A Survey of Fingerprint Classification Part I: Taxonomies on Feature Extraction Methods and Learning Models

Mikel Galar[a,*], Joaquín Derrac[g], Daniel Peralta[b], Isaac Triguero[e,f,b], Daniel Paternain[a],
Carlos Lopez-Molina[a], Salvador García[b,c,d], José M. Benítez[b], Miguel Pagola[a], Edurne Barrenechea[a],
Humberto Bustince[a], Francisco Herrera[b]

[a] *Departamento de Automática y Computación, Universidad Pública de Navarra, 31006, Pamplona, Spain.*
[b] *Department of Computer Science and Artificial Intelligence, University of Granada, 18071, Granada, Spain.*
[c] *Faculty of Computing and Information Technology - North Jeddah, King Abdulaziz University, 21589, Jeddah, Saudi Arabia.*
[d] *Department of Computer Science, University of Jaén, 23071, Jaén, Spain.*
[e] *Department of Respiratory Medicine, Ghent University, 9000 Gent, Belgium.*
[f] *VIB Inflammation Research Center, 9052 Zwijnaarde, Belgium.*
[g] *Affectv: Affectv Limited, 33-34 Alfred Place, London, WC1E 7DP, United Kingdom.*

Abstract

This paper reviews the fingerprint classification literature looking at the problem from a double perspective. We first deal with feature extraction methods, including the different models considered for singular point detection and for orientation map extraction. Then, we focus on the different learning models considered to build the classifiers used to label new fingerprints. Taxonomies and classifications for the feature extraction, singular point detection, orientation extraction and learning methods are presented. A critical view of the existing literature have led us to present a discussion on the existing methods and their drawbacks such as difficulty in their reimplementation, lack of details or major differences in their evaluations procedures. On this account, an experimental analysis of the most relevant methods is carried out in the second part of this paper, and a new method based on their combination is presented.

Keywords: Fingerprint Classification, Feature Extraction, Classification, Fingerprint Recognition, SVM, Neural Networks, Ensembles, Orientation Map, Singular Points

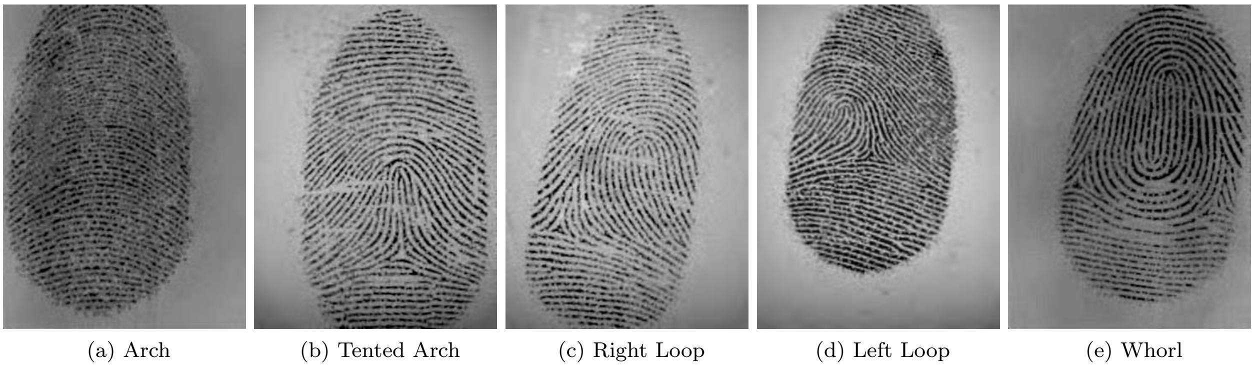
1. Introduction

Classification in Machine Learning (ML) is the problem of extracting knowledge from a set of n input examples $x_1, \ldots, x_n$ characterized by i features $a_1, \ldots, a_i \in \mathbb{A}$, including numerical or nominal values, where each instance is labeled with a desired output class label $y_j \in \mathbb{C}$ (considering a m class problem $\mathbb{C} = \{c_1, \ldots, c_m\}$) and the aim is to learn a system capable of predicting this output for a new unseen example in a reasonable way (with good generalization ability) [26]. The system generated by the learning algorithm is a mapping function defined over the patterns $\mathbb{A}^i \to \mathbb{C}$ and it is called a *classifier*.

Therefore, fingerprint classification problem consists of learning a classifier from a set of labeled fingerprints, which should be able to classify new fingerprints in the corresponding class. The most commonly used fingerprint classification model was given by Henry [37]. Most of the classification approaches reviewed

[*] Corresponding author. Tel.: +34 948 166048; Fax: +34 948 168924
Email addresses: mikel.galar@unavarra.es (Mikel Galar), jderrac@decsai.ugr.es (Joaquín Derrac), dperalta@decsai.ugr.es (Daniel Peralta), Isaac.Triguero@irc.vib-UGent.be (Isaac Triguero), daniel.paternain@unavarra.es (Daniel Paternain), carlos.lopez@unavarra.es (Carlos Lopez-Molina), sglopez@ujaen.es (Salvador García), J.M.Benitez@decsai.ugr.es (José M. Benítez), miguel.pagola@unavarra.es (Miguel Pagola), edurne.barrenechea@unavarra.es (Edurne Barrenechea), bustince@unavarra.es (Humberto Bustince), herrera@decsai.ugr.es (Francisco Herrera)

in this paper consider the five major classes shown in Figure 1: *Arch, Tented Arch, Right Loop, Left Loop* and *Whorl*. These fingerprint classes are unevenly distributed in the population (3.7%, 2.9%, 31.7%, 33.8% and 27.9%, respectively), which increases the difficulty of the classification problem from the ML point of view [33], but also makes the reduction of the search space class-dependent.



(a) Arch (b) Tented Arch (c) Right Loop (d) Left Loop (e) Whorl

Figure 1: The five major classes defined by Henry [37] considered in the fingerprint classification problem.

The fingerprint classification problem arises from the problem of fingerprint identification, which aims to claim the identity of a person by their fingerprint [77]. Unlike in the verification problem (where the aim is to check whether two fingerprints are the same), the number of matchings that need to be carried out grow along with the number of individuals in the database. Hence, a reduction of the number of comparisons is required in order to maintain the response times as short as possible. This reduction is usually quantified with the ratio of penetration [99] in the database, which measures the percentage of the database that is searched before matching the fingerprint. Classification is the most extended method to reduce the ratio of penetration, which is also directly related with the classification accuracy (percentage of correctly classified examples) obtained by the classification methods. This paper focuses on this type of methods to reduce the search space even though other techniques have also been developed such as indexing [6, 64, 14], continuous classification [15, 17, 19] or clustering and classification [47, 68, 69].

Although classification in ML typically refers to learning a classifier from a set of examples characterized by several features, fingerprint classification usually stands for the problem as a whole, including feature extraction [89]. Feature extraction (FE) consists of obtaining a set of features that are able to properly characterize an object for its posterior processing. In fingerprint classification, FE aims to describe a fingerprint as accurately as possible in order to facilitate its classification among the predefined classes. FE is a key issue, since the classification problem directly depends on the quality of the features considered.

However, from our point of view, fingerprint classification can be divided into two well-differentiated steps, which is the viewpoint considered in this paper (although some proposals [46, 104, 38, 61] consider strongly related models).

1. First, we deal with FE, that is, how to obtain a suitable representation of the fingerprint so as to achieve an accurate classification.
2. Second, we consider the classification problem as it is commonly done in ML, considering the construction of a classifier capable of classifying previously unknown fingerprints using the features extracted in FE phase.

Our aim is to review the different works proposed in the literature paying attention to both phases independently. As a result, we will put forward a taxonomy in which the different FE methods presented in the literature can be placed depending on the nature of the characteristics considered. Additionally, we will present two taxonomies in which the methods considered in fingerprint classification papers for the extraction of orientation maps and singular points can be classified (Section 3 and 4). We should emphasize that these are the two most important features for fingerprint classification. Finally, we will make an overview of the techniques that have been used to address the classification problem; we consider different groups of algorithms and their evolution in the literature will be analyzed.

The study of all these methods have led us to a thorough discussion in Section 6, where a critical view of the reviewed works is presented regarding the lack of details in their descriptions, their reimplementability and the existing differences on the way they are evaluated, among others. On this account, we aim to experimentally show this problem in the second part of this paper [34], where several relevant methods have been implemented by the authors and an exhaustive experimental evaluation is carried out in a common experimental framework. This way, their results will be objectively analyzed and their validity for their usage by other researchers will be shown.

The rest of this paper is organized as follows. Section 2 introduces the fingerprint classification problem and recalls the most important concepts on this topic. Next, Section 3 deals with the two most important processes in fingerprint classification: the extraction of orientation maps and singular points, and presents their taxonomies. Then, Section 4 puts forwards our taxonomy proposal for the classification of FE techniques and reviews the existing works in each one of the categories considered. Afterwards, Section 5 describes the different ML models that have been considered in the fingerprint classification literature. The discussion on the works reviewed along this paper is presented in Section 6, whereas Section 7 concludes this paper.

2. Background

Fingerprint features or characteristics are usually classified into three levels [77, 29]:

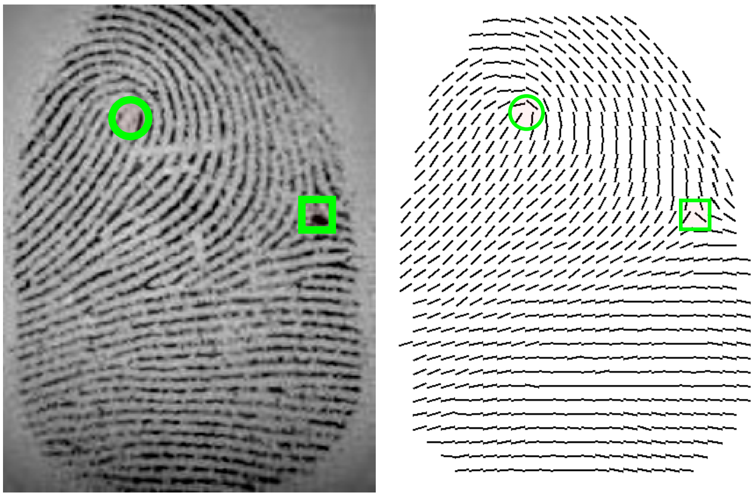
- *Level 1 (Global)* – refers to the global ridge line flow (orientations) and the features derived from it (singular points).
- *Level 2 (Local)* – considers minutiae details extracted from the ridge skeleton.
- *Level 3 (Fine-detail)* – includes intra-ridge details such as width, shape, ridge contours, sweat pores, creases, etc.

