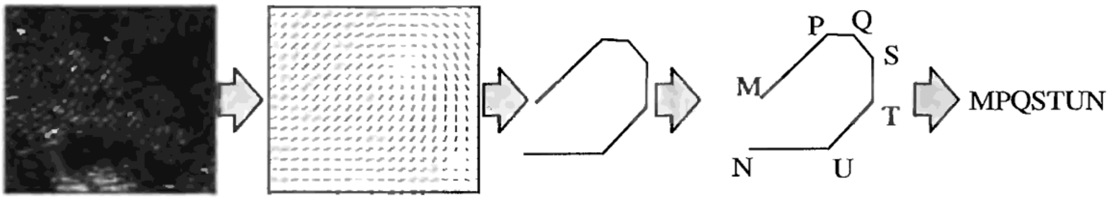


**Figure 2.7.** String-construction approach proposed in [36].

### 2.3.4 Structural Approaches

In structural approaches a relational organization of low-level features into higher-level structures is performed. It can be represented by several symbolic data structures, such as trees and graphs [37]. Usually, these approaches split the directional image into connected regions characterized by homogeneous orientations, and analyze the relations among these regions.

Maio and Maltoni in [38] compute a relational graph, summarizing the fingerprint macro-structure, from the segmentation of the directional image into regions by minimizing a cost function that takes into account the variance of the element orientations within each region (Figure 2.8). The obtained graph is compared with model graphs in order to classify the fingerprint. They don’t say which database they used.

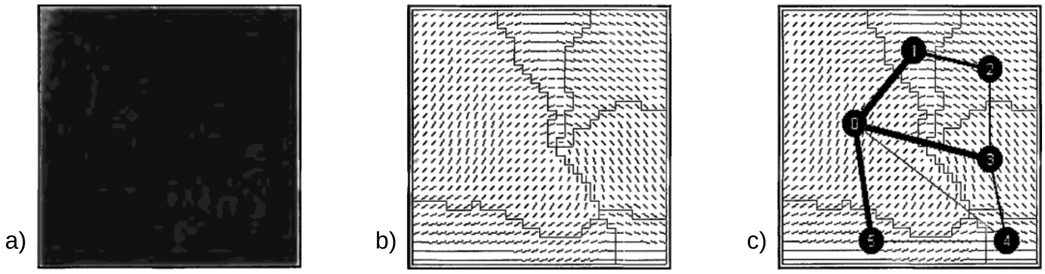


**Figure 2.8.** Approach proposed in [38]: **a)** a fingerprint image; **b)** directional image partitions; **c)** related relational graph.

### 2.3.5 Neural Network Approaches

Literature neural network approaches are generally based on multilayer perceptrons and use the orientations of directional images as inputs [39].

Authors in [40] use a neural network as a decision stage. The network is ready to perform matching process and is successfully developed to identify and classify the fingerprint using back propagation algorithm. Their methodology has been validated on a standard database of 800 images (they don’t say the database name) classified into six classes obtaining a classification rate of 80.2%.

Authors in [41] uses a pyramidal architecture constituted of several multilayer perceptrons. Each perceptron is trained to recognize fingerprints belonging to a different class.

Authors in [42] uses a Weightless Neural Network (WNN) and 30 images per class for the training phase. WNNs represent a technique for building pattern recognition systems, belonging to the memory-based architectures. They are composed by a set of discriminators and each discriminator by a set of RAMs, where each RAM corresponds to a different pattern of the input data. The used WNNs based on the WISARD (Wilkie, Stonham’s, Alexander Recognition Device) approach use as discriminators the number of the different classes. Contrary to standard multi-layer neural networks, they can be implemented in simple hardware structures and trained very rapidly. Figure 2.9 shows the discriminator and the WISARD-based WNN architectures, where the 0 or 1 values are stored in RAMs: a value of 1 corresponds to a specific feature of the training set for a specific class. The RAM output value corresponds to a partial input data, while the discriminator output value is linked to the whole input data. The Winner-Take-All (WTA) module, on the basis of the maximum between the discriminators output value, performs the input pattern classification.

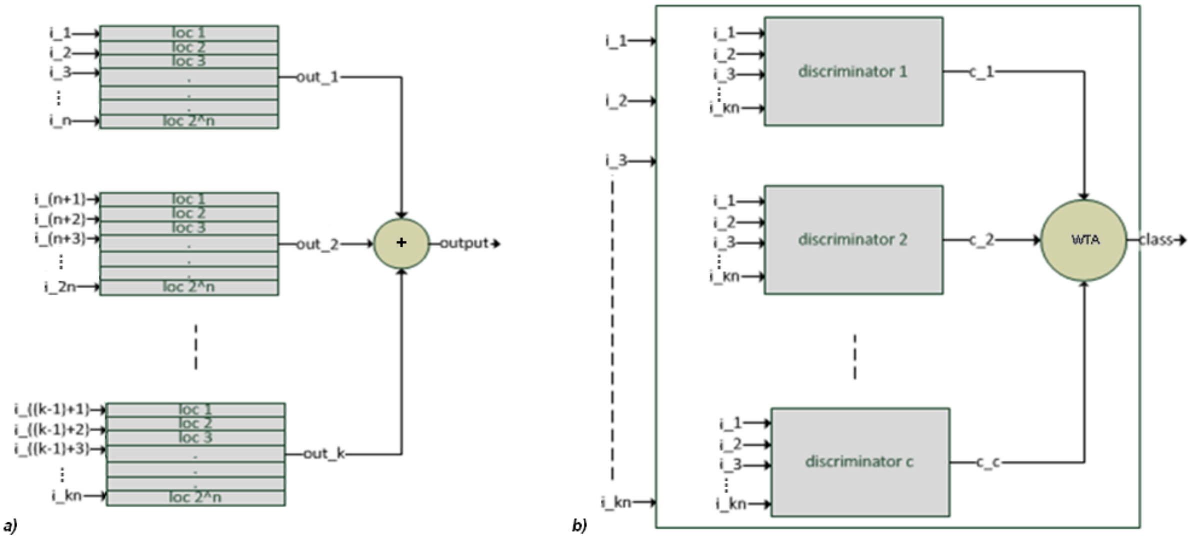


**Figure 2.9. a)** Discriminator architecture; **b)** WISARD-based WNN architecture.

In [43] authors use a feed-forward neural network (Figure 2.10 a)). Figure 2.10 b) shows the class separation obtained by their neural network.

Authors in [44] uses two disjoint neural networks. The singularity points location and a 20 x 20 orientation image are used for the training phase. The outputs of the two disjoint neural networks are then passed to a third network for the fingerprint classification.

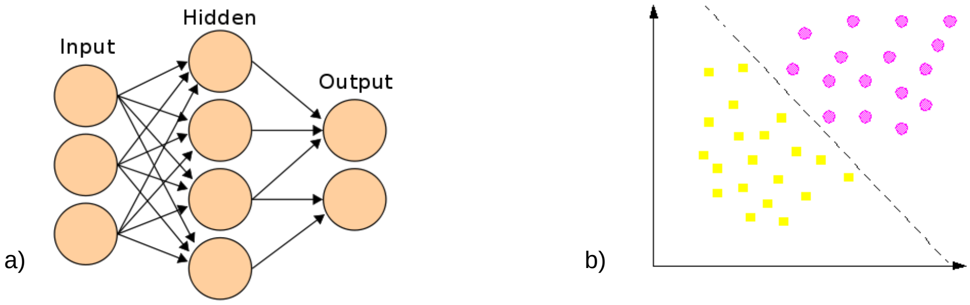


**Figure 2.10. a)** feed-forward neural network; **b)** the obtained class separation.

# THE PROPOSED INNOVATIVE TECHNIQUES AND APPROACHES

Emerging ubiquitous technologies, such as Wireless Sensor Networks (WSNs) and Internet of Things (IoT), have attracted a lot of attention and are expected to bring benefits to several application areas. However, they are changing the way people communicate and use network services and functionalities. So, it is arguably required to investigate novel design techniques and technologies for user recognition, in order to provide a secure access to systems, data and resources.

The traditional authentication systems based on username and password are not able to guarantee a suitable protection level. Unlike passwords, instead, user biometric information is unique and unchangeable; therefore the biometric identity has the advantage to guarantee that only the authorized users have access to available resources and services. So, biometric recognition systems are a valid alternative to this traditional approach. Fingerprint authentication systems are the most commonly used. They are a rapidly evolving technology in mobile devices, with a very strong potential to be widely adopted in a broad range of human scenarios. However, there are many challenges to overcome in designing completely automatic and reliable systems, especially when input data are of poor quality and contains partial information.

Traditional biometric approaches involves interactions among a large number of entities: passive access points for user biometric trait acquisition, networked databases for user biometric identity storing, and trusted servers running the user recognition systems. So, traditional systems usually undergo several types of attacks. Embedded architectures, instead, provide a more secure and flexible infrastructure, since all elaboration steps are performed on board, so biometric identities are securely managed and stored inside the system without any data leaking out.

# Chapter 3 – The Proposed Innovative Embedded Sensor

## 3.2 Introduction

The growing number of mobile users has deeply influenced scenarios such as commercial, banking, and government applications. Due the increasing security requirements, the way people access information resources, data communication and processing, is radically changing. In this field, biometric recognition systems are a good solution for the security issues. They are a rapidly evolving technology in mobile devices: for example, they are recently available in the new generation of Apple [45] and Samsung smart phones [46].

Depending on the application context, a biometric recognition system may be used as authentication or identification system. An authentication system checks the person's identity by comparing the captured biometric characteristic with his/her own biometric template enrolled in the system. It conducts a one-to-one comparison to determine whether the identity claimed by the individual is true. An identification system recognizes the subject by searching the entire template database for a match. It conducts one-to-many comparisons and establishes person's identity or fails if he/she is not enrolled in the system database, without the subject having to claim an identity. A biometric recognition system may be further classified as unimodal, when one or more instances of a single biometric trait (e.g., multiple impressions of a finger) are processed. The system is classified as multimodal, when it uses one or more instances of multiple biometric characteristics (e.g., fingerprint and face images) [21]. Multi-algorithmic systems represent a particular multimodal systems class, where the same biometric trait is processed with different algorithms [47].

To reduce the processing time in identification systems, biometric characteristics can be classified in an accurate and consistent way such that the input needs to be matched only with a database subset. Fingerprint classification, for example, can be performed using a wide variety of algorithms, almost all based on one or more of the following features: neural network [42], Gabor filter and support vector machine [48], genetic programming [49], singularity points [50], etc. Unfortunately, singular points are not always present in a fingerprint image (e.g. in the partially fingerprint image acquisition). In that case, it may be useful the approach proposed in [51], where pseudo-singularity points are detected and extracted for fingerprints classification and matching.

Fingerprint recognition is a well-researched problem, and automatic techniques could be successfully adapted in a broad range of human scenarios. However, there are many challenges to overcome in designing completely automatic and reliable systems, especially when input data are of poor quality. For example, fingerprint acquisitions not correctly performed, because of skin humidity, impressing pressure, large translation on sensor area, sensing mechanism and so on, could lead to the following issues [52]:

- Quite different ridges quality;
- Ridges and valleys pattern deformation;
- Insufficient contrast;
- Small foreground area;
- Inadequate overlapping area between different images although they are captured from the same finger.

In this thesis, a novel embedded Automatic Fingerprint Authentication System (AFAS) for mobile users is described. The goal of the proposed approach is to improve the performance of a standard embedded AFAS, in terms of used resources, execution time and working frequency, in order to enable its employment in mobile devices architectures. Starting from the work described in [53], focused only on an advanced matching technique for partial fingerprints, the novel embedded AFAS has been prototyped adding the proposed fingerprint image quality evaluation module. This module is designed to find a measure that can characterize the quality of raw fingerprint images, only using the information achieved in the acquisition step. The quality index calculates and merges six different global quality indexes based on: image contrast, ridges orientation certainty level, fingerprint's center position, impressing pressure and fingerprint size over the entire image. It is also specialized in identifying areas of poor quality. If the image overcomes the quality constraints only good areas are processed reducing the potential false minutiae number. Otherwise, if the image is rejected, the system suggests to user a set of information about the not correct acquisition step, helping him to follow correct guidelines to obtain a better image quality in the next fingerprint acquisition task (Figure 3.1).

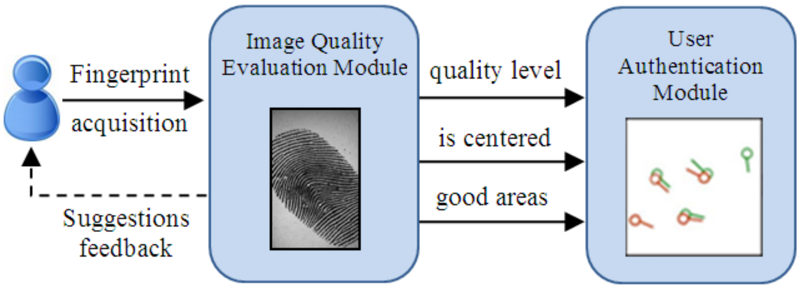


**Figure 3.1.** Image Quality Evaluation Module classifies the fingerprint image quality, identifies high quality areas. It checks if the fingerprint is centred over the image. If an image is rejected, a suggestions feedback is given, to the user for the next fingerprint acquisition tasks.

The proposed AFAS architecture has been designed for Field Programmable Gate Array (FPGA) devices using pipeline techniques and parallelisms in order to reduce the execution time. It has been prototyped on the Agility RC2000 development board, equipped with a Xilinx Virtex-II xc2v6000 FPGA [54]. To evaluate the effectiveness of the proposed embedded sensor, three tests have been conducted starting from two different free databases, chosen for their different characteristics in terms of resolution and image dimensions.

The AFAS described in [55] has been extended with the proposed fingerprint image quality evaluation module and the efficient features extractor described in [70]. Experimental trials on the FVC2002 DB2-B database [56] show that the accuracy performance has been strongly increased. Then, the matching algorithm has been replaced with the advanced technique for partial fingerprints proposed in [53]. Experimental results on the PolyU database [57] show an interesting trade-off between required hardware resources, authentication time and accuracy rate. Finally, the fingerprint image quality evaluation module has been replaced with a pre-processing module to enhance fingerprint images, a Gabor filter, and the system has been tested on the same PolyU database. The obtained experimental results prove the validity of the proposed novel AFAS.

### 3.2.1 Remarks on Fingerprint Image Quality Evaluation Methods

One of the main techniques to test the performance of an automatic fingerprint recognition system relies heavily on the quality analysis of the acquired fingerprint image [58]. In literature many researchers have studied, proposed and implemented different methods for evaluating the images quality, using for example artificial neural networks, micro and macro features analysis, texture feature estimates and so on.

In [52] the authors propose a hybrid scheme to measure the quality of fingerprint images by combining both local and global characteristics. It uses local texture features and some global factors such as the standard deviation of Gabor features, the foreground area and central position, the number of minutiae and the existence of singular points. The authors define seven quality indexes and also two weighting methods, an overlapping area based method and a linear regression method, for computing the correlation between the final quality value and each quality index.

In [76] the authors present a fast fingerprint enhancement algorithm, based on the estimated local ridge orientation and frequency, which can adaptively improve the clarity of ridge and valley structures of input fingerprint images. It models the ridge and valley patterns as a sinusoidal wave, and then calculates the amplitude, frequency, and variance of the wave to determine the quality of the fingerprint regions.

In [59] the authors define a method not aimed at selecting images of good visual appearance, but aimed at identifying poor quality as well as invalid fingerprints for automatic fingerprint identification systems. It analyses the image in the spatial domain and uses the orientation certainty to certify the localized texture pattern, while ridge and valley structure to detect invalid images.

In [60] the authors implement an effective quality classification method for fingerprint images based on neural networks. It uses effective area, energy concentration, spatial consistency and directional contrast as quality indexes. A comparison with individual quality index thresholding and linear weighted sum method, on a private database, shows the higher quality classification accuracy of their method.