Among these levels, only the first one is used for fingerprint classification (with few exceptions [95]), since fingerprint classes are intuitively defined from global characteristics. Otherwise, level 2 and 3 features are commonly considered for fingerprint matching [48, 23, 13, 94] as they allow to claim for the individuality of a fingerprint. Therefore, FE for classification is mainly carried out with level 1 features, that is, fingerprint features for classification are closely related to fingerprint orientations and Singular Points (SPs). Fingerprint orientations are represented in an Orientation Map (OM), which is the representation of the local ridge flow in the fingerprint. SPs are defined as the locations in the fingerprint with the greatest ridge orientation variance, i.e., where the ridges vary more abruptly. There are two types of SPs known as *cores* and *deltas*. Figure 2 shows a fingerprint image (2a), its OM (2b) and its SPs are marked in both images. Notice that *Arch* class fingerprints do not have any SP.

Once the principal features of fingerprints used for classification have been presented, each fingerprint class can be briefly defined as follows.

- *Arch*: Fingerprints that have no SPs and ridges flow from one side to the other forming a small bump.
- *Tented Arch*: Fingerprints that have one core and one delta (the delta is under the core), and ridges flows similarly to Arch, but at least one ridge shows a high curvature.
- *Right Loop*: Fingerprints that have one core and one delta (the delta is under and to the left of the core), and one or more ridges flows from the right side curve back and go out from the same side.
- *Left Loop*: Fingerprints that have one core and one delta (the delta is under and to the right of the core), and one or more ridges flows from the left side curve back and go out from the same side.
- *Whorl*: Fingerprints that have two cores and two deltas, and at least one of its ridges makes a full turn around the center of the fingerprint.

In addition to fingerprint classification there are other options to reduce penetration rate. Fingerprint indexing using minutiae has been proposed by several authors [6, 64, 14], where fingerprints in the database



(a) Original fingerprint (b) Orientation map

Figure 2: Two global characteristics of the fingerprints: OM and SPs. The core point is marked with a circle, whereas the delta point is highlighted with a square.

are ordered depending on their similarity with the input one observing different minutiae features. This approach has the disadvantage of being highly correlated with the posterior matching process, which generally considers minutiae. Continuous classification [15, 17, 19] has several similarities with exclusive classification and fingerprint indexing. In this case global features (such as orientations) are used instead of minutiae, but the way in which the penetration rate is reduced is the same as in indexing, where fingerprints are sorted by their distance with respect to the input one. Finally, methods based on clustering and classification [47, 68, 69] also perform an exclusive classification, but in this case the classes are created by a clustering algorithm, which need not agree with Henry ones. Consequently, the difficulty appears at the clustering phase, where the number of generated classes has to be decided. We should notice that some FE methods have been used equally for continuous and exclusive classification [15, 17, 19, 61, 60], and hence they are considered in this work. Anyway, this review is focused on exclusive classification approaches, leaving out of the scope of this work those only considering continuous and clustering-based classification.

Table 1 lists in a compact manner the most relevant works for fingerprint classification, sorted in chronological order of appearance. For each of them, we include the most relevant data for fingerprint classification, *i.e.* the features used for classification, the OM extraction model, SP detection method (or reference point, see Section 3), classification technique considered, databases used for testing and the corresponding reference. Each approach is termed by the name of its first author, to which the publication year has been appended if necessary.

3. Feature extraction techniques: singular points and orientation map

Before analyzing the FE methods to generate the feature vectors for classification, we review the two most important phases in any FE algorithm for fingerprint classification: OM extraction and SP detection. Both phases are present in most of the approaches as it can be seen in Table 1. Even though the final features used for classification are neither orientations nor SPs, they are used in previous processing steps as we explain in Subsections 3.1 and 3.2, respectively.

3.1. Extraction of the orientation map

The extraction of the OM consists of obtaining a representation of the local ridge flow for every subregion (commonly named block) in the fingerprint. Figure 2b shows an example of an OM. Given an image of $N \times M$

pixels and considering orientation blocks of $n \times m$, an OM is a matrix of $N/n \times M/m$, which usually stores the angles in radians in the range $[0, \pi)$ or $[-\pi/2, \pi/2)$. The size of the block should be enough as to be able to obtain a good estimate of the local ridge flow, but it cannot be too large since the changes in the local orientations need to be captured in order to properly locate the global characteristics of the fingerprints.

The importance of the OM extraction comes from the fact that it is used for almost any processing that is carried out with the fingerprint. For example, SP detection methods analyze the behavior of the orientations [3, 88]; similarly, ridge structure can be obtained from the OM [101, 102]. Hence, this phase is always the first step in any fingerprint classification system. In the case of fingerprint matching, accurately estimating the orientation of the minutiae is a key issue. Otherwise, in classification the OM does not require the same accuracy in terms of individual errors in the estimated degrees, but focuses on being accurate in terms of global structure, which is used to classify the fingerprint. For this reason, the OM extraction methods used for fingerprint classification are usually simpler than those dedicated to matching (see [108, 41] for dedicated revisions).

In order to classify the different OM extraction methods, we propose the taxonomy shown in Figure 3, which is explained hereafter. Each proposal is classified to its corresponding category, and the size of the orientation block used by the authors is provided. Those proposals that used more than one OM extraction method from different categories are pointed out by a black circle.

Table 1: List of the most relevant fingerprint classification works (1975-2015).

Name	Year	Features	Orientations	Singular points	Reference point	Classification technique	Test databases	Reference
Moayer75	1975	Orientations (Syntactic-coding)	Skeleton tracing (12 × 12)			Syntactic	Other	[82]
Moayer76	1976	Orientations (Syntactic-coding)	Skeleton tracing (12 × 12)			Syntactic	Other	[83]
Rao76	1976	Ridge structure (Ridge-tracing (generic))	N/A			Fixed Classification	Other	[98]
Rao80	1980	Orientations (Syntactic-coding)	Diferentiation operator (32 × 32)			Syntactic	Other	[97]
Kawagoe	1984	Ridge structure (Ridge-tracing (from SPs))	Micro-patterns (16 × 16)	Poincarè Label-structured		Fixed Classification	Other	[57]
Hughes	1991	Orientations (Direct/unaltered)	Micro-patterns (64 × 64)			Neural Networks	Other	[43]
Wilson92	1992	Orientations (Registered)	Slits sum (16 × 16)		Rule-based (R92)	Neural Networks	Other	[122]
Bowen	1992	Orientations (Direct/unaltered) Singular Points (Vorticity map)	W/O details (N/A)	Vorticity map		Neural Networks	Other	[9]
Kamijo	1993	Ridge structure (Ridge-tracing (generic))	W/O details (16 × 16)			Neural Networks	Other	[54]
Moscinska	1993	Orientations (Direct/unaltered)	Skeleton tracing (32 × 32)			Neural Networks	Other	[85]
Candela	1995	Orientations (Registered) Ridge structure (Ridge-tracing (Generic))	Slits sum (16 × 16)		Rule-based (R92)	Neural Networks	NIST-14	[10]
Wilson95	1995	Orientations (Registered)	Slits sum (16 × 16)		Rule-based (R92)	Neural Networks	NIST-4	[121]
Karu	1996	Singular points (Number/positions)	Slits sum (8 × 8)	Poincarè		Fixed Classification	NIST-4/9	[55]
Halici	1996	Orientations (Registered)	Slits sum (16 × 16)			Neural Networks	Other	[36]
Maio	1996	Orientations (Segmentation)	Level-curves (32 × 32)			Graph Matching	Other	[73]
Geng	1997	Orientations (Direct/unaltered)	W/O details (24 × 24)			Neural Networks	NIST-4	[35]
Senior97	1997	Ridge structure (Skeleton)	Gradient (16 × 16)			Structural Models	NIST-4	[101]
Ballan	1997	Singular points (Number/positions)	Gray-level consistency/variance (5 × 5)	Directional histograms		Fixed Classification	NIST-4	[2]
Chong	1997	Ridge structure (Geometric/global)	W/O orientations			Fixed Classification	Other	[25]
Sarbadhikari	1998	Filter responses (FFT)	W/O orientations			Neural Networks	Other	[100]
Qi	1998	Orientations (Registered)	Slits sum (16 × 16)		Rule-based (R92)	Neural Networks	NIST-14	[120]
Cappelli99a	1999	Orientations (Segmentation)	Slits sum (16 × 16)			Graph Matching	NIST-4/14	[15]
Hong99	1999	Singular points (Number/positions) Ridge structure (Skeleton)	Gradient (16 × 16)	Poincarè		Fixed Classification	NIST-4	[39]
Jain	1999	Filter responses (Gabor)	Gradient (16 × 16)	Poincarè	Covariances	Multiple Techniques	NIST-4	[46]
Cappelli99b	1999	Orientations (Registered)	Slits sum (16 × 16)	Poincarè	Rule-based (R92)	Nearest Neighbor	NIST-4	[17]
Cho	2000	Singular points (Number/positions)	Gray-level consistency/variance (8 × 8)	Poincarè		Fixed Classification	Other	[24]
Cappelli00	2000	Orientations (Registered) Orientations (Segmentation)	Slits sum (16 × 16)	Poincarè	Rule-based (R92)	Nearest Neighbor	NIST-14	[18]
Marcialis	2001	Orientations (Segmentation) Filter responses (Gabor)	Slits sum (16 × 16) Gradient (16 × 16)	Poincarè	Covariances	Multiple Techniques	NIST-4	[78]
Yao01	2001	Filter responses (Gabor)	Gradient (16 × 16)	Poincarè	Covariances	Support Vector Machines	NIST-4	[123]

Continued on next page...

Table 1: (continued).