In [61] the authors describe a novel method for estimating the quality of fingerprint images using both local and global analyses. They propose a fusion method mixing the information from ridge and valley line resolution, fingerprint area and gray levels average and variance, using the golden section method to select the relevant weights value.

In [62] the authors propose a novel quality-checking algorithm which considers the condition of the input fingerprints and the orientation estimation errors. First, the 2-D gradients of the fingerprint image is separated into two sets of 1-D gradients, and then, the shape of the probability density functions of these gradients is measured in order to determine the fingerprint quality.

In [63] the authors present an image quality assessment technique for a novel fingerprint multimodal algorithm to provide high accuracy under non-ideal conditions. It uses the Redundant Discrete Wavelet Transform to assess the image quality, for high resolution fingerprint databases, by determining the presence of noise, smoothness, and edge information in a fingerprint image. Successively, in [64] the authors extend this technique designing a local image quality assessment algorithm. They use it as the first step of a novel algorithm for fast extraction and identification of level-3 features, such as pores, ridge contours, dots and incipient ridges.

After an exhaustive analysis of the above described methods for fingerprint image quality evaluation and in order to achieve the best trade-off between execution time and used resources for mobile devices, a mixed method has been designed and integrated in the proposed novel embedded AFAS. It is based on a fingerprint image global analysis in the spatial domain and it is inspired by works described in [52][59].

## 3.3 The Proposed Novel Embedded Fingerprint Authentication System

The proposed minutiae based AFAS is focused on the acquired raw image quality evaluation identifying poor quality areas, such as dry and moist portions, in order to overcome the common problems in wrong acquisitions on mobile devices. The system checks if the distance between image center and the fingerprint center coordinates is lower than an experimental fixed threshold in order to extract the maximum number of corresponding minutiae. If this condition is verified and the image overcomes the quality constraints, only high quality image portions are processed. Otherwise, the image is rejected and the system gives to the user suggestion feedbacks about the wrong acquisition step, helping him to obtain a better image quality in the next fingerprint acquisition task. In addition, an advanced matching technique for user recognition, based on partial fingerprints, is performed to improve the system accuracy [53]. This technique calculates a likelihood ratio by trying every possible overlaps of the acquired fingerprint with the enrolled one. The roto-translation parameters computation is based on the similar minutiae pairs identification belonging to both fingerprints.

Considering the functionalities of the proposed system, three main components can be identified: the User Interface (UI), which enables the user to interact with the system, the Acquisition Module (AM), which deals with the fingerprint image acquisition, and the Processing Module (PM), based on the FPGA processing engine implementing the authentication phase (Figure 3.2).

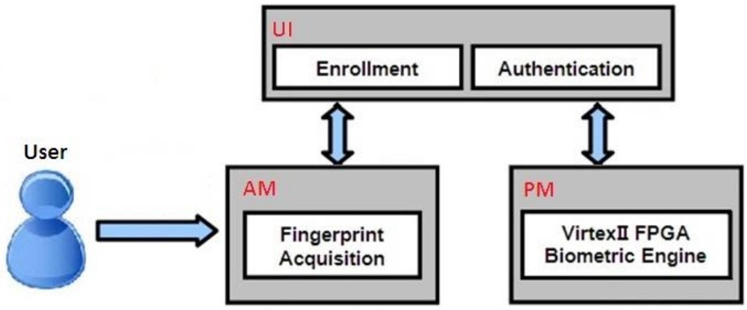


**Figure 3.2.** System's components: the User Interface (UI), the Acquisition Module (AM) and the Processing Module (PM).

Using the proposed PM, no image enhancement after fingerprint acquisition is performed. Therefore, a considerable savings in terms of execution time and hardware resources has been achieved with respect to a standard AFAS implementation. With more details, the proposed AFAS requires an Image Quality Evaluation module, including a Binarization module, a Thinning module, a Feature Extraction module, an Alignment module, and, finally, a Matching module. Despite to a standard AFAS implementation no Normalization, Enhancement, Field Orientation, Filtered Orientation and Smoothing tasks are required (Figure 3.3).

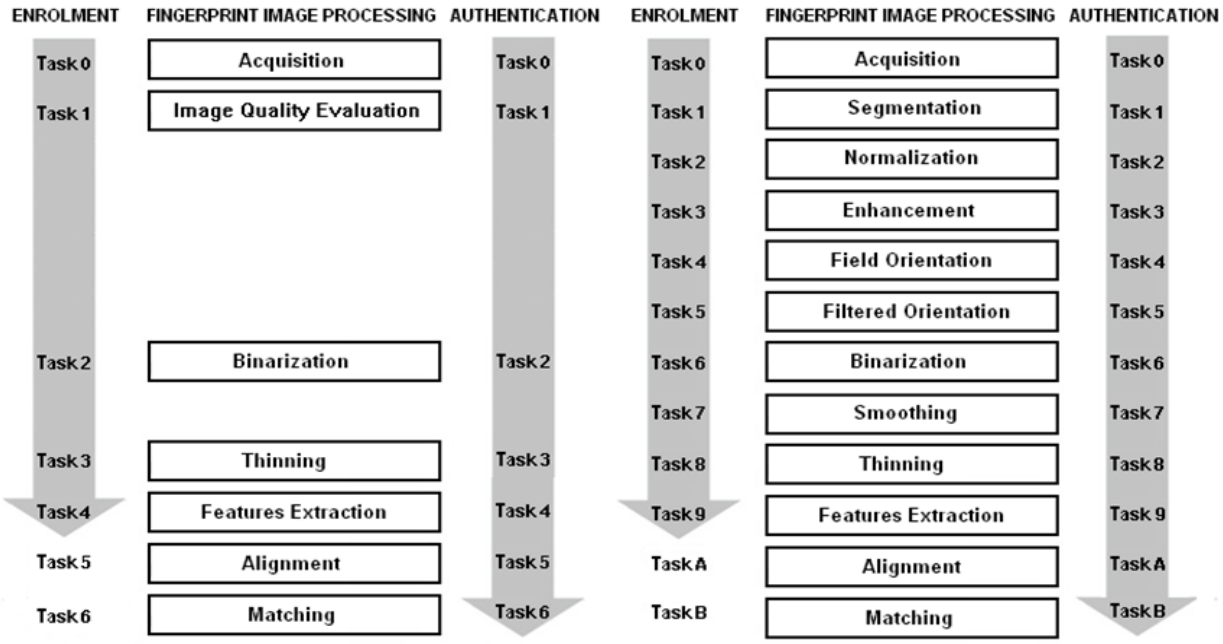


**Figure 3.3.** Comparison between the proposed AFAS (on the left) and the standard AFAS (on the right).

In the following subsections the main sub-modules of the proposed novel AFAS will be described.

### 3.3.1 Fingerprint Image Quality Evaluation module

This module, inspired by works described in [52][59], evaluates the fingerprint image quality through a global analysis in the spatial domain. With more details, it analyses the image by blocks, calculates the fingerprint central position, identifies the dry and moist blocks, and classifies the image quality into two levels.

Figure 3.4 shows the architecture of the proposed module. It is composed of Image Blocks Analyzer sub-module, Indexes Calculator sub-module, Quality Level Evaluator sub-module. The Image Block Analyzer sub-module is composed of Max Min Level Calculator sub-module, Orientation Certainty Level Calculator sub-module, Average Calculator sub-module, Variance Calculator sub-module, Block Analyzer sub-module and Fingerprint Center Calculator sub-module.

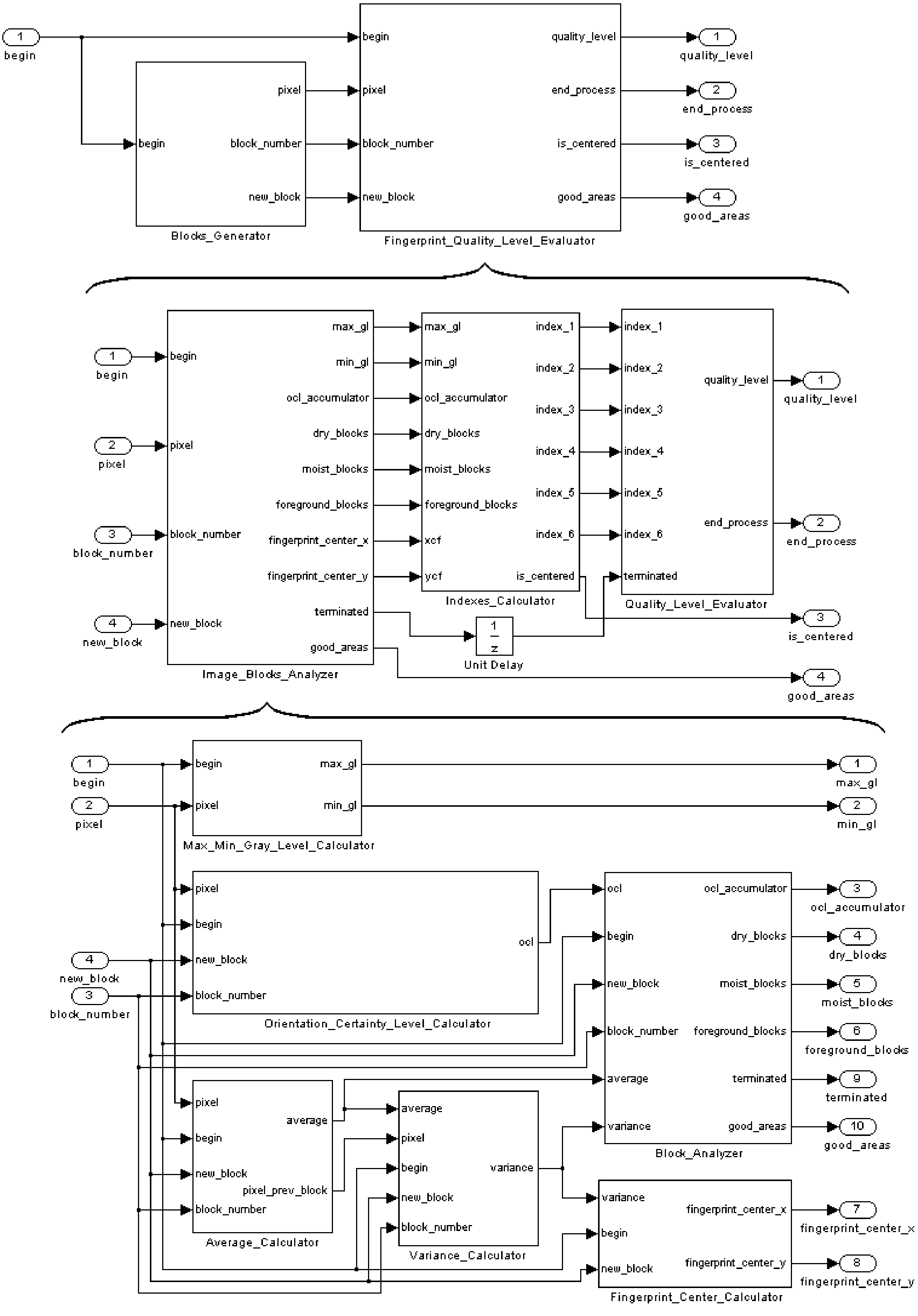


**Figure 3.4.** The proposed architecture evaluates the fingerprint image quality level.

In the follow, the sub-modules are described.

***Image_Blocks_Analyzer sub-module***

This sub-module is able to process block by block the fingerprint image. For each block it calculates, in a concurrent way, the following features:

- Max and min gray level. These local values are used to calculate the global max and min gray level of the entire image;
- Gray levels average and variance. These values are used to classify blocks as foreground/background and as dry/moist/good;
- Ridges orientation certainty level (ocl). This value, only for foreground blocks, is added to the ocl_accumulator signal, subsequently used for the calculation of the $2^{nd}$ index.

After that, in a concurrent way, it identifies the fingerprint high quality areas and calculates the fingerprint central position. The following subsections describe the main sub-modules of the proposed Image_Blocks_Analyzer sub-module.

➢ Orientation_Certainty_Level_Calculator sub-module

A fingerprint image block generally consists of ridges separated by valleys with the same orientation. Ridges and valleys constant structure and regular orientation can be used to evaluate the quality of each considered block. They are analytically calculated through the gradient of the gray levels along the x and y directions of a pixel [59]. The covariance matrix C of the gradient vector for an image block of M points is given by:

$$\mathrm{C} = \mathrm{E}\left\{\begin{bmatrix}\mathrm{dx}\\ \mathrm{dy}\end{bmatrix}\begin{bmatrix}\mathrm{dx} & \mathrm{dy}\end{bmatrix}\right\} = \begin{bmatrix}\mathrm{a} & \mathrm{c}\\ \mathrm{c} & \mathrm{b}\end{bmatrix}$$

where

$$\mathrm{E}\{\bullet\} = \frac{1}{\mathrm{M}}\sum_{\mathrm{M}} \bullet$$

The ridges orientation certainty level (ocl) is calculated as shown in Eq. (3.1).

$$\mathrm{ocl} = 100 * \frac{(\mathrm{a}+\mathrm{b}) - \sqrt{(\mathrm{a}-\mathrm{b})^2 + 4\mathrm{c}^2}}{(\mathrm{a}+\mathrm{b}) + \sqrt{(\mathrm{a}-\mathrm{b})^2 + 4\mathrm{c}^2}} \qquad (3.1)$$

With low (high) ocl values, the local structure and orientation of ridges and valleys are very regular (irregular), and therefore the block has good (wrong) quality (Figure 3.5). With more details, this sub-module is further composed of two sub-modules, implementing a two stage pipeline (Figure 3.6). While the first sub-module calculates the covariance matrix C of block j, the second sub-module calculates the ocl value of j-1 block.

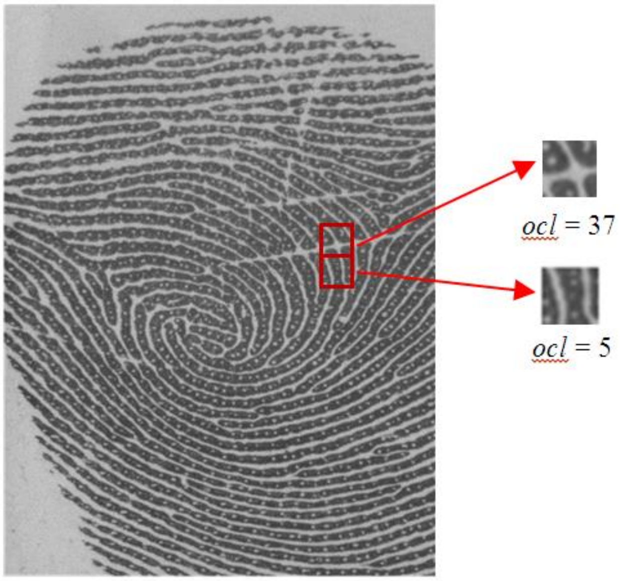


**Figure 3.5.** Examples of different ocl values.

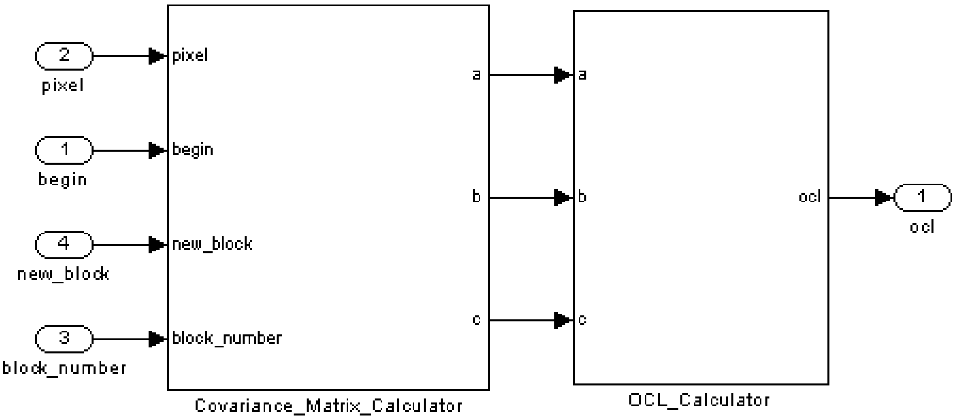


**Figure 3.6.** Orientation_Certainty_Level_Calculator sub-module.

- Average_Calculator and Variance_Calculator sub-modules

Average and variance are important characteristics for evaluating the block quality: average measures the luminosity, while variance measures the contrast. A low average value is linked to a block prevalently containing ridges (because it is dark), while a low variance value entails the block doesn’t contain any useful portion of the fingerprint (because it has a low contrast).

The Average_Calculator sub-module stores the incoming block pixels on a shift register and sends the pixels of the previous block in order to achieve the best trade-off between requested resources and execution time.

- Block_Analyzer sub-module

The ocl characteristic is not sufficient to quantify the clearness of the fingerprint ridges and valleys pattern when the skin humidity is also considered. For a moist block the ridges are too thick, since it has low average value. On the other hand, the ridges are too thin for a dry block, since it has a high average value. So, the average value is heavily influenced by the background gray level intensity (Figure 3.7). In this work, the gray level intensity of the image background is fixed to be the average value of the first image block, since it does not usually contain part of the fingerprint. If the block contains part of the fingerprint (i.e. the fingerprint covers the entire image) the background gray level is assumed as dark.

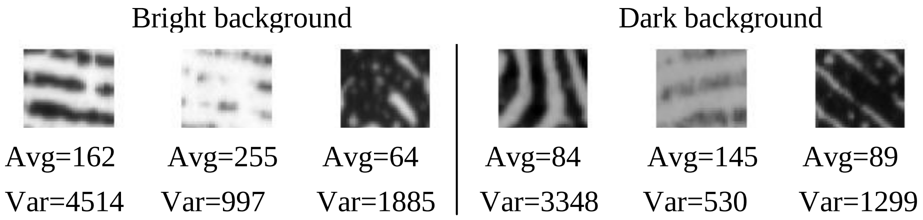
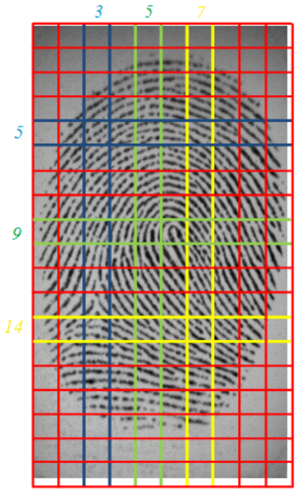


**Figure 3.7.** Examples of average and variance values with dark and bright background.

This sub-module compares the average value of the first block with an experimental fixed threshold classifying the background as dark or bright and setting moist and dry thresholds. These values are experimentally fixed and depend on the used database. As example, on the FVC2002 DB2-B, the dry thresholds are 140 and 180 for bright and dark background, respectively, while the moist thresholds are 80 for dark background and 100 otherwise. Successively, it classifies each block as foreground or background using the incoming variance value. The foreground threshold is not influenced by the background gray level and it is experimentally fixed to 190.

- Fingerprint_Center_Calculator sub-module

This sub-module calculates, in a concurrent way, the fingerprint central position (Figure 3.8). It checks if the considered block belongs to the column Nx/2, Nx/4 or 3Nx/4. If so then, if it is of foreground, a value equal to the block size is added to the relevant column foreground accumulator (an accumulator for each considered column), otherwise only if this accumulator value is zero, the same value is added to the relevant column background accumulator (i.e. the background blocks below the fingerprint are discharged). Concurrently, the same check is performed on the rows Ny/2, Ny/4 or 3Ny/4, and, in the same way, the relevant row background or foreground accumulator is increased. Finally, for the last block, the column foreground accumulator with the highest value is selected and the y-coordinate of the fingerprint’s center is calculated as the sum of the half value stored in the selected foreground accumulator and the relevant column background accumulator value. Concurrently, the x-coordinate of the fingerprint's center is calculated in the same way.



$$N_x = 9; \quad N_y = 18$$

$$f\_N_{x5} = 480; \quad b\_N_{x5} = 30$$
$$f\_N_{x3} = 390; \quad b\_N_{x3} = 90$$
$$f\_N_{x7} = 450; \quad b\_N_{x7} = 30$$

$$f\_N_{y9} = 270; \quad b\_N_{y9} = 0$$
$$f\_N_{y5} = 240; \quad b\_N_{y5} = 30$$
$$f\_N_{y14} = 270; \quad b\_N_{y14} = 0$$

$$fingerprint\_center\_x = 270/2 + 0 = 135$$
$$fingerprint\_center\_y = 480/2 + 30 = 270$$

**Figure 3.8.** Example of a FVC2002 fingerprint's center calculation.

***Indexes_Calculator sub-module***

Among common quality indexes present in literature and reported in the related works section, this subsystem concurrently calculates six global indexes, designed in order to realize a module reducing used resources and execution time. To make all indexes compatible, they have normalized in the range of [0, 100]. High index value entails a good image quality.

➢ Index1_Calculator sub-module

The first index measures the contrast between fingerprint and background. This value is calculated as the difference between the maximum and the minimum gray level value of the entire image, Eq. (3.2):

$$\text{index\_1} = 100 * (\text{max\_gl} - \text{min\_gl})/255 \qquad (3.2)$$

➢ Index2_Calculator sub-module

The second index extends to the whole image the considerations about the block orientation certainty level estimation, thus, globally measuring the clarity and continuity of ridges and valleys orientation. It is calculated by averaging all the ocl values relating to only foreground blocks, Eq. (3.3):

$$\text{index\_2} = 100 - \frac{\text{ocl\_accumulator}}{\text{foreground\_blocks}} \qquad (3.3)$$

- Index3_Calculator sub-module

The third index measures the humidity of the entire image and it is calculated as the ratio between the number of moist blocks and the number of foreground blocks, Eq. (3.4):

$$\text{index\_3} = 100 - (100 * \frac{\text{moist\_blocks}}{\text{foreground\_blocks}}) \qquad (3.4)$$

- Index4_Calculator sub-module

The fourth index measures the dryness of the entire image and it is calculated as the ratio between the number of dry blocks and the number of foreground blocks, Eq. (3.5):

$$\text{index\_4} = 100 - (100 * \frac{\text{dry\_blocks}}{\text{foreground\_blocks}}) \qquad (3.5)$$

- Index5_Calculator sub-module

The fifth index measures the image area occupied by the foreground blocks. It is an estimate of the fingerprint size over the entire image and it is calculated as the ratio between the number of foreground blocks and the total number of blocks, Eq. (3.6):

$$\text{index\_5} = 100 * \frac{\text{foreground\_blocks}}{\text{N}} \qquad (3.6)$$

- Index6_Calculator sub-module

The sixth index measures the position of the fingerprint over the entire image: too large translation caused by human behavior can generate an insufficient overlapping area between images captured from the same finger. It is calculated as the average of two values, $\text{i6}_\text{x}$ and $\text{i6}_\text{y}$, Eq. (3.7):

$$\text{index\_6} = \frac{\text{i6}_\text{x} + \text{i6}_\text{y}}{2} \qquad (3.7)$$

with

$$\text{i6}_\text{x} = 100 - \left(100 * \frac{|\text{x}_\text{cf} - \text{x}_\text{ci}|}{\text{x}_\text{ci}}\right), \qquad \text{i6}_\text{y} = 100 - (100 * \frac{|\text{y}_\text{cf} - \text{y}_\text{ci}|}{\text{y}_\text{ci}})$$

where xcf and ycf are the coordinates of the fingerprint’s center, while xci and yci are the coordinates of the image’s center.

In addition, this subsystem checks if the distance between the respective coordinates of the image's center and the fingerprint's center is lower than a threshold (experimentally fixed to 100) and then sets the is_centered signal.

***Quality_Level_Calculator sub-module***

First, this subsystem calculates the final fingerprint quality index as linear combination of the previous six indexes. As described in [52], a linear regression method is used for weights calculation. They are experimentally determined by performing tests to observe the behavior of the change in the final quality index while one index is changing and the others are constant. Experimental results show that the most relevant indexes are ocl, fingerprint moisture and fingerprint dryness. Then, by comparing the final quality index value with a threshold (experimentally fixed to 65), this subsystem classify the image quality level as Good or Bad. Finally, the subsequent tasks are performed only if the quality is Good and the fingerprint is centered over the image.

### 3.3.2 Fingerprint Features Extraction module

For the fingerprint features extraction, the module described in [70] has been modified: no image enhancement is performed. Figure 3.9 shows the proposed schema. It is composed of four blocks: three processing blocks and one master controller block. The Image Pre-processor performs an adaptive binarization. The Macro-Features Extractor block extracts the singularity points. The Micro-Features Extractor block performs an adaptive thinning, extracts the minutiae, around the singularity points, and executes a post-processing phase to erase potential false minutiae. The Master Controller block synchronizes and coordinates the two extractor blocks. With more details, the Macro-Features Extractor module computes the directional image and then extracts the Core and Delta points using the Poincarѐ indexes algorithm [71]; while the Micro-Features Extractor module performs an adaptive thinning, of the areas centered on the extracted singularity points, and then extracts the minutiae (termination and bifurcation points) only in good areas. The Master Controller module plays an important role, since it synchronizes and coordinates the two features extractor modules:

- While storing in memories the incoming binary image, the Master Controller module divides it in 9x9 non overlapping blocks and sends, to the Macro-Features Extractor module, a 17x17 mask centered in the considered block;
- Every time the Macro-Features Extractor module detects a singularity point, only in the User Enrollment, Authentication or Identification phase, the Master Controller module sends to

the Micro-Features Extractor module a 16Nx*16Nx window centered on the detected point. If the Macro-Features Extractor module does not extract singularity points, a central 32Nx*32Nx window, organized as 4 non overlapping windows, is sent.

Figure 3.10 shows the proposed architecture.

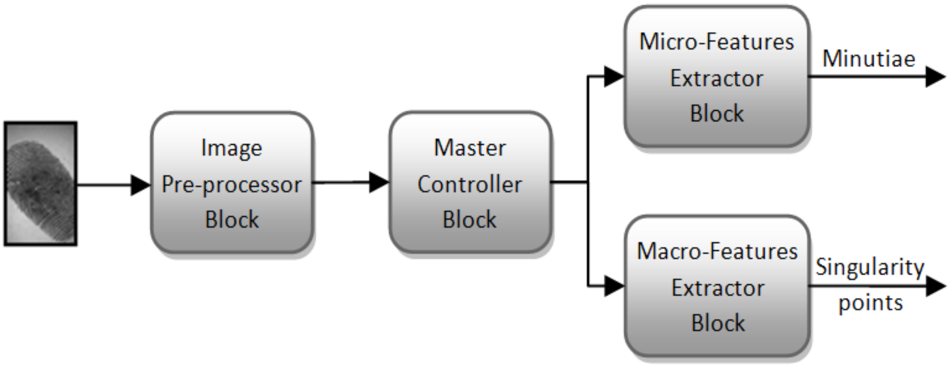


Image Pre-processor Block
Master Controller Block
Micro-Features Extractor Block
Minutiae
Macro-Features Extractor Block
Singularity points

**Figure 3.9.** The proposed Fingerprint Features Extractor schema.

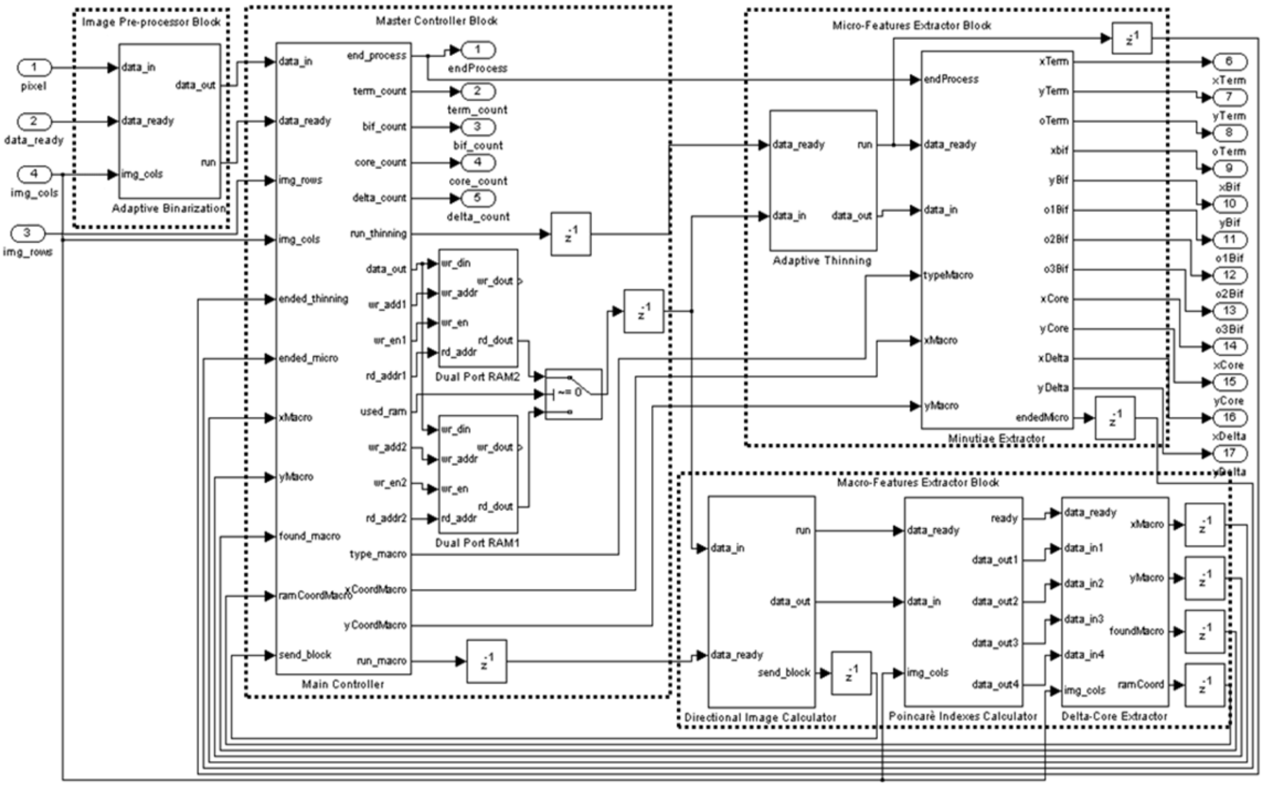


**Figure 3.10.** The proposed architecture for fingerprint features extraction.

The proposed Fingerprint Features Extractor is focused on adaptive processing allowing to acquire the user image from different fingerprint sensors. In order to reduce execution times, power consumption and used resources, the Features Extractor device realizes a four stages pipeline:

- Stage 1

The first stage performs the adaptive binarization proposed in [70], giving out an image where pixels assume a binary value: white as background or black as foreground. The binarization process is based on the software adaptive technique described in [65], setting the threshold as the average of the maximum and minimum gray level values in a local window. The relevant hardware implementation, using for example a 8x8 window, needs to store 7 rows and 8 pixels to realize the pipeline. In addition, it needs to calculate a new average value for each pixel. Despite, the novel hardware approach proposed in [70] processes the image by row: it uses a 60 pixels buffer, belonging only to the working row, and calculates its average value, used for all the buffered pixels (Figure 3.11).