Name	Year	Features	Orientations	Singular points	Reference point	Classification technique	Test databases	Reference
Senior01	2001	Orientations (Registered) Ridge structure (Ridge-tracing (generic)) Ridge structure (Skeleton)	Slits sum (16 × 16) Gradient (16 × 16)		Rule-based (R92)	Multiple Techniques	NIST-4	[102]
Nagaty01	2001	Orientation (Syntactic-coding)	Gray-level consistency/variance (32 × 32)	Poincarè		Neural Networks	Other	[86]
Pattichis	2001	Orientations (Registered)	Amplitude-modulated (16 × 16)		Rule-based (R92)	Neural Networks	Other	[93]
Zhang01	2001	Singular points (Number/positions) Ridge structure (Ridge-tracing (from SPs))	Gradient (16 × 16)			Fixed Classification	NIST-4	[125]
Bernard	2001	Orientations (Registered)	Gabor (16 × 16)	Poincarè		Neural Networks	Other	[4]
Cappelli02	2002	Orientations (Registered) Orientations (Segmentation)	Slits sum (16 × 16)	Poincarè	Rule-based (R92)	Nearest Neighbor	NIST-14	[20]
Chang	2002	Ridge structure (Geometric/global)	W/O orientations			Syntactic	NIST-4	[22]
Jain02	2002	Ridge structure (Geometric/global)	Gradient (16 × 16)			Graph Matching	NIST-4	[45]
Mohamed	2002	Singular points (Number/positions) Orientations (Direct/unaltered)	Gray-level consistency/variance (5 × 5)	Poincarè		Neural Networks	NIST-4	[84]
Cappelli03	2003	Orientations (Registered)	Slits sum (16 × 16)	Poincarè	Rule-based (R92)	Multiple Techniques	NIST-4	[21]
Yao03	2003	Orientations (Segmentation) Filter responses (Gabor)	Slits sum (16 × 16) Gradient (16 × 16)	Poincarè	Covariances	Multiple Techniques	NIST-4	[124]
Cappelli04	2004	Orientations (Registered)	Slits sum (16 × 16)	Poincarè	Rule-based (R92)	Nearest Neighbors	NIST-4/14	[16]
Zhang04	2004	Singular points (Number/positions) Ridge structure (Ridge-tracing (from SPs))	Gradient (16 × 16)	Poincarè		Fixed Classification	NIST-4	[126]
Nyongesa	2004	Singular points (Number/positions) Singular points (Relative measures) Orientations (Direct/unaltered)	Gray-level consistency/variance (5 × 5)	Poincarè		Neural Networks	NIST-4	[91]
Shah	2004	Orientations (Direct/unaltered) Orientations (Registered)	Line detection (16 × 16)		Entropy	Nearest Neighbor	Other	[104]
Wang04	2004	Singular points (Number/positions) Ridge structure (Ridge-tracing (from SPs))	Gray-level consistency/variance (N/A)	Poincarè		Fixed Classification	NIST-4	[113]
Senior04	2004	Orientations (Registered) Ridge structure (Ridge-tracing (generic)) Ridge structure (Skeleton)	Slits sum (16 × 16) Gradient (16 × 16)			Multiple Techniques	NIST-4	[103]
Klimanee	2004	Singular points (Number/positions)	Gradient (4 × 4)	Poincarè Variance		Fixed Classification	Other	[58]
Neuhaus	2005	Orientations (Direct/unaltered)	Gradient (N/A)			Graph matching	NIST-4	[87]
Park	2005	Orientations (Registered)	FFT (16 × 16)		Consistency	Other	NIST-4	[92]
Tan	2005	Orientations (Direct/unaltered)	Gradient (32 × 32)			Other	NIST-4	[106]
Min06	2006	Filter responses (Gabor)	Gradient (16 × 16)	Poincarè	Covariances	Support Vector Machines	NIST-4	[80]

Continued on next page. . .

Table 1: (continued).

Name	Year	Features	Orientations	Singular points	Reference point	Classification technique	Test databases	Reference
Wang07	2007	Singular points (Number/positions) Ridge structure (Ridge-tracing (from SPs))	Gradient (16×16)	Poincarè		Fixed Classification	NIST-4	[111]
Kristensen	2007	Filter responses (Gabor)	Gradient (16×16)	Poincarè	Covariances	Neural Networks	Other	[59]
Hong08	2008	Singular points (Number/positions) Ridge structure (Ridge-tracing (from SPs)) Filter responses (Gabor)	Gradient (16×16)	Poincarè	Covariances	Multiple Techniques	NIST-4	[38]
Li	2008	Orientations (Registered) Singular points (Number/positions) Singular points (Relative measures)	Gradient (8×8)	Complex filters		Support Vector Machines	NIST-4	[63]
Liu08	2008	Orientations (Direct/unaltered) Singular points (Number/positions)	Gradient (16×16)	Shrinking and spanning		Fixed Classification	NIST-4	[66]
Fan	2008	Singular points (Number/positions) Ridge structure (Ridge-tracing (from SPs))	Gradient ($4 \times 4, 8 \times 8, 16 \times 16$)	Zero-pole Poincarè		Fixed Classification	NIST-4	[28]
Jin	2009	Filter responses (DCT)	W/O orientations			Neural Networks	NIST-24	[49]
Liu09	2009	Singular points (Number/positions) Ridge structure (Skeleton)	W/O details (N/A)	Poincarè		Fixed Classification	NIST-4	[70]
Wang09	2009	Singular points (Number/positions)	Slits sum (17×17) Gradient (16×16)	Poincarè		Fixed Classification	FCV2002/04	[112]
Jung09	2009	Ridge structure (Skeleton)	W/O orientations			Structural Models	FCV2000/02	[52]
Liu10	2010	Singular points (Relative measures)	Gradient (10×10)	Complex filters		Structural Models	NIST-4	[67]
Min10	2010	Filter responses (Gabor)	Gradient (16×16)	Poincarè	Covariances	Support Vector Machines	NIST-4	[81]
Rajanna	2010	Orientations (Registered) Ridge structure (Minutiae) Filter responses (Gabor)	Slits sum (16×16) Gradient (8×8)	Poincarè	Ridge normals	Nearest Neighbor	NIST-4	[95]
Hu	2010	Orientations (Direct/unaltered)	Gradient (16×16)			Multiple Techniques	FCV2004	[42]
Leung	2011	Filter responses (Gabor)	Gradient (16×16)		Rule-based (R92)	Other	FCV,NIST-4	[61]
Wang11	2011	Filter responses (GHM multiwavelet)	W/O orientations			Other	NIST-4	[110]
Le	2012	Filter responses (Gabor)	Gradient ($6 \times 6, 12 \times 12, 18 \times 18$)	Complex filters		Nearest Neighbor	FCV2004	[60]
Cao	2013	Orientations (Direct/unaltered) Singular points (Number/positions) Ridge structure (Ridge-tracing (from SPs))	Gradient (8×8)	Complex filters		Support Vector Machines Fixed Classification Nearest Neighbor	NIST-4	[11]
Luo	2014	Filter responses (Curvelet)	W/O orientations			Nearest Neighbor	NIST-4	[72]
Jung15	2015	Orientations (Registered)	FFT (16×16)		Markov	Nearest Neighbor	FCV2000/02/04	[53]

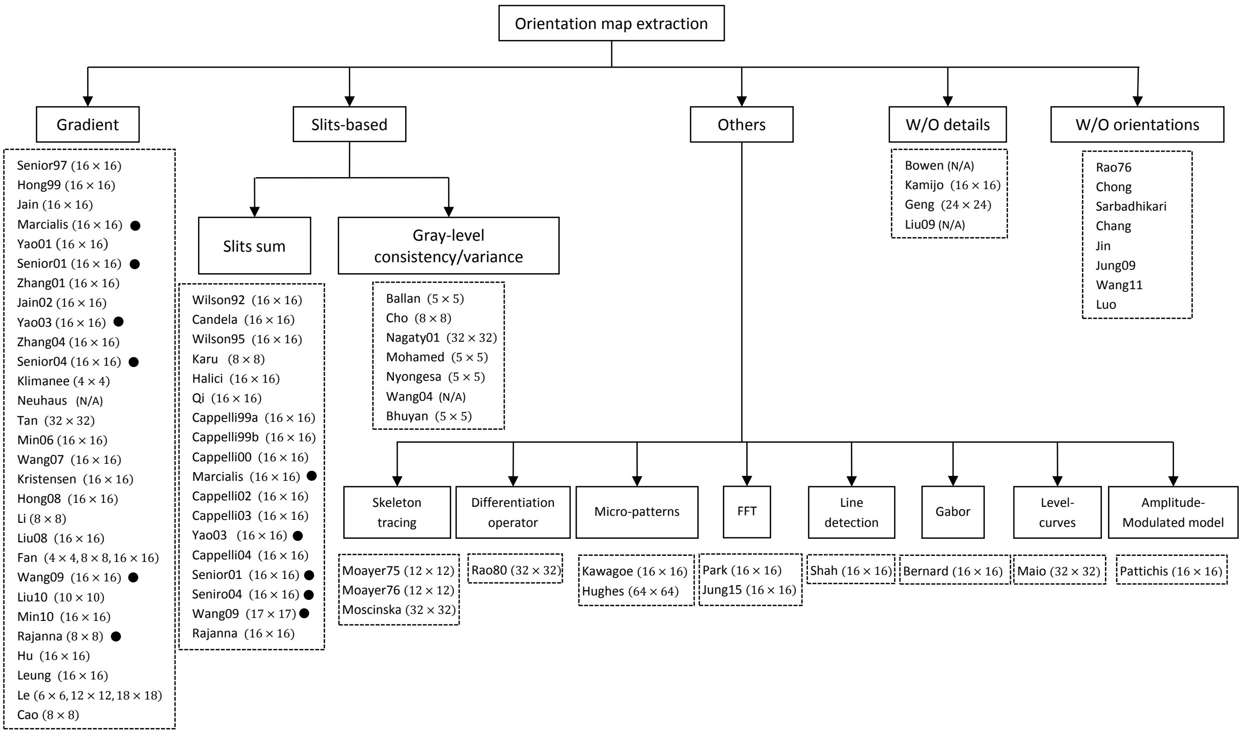


Figure 3: Taxonomy of orientation map extraction methods used in fingerprint classification.

3.1.1. Gradient method

Mathematically, the gradient is the vector of partial derivatives. One of its properties is the fact that the gradient of an image points to the direction of maximum tonal change, its magnitude being proportional to that change. That is, the gradient in an image block indicates the direction in which the gray-level intensity changes more rapidly. Consequently, in the context of OM extraction, the gradient at each pixel of the image is expected to be perpendicular to the appearance of ridges (or valleys). Hence, the orientation of the ridges or valleys in a block is perpendicular to the overall direction of the gradients in that same block (this idea is shown in Fig. 4).

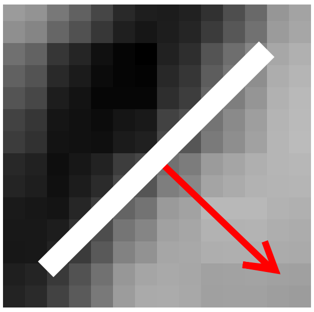


Figure 4: Gradient of an image block (in red) and its corresponding orientation of the block (in white).

One should take into account that a gradient is computed for each pixel in the block, but the orientation is then assigned to the whole block. Hence, pixel gradients need to be averaged, which is not straightforward since directional vectors (angles) are being averaged. In order to solve this problem, Kass and Witkin [56] proposed the usage of squared gradient vectors, which can be easily averaged. Following this method, Rao

[96] proposed the computation of fingerprint OM following this idea, which have been widely used since then [101, 39, 45, 38, 67, 11]. For details on the implementation of the gradient method we refer the readers to [3]. Furthermore, the authors of [3] presented an equivalent method based on principal component analysis. In fact, this algorithm is the origin of the squared representation of the orientation map, which is used for smoothing, besides other purposes. In this representation, each orientation θ is transformed to $2 \cdot \theta$ and is represented by the two components $[\sin(2 \cdot \theta),\ \cos(2 \cdot \theta)]$, which can be independently averaged without losing the radial properties of the angles.

The gradient method method is the most extended one for fingerprint classification as it can be shown in Fig. 3. Senior [101] was the first author considering this method, whose usage has been then considered by many others including some of the most recent papers [61], [60]. Gradient extraction is sensitive to noise and imperfections [107, 65], hindering the OM estimation. However, this problem is usually overcome by a posterior smoothing of the OM, as well as in the rest of the methods, which allows for a better representation of the local ridge flow for classification.

3.1.2. Slits sum method

Slits sum methods [105] attempt to find out the most probable orientation of a ridge or valley at each pixel. In order to do so, a number of slits (usually eight in a 9×9 block, see Fig. 5) is defined in different orientations with respect to the center pixel. Then, the maximum slit sum, or the minimum one, is considered as orientation depending on whether the center pixel belongs to a valley or to a ridge, respectively. This is due to the fact that sums along pixels in a valley (bright pixels) are higher, whereas the sums along pixels in a ridge (dark pixels) are lower.

7		8		1		2		3
6		7	8	1	2	3		4
		6				4		
5		5		**C**		5		5
		4				6		
4		3	2	1	8	7		6
3		2		1		8		7

Figure 5: Slit configuration used for the computation of the orientation map in the slits sum method. **C** refers to the center pixel, whereas each number correspond to a slit.

A posterior smoothing is usually carried out in order to improve the OM estimation. This method has been widely considered for fingerprint classification since 1992 [122], which was the beginning of PCASYS (Pattern-level Classification Automation SYStem) [10]. Owing to PCASYS this method became popular, since it was made publicly available and has been a reference tool for researchers in the area.

3.1.3. Gray-level consistency/variance method

This method is similar to slits sum method (in fact, they belong to the same category). However, in this method proposed by Mehtre [79] it is assumed that the intensity change of the pixels along the orientation is low. On this account, the pixels in different slits (orientations) are compared with the center pixel and the slit with the lowest variation is considered as orientation. In order to define the slits, most of the methods consider a 5×5 block (see Fig. 6).

3.1.4. Others

Among the number of papers in the literature, other OM extraction mechanisms have been considered such as skeleton tracing [82, 83, 85], micro-patterns [57, 43], Fast Fourier Transform (FFT) [92, 53] and line detection [104] (see Others in Fig. 3). However, these approaches are non-standard approaches and have received low attention in the literature.

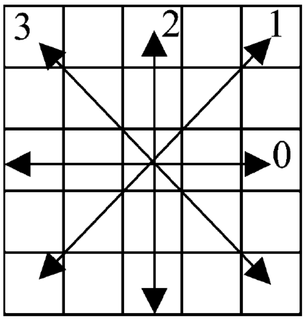


Figure 6: Block used for the computation of the gray-level variance in 4 different orientations in the Gray-level consistency-variance method.

There is also a fact which reflects the lack of details in many papers on fingerprint recognition as those in which orientations are used but no details are given [9, 54, 35, 70] (W/O details in Fig. 3).

Otherwise, there are few papers which do not make use of orientations [98, 25, 100, 22, 49, 52, 110, 72] (W/O orientations in Fig. 3). Nevertheless, they are exceptions and in most of the cases, they consider orientations in some way, despite not explicitly extracting an OM.

3.2. Extraction of singular points

SPs are the points in the fingerprint where the ridges vary more rapidly, that is, where the ridge curvature is higher than usual (Fig. 2b). These points are important because they determine the topological structure of the fingerprint and consequently, its class. As we have already mentioned, there are two types of SPs:

- *Core*: The upper-most point of the inner-most ridge of the loop, that is, the point where ridges tend to converge.
- *Delta*: The triangular-shaped pattern where the ridge flow diverges, in other words, the point where ridges with three different orientations meet.

SPs serve for classification purposes, since assuming that SPs are perfectly located, the fingerprint can be labeled following simple rules that stem from the definition of each fingerprint class. Obviously, the problem comes from the difficulty in finding these points, which is a hard task that highly depends on the extracted OM, and therefore on the quality of the fingerprint. In addition to the usage for classification purposes, SPs, and more specifically core points, are usually considered as the center of the fingerprint; hence, they can be used as registration point for methods requiring fingerprint alignment both for classification or matching [45, 20, 63].

The concept of reference point (which arises from the latter usage) refers to the point considered as the center of the fingerprint, that is, the one used for fingerprint alignment. In those cases where a core exists, the reference point and the core are usually the same point (but this need not always occur [119, 45]). However, in Arch type fingerprints there is no core, and hence the reference point needs to be established following other rules. The reference point is usually considered to be the position where the orientation change is maximum. Fig. 7 shows an example of the reference point in an Arch type fingerprint.

On account of the differences between SP detection and reference point detection methods, our taxonomy is divided into these two differentiated categories. Fig. 8 shows the classification of both types of methods. Notice that as in the case of OM extraction methods, there are papers that consider more than one model (e.g., SPs and reference point extraction) in which case a black dot indicates that the paper is classified into more than one category. In the first category, Poincarè method is without any doubt the most extended method for SP detection in the literature, being used in almost every paper that considers SPs. Methods based on complex filters have been the second most common ones (and mainly in recent years [67, 60, 11]). Other mechanisms have been also considered, but without much success. Similarly, reference point detection methods have been dominated by two methods (despite other few models have been also considered [104, 92,

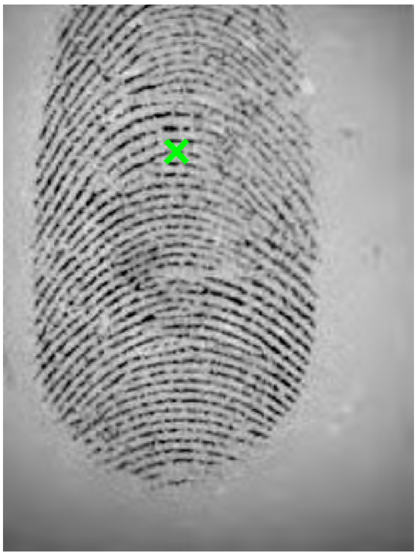


Figure 7: Reference point in an arch type fingerprint.

95]). These methods are the rule-based R92 method [116, 119], which success is due to its usage in PCASYS software tool, and the method based on orientation covariances, which is considered in the widespread FingerCode model [46] (explained in Section 4). These four methods are briefly explained hereafter.

3.2.1. Poincarè method for SP detection

The Poincarè method was firstly applied for SP detection in [57]. It is based on the concept of Poincarè index, which is defined as the total rotation of the vectors along a curve in a vector field. In its application to fingerprints, the OM is taken as a discrete vector field. For each block in the OM, the Poincarè index is computed using a digital curve around the block, considering the surrounding orientations as vectors. Obviously, orientations have no direction, and therefore the direction of the first orientation is selected arbitrarily, whereas the remaining directions are taken so that the difference with the preceding one is minimum. Being in a discrete environment, the resulting total rotation can only take three values: 0, a positive value (which depends on the implementation, e.g., π) or a negative value (e.g., $-\pi$), meaning that there is no SP, there is core point or a delta point, respectively. Fig. 9 shows these three cases in the case of a digital curve of 9 pixels (3×3); the arrows refer to the direction that is used in the computation of Poincarè in each orientation.

It is common, depending on the size of the digital curve considered, to label more than one neighboring orientation block as a core or delta, in which case they are usually clustered, being the SP located in its center. Moreover, the OM extracted can produce the detection of too many SPs (more than the maximum number of cores and/or deltas, i.e., two of each type). These cases are meant to be produced by aberrations in the OM, typically as a consequence of image imperfections. The most common practice is to smooth the OM applying the Poincarè method once again until an acceptable number of SPs is found.

The Poincarè method is the most applied one for SP detection. It has been considered for different aspects such as validation of SPs [57], direct classification based on SPs [55], reference point location [46] and initial point location for ridge-tracing [126], among others.

3.2.2. Complex filters-based SP detection

The usage of complex filters for SP detection comes from the symmetries that appear at the SPs positions when the squared orientation map is considered (every orientation θ is transformed to $2 \cdot \theta$) [88] (see Fig. 10). These symmetries were originally studied by Bigün in [7], where general curve families were detected measuring the amount of symmetry in each area. These ideas were applied to the symmetries of the SPs,

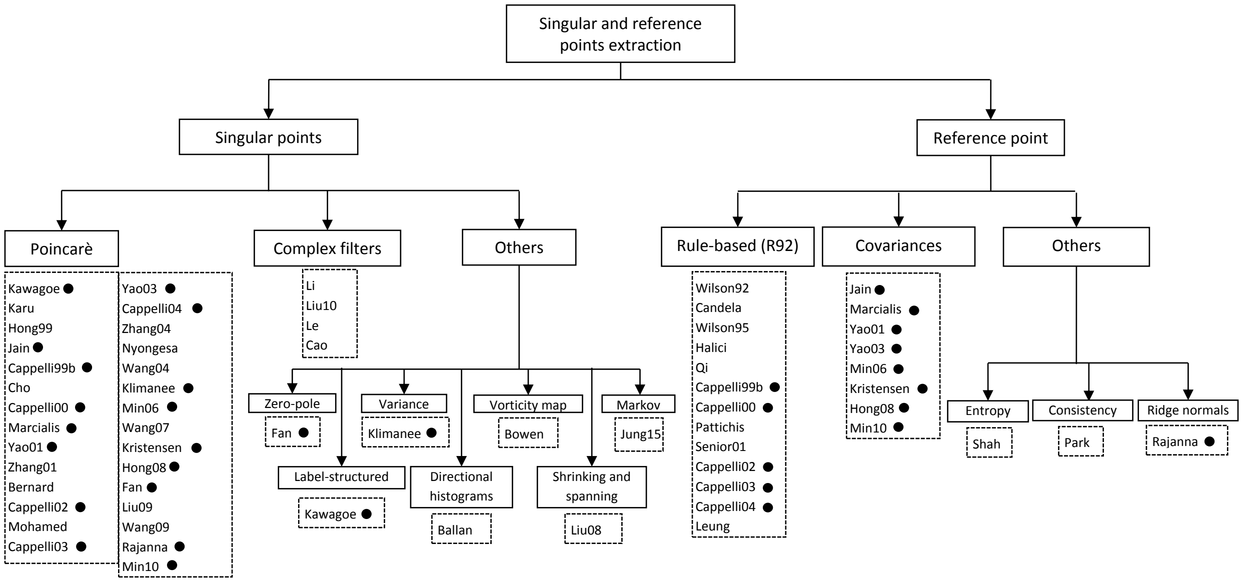
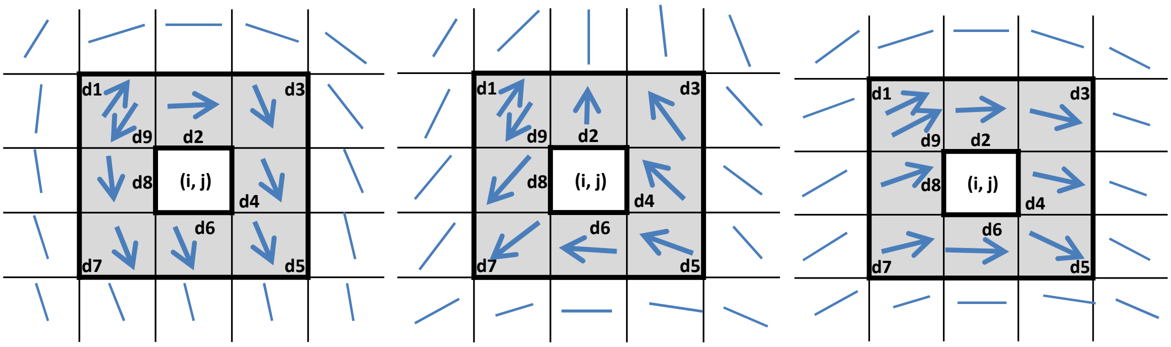


Figure 8: Taxonomy of singular and reference points extraction methods used in fingerprint classification.