In this implementation, after an exhaustive analysis on images of different size, a general equation based on the ratio between the image width and block size is proposed for the pixel buffer size (Eq. 3.8). This equation obtains the best trade-off between used resources and execution times:

$$buffer\_size = 7 * (img\_width/block\_size) \approx 7 * N_x \tag{3.8}$$

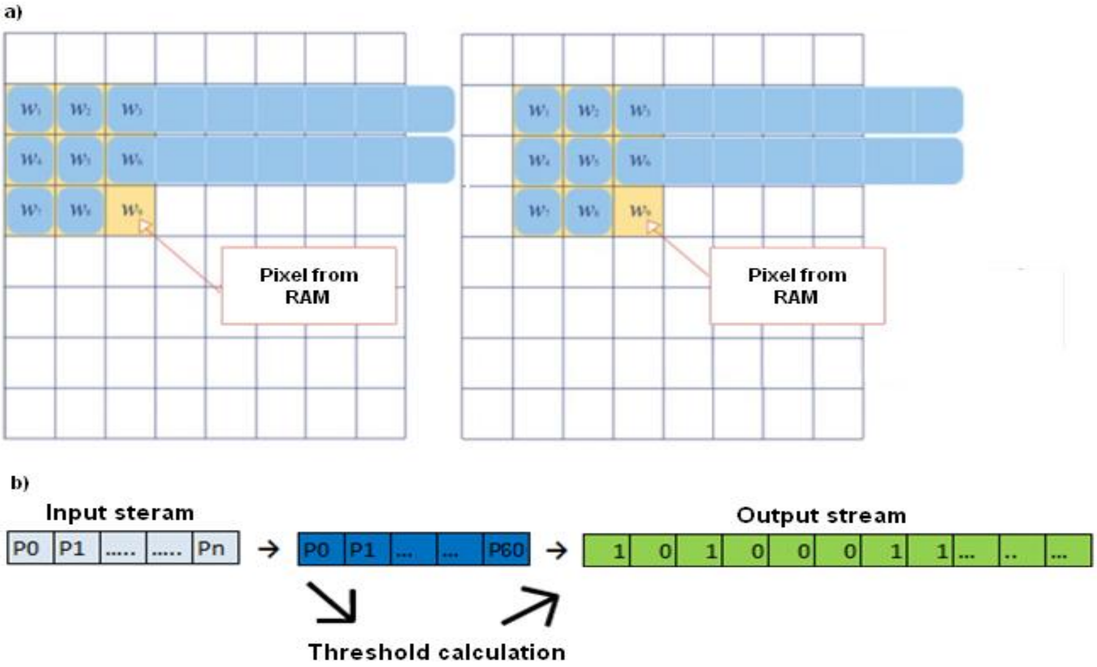


**Figure 3.11.** A comparison between a) the software approach described in [65] and b) the novel hardware approach proposed in [70].

- Stage 2

The second stage computes the directional image and then extracts the singularity points (Core and Delta) using the Poincarè indexes algorithm [71]. It computes the directional image in three steps: extraction of the direction for each pixel; processing of the previous step output assembling the pixels in 8x8 blocks; computing of the predominant direction for each block (in every 8x8 block, the direction with greater frequency is attributed to the considered block). The proposed module extracts 8 directions, from 0° to 180° as shown in Figure 3.22, and codified as a number in [0, 7].

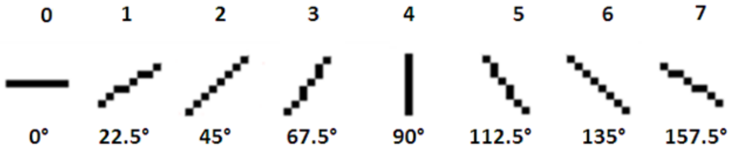


**Figure 3.22.** The 8 directions used to build the directional image

In order to extract the direction $\mathrm{D(i,j)}$ of the point $\mathrm{(i,j)}$ a vector $\mathrm{v}$ is calculated by the following equation (3.9):

$$v[k] = \sum_{d=1}^{q}\left[C\left(i_{d,}\, j_d\right) - C(i,j)\right] \qquad with\ k = 0..7 \qquad (3.9)$$

where $\mathrm{C(i,j)}$ and $\mathrm{C\left(i_{d,}\, j_d\right)}$ indicate the gray level of points $\mathrm{(i,j)}$ and $(\mathrm{i}_{\mathrm{d},}\ \mathrm{j}_{\mathrm{d}})$, respectively, while q=16 is the number of selected pixels along a considered direction. The direction $\mathrm{D(i,j)}$ is obtained as the position of the minimum value in the vector v. However, acquisitions not correctly performed can affect the calculation of predominant directions inside spoiled zones; therefore, a smoothing algorithm is applied. This is achieved by calculating the directional histogram, comparing the directions in areas of 3x3 blocks: the direction of the central block is replaced by the higher frequency direction of the neighboring blocks.

➢ Stage 3

The third stage performs an adaptive thinning, reducing the ridge thickness to the unitary value. The sub-module proposed in [70] has been optimized in order to reduce hardware resources and execution times (Figure 3.12). The adaptive processing is performed comparing the current thinned image with the previous thinned image stored in memory.

➢ Stage 4

The last stage performs a minutiae (termination and bifurcation points) extraction, filtering the thinned area in a 3x3 sliding window. For the minutiae extraction, the architecture proposed in [70] has been used. However, in order to reduce the system execution time and the potential false minutiae number, only the good areas are processed. Figure 3.13 shows the proposed architecture.

Each minutia is described by the type (termination or bifurcation), the x-y Cartesian coordinates and one angle for terminations or three angles for bifurcations with respect to the x axis. In order to calculate the minutiae orientations, it compares the position of the considered minutia with the pixels of the thinned image in a 5x5 window (Figure 3.14).

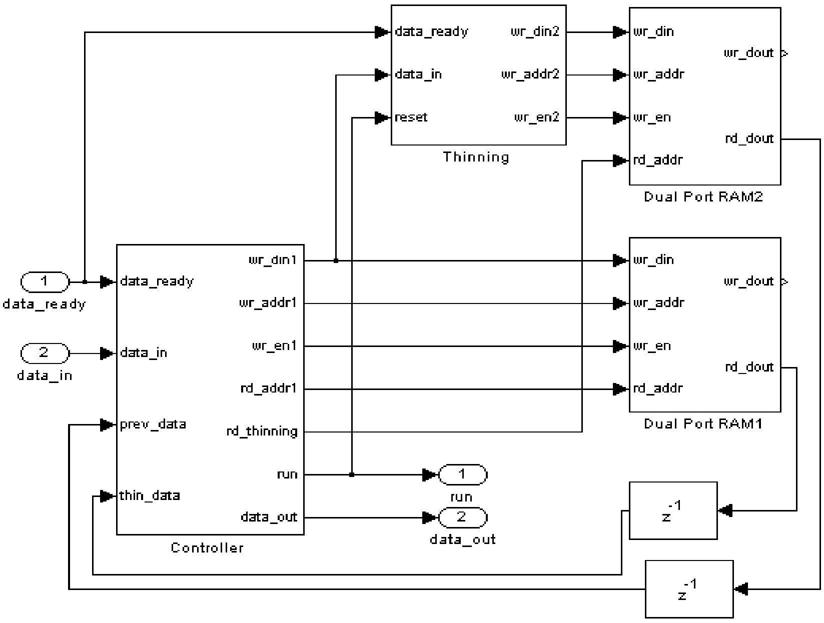


**Figure 3.12.** The proposed Adaptive Thinning sub-module

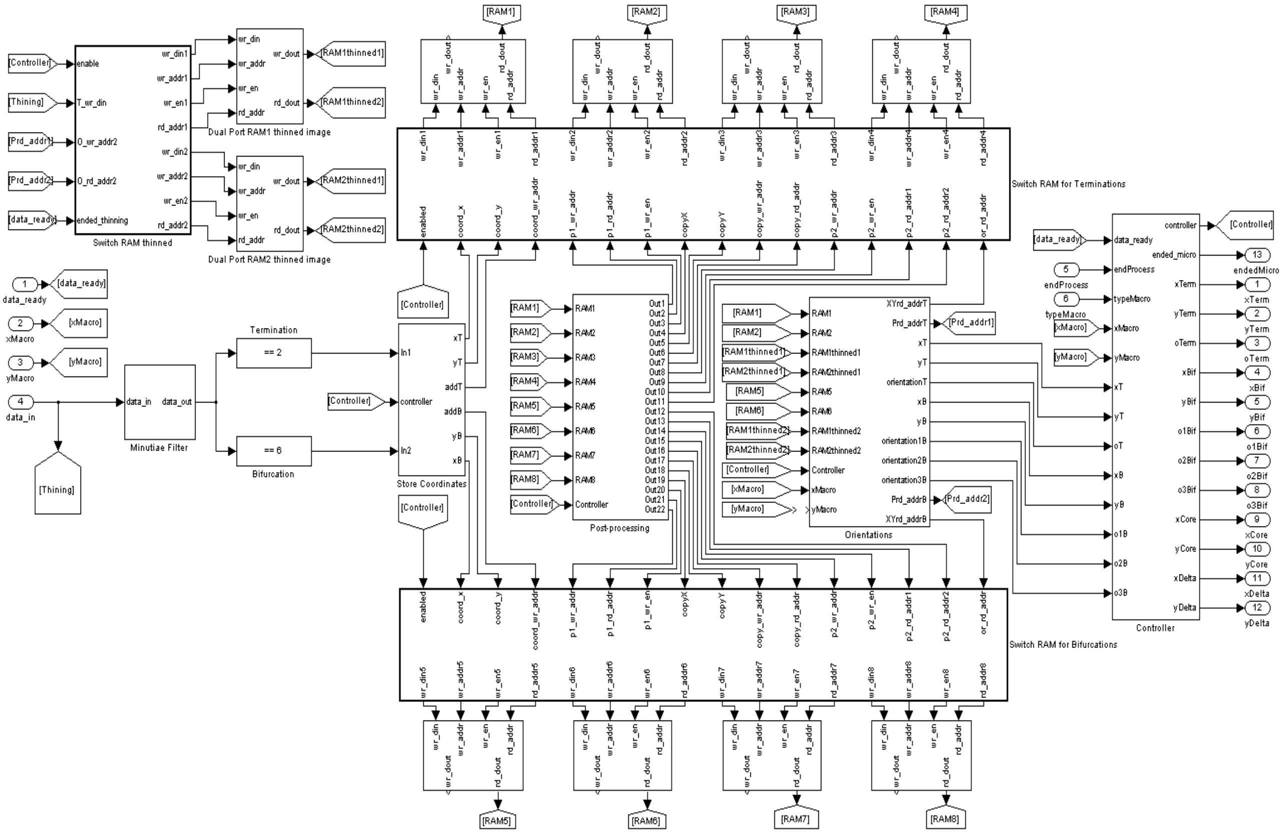


**Figure 3.13.** The proposed Minutiae Detection module

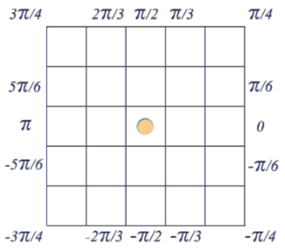


**Figure 3.14.** The 5x5 window used to calculate the minutia orientations

***Post-processing Phase***

In order to remove potential false minutiae the algorithm proposed in [72] has been implemented. It checks if the Euclidean distances between termination-termination, bifurcation-bifurcation and termination-bifurcation are lower than three different thresholds, experimentally fixed. Figure 3.15 shows the proposed architecture.

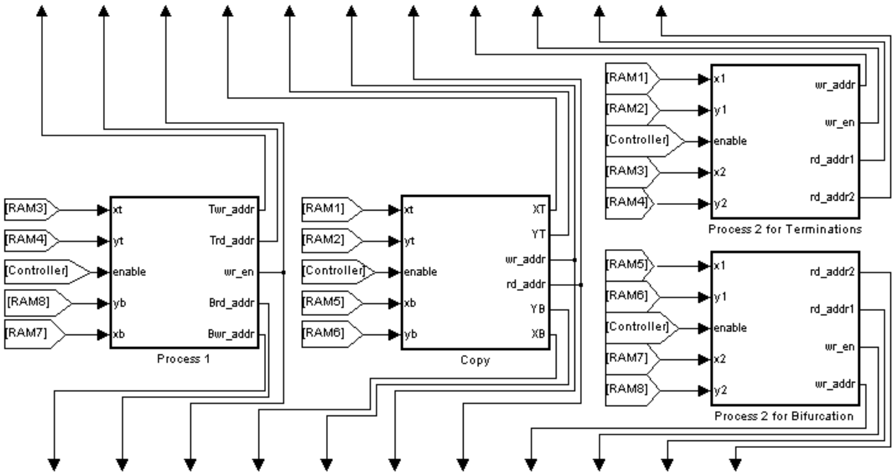


**Figure 3.15.** The proposed architecture of the Post-processor module

### 3.3.3 Minutiae-based Matching module

For the minutiae matching, the algorithm proposed in [53] has been used. The computation of a likelihood ratio in fingerprint authentication is obtained by trying all the possible overlapping of the acquired fingerprint with the one enrolled in the system. The roto-translation parameters computation is based on the identification of two similar pairs of minutiae belongs to both fingerprints (Figure 3.17). A threshold (experimentally fixed to 175) based on Euclidean distance is used to generate the minutiae pairs.

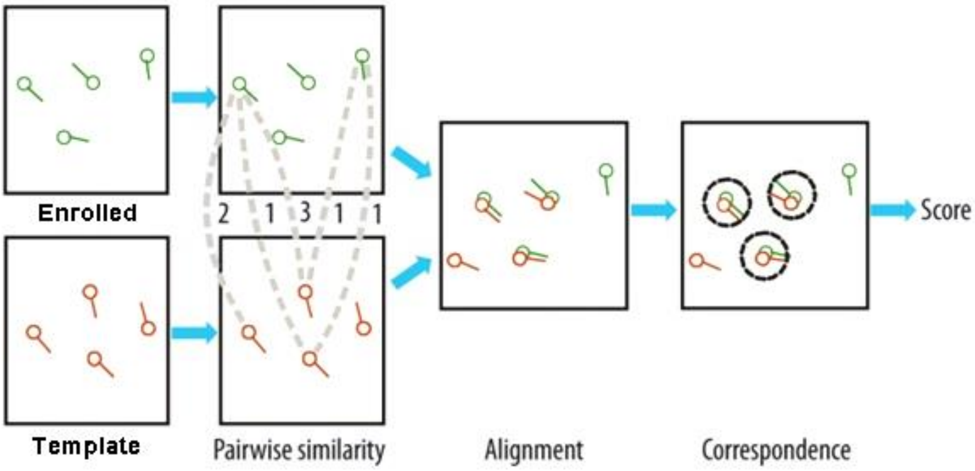


**Figure 3.17.** The Roto-translation parameters computation

First, roto-translation parameters are computed only if the value of Euclidean distance between each minutiae pair of both fingerprints is lower than a threshold (experimentally fixed to 20). The rotation parameter is based on the differences between the corresponding angles in the selected minutiae pairs. If the gap between each of these differences with respect to the other is lower than a threshold (experimentally fixed to 1.5) the rotation parameter is the average of the calculated differences. In the same way, the translation parameter is based on the differences between the respective Cartesian coordinates in the selected minutiae pairs. If the gap between each coordinate distance is lower than a threshold (experimentally fixed to 30) the translation parameter is the average of the respective calculated differences.

Then, the roto-translation is performed and, for each minutia, differences between respective coordinates x-y (diffxy) and angles (difftheta) are calculated. Only when these differences are lower than two thresholds (xythreshold and thetathreshold, experimentally fixed to 15 and 0.785, respectively) a first partial score is obtained and normalized in the range of [0, 1]. The complete score is calculated as Eq. (3.9):

$$\mathrm{s_i} = 0.75 * \left(1 - \frac{\max\left(\mathrm{diff_{xy}}\right)}{\mathrm{xy_{threshold}}}\right) + 0.25 * \left(1 - \frac{\max(\mathrm{diff_{theta}})}{\mathrm{theta_{threshold}}}\right) \qquad (3.9)$$

where higher importance has been made to the differences between respective coordinates rather than to angles, due to rounding problems on data.

Finally, among all complete scores, only the greater is considered. Therefore, the final matching score is calculated adding the 12 highest obtained scores. In accordance with the USA guidelines in the forensic field, when two fingerprints have a minimum of 12 corresponding minutiae, these are regarded as coming from the same finger [66].