(a) Core point

(b) Delta point

(c) No SP

Figure 9: Example of the three possible outcomes of the Poincarè method in a digital curve of 9 pixels (3 × 3).

since the squared orientations can be easily represented by means of complex numbers. The filters, which represent the symmetries that are expected to be found if a core or a delta is present, are also defined as templates of directional values represented by means of complex numbers.

The detection of the SPs is performed carrying out the convolution of the squared OM with the complex filters, selecting those blocks for which the response is over a certain threshold. Obviously, there exist dedicated filters for each type of SP. It is relevant to point out that these filters do not only detect the presence of SPs, but can also provide their orientations, which cannot be directly computed using Poincarè method.

Complex-filters based SP detection has been considered for different purposes by different authors. In [63], the authors used the SPs for classification, but also for fingerprint alignment. Liu [67] considered relative measures obtained from the SPs as features, also taking into account the responses obtained by the SPs. In the case of Le [60], a modification of the original method was considered were only concave cores were sought in order to use them as reference point for further processing. In the first two works (Li and Liu), the authors considered a multi-scale technique in order to increase the robustness of this SP detection method as proposed in the original work [88]. In [11], the authors do not only used complex-filters to align

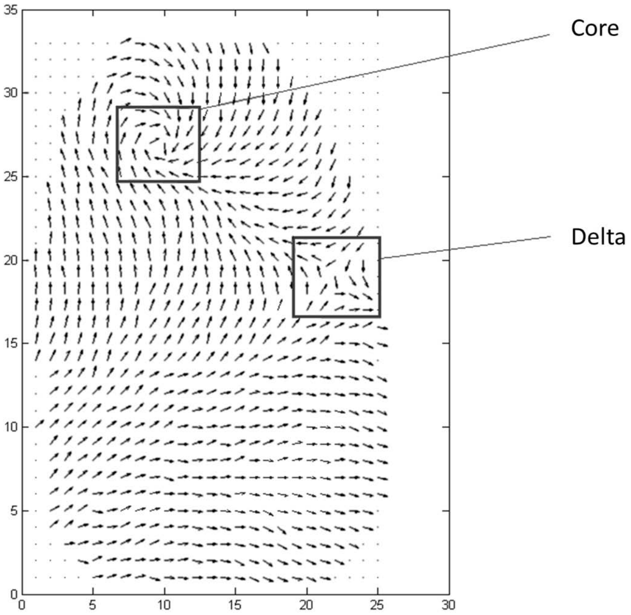


Figure 10: Symmetries that can be found in a squared OM in the location of the SPs.

the fingerprint but also their response in the classification phase.

3.2.3. Rule-based reference point detection (R92)

R92 algorithm does not seek for SPs, but sets a reference point in the fingerprint image instead [116, 119]. In order to do so, a consistent feature is defined, which can be found in any type of fingerprint (right/left loop, whorl, arch and tented arch). In the case of the fingerprint types where a core exists, this feature is located approximately in the same place. Otherwise, in the case of arch type fingerprints, despite not having a true core, R92 is able to find a reference point, which is defined as the area with the greatest orientation variance.

In order to find the reference point a scoring mechanism based on rules considering the neighborhood of each orientation block is followed, which aims to give the largest score to the reference point. In first place, the rows of the orientation map are scanned looking for changes in the slope of consecutive orientations from positive to negative. Secondly, a set of rules are applied increasing the scores of the corresponding positions if these rules are fulfilled. Finally, the computation of the position of the reference point is carried out from the scores obtained (see [116, 119]).

This method has been widely used in the literature, mainly when the feature extraction model required the alignment of the fingerprint (e.g., [122, 10, 121, 17, 18]). We have only found one case where this method was not used for registration but for the establishment of the area for the feature extraction [61].

3.2.4. Covariance-based reference point detection

This method was first used in the FE method of Jain *et al.* [46], where a reference point was needed in order to establish the area for the FE process. It was only considered for fingerprints in which no core points were found (i.e., mainly Arch fingerprints). Its popularity and usage in several papers resides in the fact that any method based on Jain's one makes use of this method for setting the reference point when no core is found.

In this method, likewise in R92 one, a point were the neighborhood presents the largest variation of its orientations is sought. Different orientations in a neighborhood are represented by large values in the covariance matrix of its orientations, and therefore the point reaching the largest variation is taken as reference point.

More specifically, the covariance matrix of each block in the OM is computed using its local neighborhood of a given size. Then, the largest eigenvalue of each covariance matrix is obtained and a new image is formed, which is composed of these values. This image is thresholded and finally, the largest connected component in the image is taken to compute the location of the reference point by its centroid. Notice that greater values of the eigenvalues indicate greater variations in the orientations of the neighborhood and this is why a reference point can be found in this manner.

3.2.5. Others

Although most of the papers work with the aforementioned methods for SPs and reference point detection, there are other few models which aim to extract these type of points. Regarding the extraction of SPs, there are only three methods that do not consider any of the previous ones: Bowen [9], which made use of a vorticity map to find cores and deltas, Ballan [2], which considered directional histograms together with a set of rules, and Liu08 [63], which discretized the orientation in order to use a shrinking and spanning algorithm where the neighboring orientations were analyzed. Besides, other three works considered alternative models such as the zero-pole model combined with Hough transform [28], a label-structured model using rules to define SPs[57] and a the variance method [58], where orientations were codified into six codes and the locations where all the six codes were present in the OM were labeled as SPs. However, these three methods used the Poincarè method after the proposed one as a verification step.

With respect to the reference point location methods, few approaches differ from R92 and covariance-based ones. Shah [104] considered the point of maximum entropy of the orientations as reference point, which also aimed at seeking for the greatest area of variation. Park [92] used a consistency based model, in which the minimum consistency was sought, that is, the area with the greatest orientation changes. Finally, Rajanna [95] used the method presented in [90], which relied on the intersection of the ridge normal lines so as to define a reference point in the fingerprint image, and Jung [53] considered previously trained Markov models to detect reference points.

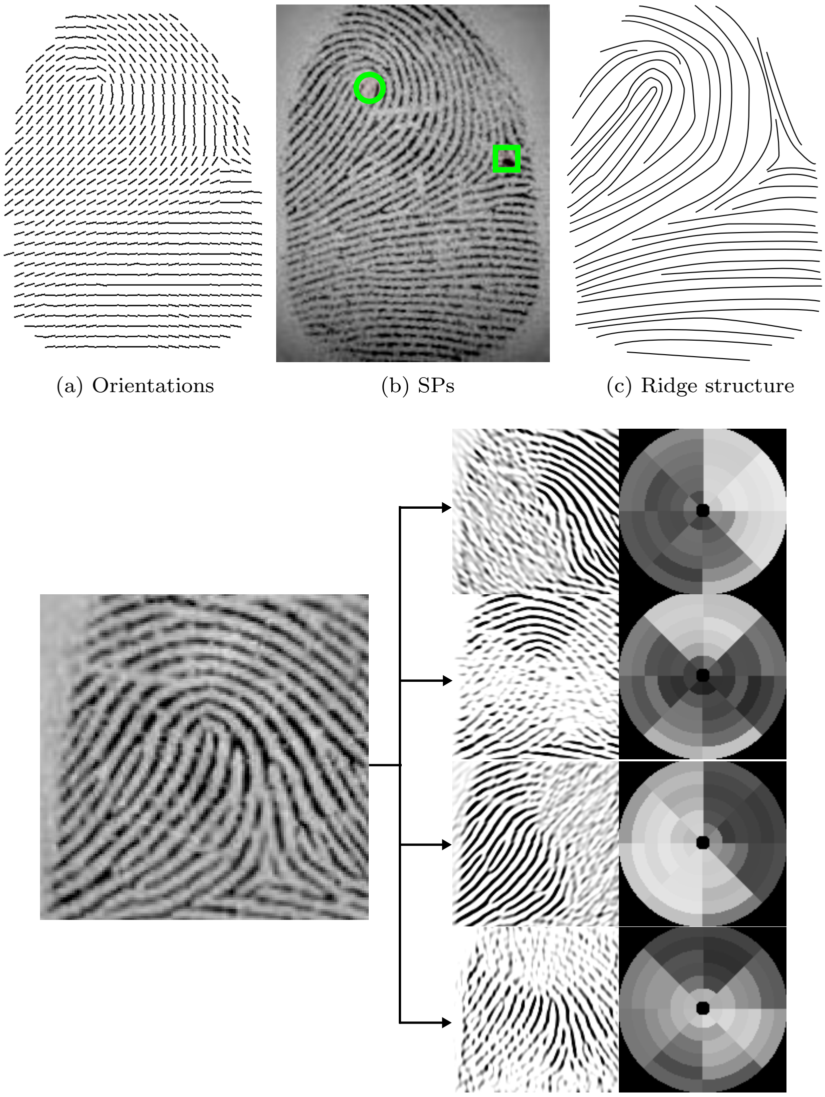
4. Feature extraction methods for classification: feature vectors and taxonomy

Once we have presented the basis of the two most important features for fingerprint classification, in this section we present and describe our taxonomy of FE methods for fingerprint classification. This taxonomy is a refinement of that considered in [77] where the FE methods were classified into four categories depending on the features used (a method could be classified into one or more categories): orientation image, SPs, ridge line flow and Gabor filter responses. In this paper, we have gone through the different methods and we have subdivided these coarse categories into more refined subcategories. Moreover, we have adapted some of them so as to be able to cover a larger range of methods.

Finally, we consider a similar set of four major groups (a representation of these type of features can be observed in Fig. 11):

- *Orientations*: Orientations or information obtained from the orientations (OM) are used to form the feature vector.
- *Singular Points*: Features obtained from the number and position of the SPs are considered.
- *Ridge structure*: Global ridge structure is used to obtain the features.
- *Filter responses*: Features are obtained from the responses of the fingerprint to different filters.

Hence, the last two categories include the last two ones of the previous taxonomy [77]. In order to perform a finer classification, we have then subdivided each one of these four classes into several subclasses depending on how these fingerprint features have been transformed into features for classification. In Fig. 12 our proposed taxonomy is presented. Each method has been classified into at least one category, and as in the previous taxonomies, we have added a black dot near the method if it considers more than one feature (i.e., it appears more than once in the schema). In the rest of this section, we describe each one of the four major categories and its subcategories in detail, focusing on the different works considering them.



(a) Orientations

(b) SPs

(c) Ridge structure

(d) Filter responses

Figure 11: Visual representation of the four types of fingerprint features used for classification considered in our taxonomy.

4.1. Orientations

Orientations, and more specifically, the extraction of OMs have been already described in Section 3.1. This global representation allows for the classification of fingerprints as long as it is well-defined. However, a series of problems exists such as the practical difficulty in accurately extracting the OM, fingerprint rotations and geometric perturbations.