### 3.3.4 Experimental Results

The proposed approach introduces interesting characteristics for mobile devices. The architectural implementation on FPGA, considering its working frequency (50 MHz), achieves the performance of the highly competitive systems, realizing a good trade-off between accuracy rate, used resources and execution time. To evaluate the accuracy performances of the proposed authentication system, the FRR and FAR indexes have been used and two different free databases with different characteristics in terms of image resolution and dimensions have been used.

The following subsections report the used databases and datasets description, the execution time, the required hardware resources and the authentication performance of the proposed AFAS.

***Databases description***

*FVC2002 DB2-B database*

This free downloadable database has been made available for the second edition of the International Fingerprint Verification Competition [67]. It contains 80 fingerprint images of 296x560 pixels, with a resolution of 569 dpi. The images has been acquired from 10 users (8 acquisitions for user of the same finger), via the scanner Biometrika FX2000 [68], with a maximum rotation of about 35 degrees between impressions (Figure 3.18).

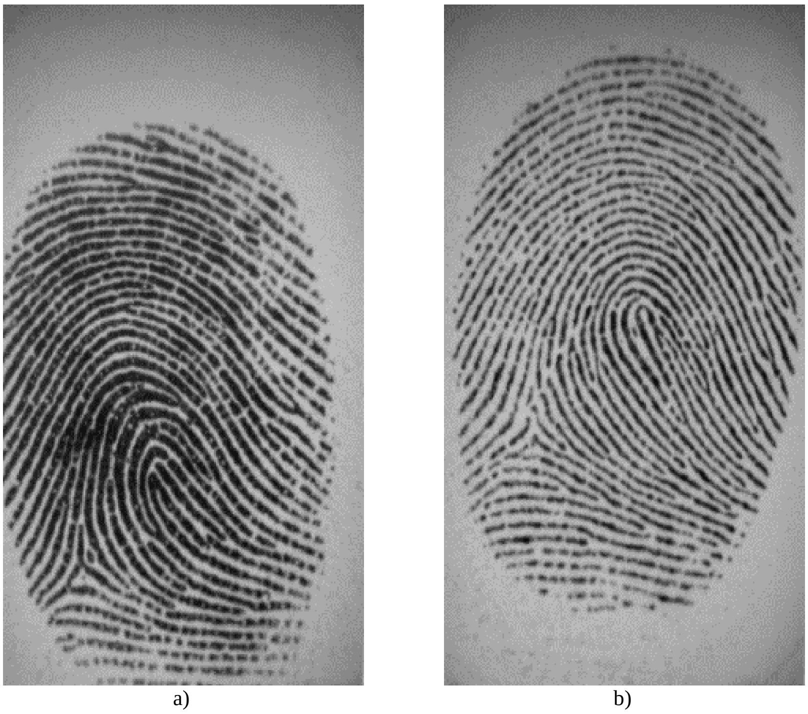


**Figure 3.18.** Two example images of the FVC2002 DB2-B acquired by Biometrika FX2000 sensor

*PolyU database*

This free downloadable database has been built at the Hong Kong Polytechnic University [57]. It contains 1480 fingerprint images of 480x640 pixels, with a resolution around 1,200 dpi of 148 users (10 acquisitions for user of two fingers, Figure 3.19). Each image name has been described using three numbers in the following way: first number represents the user, second number represents the finger, and third number represents the different acquisition.

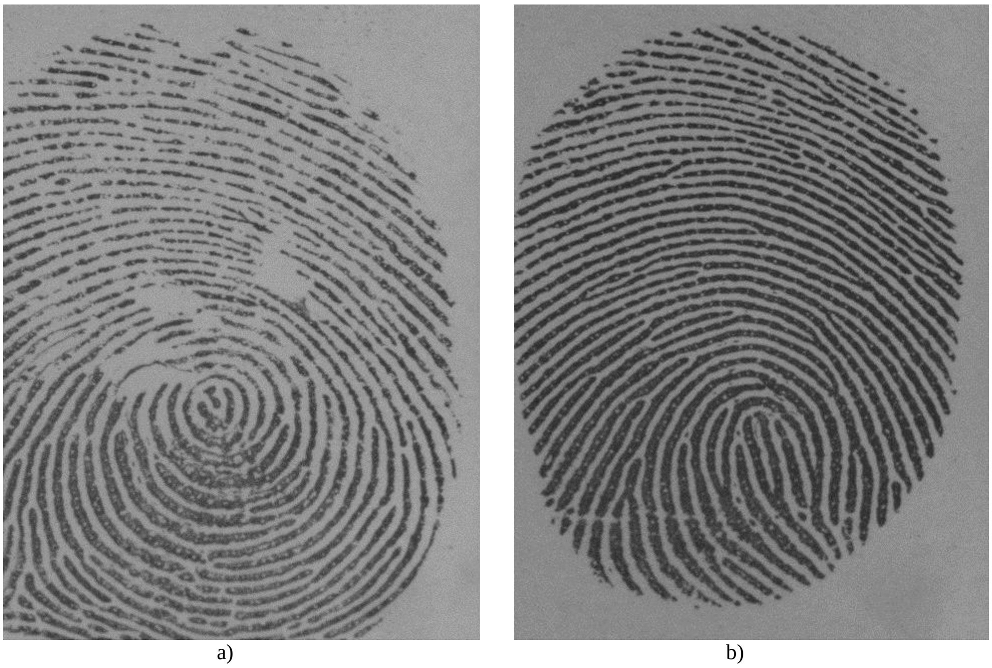


**Figure 3.18.** Two example images of the Hong Kong Polytechnic University database

*Datasets description*

Starting from the above description databases, two different datasets have been built:

- the *datasest1* has been generated using the entire FVC2002 DB2-B database (10 users, 8 acquisitions for user);
- the *datasest2* has been generated using a consistent subset of the PolyU database (100 users with 5 acquisitions for user of the same finger).

***Authentication Performance***

Three different tests have been conducted, starting from the AFAS described in [55] where the feature extract has been replaced by the module described in [70]:

1. the AFAS has been extended with the proposed fingerprint image quality evaluation module and tested on the *dataset1*;

2. the AFAS has been extended with the proposed fingerprint image quality evaluation module and, moreover, the matching algorithm has been replaced with the advanced technique, based on partial fingerprints, proposed in [53] and tested on the *dataset2*;
3. the AFAS has been extended with a pre-processing task, based on the Gabor filter, to enhance fingerprint images and, moreover, the matching algorithm has been replaced with the advanced technique, based on partial fingerprints, proposed in [53] and tested on the *dataset2*.

Table 3.1 illustrates the authentication performance in terms of FAR and FRR indexes for the three performed tests.

**Table 3.1.** FAR and FRR indexes of the three performed tests.

| Test Number | FAR | FRR |
|---|---|---|
| 1. | 0% | 6.25% |
| 2. | 0% | 8.00% |
| 3. | 0% | 9.00% |

***Execution Time***

The following tables (Tables 3.2-3.4) and Figure 3.19 illustrate the elaboration times, for the three performed tests, required by each single task, with a working frequency of 50 MHz.

**Table 3.2.** Execution times of test n. 1.

| Task | Execution Time (msec) |
|---|---|
| Image Quality Evaluation | 3.9 |
| Binarization | 2.2 |
| Thinning | 39.0 |
| Minutiae Extraction | 13.7 |
| Matching | 3.8 |
| **TOTAL** | 62.6 |

**Table 3.3.** Execution times of test n. 2.

| Task | Execution Time (msec) |
|---|---|
| Image Quality Evaluation | 3.9 |
| Binarization | 2.2 |
| Thinning | 39.0 |
| Minutiae Extraction | 13.7 |
| Matching | $2.35\text{x}10^3$ |
| **TOTAL** | $2.4\ \text{x}10^3$ |

**Table 3.4.** Execution times of test n. 3.

| Task | Execution Time (msec) |
|---|---|
| Gabor Filter | $2.4\text{x}10^3$ |
| Binarization | 2.2 |
| Thinning | 39.0 |
| Minutiae Extraction | 13.7 |
| Matching | $2.35\text{x}10^3$ |
| **TOTAL** | $4.8\ \text{x}10^3$ |

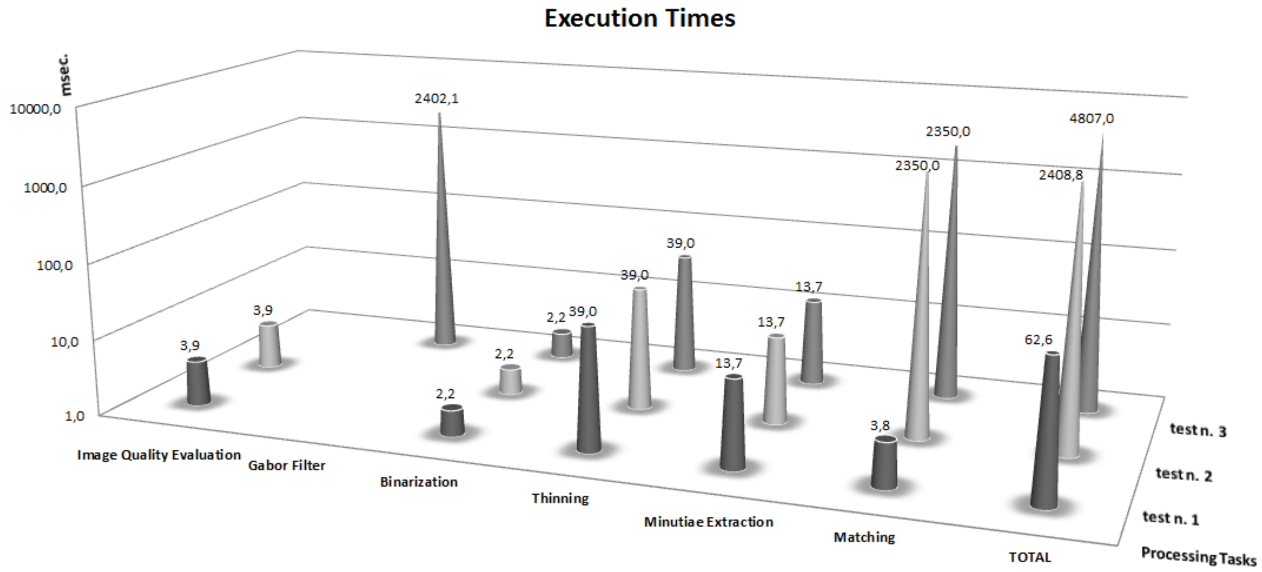


**Figure 3.19.** Elaboration times required by each processing task for the three performed tests

***Hardware Resources***

The following tables (Tables 3.5-3.7) depict the required hardware resources, for the three performed tests, used by each single task on the Agility RC2000 development board. Figure 3.20 illustrates the total used hardware resources for the three performed tests.

**Table 3.5.** Used resources of test n. 1.

| Resource Type | Image Quality Evaluation | Binarization | Thinning | Minutiae Extraction | Matching |
|---|---|---|---|---|---|
| Slices | 5.83% | 0.14% | 0.67% | 21.80% | 0.34% |
| Multiplier Blocks | 4.17% | 0.00% | 0.00% | 19.44% | 0.69% |
| RAM Blocks | 0.69% | 0.69% | 0.69% | 14.58% | 6.25% |
| IOBs | 4.98% | 4.98% | 0.00% | 0.00% | 10.57% |

**Table 3.6.** Used resources of test n. 2.

| Resource Type | Image Quality Evaluation | Binarization | Thinning | Minutiae Extraction | Matching |
|---|---|---|---|---|---|
| Slices | 5.83% | 0.14% | 0.67% | 21.80% | 65.97% |
| Multiplier Blocks | 4.17% | 0.00% | 0.00% | 19.44% | 1.39% |
| RAM Blocks | 0.69% | 0.69% | 0.69% | 14.58% | 0.69% |
| IOBs | 4.98% | 4.98% | 0.00% | 0.00% | 10.57% |

**Table 3.7.** Used resources of test n. 3.

| Resource Type | Gabor Filter | Binarization | Thinning | Minutiae Extraction | Matching |
|---|---|---|---|---|---|
| Slices | 11.41% | 0.14% | 0.67% | 21.80% | 65.97% |
| Multiplier Blocks | 2.78% | 0.00% | 0.00% | 19.44% | 1.39% |
| RAM Blocks | 1.39% | 0.69% | 0.69% | 14.58% | 0.69% |
| IOBs | 4.98% | 4.98% | 0.00% | 0.00% | 10.57% |

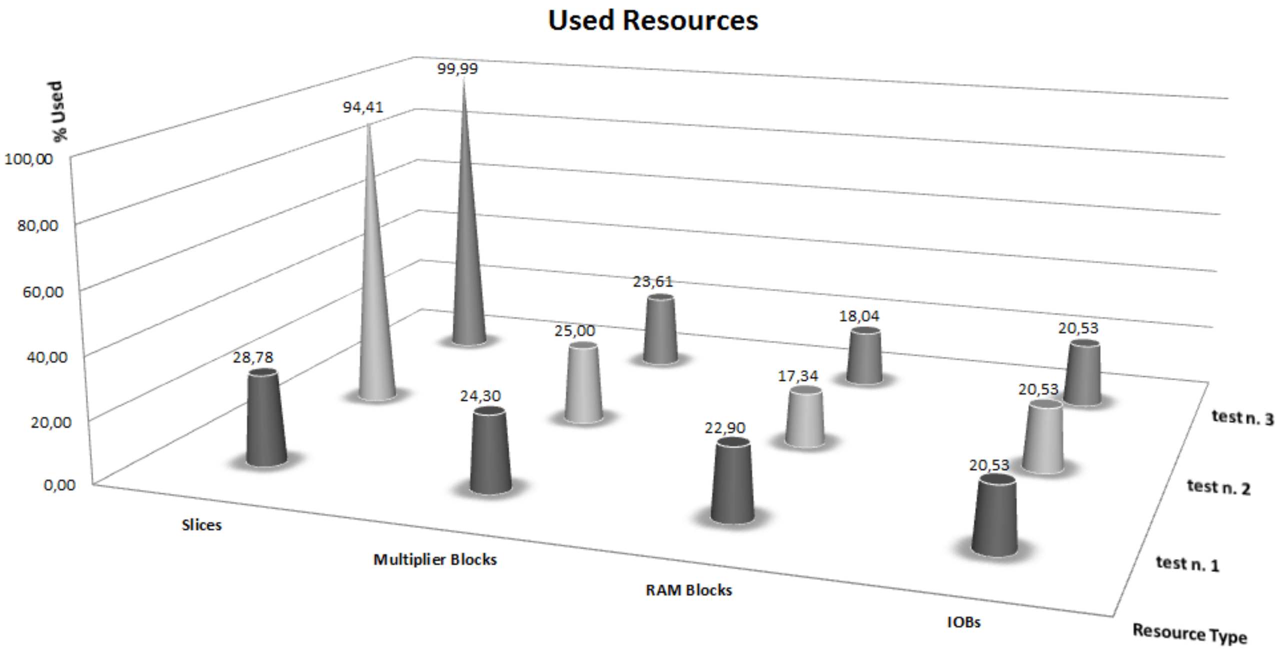


**Figure 3.19.** Used hardware resources, for the three performed tests

### 3.3.5 Sensor Storage Capabilities of User's Templates

The embedded biometric sensor uses the on board RAM to store and manage user biometric identity. With more details, one memory bank has been used for temporary storage process required by fingerprint processing tasks, while the remaining memory banks (6MB) have been used for to store user's biometric templates. Each user template requires about 512 byte, considering a variable number of templates. Table 3.8 shows the maximum number of users that can be enrolled on the sensor.

**Table 3.7.** Maximum number of user templates stored in the board memory.

| **Template Number for each user** | **Sensor Storage Capability (number of users)** |
|---|---|
| 2 | **6144** |
| 3 | **4096** |
| 4 | **3072** |

### 3.3.6 Extracted Minutiae Comparison

The proposed approach improves the performance of a standard embedded AFAS, such as would a Gabor filtering process in order to reconstruct the poor quality areas. Figure 3.16 shows the minutiae extracted using the Gabor filter, to reconstruct image areas of poor quality, and using the image quality evaluation, to discard those areas. As depicted, the Gabor filter approach introduces two false bifurcations and discards two terminations, while the proposed approach discards two bifurcations and one termination. Statistical analysis performed on a 50 images subset of both databases shows a rate of false minutiae discharged of 5% on the PolyU and 4% on the FVC database.

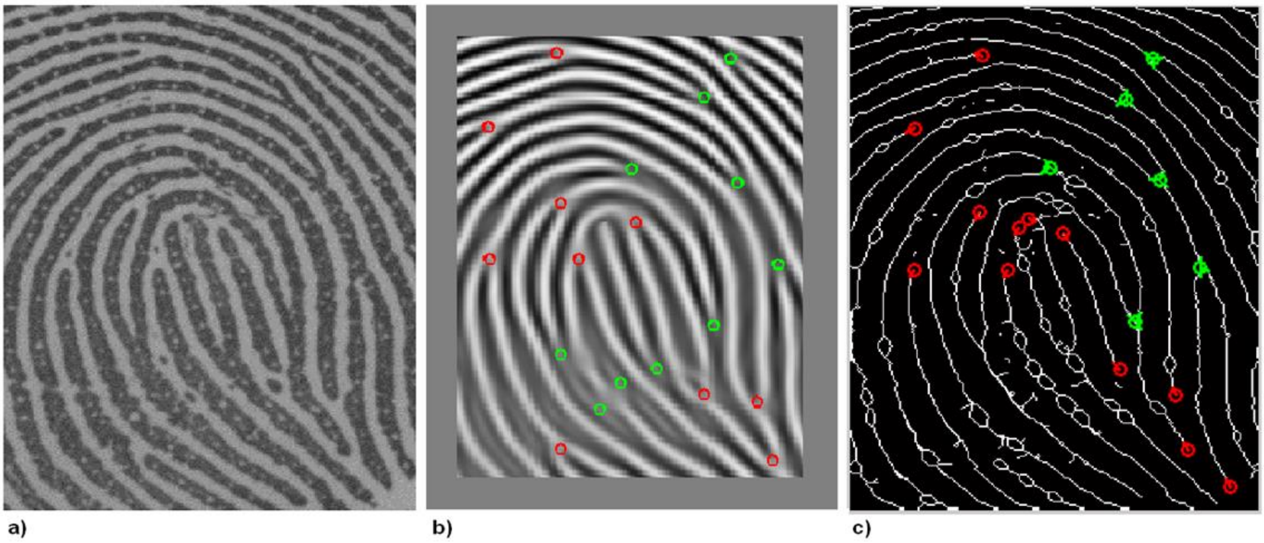


**Figure 3.16.** a) Image 2_1_5 from PolyU database; b) Minutiae extracted with a Gabor filter and without the image quality evaluation; c) Minutiae extracted with the image quality evaluation and without a Gabor filter

### 3.3.7 Discussion and Comparisons

User authentication is one of the most challenging issues for system and network security. A robust authentication mechanism is based on the use of biometric access control methods, processing one or more biometrics (such as a fingerprint). There are many approaches to deal with fingerprint verification. In recent literature publications, few findings have been on design and prototyping of an embedded biometric recognizer.

In [69] the authors proposed an implementation of a hardware identification system. However, the fingerprint matching phase was not developed and presented, so that no direct comparison with this work can be addressed. The remaining fingerprint processing tasks had been implemented in a FPGA device with a clock frequency of 27.65MHz and a processing time of 589.6 msec. Compared with this system, the achieved execution times denote high performance levels.

In [52] the authors use local texture features as well as some global factors such as the standard deviation of Gabor features, the foreground area and central position, the number of minutiae, and the existence of singular points. They produce a good analysis about Equal Error Rate (EER) for three databases: FVC2002 DB2A, Fujitsu database and FVC2002 DB4A.

In [19] the authors have developed a software fast fingerprint enhancement algorithm which can adaptively improve the clarity of ridge and valley structures based on the local ridge orientation and ridge frequency. Experimental results show that their enhancement algorithm is capable of improving both the goodness index and the verification performance. The whole execution time of the enhancement algorithm on a Pentium 200MHZ is 2.49 sec, with FAR=0.01% and FRR=27% (without enhancement) and FRR=9% (with enhancement) using the MSU fingerprint database (700 live-scan images; 10 per individual each).

In [63, 64] the authors present an image quality assessment software technique for a novel fingerprint multimodal algorithm to provide high accuracy under non-ideal conditions. Their study was based on a small number of minutia features. This is likely to be the case with latent fingerprints collected at a crime scene. Specifically, the performance of their fusion algorithm is studied when the number of minutiae is between 5 and 10. Experimental results show that while the performance of existing fusion algorithm decreases if compared to the performance of complete rolled fingerprints, the proposed approach is able to compensate for the limited partial information. The approach shows FRR between 91.35% and 97.98% with FAR=0.01%, using a comprehensive database with rolled and partial fingerprint images of different quality and arbitrary number of features.

## 3.4 The Proposed Novel Fingerprint Classification System

The proposed innovative Fingerprint Classification System [73] uses an heuristic approach, inspired by the work described in [74]. It is an efficient and effective method to optimize the training phase in fingerprint classification tasks, using only the directional image information. It combines a Fuzzy C-Means clustering method and a Naive Bayesian Classifier, and it is composed of three modules: Datasets Building, Training Dataset Extraction and Fingerprint Classification (Figure 3.20). Unlike literature approaches using a lot of training examples (e.g., in [42] authors use 30 images per class, while in [31] half of each class in the whole database), the proposed one requires only the use of 18 directional images per class. Figure 3.21 shows the proposed architecture.

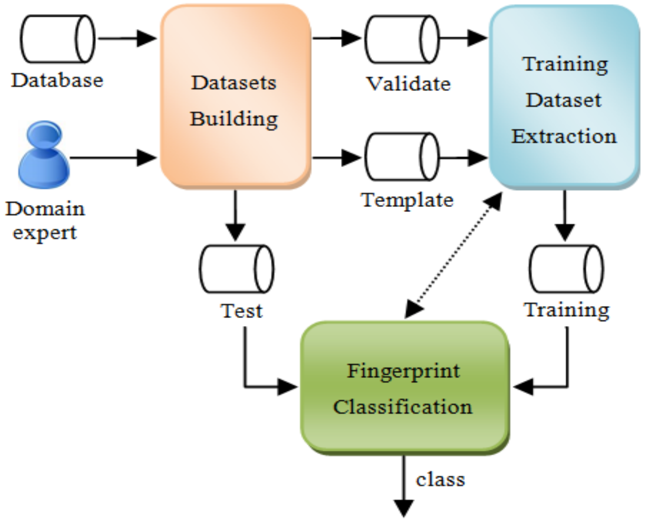


**Figure 3.20.** The innovative Fingerprint Classification System.

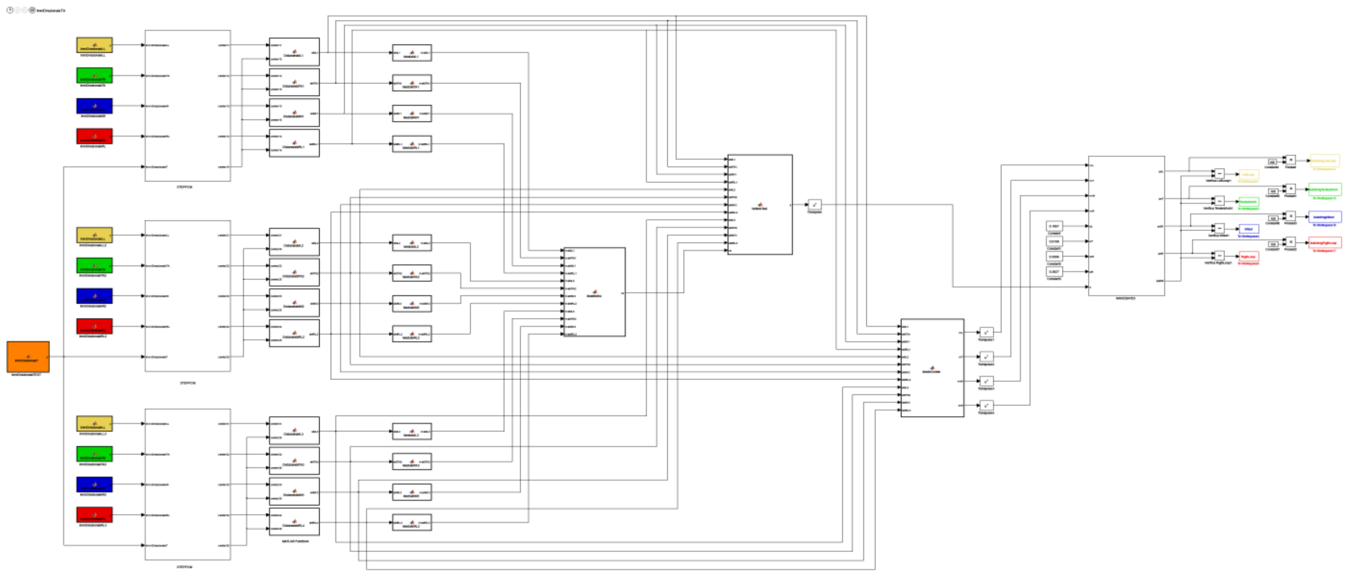


**Figure 3.21.** The proposed Fingerprint Classification System architecture.

The following subsections describe the three modules.

### 3.4.1 Datasets Building Module

This module plays an important role because it requires the contribution of a domain expert to choose the Template dataset from which to extract the best training set. It is composed of three sub-modules following described (Figure 3.21).

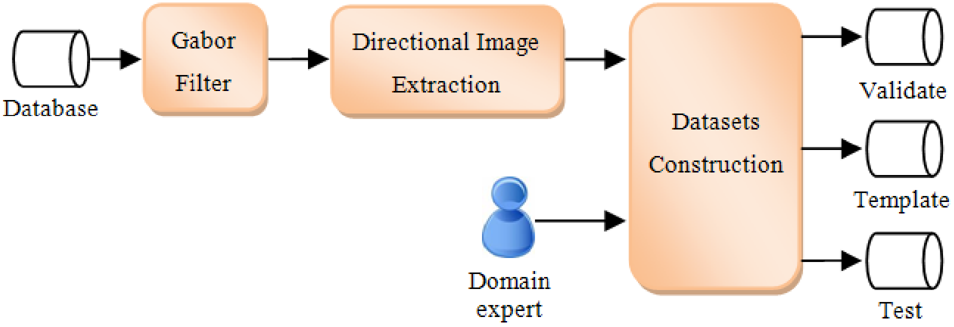


**Figure 3.21.** The proposed Datasets Building Module

***Gabor Filter sub-module***

It is applied to all images of the used database to enhance their quality.

***Directional Image Extraction sub-module***

It is the Directional Image Calculator sub-module of the AFAS system previously described.

***Datasets Construction sub-module***

It builds three different datasets: the 150 images Template dataset requires a domain expert, since it is hand selected; the 100 images Validate dataset is randomly selected, following the common distribution in nature of fingerprint classes, and then it is hand divided into the considered four classes by the domain expert; the Test dataset consists of the remaining images of the original database.

### 3.4.2 Training Dataset Extraction Module

It is composed of two sub-modules (Figure 3.23) and it works in cooperative way with the Fingerprint Classification Module. The following subsections will describe the involved sub-modules.

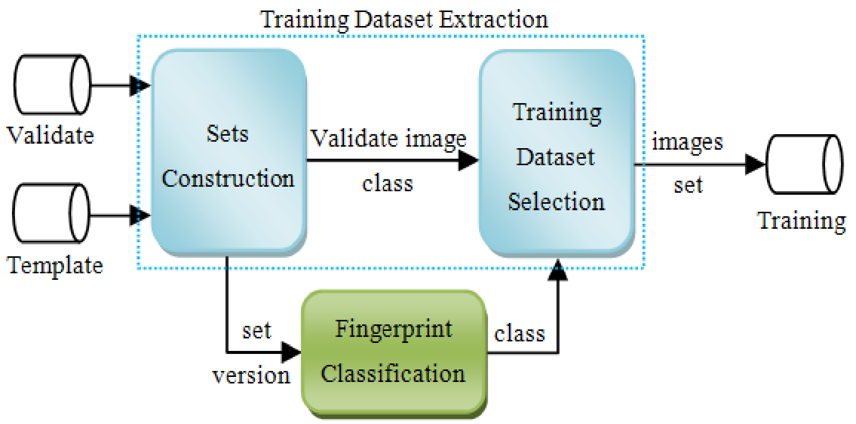


**Figure 3.23.** The proposed Training Dataset Extraction Module.

***Sets Construction sub-module***

From the four clusters obtained applying the Fuzzy C-Means clustering method to the Template dataset, it builds 250 collections of 18 randomly selected images per cluster: 12 images near the cluster center and 6 images near the boundary. With more details, every boundary is identified calculating the Euclidean semi-distance among each cluster centers pair. Successively, for each collection, it builds 200 different sets, each of one composed of 3 groups of 6 randomly selected images per cluster (Figure 3.24). Finally, for each set, it creates 100 different set versions, adding one Validate image per group.

***Training Dataset Selection sub-module***

It stores the accuracy rate of each set. Successively, it selects the one with the highest value over a threshold, experimentally fixed to 80%. The threshold is used to fix the number of items of the Validate sets and collections.

## 3.4.3 Fingerprint Classification Module

It is composed of five sub-modules (Figure 3.25). The following subsections will describe the involved sub-modules.

***Fuzzy C-Means sub-module***

It is composed of three components, each processing one group. Each component calculates five centroids: one centroid for the Test image and four centroids for the Training images (applying the average function on the elements of the same cluster).