For these reasons, the most commonly used methods which consider orientations as features also make use of preprocessing mechanisms in which the fingerprint (or the OM) is aligned before encoding the final feature vector [10]. However, there are also authors proposing classification with unaltered (not registered) orientations [43, 91]. Otherwise, earlier methods were focused on syntactic-coding of the OM [82, 83, 97], but recent methods have not considered these types of features. Another type of FE with orientations considers the segmentation of the OM [73, 15], where areas with different orientations are separated and thereafter used for classification. Hence, we have considered these four subclasses inside orientations-based features: syntactic-coding (earlier approaches), direct/unaltered, registered and segmentation.

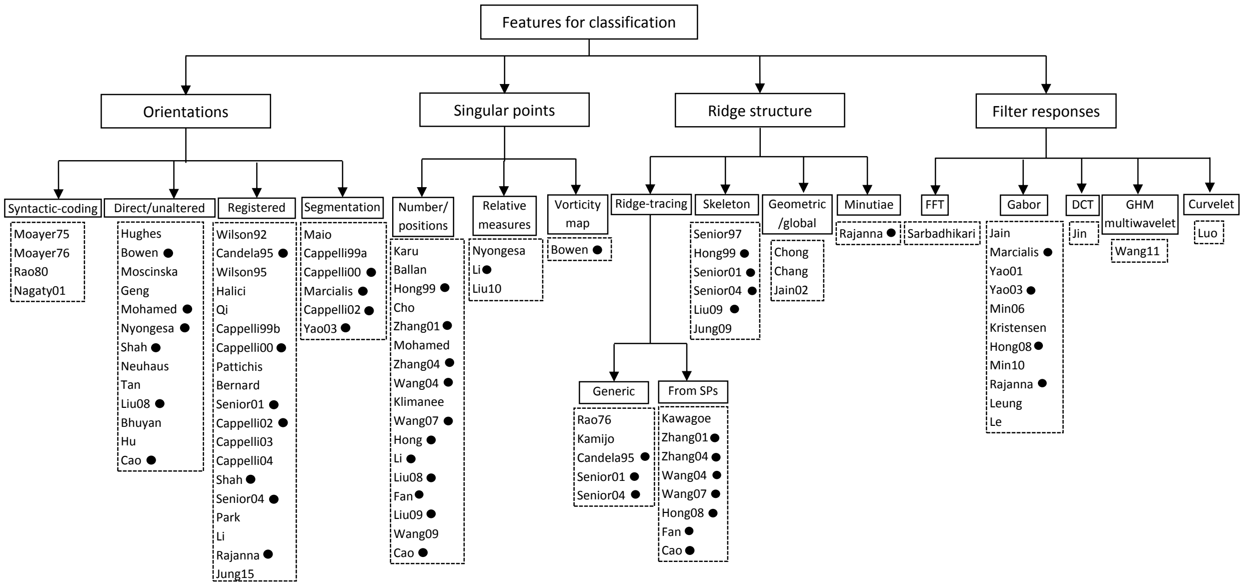


Figure 12: Taxonomy of the features used for fingerprint classification in the literature.

Anyway, before introducing them, we should remark that in most of the cases and independently of the category considered, orientations by themselves are hardly ever used as features in the feature vector, but a method to reduce the number of features is considered. These methods can be both feature reduction methods (selection/transformation) or different measures taken from the orientations. Notice that even in a coarse OM (e.g., the OM generated by PCASYS, whose dimensions are 28×30), if every orientation is codified as a feature, the feature vector would be excessively large (840 features or 1680 features in the case of PCASYS where both components of the squared OM are codified). In addition, notice that each method may have used a different OM extraction method, which can be found in Table 1.

4.1.1. Syntactic-coding

These methods were the first approaches for FE in automatic fingerprint classification, and therefore their aim is focused on reducing the number of features that were used to codify the OM with different symbols to simplify its processing. In order to perform this codification, these methods usually identify specific ridge patterns, whose symbols are previously established. Once this codification is carried out, the fingerprints are classified by finite automatons or grammars.

All the methods in this category have a first processing step in common, which consists of transforming every orientation into the corresponding directional code. That is, the OM is discretized into four values $\{-\pi/4, 0, \pi/4, \pi/2\}$. Afterwards, two different approaches can be found.

Moayer and Fu [82, 83] took disjointed windows of four orientation blocks and a symbol out of the 17 possible symbols (each one with four possible rotations) was assigned to each window. These symbols were concatenated forming a feature vector of length 64, due to the 16×16 OMs.

In [97], Rao started from the directional codes, but used a preprocessing mechanism in order to represent the directional codes in an easier-to-trace matrix with values in $\{0, 1\}$. Using some operators which act as filters in the tracing of the new matrix, a number of strings formed of symbols were obtained. Symbols represented both the different directions of the endpoints and the angles that can be formed by the orientation in the matrix. The tracing was carried out from left to right and top to bottom. The features used for classification were the set of strings from which those representing straight lines were removed. Following this approach, Nagaty [86] considered to process these strings further, transforming them into binary codified ones.

4.1.2. Direct/unaltered

The global ridge flow in a fingerprint, i.e., the OM can be used for classification if it is properly interpreted. For this reason, some authors considered its direct codification as a feature vector [43, 9]. In these cases, either coarse orientations blocks were used as in [43], where the OM size used was of 8×8, or a large amount of features need to be considered [9] (20×20 OM, that is, 400 features).

In order to reduce the number of features in this latter case, Geng and Shen [35] used Karhunen-Loève (KL) transform [44] (also known as Principal Component Analysis, PCA, [51]) for dimensionality reduction, but the number of features that were finally used was not reported by the authors, and the same occurs with [11]. Tan *et al.* [106] also considered the usage of a coarse OM (12×13), but in this case, this map went through 16 different processes so as to extract various features, which altogether form the final feature vector of 2496 elements. In relation to these approaches, more recently Hu and Xie [42] presented a method in which the orientations of the OM were used as primitive features, and new features were generated via Genetic Programming [27] aiming to maximize Fisher's criterion, i.e., the discrimination among classes.

In [85], a different approach considering the usage of orientations was proposed.The authors considered the classification problem as a problem of classifying delta patterns from windows of 3×3 orientation blocks. Before attempting to classify these blocks, the OM was codified with directional codes (explained in the previous subsection). Hence, the feature vector is the discretized OM (which is then computed in windows of 3×3). Also related with the SPs, in [66] the orientations around them were used for classification purposes. Otherwise, in [87] the authors presented a process of what they called a normalization of the OM establishing all the angles in $[-\pi/2, \pi/2]$. Then, they computed the angle variance, but taking the average of the angles as their sums in such a way that greater variance was given by vertical ridges and SPs areas. The binarized map of the variance was finally used for classification.

Rather than considering all the blocks in the OM as features, in [84] and in its extension [91], the global orientation of the fingerprint, that is, the averaged orientation of all the blocks of the OM, was used as a feature among others. Similarly, Shah and Sastry [104] considered the two most predominant orientations and their percentage of occurrence in 24 areas in which the OM was divided (the division was carried out considering the center of the image as the center of the fingerprint).

4.1.3. Registered

The spatial transformations of fingerprints in the image and the corresponding displacement of the OM could hinder the classification performance. On this account, a number of methods have considered the registration of the OM with respect to a reference point. Even two of them have also carried out the alignment with the orientation of the reference point [4, 63].

In this category, most of the works are directly influenced by PCASYS [10] and its preceding works [122, 121]. In all these works, the R92 algorithm was used to find a reference point. Then, the OM was registered and the KL transform was applied so as to reduce the number of features in the final feature vector. Differently from previous approaches, the vertical and horizontal components of the squared OM were considered separately in the feature vector. In [122] and [121], 640 features were reduced to 106 features, whereas in PCASYS [10], 1680 features (28×30 blocks with two components per block) were reduced to 64 and 128 features, depending on the classification method that was used afterwards.

Other authors have followed the same scheme of PCASYS, in some cases with variations. In [36], the authors also codified the registered orientations (with respect to the reference point extracted with R92 method) in the feature vector, which was thereafter reduced using the KL transform. They worked with OMs of 16×16 blocks, which produced 256 features (using one feature for each orientation), whose reduction lead to a feature vector of length 40. Pattichis *et al.* [93] proposed to substitute the enhancement phase carried out in PCASYS (using the FFT before extracting the OM) and the extraction itself with the usage of amplitude and frequency modulated functions. The same feature vector as that used in PCASYS was used in [120, 102, 103, 95]. Additionally in the case of [95], a variant using registered orientations obtained with the gradient method instead of the slits sum one was considered; however, no details on the location of the reference point were given in this case.

A number of works from Cappelli *et al.* [17, 18, 20, 21, 16] considered the usage of registered orientations as a part of the feature vector for classification, which in most of cases were reduced via an extension of the

KL transform to multiple subspaces, i.e., Multi-space KL (MKL) transform. The most remarkable differences with respect to PCASYS (apart from the MKL transform) were the method used for OM extraction (the gradient method), the usage of R92 method only in case that the Poincarè one was not able to find a core point and the preprocessing of the orientations by a enhancement step which consisted of three phases: regularization, attenuation and strengthening [15]. In all except for [21], the features obtained were followed by the MKL transform; in this exception, the MKL was also used, but the orientations without processing were also considered as direct features for classification.

Other methods using registered orientations highly differ from the approach of PCASYS. Bernard *et al.* [4] used a Gabor-based technique to extract the OM. The reference point for the registration was obtained using Poincarè method in combination with the orientation variance method in which the variance of the orientations need to exceed a threshold so as to be taken as a SP. The 32×32 OM was registered and compensated for the angle difference with respect to the reference point, whose angle was computed using the PCA method. Finally, all the blocks were codified in the feature vector.

Also in [104], the authors considered a different model for the computation of the reference point based on the orientation entropy. Once this point was calculated, orientation blocks of 15×15 pixels were formed in a cropped 120×120 pixels window. The feature vector was composed of the two predominant orientations in the pixels of each block, making a total of 128 features (2 features per block). Notice that in this method, orientations were computed pixel-wise via a line detector.

Park and Park [92] obtained the orientations with FFT and found the reference point using the minimum consistency method. Around this point, they centered a 21×21 window in the OM; then, each block was converted to an image of 5×5 pixels which represents the orientation and the 105×105 features where finally reduced to only 4 features by a KDA/GSVD model (Kernel Discriminant Analysis / Generalized Singular Value Decomposition).

The part of the feature vector in [63] devoted to registered orientations is the unique one in which instead of considering orientations, a model that represented the OM with several coefficients was used. In fact, these coefficients were the ones considered in the feature vector. It was obtained following a complex constrained nonlinear phase portrait model previously developed by the authors in [62]. Since the model of order 6 was used, 54 features were obtained. However, before estimating this model, the gradient method was used to obtain the OM, the complex filter method was used to extract cores and deltas, the fingerprint was aligned with respect to the translation and rotation and it was cropped finally considering an image of 320×320 pixels.

Jung and Lee [53] proposed a different approach where the OM was divided into four quadrants centered on the core (or reference) point of the fingerprint. Then, local models in each quadrant were computed (histograms of discretized orientations), which formed the feature vector for classification.