Fuzzy C-Means is an iterative algorithm and its purpose is to find cluster centers to minimize the objective function described by the formula (3.11):

$$J_{FCM} = \sum_{i=1}^{C} \sum_{k=1}^{N} u_{ik}^{p} \, \|x_k - v_i\|^2 \qquad (3.11)$$

where p = [1, ∞), the constant that determines the fuzziness degree of the classification process, has been experimentally fixed to 6. The algorithm stop condition is described by the relation (3.12), where $\epsilon$ has been experimentally fixed to 0.1.

$$\left| J_{FCM}\left[i^{th} \text{ iteration}\right] - J_{FCM}\left[(i-1)^{th} \text{ iteration}\right] \right| < \epsilon \qquad (3.12)$$

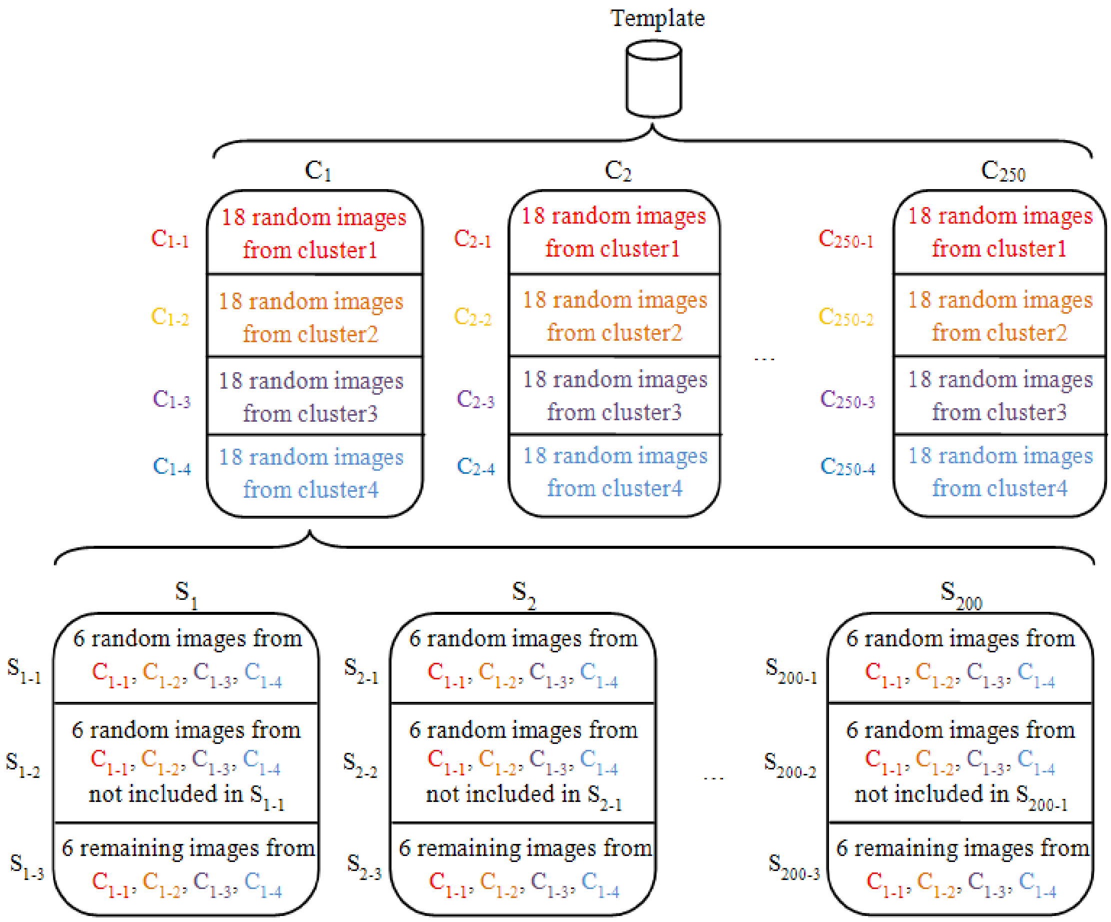

Template
C1
C2
C250
C1-1
18 random images from cluster1
C1-2
18 random images from cluster2
C1-3
18 random images from cluster3
C1-4
18 random images from cluster4
C2-1
C2-2
C2-3
C2-4
C250-1
C250-2
C250-3
C250-4
...
S1
S2
S200
S1-1
6 random images from C1-1, C1-2, C1-3, C1-4
S1-2
6 random images from C1-1, C1-2, C1-3, C1-4 not included in S1-1
S1-3
6 remaining images from C1-1, C1-2, C1-3, C1-4
S2-1
S2-2
not included in S2-1
S2-3
S200-1
S200-2
not included in S200-1
S200-3
...

**Figure 3.24.** The proposed set construction approach. The different colors represent the 4 clusters.

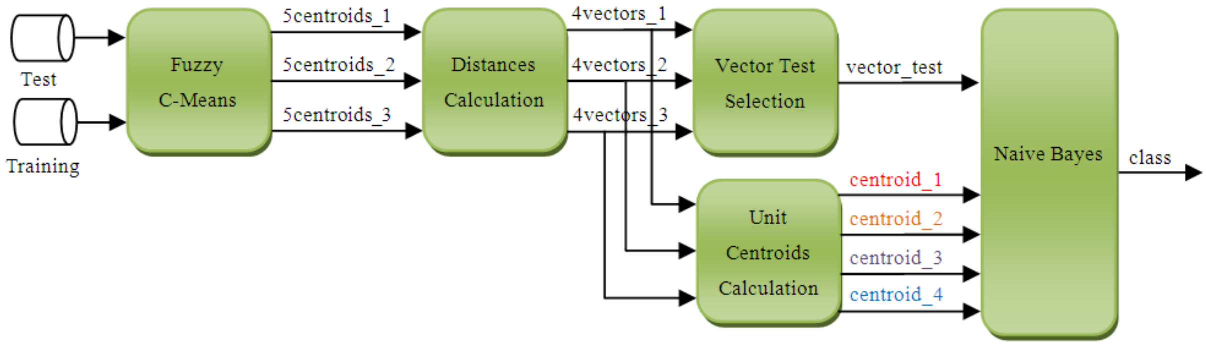


**Figure 3.25.** The proposed Fingerprint Classification Module.

### *Distances Calculation sub-module*

It calculates for each group the distances, element by element, between the centroid of the Test image and the four centroids of the Training images. By comparing such distances with a threshold, experimentally fixed to 0.1, it creates 4 binary vectors, for each group (Figure 3.26).

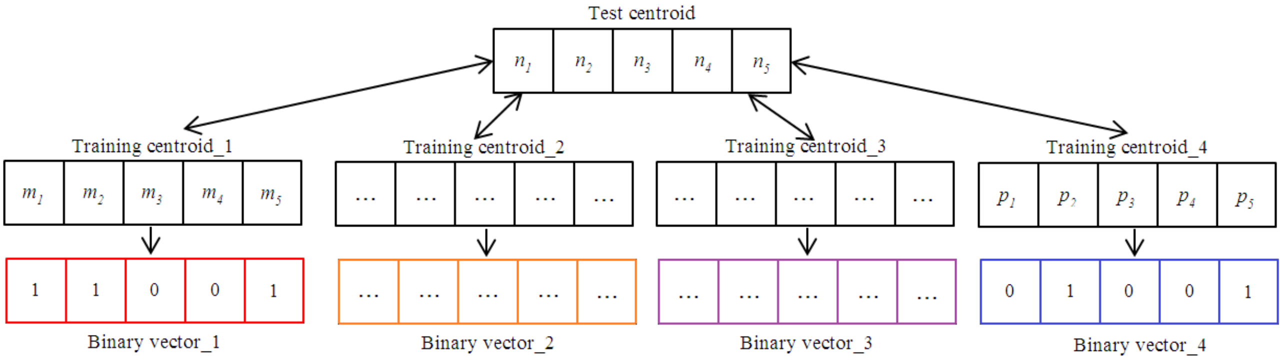


**Figure 3.26.** The 4 colored vectors represent the binary vectors. The cells of each vector of black color (Test centroid vector and 4 training centroid vectors) contain the centroid coordinates.

### *Vector Test Selection sub-module*

It analyzes the 12 binary vectors and identifies the best one representing the Test image. It chooses the first among those vectors containing more 1 than 0 values.

### *Unit Centroids Calculation sub-module*

It reorganizes the 12 binary vectors in 4 units, so that each unit is composed of the 3 vectors of the same cluster. Successively, it calculates 4 centroids, one per unit, computing the average of the respective elements (Figure 3.27).

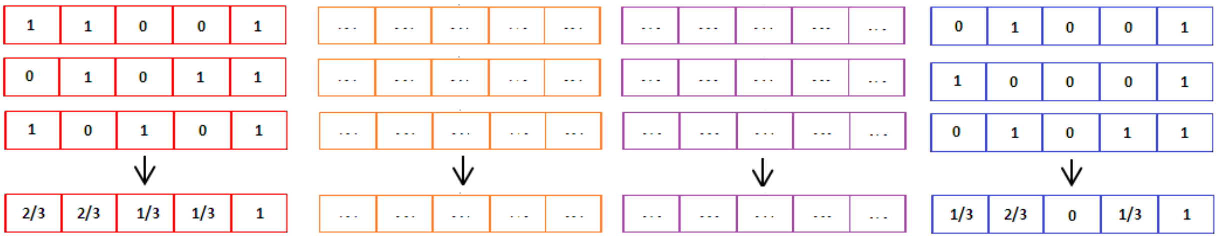


**Figure 3.27.** The 4 unit centroids obtained applying the average function to the 3 vectors of the same cluster (a different color for each cluster)

### *Naive Bayes Classifier sub-module*

It classifies the Test image using the 5 vectors, obtained by the two previous sub-modules. A Naive Bayes Classifier is a simple probabilistic classifier based on the Bayes theorem with strong independence assumptions [74]. It assumes that the domain variables are independent, given the class, and each variable has a finite number of values. Usually, the model parameters (e.g., prior class probabilities and feature probability distributions) are approximated with the relative frequencies from the training database. In the proposed work, the prior class probabilities of the used NIST classes have been experimentally fixed as in Table 3.8, following the common distribution in nature of fingerprint classes.

**Table 3.8**. Prior class probabilities of the used NIST classes.

| NIST class | Value |
|---|---|
| Tented Arch | 0.07 |
| Left Loop | 0.20 |
| Right Loop | 0.25 |
| Plain Loop/Central Pocket Loop | 0.48 |

### 3.4.4 Experimental Results and Comparison

To test the effectiveness of the proposed approach the free downloadable database PolyU [57] has been used. It contains 1480 fingerprint images belonging to the following NIST classes: Left Loop, Right Loop, Arches (Plain and Tented) and Whorl (Plain Loop, Central Pocket Loop, Accidental Loop and Double Loop) [19]. However, in the proposed work, a consistent PolyU subset of 1185 images, containing the Left Loop, Right Loop, Tented Arch, Plain Loop and Central Pocket Loop images, has been used. The obtained accuracy rate is 87.59 %, the classification time is 0.38 msec. (with a 50 MHz working frequency) and used resources on the Agility RC2000 development board are detailed in Table 3.9.

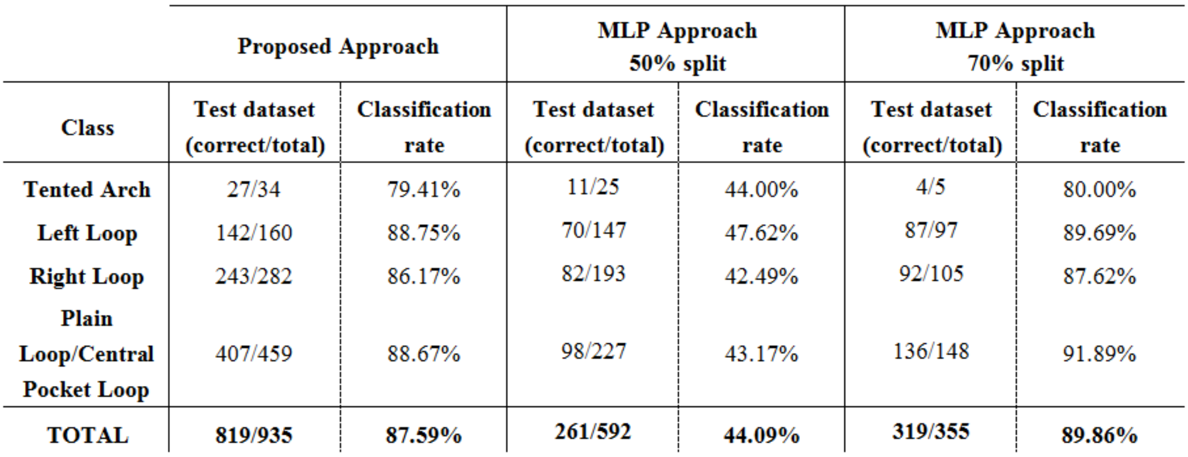
**Table 3.9.** Used resources.

| Resource Type | Percentage |
|---|---|
| Slices | 94% |
| Multiplier Blocks | 17% |
| RAM Blocks | 4% |
| IOBs | 14% |

Since in the literature no classification systems has been tested using the PolyU database, we have performed a comparison (reported in Table 3.10) between the proposed approach and a standard

Multilayer Perceptron (MLP) approach (1200 input, 50 hidden, 4 output), in terms of classification rate. The used Training and Test sets are described in Table 3.11.

**Table 3.10.** The comparison between the proposed approach and a standard Multilayer Perceptron (MLP) approach, using the PolyU database.



| | Proposed Approach | | MLP Approach<br>50% split | | MLP Approach<br>70% split | |
|---|---|---|---|---|---|---|
| **Class** | **Test dataset (correct/total)** | **Classification rate** | **Test dataset (correct/total)** | **Classification rate** | **Test dataset (correct/total)** | **Classification rate** |
| **Tented Arch** | 27/34 | 79.41% | 11/25 | 44.00% | 4/5 | 80.00% |
| **Left Loop** | 142/160 | 88.75% | 70/147 | 47.62% | 87/97 | 89.69% |
| **Right Loop** | 243/282 | 86.17% | 82/193 | 42.49% | 92/105 | 87.62% |
| **Plain Loop/Central Pocket Loop** | 407/459 | 88.67% | 98/227 | 43.17% | 136/148 | 91.89% |
| **TOTAL** | **819/935** | **87.59%** | **261/592** | **44.09%** | **319/355** | **89.86%** |

**Table 3.11.** Training and Test sets description.

| | **Proposed Approach** | **MLP Approach**<br>**50% split** | **MLP Approach**<br>**70% split** |
|---|---|---|---|
| **Training set** | 250 images (21%) [1] | 593 images | 830 |
| **Test set** | 935 images (79%) | 592 images | 355 |

[1] 150 and 100 images are used as Template dataset and Validate dataset, respectively

# CONCLUSIONS

In this thesis a novel embedded sensor has been proposed. It is composed of an Authentication system and a Classification system.

The proposed embedded AFAS improves the performance of a standard AFAS, in terms of both used resources and execution time. It is focused on the raw image quality evaluation of the acquired fingerprint, identifying areas of poor quality. It is designed to find a measure to characterize the quality of raw fingerprint images, using only the information obtained in the acquisition step. In addition, an advanced matching technique for user recognition using partial fingerprints has been developed to increase system accuracy. The best achieved FAR and FRR indexes are 0% and 6.25%, respectively. The required elaboration time is 62.6 msec. with a working frequency of 50 MHz. The proposed prototype has been implemented on the Agility RC2000 development board, addressing interesting characteristics for security in mobile device applications and enabling its use in commercial, banking and government scenarios.

The proposed Classification system uses a heuristic approach to optimize the training phase in fingerprint classification tasks. The approach combines the classification properties of a Fuzzy C-Means clustering method and a Naive Bayesian Classifier on directional image information. Unlike literature approaches using a lot of training examples, the proposed approach requires only 18 directional images per class. Experimental results, conducted on a consistent subset of the free PolyU database, show a classification rate of 87.59%. The architectural implementation on FPGA, considering its working frequency (50 MHz), achieves the performance of the highly competitive systems, realizing a good trade-off between accuracy rate, used resources and execution time. Two main issues of this system are currently works in progress: 1) the study and implementation of an automatic technique for the training dataset extraction and 2) a test phase with other different dataset.

All the innovative approaches presented in this thesis have been published in conferences and international journals.