4.1.4. Segmentation

OM segmentation-based methods highly differ from the approaches using orientations as features. Instead of codifying the orientations themselves in the feature vector, these methods first carry out a segmentation of the OM, that is, they group areas with similar orientations, under the assumption that such segmentation can be used for classification (this idea can be observed in Fig. 13). Two types of methods can be found in this category depending on how they carry out the segmentation.

Maio and Maltoni [73] presented the first segmentation approach. The segmentation of the OM (computed using level curves) was obtained by a dynamic clustering method, and then a relational graph (invariant with respect to translation and rotation) was extracted. Finally, an inexact graph matching process was carried out in order to find the similarity of the graph extracted with respect to the templates of the different classes. However, this final step was not detailed in the paper. The authors proposed to use the similarity (matching) values obtained for each class as feature vector. The same segmentation model was also considered in [78, 124], but instead of performing the inexact graph matching, the authors extracted several features from the graph nodes such as local information and geometric and spectral relations (the OM computation also varied); however, few details were given on their extraction.

In [15] an evolution of the approach in [73] was presented, which was also used as a part of the feature vectors in [18] and [20]. In this method the authors used the gradients for the OM estimation and considered

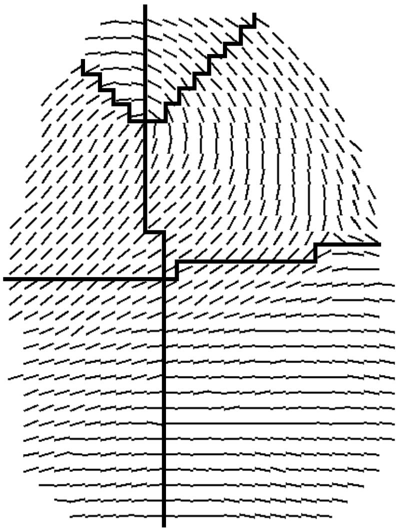


Figure 13: Segmentation of an OM.

their enhancement with the three phases mentioned in the previous subsection. Anyway, the major difference was the definition of predefined masks (graphs templates) for each fingerprint class. These masks were fitted to the OM minimizing the segmentation cost, which depended on the variance in the different regions and another term acting when the orientation difference between two regions could be a discriminant factor. Finally, the lowest segmentation cost found for each class formed the feature vector for classification (5 features).

4.2. Singular points

SPs and the different methods for their extraction have been described in Section 3.2. If correctly characterized, SPs can straightforwardly lead to accurate classification of fingerprints. For this reason, most of the methods in this category consider the usage of the number and/or positioning of the SPs as features for classification purposes (simple rules can be then derived so as to classify fingerprints). Otherwise, other methods [84, 91, 63, 67] complement or directly use relative measures obtained from SPs. In addition, there is a discordant work [9] in which a vorticity map was used claiming that it represents the existence of core and delta points among other characteristics. Anyway, any type of classification based on SPsis highly affected by the quality of the SP detection method and of the fingerprint image, since both false (spurious) SPs or non-detected ones can hinder the results obtained. Recall that for each method, Table 1 shows the SP detection method used.

On this account, we have identified three subcategories depending on how SPs are encoded in the feature vector: number/positions, relative measures and vorticity map. Hereafter, we detail the two most important ones (the first two subcategories), since the absence of details in [9] does not allow us to properly extend the description of this last category.

4.2.1. Number / positions

Almost all the methods in this category combine the usage of the number and/or positions of the SPs with other type of feature, which shows both the usefulness of this feature for classification purposes but also the fact that this information may not be enough to perform an accurate classification.

All the works listed in this category in Fig. 12 use the number of cores and deltas in the fingerprint as features. In addition, some of them add more information in terms of relative position of the deltas with respect to the core or the direction of the SPs. In fact, there are only three methods, in which the number of cores and deltas are used as unique features [113, 66, 28].

Karu and Jain [55] added to these values the difference of slope between the line connecting the core/delta pair (if only one pair was found) and the local orientations along the line. Besides this characteristic, the authors also considered the relative position of the delta with respect to the core, which was also considered in [2, 112]. In [70] the same features as in [55] were used. Similarly, Hong and Jain [39] used these characteristics, but included the number of ridges crossing the connecting line.

Other authors took into account the direction of the core points. For example, in [24] the curvature of the core (concave or convex) and its direction (left or right) were used as features. Zhang *et al* [125, 126] also included information of the direction of the main core point. The same information was used in [84] together with the relative position of the core-delta pair.

Klimanee and Nguyen [58], instead of only considering the direction of the main core point, used the direction of all the SPs. A different approach was presented in [111], where a new type of SP was put forward, namely *coredelta*, aiming to ease the detection of tented arch fingerprints. Li *et al.* approach [63] differs from the previous ones because they used the complex filter method to detect SPs, and consequently introduced the confidence in the SP detection in the feature vector (the number of cores and deltas were multiplied by the confidence in their estimation). Cao *et al.* [11] also made use of this information in the classification phase. A more common approach was presented in [38], where the positioning of the deltas with respect to the main core (in discretized values) and the distance among them (also discretized) were considered as features.

4.2.2. Relative measures

Among SPs-based features, some authors proposed the usage of measures taken from the SPs instead of their number or positioning as we have shown in the previous subsection. In these cases, the number of cores and deltas are implicitly included in the feature vector as well as the relative position among SPs.

In [91], the authors considered all the possible combinations between SPs and extracted the corresponding angles as features. It consisted of 6 elements; these elements took the value zero whenever the corresponding measure could not be taken (one of the SPs was not found), and hence the number of SPs was implicitly introduced. Similarly, Li *et al* [63] included the angles between the primary core point and the two deltas in the feature vector (if a delta was not found, a zero was codified).

Finally, Liu [67] considered a multi-scale method in which the SPs were extracted in each one of the four scales of the OM. For each scale, the same 16 measures (or values), 4 for each possible SP, were considered in the final feature vector of 64 elements. Most of these measures were obtained in relation to the primary core (the upper one) and included: the position of the primary core, its direction, the confidence in the estimation of the SPs and the rest of the features were obtained as distances to the primary core, direction differences between SPs and directions difference between the line connecting the primary core with each SP and the direction of the primary core. Notice that some of the features were available due to the usage of the complex filter method to obtain the SPs.

4.3. Ridge structure

This category encompasses those FE methods in which the features are obtained from the global ridge structure. This structure can be given by a set of curves, a mathematical model or simpler (processed) representations than the original fingerprint image, whose aim is to represent the global ridge structure.

Even though these methods share the type of feature used for classification, they can be divided into four well-differentiated subcategories depending on how the global structure is represented. Ridge-tracing methods trace the ridge flow following the same local orientation and creates a set of curves which are usually referred as pseudo-ridges [98, 126, 38]. Other methods extract features from the skeletonized fingerprint image [101, 39, 70], whereas geometric models aiming to describe the ridge flow have been also considered [25, 22, 45]. Otherwise, in [95] the authors considered the presence of minutiae as features, which could also be considered as a characteristic of the ridge structure (despite being a local characteristics). These four subcategories are the ones that we have considered in our taxonomy (Fig. 12). Next, we focus on the details of each one of them.

4.3.1. Ridge-tracing

Ridge-tracing consists of following the local ridge orientations in order to obtain a representation of the global behavior of the ridges in the fingerprint are obtained. These curves maintain the local orientation of the ridges and valleys, but they do not necessarily coincide with them. This category can be further divided depending on the starting point of the traces (generic and starting from SPs as it can be observed in Fig. 8).

Initially generic methods were proposed, without a specific starting point for the ridge-tracing. In [98], the authors used ridge-tracing technique around a manually labeled reference point and a number of rules were used to classify the fingerprint following the trace obtained. Kamijo [54] then considered the usage of the ridge-tracing technique, storing the final trace in a 16 × 16 matrix whose values formed the feature vector. Otherwise, in PCASYS [10] software, a ridge-tracing module was developed in order to improve the classification of whorl type fingerprints. The fingerprint was traced trying to find a pseudo-ridge with a concave upward shape, in which case it was labeled as a whorl without taking into account the output of the classifier (which was based on orientation features). The same technique was used by Senior in [102, 103], since PCASYS was included in his proposals.

Ridge-tracing from SPs was first proposed by Kawagoe and Tojo in [57]. In this work, the authors carried out different ridge traces depending on a previous classification based on SPs. Ridge-tracing was used to differentiate between plain whorl and twin loop fingerprints using measures of twinness and flatness taken from the endings of the traces; similarly, it was considered to distinguish between loops and tented arch fingerprints but also to refine the classification of loops further by tracing pseudo-ridges from the core point and comparing them to the trace of the central trace that lies between both parts of the pseudo-ridges.

Zhang *et al.* [125] used pseudo-ridges (Fig. 14) to distinguish between the fingerprint types with one core point. A pseudo-ridge was traced from this point (in both possible directions) and depending on the position of the two ending points of the pseudo-ridge the fingerprint was classified (if more than one turning point was found it could also be classified as a whorl). This work was extended in [126] including the ridge-tracing from delta points if a core was not found (setting the turning point of the pseudo-ridge as a core point) and including a computation of the area under the line connecting the core point and the end point of the medial axis of the pseudo-ridge (the line lying on the middle of both parts of the pseudo-ridge) and the medial axis itself. Following a threshold and depending on the size and the sign of the area, fingerprint class was estimated using this area as feature. In [28], the authors considered ridge-tracing aiming to find the presence of an undetected core in whorl type fingerprints, and hence used it when only one core was found. Pseudo-ridges were also considered in [11] to differentiate Whorls from the rest of the classes as well as to decide whether a loop was Right or Left.

Also starting from SPs but with a block-wise tracing instead of being pixel-wise, Wang and Xie [113] computed the number of blocks in the trace with right and left tendency (orientation), whose values were used for classification. Otherwise, Hong *et al.* [38] were the first ones explicitly including ridge-tracing information into the feature vector. They traced the ridges from the primary core point (the closest one to the image center in their work) following the proposal of Zhang [126]. However, instead of using the pseudo-ridge directly for classification, the position and the distance (as nominal values) of the endpoints of the pseudo-ridge relative to the primary core point were encoded in the feature vector.

Observing Fig. 12, one can notic that ridge-tracing method are used, in most of the cases (those with the black circle), as additional features. Hence, they can help improving the performance of classification techniques in combination with the previous ones (logically, those starting from SPs already need to extract them, and consequently used them as features). Moreover, a number of approaches (all the generics and some of those starting from SPs) did not extract specific features for classification (classifier learning) from the pseudo-ridges, but used them directly for labeling the fingerprints.