# REFERENCES

1. P. Canali, S. Ciampoli, G. D'Ammassa, D. Meoli, A. Stefani, "Le Tecniche Biometriche", 2004.
2. A. K. Jain, L. Hong, S. Pankanti, "Biometric Identification", Communications of the ACM, February 2000.
3. "A Performance Evaluation of Biometric Identification Devices, Technical Report", SAND91-0276, Sandia National Laboratories, Albuquerque, NM and Livermore.
4. V. Conti, G. Pilato, S. Vitabile, F. Sorbello, "Verification of Ink-on-paper Fingerprints by Using Image Processing Techniques and a New Matching Operator", AI*IA Siena 10-13 September 2002, pp. 594 – 601.
5. J. K. Anil: "A Multichannel Approach to FingerPrints Classification". IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol.21, n.4 (1999) 348-358.
6. D. R. Sidlauskas, "3D hand profile identification apparatus", U.S. Patent No. 4736203, 1988.
7. A. Pentland, T. Choudhury. "Face recognition for smart environments", IEEE computer, special issue, February 2000.
8. F. K. Prokoski, "Disguise detection and identification using infrared imagery". In the Proceedings of SPIE, Optics, and Images in Law Enforcement II. A.S. Hecht, Ed. (Arlington, VA, May, 1982), 27–31.
9. M. Negin, T. Camus, "An iris biometric system for public and personal use", IEEE computer, special issue, February 2000.
10. B. Fang, Y. Yan Tang: "Elastic registration for retinal images based on reconstructed vascular trees", IEEE Transactions on Biomedical Engineering, Volume 53, Issue 6, June 2006 Page(s):1183-1187.
11. V. Nalwa, "Automatic on-line signature verification". In Proceedings of the IEEE 85, 2 (1997), 213–239.
12. S. Furui, "Recent advances in speaker recognition". Pattern Recognition Letters 18 (1997) 859–872.
13. UK Biometrics Working Group (BWG): "Biometrics Security Concerns" (2003).
14. P. Ambalakat, "Security of Biometric Authentication Systems", 21st Computer Science Seminar. SA1-T1-1. Page 2, www.rh.edu/~rhb/cs_seminar_2005/SessionA1/ambalakat.pdf

15. V. Conti, G. Pilato, S. Vitabile, F. Sorbello, “A Robust System for Fingerprints Identification”, Knowledge-Based Intelligent Information Engineering System and Allied Technologies, Crema 16-18 September 2002, pp. 1162-1166.
16. Y. Chen and A. K. Jain, “Dots and incipients: extended features for partial fingerprint matching” proc. of Biometric Symposium, 2007, pp. 1-6, ISBN: 978-1-4244-1549-6, DOI 10.1109/BCC.2007.4430538
17. A. K. Jain, “On-Line Fingerprint Verification”, IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol.19, n. 4 (1997) 302-314.
18. S. Prabhakar, A. K. Jain, W. Jianguo, “Minutiae Verification and Classification”, Department of Computer Engineering and Science, University of Michigan State, East Lansing (1998) MI 48824.
19. National Institute of Standards and Technology: www.nist.gov
20. A. M. Bazen, G. T. B. Verwaaijen, S. H. Gerez, L. P. J. Veelenturf, B. J. van der Zwaag, “A Correlation-Based Fingerprint Verification System”, ProRISC 2000, Workshop on Circuits, Systems and Signal Processing, Veldhoven, The Netherlands, November 2000.
21. D. Maltoni, D. Maio, A. K. Jain, S. Prabhakar, “Handbook of Fingerprint Recognition”. Springer (New York),, 2003, ISBN: 0-387-95431-7.
22. K. Ito, H. Nakajima, K. Kobayashi, T. Aoki, T. Higuchi, “A Fingerprint Matching Algorithm Using Phase-Only Correlation”, IEICE Trans, Fundamentals, Vol. E87-A, No. 3, March 2004.
23. B. M. Mehtre, B. Chatterjee, “Segmentation of fingerprint images - a composite method”, Pattern Recognition, Volume 22, Issue 4 (1989), Pages: 381 – 385, ISSN:0031-3203
24. B. M. Mehtre, N. N. Murthy, S. Kapoor, B. Chatterjee, “Segmentation of fingerprint images using the directional image”, Pattern Recognition, Volume 20, Issue 4 (1987), Pages: 429 – 435, ISSN:0031-3203
25. A. R. Rao, B. G. Schunk, “Computing oriented texture fields” - Computer Vision and Pattern Recognition, Proceedings CVPR '89, IEEE Computer Society Conference on Digital Object Identifier.
26. Jingjing Wang, Xiaoyan Sun, “Fingerprint Image Enhancement using a fast Gabor filter“, proc. of the 8th IEEE World Congress on Intelligent Control and Automation (WCICA), 2010, pp. 6347 – 6350, DOI: 10.1109/WCICA.2010.5554350, ISBN 978-1-4244-6712-9
27. Manhua Liu ; Xudong Jiang ; Kot, A.C., “ Nonlinear Fingerprint Orientation Smoothing by Median Filter“, proc. of 5th IEEE International Conference on Information, Communications and Signal Processing, 2005, pp. 1439 – 1443, DOI: 10.1109/ICICS.2005.1689296, ISBN 0-7803-9283-3

28. Wang Yao, Yu Jian-ping, Liu Hong-wie, Zhang Peng, “Fingerprint Image Enhancement Based on Morphological Filter“, proc. of IEEE International Conference on Computational and Information Sciences (ICCIS), 2011, pp. 34 – 37, DOI: 10.1109/ICCIS.2011.152, ISBN 978-1-4577-1540-2
29. T. Y. Zhang, C. Y. Suen: “A fast parallel algorithm for thinning digital patterns”, Communications of the ACM., vol. 27, issue 3, pp. 236-239, 1984.
30. V. Conti, C. Militello, S. Vitabile and F. Sorbello, “Introducing Pseudo-Singularity Points for Efficient Fingerprints Classification and Recognition”, proc. of the 4th IEEE International Conference on Complex, Intelligent and Software Intensive Systems (CISIS), 2010, pp. 368-375, ISBN: 978-0-7695-3967-6, DOI: 10.1109/CISIS.2010.134
31. H.-W. Jung and J.-H. Lee, “Fingerprint Classification Using the Stochastic Approach of Ridge Direction Information”, proc. of the IEEE International Conference on Fuzzy Systems, 2009, pp. 169-174, ISSN: 1098-7584, DOI: 10.1109/FUZZY.2009.5277309
32. A. Senior, “A Combination Fingerprint Classifier”, IEEE Transaction on Pattern Analysis and Machine Intelligence, 2001, Vol. 23, No. 10, pp. 1165-1174
33. **G. Vitello**, V. Conti, A. Gentile, S. Vitabile, F. Sorbello, “Design and Implementation of an Efficient Fingerprint Features Extractor”, proc. of 17th Euromicro Conference on Digital Systems Design (DSD), 2014, pp. 695-699, DOI: 10.1109/DSD.2014.101
34. M. Ballan, F. Ayhan Sakarya, and B. L. Evans “A Fingerprint Classification Technique Using Directional images”, proc. of the Thirty-First Asilomar Conference on Signal, System and Computer, 1997, Vol. 1, pp. 101-104, 1997
35. S. M. Mohamed and H. O. Nyongesa, “Automatic Fingerprint Classification System Using Fuzzy Neural Techniques”, proc. of the IEEE International Conference on Fuzzy System, 2002, Vol. 1, pp. 358-362
36. K. Rao, K. Balck, “Type Classification of Fingerprints: A Syntactic Approach”, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 2, no. 3, pp.223-231, 1980.
37. H. Bunke, “Structural and Syntactic Pattern Recognition”, in Handbook of Pattern Recognition & Computer Vision, C.H. Chen et al. (Eds.), World Scientific, River Edge, NJ, 1993
38. D. Maltoni, D. Maio, “A structural Approach To Fingerprint Classification” proc. of the 13th International Conference on Pattern Recognition, ICPR’96, Vol. 3, pp. 578-585, 25-29 August 1996
39. H. V. Neto, D. L. Borges, “Fingerprint Classification with Neural Networks”, proc. of 4th IEEE Brazilian Symposium on Neural Networks, pp. 66–72, 3-5 December 1997.

40. S. R. Patil and S. R. Suralkar, “Fingerprint Classification using Artificial Neural Network”, International Journal of Emerging Technology and Advanced Engineering, 2012, Vol. 2, Issue 10, pp. 513-517, ISSN: 2250-2459
41. M. Kamijo, “Classifying Fingerprint Images Using Neural Network: Deriving The Classification State”, IEEE International Conference on neural networks, 1993, Vol. 3, pp. 1932-1937, 28 March-1April 1993
42. V. Conti, C. Militello, S. Vitabile, F. Sorbello, “An Embedded Fingerprints Classification System based on Weightless Neural Networks”, in New Direction in Neural Networks, 2008, IOS Press, pp. 67-75, ISBN/ISSN: 978-1-58603-984-4, DOI: 10.3233/978-1-58603-984-4-67.
43. A. K. Jain, S. Prabhakar, L. Hong, “A Multichannel Approach to Fingerprint Classification”, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 21, no. 4, pp. 348-359,1999.
44. J. Bowen, “The Home Office Automatic Fingerprint Pattern Classification Project”, in Proc. IEE Colloquium on Neural Networks for Image processing Applications, 1992.
45. Apple Inc., website: http://support.apple.com/kb/HT5883
46. Samsung, website: http://www.samsung.com/it/consumer/mobile-devices/smartphones/smartphones/ SM-G900FZKAITV-spec
47. V. Conti, C. Militello, F. Sorbello, S. Vitabile, “A Multimodal Technique for an Embedded Fingerprint Recognizer in Mobile Payment Systems”, International Journal of Mobile Information Systems (IJ-MIS), vol. 5, No. 2, pp. 105-124, ISSN: 1574-017X, DOI: 10.3233/MIS-2009-0076.
48. D. Batra, G. Singhal, S. Chaudhury, “Gabor Filter based Fingerprint Classification using Support Vector Machines”, proc. of the 1st IEEE India Annual Conference (INDICON), 2004, pp. 256-261, DOI: 10.1109/INDICO.2004.1497751.
49. J. Hu, M. Xie, “Fingerprint classification based on genetic programming”, proc. of 2nd International Conference on Computer Engineering and Technology (ICCET), 2010, vol. 6, pp. 193-196, DOI: 10.1109/ICCET.2010.5486315.
50. A. Tariq, M.U. Akram, S.A. Khan, “An automated system for fingerprint classification using singular points for biometric security”, proc. of International Conference for Internet Technology and Secured Transactions (ICITST), 2011, pp. 170-175.
51. V. Conti, C. Militello, F. Sorbello, S. Vitabile. “A Frequency-based Approach for Features Fusion in Fingerprint and Iris Multimodal Biometric Identification Systems”, IEEE Transactions on Systems, Man, and Cybernetics (SMC) Part C: Applications & Reviews, 2010, vol. 40, N°4, p. 384-395, ISSN: 1094-6977, DOI:10.1109/TSMCC.2010.2045374.

52. J. Qi, D. Abdurrachim, D. Li, H. Kunieda, “A Hybrid Method for Fingerprint Image Quality Calculation”, proc. of 4th IEEE Workshop on Automatic Identification Advanced Technologies, 2005, pp. 124-129, ISBN: 0-7695-2475-3, DOI: 10.1109/AUTOID.2005.3.
53. V. Conti, **G. Vitello**, F. Sorbello, S. Vitabile, “An Advanced Technique for User Identification using Partial Fingerprint”, proc. of 7th International IEEE Conference on Complex, Intelligent and Software Intensive Systems (CISIS) 2013, pp. 236-242, ISBN: 978-0-7695-4992-7, DOI: 10.1109/CISIS.2013.46.
54. Xilinx Inc., website http://www.xilinx.com/support/documentation/data_sheets/ds031.pdf.
55. V. Conti, S. Vitabile, **G. Vitello**, F. Sorbello, “An Embedded Biometric Sensor for Ubiquitous Authentication”, proc. of AEIT Annual Conference 2013 - Innovation and Scientific and Technical Culture for Development, ISBN: 9788887237337.
56. FVC Databases, website http://bias.csr.unibo.it/fvc2002/databases.asp.
57. PolyU Database, website http://www.comp.polyu.edu.hk/~biometrics/HRF/HRF.htm.
58. E. Tabassi, C. Wilson, C. Watson, “Fingerprint image quality”, NIST. Res. Rep. NISTIR7151, 2004.
59. E. Lim, X.D. Jiang, W.Y. Yau, “Fingerprint Quality and Validity Analysis”, proc. of International IEEE Conference on Image Processing, 2002, pp. 469-472, vol. 1, ISSN: 1522-4880, DOI: 10.1109/ICIP.2002.1038062.
60. X. Yang, Y. Luo, “A classification method of fingerprint quality based on neural network”, proc. of International IEEE Conference on Multimedia Technology (ICMT), 2011, pp. 20-23, ISBN: 978-1-61284-771-9, DOI: 10.1109/ICMT.2011.6001832.
61. F.J. Ani, X.P. Cheng, “Approach for Estimating the Quality of Fingerprint Image based on the Character of Ridge and Valley Lines”, proc. of International IEEE Conference on Wavelet Active Media Technology and Information Processing (ICWAMTIP), 2012, pp. 113-116, ISBN: 978-1-4673-1684-2, DOI: 10.1109/ICWAMTIP.2012.6413452.
62. S. Lee, H. Choi, K. Choi, J. Kim, “Fingerprint-Quality Index Using Gradient Components”, IEEE Transactions on Information Forensics and Security, vol. 3, No. 4, pp. 792-800, 2008, ISSN: 1556-6013, DOI: 10.1109/TIFS.2008.2007245.
63. M. Vatsa, R. Singh, A. Noore, M.M Houck, “Quality-augmented fusion of level-2 and level-3 fingerprint information using DSm theory”, International Journal of Approximate Reasoning, vol. 50, issue 1, 2009, pp. 51–61.
64. M. Vatsa, R. Singh, A. Noore, S.K. Singh, “Quality Induced Fingerprint Identification using Extended Feature Set”, proc. of 2nd IEEE International Conference on Biometrics: Theory,

Applications and Systems, 2008, pp. 1-6, ISBN: 978-1-4244-2729-1, DOI: 10.1109/BTAS.2008.4699327.

65. J. Bernsen, “Dynamic thresholding of gray-level images”, proc. of 8th International Conference on Pattern Recognition, Paris, 1986, pp. 1251–1255.
66. NSCT – Fingerprint Recognition, website http://www.biometrics.gov/documents/fingerprintrec.pdf.
67. D. Maio, D. Maltoni, R. Cappelli, J.L. Wayman, A.K. Jain, “FVC2002: the Second International Competition for Fingerprint Verification Algorithms”, website http://bias.csr.unibo.it/fvc2002/
68. Biometrika FX2000, website http://www.biometrika.it/eng/fx2000.html.
69. V. Bonato, R.F. Molz, J.C. Furtado, M.F. Ferrão, F.G. Moraes, “Propose of a hardware implementation for fingerprint systems”, UNISC - Departamento de Informatica Santa Cruz-Brazil, PUCRS - Faculdade de Informatica porto Alegre – Brazil.
70. **G. Vitello**, V. Conti, A. Gentile, S. Vitabile, F. Sorbello, “Design and Implementation of an Efficient Fingerprint Features Extractor”, proc. of 17th Euromicro Conference on Digital Systems Design (DSD), 2014, pp. 695-699, DOI: 10.1109/DSD.2014.101
71. H. Zhang, Y. Yin, G. Ren, “An Improved Method for Singularity Detection of Fingerprint Images”, Book Advances in Biometric Person Authentication, Springer Berlin/Heidelberg, vol. 3338/2004, pp. 516-524, ISBN: 978-3-540-24029-7.
72. N. Ratha, S. Chen, and A.K. Jain, “Adaptive Flow Orientation Based Feature Extraction in Fingerprint Images” Pattern Recognition, Vol. 28, No. 11, pp. 1,657-1,672, 1995, ISSN: 0031-3203
73. **G. Vitello**, G.I.M. Migliore, V. Conti, S. Vitabile, F. Sorbello, “A Novel Technique for Fingerprint Classification based on Fuzzy C-Means and Naive Bayes Classifier”, proc. of 8th IEEE International Conference on Complex, Intelligent and Software Intensive Systems (CISIS), 2014, pp. 155-161, DOI: 10.1109/CISIS.2014.23.
74. Yongchuan Tang, Wuming Pan, Haiming Li and Yang Xu, “Fuzzy Naive Bayes Classifier Based on Fuzzy Clustering”, proc. of IEEE International Conference on Systems, Man and Cybernetics, 2002, Vol. 5, ISSN: 1062-922X, DOI: 10.1109/ICSMC.2002.1176401
75. Militello, C.; Conti, V.; Vitabile, S.; and Sorbello, F.; “A Novel Embedded Fingerprints Authentication System based on Singularity Points”, proc. of the 2$^{nd}$ IEEE International Conference on Complex, Intelligent and Software Intensive Systems, 2008, pp. 72-78, ISBN: 0-7695-3509-1, DOI:10.1109/CISIS.2008.56.
76. L. Hong, Y. Wan, A. K. Jain, “Fingerprint Image Enhancement, Algorithm and Performance Evaluation”. IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol.20, n.8, 1998, pp. 777-789.