4.3.2. Skeleton

We have included in this category those methods which represent the ridge structure with the skeletonized fingerprint image (Fig. 15). That is an image where the ridges are represented by an eight-connected, one pixel-wide line. This representation has the advantage of being independent of the OM extraction and the

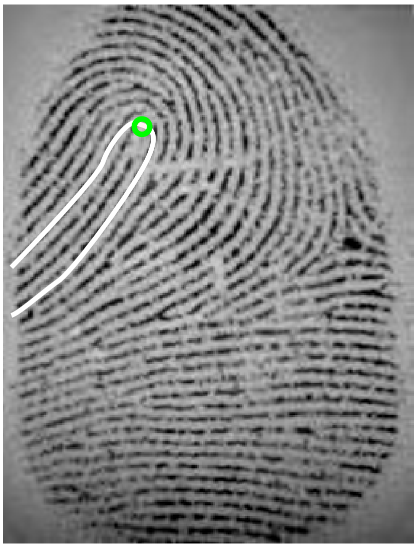


Figure 14: Pseudo-ridge obtained from ridge-tracing starting in the core point.

SPs, but the extraction of the skeleton is a difficult task, since poor quality images affect the skeletonization phase arising problems such as spurious, broken or missing ridges.

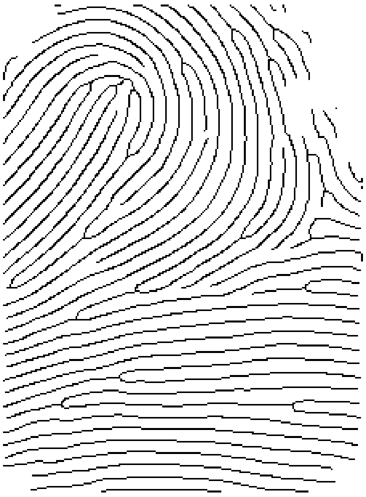


Figure 15: Skeletonized fingerprint image.

Senior [101] made the first attempt to include information extracted from the skeleton in the feature vector. He traced horizontal parallel lines along the skeletonized image and extracted four features for each intersection between these lines and ridge skeleton ones: the angle, the angle change since the last intersection, the distance to the last intersection and the curvature of the ridge. He extended this work in [102, 103] including new features extracted from the ridge skeleton. More specifically, features from the salient points of the ridges where extracted: curvature maxima and four axis-parallel turning points.

A different approach was considered in [39] by Hong and Jain. They count the number of reccurring ridges, which were divided into three types, type 0, 1 or 2, depending on the cumulative orientation of their subridges. These features where then used to classify the fingerprints. This method was also followed in

[70]. Otherwise, Jung and Lee [52] developed a chaincode in which the slopes of the ridges taken from the skeleton image were codified in 16 orientations and added to the chain.

4.3.3. Geometric/global

This category is formed of those methods using a geometric or global representation model of the fingerprint, which is then used to obtain the features used for classification. There are three works [25, 22, 45] considering this type of representation, which have not had continuity in the literature. Even though a global representation of the fingerprint has several advantages, this methods also has some drawbacks such as being quite ad-hoc and complex methods.

Chong *et al.* [25] presented a geometric framework in which the ridges of the fingerprint were modeled by B-spline curves (similar to Fig. 11c). The B-splines were fitted to the ridge shape and similarly-oriented but non-overlapping proximate ridges were grouped. Finally, a global ridge shape analysis on the image of grouped ridges was performed, taking into account that each class was defined with a different shape. These shapes were obtained by previous observations made by the authors. The method of Jain and Minut [45] also considered the usage of splines to define kernel curves for each class. These curves also modeled the shape of each corresponding class. However, in this case the new fingerprint was fitted to each one of the kernels and the fitness value was used for classification.

A different approach was proposed in [22] where 10 basic ridge patterns were defined and named fundamental ridges. These fundamental ridges were sought in the fingerprint by ridge-tracing and a ridge pattern discrimination phase was carried out from which a feature vector containing the sequence of fundamental ridge patterns in the fingerprint was obtained.

4.3.4. Minutiae

The unique attempt to use local fingerprint characteristics can be found in [95], where minutiae maps are used for classification purposes. Minutiae can be considered in ridge structure category, since they are defined as ridge discontinuities. The usage of this type of characteristics in fingerprint classification is rare, but the authors stated that this was carried out to study whether any correlation exists between minutiae and fingerprint classes. In order to do so, the authors defined a circular region around the reference point of the fingerprint and divided it into 3 bands with 12 sectors each one. At each sector two different features were computed. First, the normalized number of minutiae in the sector (with respect to the sector with the largest number of minutiae). Second, the average orientation of the minutiae. The authors showed that minutiae maps were not useful to discriminate between classes by themselves properly, but some correlation between them was found.

4.4. Filter responses

Some FE methods are based on the response of the fingerprint image to different filters. The filter responses in each pixel usually depend on the local orientation of ridges and valleys in the image, and therefore they represent valuable information for classification. There is one technique that excels among them in this category, which was developed by Jain *et al.* [46] and used Gabor filters with different orientations, which was named FingerCode. This technique has been widely spread and many authors have used it (see Fig. 12). In this subsection, we also briefly cover other approaches considering different filters such as the Fast Fourier Transform (FFT) [100], the Discrete Cosine Transform (DCT) [49] or Geronimo-Hardin-Massopust (GHM) discrete multiwavelet transform [110].

4.4.1. Gabor

The FingerCode method [46] is the method used by the rest of the works in this category (it is represented in Fig. 11d). In this method, a global representation of the ridges and valleys obtained with the Gabor filter was used, which was independent of the presence of minutiae. First, the region of interest was computed as a location around the core (computed with Poincarè) or the reference point (measuring local covariances). This region was centered 40 pixels down from the corresponding point and it was a spatial tessellation of a circular form composed of a number of sectors divided in different bands (in [46] a total of 48 sectors were used, 6 bands and 8 sectors per band). Once this region was established, it was decomposed into four different components using the Gabor filter with four orientations $(0, \pi/4, \pi/2, 3\pi/4)$. In this manner, ridges and valleys become more or less accentuated depending on how similar their orientation with respect to that of the Gabor filter was (see Fig. 11d). Hence, in the areas where the orientation was the same, the ridges and valleys became more defined, showing a larger gray level variance in the sector. Otherwise, in the other areas, the variance was low, since orientations differ. This gray level variance in each one of the previously defined sectors was the one used to form the FingerCode, which consisted of 192 values representing the standard gray level deviation in the 48 sectors of every component (there were 4 components). As a result, each value indicated the answer of the sector to the corresponding filter orientation. Exactly the same implementation and parameters were used in several works [78, 123, 124, 80, 38, 81]. It should be noticed that in the FingerCode method, a fingerprint was rejected for FE whenever all the region of interest could not be located in the fingerprint image.

Other works have considered the usage of FingerCode model with some variations. In [59], a different set-up was considered with 80 sectors (5 bands and 16 sectors per band) and 8 orientations, making a total of 640 features. Rajanna *et al.* [95] followed the original approach but a different method to locate the reference point was used (it is not clear which of the proposed methods for reference point location was used in this case).

Greater variations of the original method have been considered in [60, 61]. Le and Van [60] produced a feature vector of 1280 elements, that is, they considered 10 bands with 16 sectors in each one and 8 different orientations. Moreover, they used a complex filter-based method to detect a concave core (or reference point) to locate the region of interest and performed a correction with respect to the rotation of the fingerprint prior to the FingerCode extraction. In the case of Leung and Leung [61] the major difference was the way in which the region of interest was divided into sectors. They used a squared mesh with equally-sized square sectors. 12 orientations were used and 68 sectors were considered for each orientation making a total of 816 features. Moreover, R92 method was used for reference point location instead of Poincarè and covariances. Another important difference was the fact that in [61] fingerprints were not rejected when a part of the region of interest fell out of the fingerprint image or the segmented fingerprint area, but they set the corresponding sector with a missing value which is then treated by the classifier learning algorithm.

4.4.2. Other filters

As we have shown in our taxonomy in Fig. 12, filters other than Gabor ones have been used by different authors for fingerprint classification. These filters are the FFT, the DCT, the GHM discrete multiwavelet transform, and the Curvelet transform.

The FFT was used by Sarbadhikari et al. [100] to obtain the frequency variations of the gray tones in the image along bands with different orientations. They assumed that each class has different frequencies in each band. The four orientations considered were the same as those of FingerCode and 256 values (power spectrum of the FFT) were obtained for each orientation, forming a feature vector of 1024 values.

Instead of considering bands of the image, Jin and Jin [49] processed the whole image with the DCT in disjoint blocks of 8 × 8 pixels obtaining 64 coefficients for each block. Thereafter, the low-frequency components were discarded (14 coefficients) and the rest went through a dimensionality reduction phase were the Fuzzy Cluster Means [5] and the Fisher Discriminant Analysis (FDA) [30] were used. Finally a feature vector of 30 features was obtained, but few details on the whole process were given. Even less implementation details were given in [110] were the GHM multiwavelet transform was proposed for fingerprint classification and a comparison with the scalar Daubechies wavelet was carried out. They were also applied to the whole image and a feature selection process based on a co-evolutionary algorithm was carried out to reduce the number of features (which was not reported). Curvelet transform has also been considered for FE in [72], where five Curvelet scales were obtained from which 16 texture-based features were computed using gray-level co-occurrence matrices in addition to the 49 direction features that were obtained from the coefficients of the transform.

5. Classification approaches

After FE, the classification phase can be carried out. In this phase, the set of previously labeled fingerprints processed with the corresponding FE method form the training set used to learn a classifier. Once the classifier is obtained, it can be used to classify new fingerprints, providing that their features have been also successfully captured.

Many classification approaches have been proposed in the literature, incorporating different ML approaches. Recall that Table 1 summarizes the main approaches published showing the type of classifier used and the databases used for testing the classifier. As it can be observed in the table, most of the experimental studies conducted have considered the National Institute of Standards and Technology (NIST) databases for benchmarking since their public release. Among them, NIST Special Database 4 (NIST-4) [117] and NIST Special Database 14 (NIST-14) [115] are the most popular ones, although NIST Special Database 9 (NIST-9) [118] has also been considered by some authors. Recent approaches have also adapted the fingerprints of the Fingerprint Verification Competition (FVC) [74, 75, 76, 12] to the evaluation of classification systems. The rest of the studies (mostly the pioneering ones) have used other databases, including private databases and other NIST releases, whose specific version was not fully specified.

Fig. 16 shows an snapshot of the evolution of the field: It represents each different approach as a circle, which is white if the approach consists of a single classifier, or black if it represents a multiclassifier (integrating several classifiers from the same or from distinct types). The location of the point determines both its year of publication and the family of ML techniques in which it can be categorized. If an approach includes techniques from more than one family, it is replicated as necessary. The families defined are:

- **Syntactic**: This family includes approaches based on the extraction of symbols from the features of the fingerprints. These symbols are analyzed by finite automaton or grammars to infer a class for the fingerprint (it is strongly related with syntactic-coding FE methods).
- **Artificial Neural Networks (ANNs)**: This family encompasses approaches based on ANNs [8].
- **Graph Matching**: Methods that incorporate graph matching techniques (such as the use of masks or graph-based prototypes) are assigned to this family.
- **Structural Models**: This family includes approaches which build structural models to represent the knowledge of the classes of fingerprints. Typical examples include Decision Trees (DTs) and Hidden Markov Models (HMMs).
- **Nearest Neighbor (NN)**: The approaches in this family are based on the computation of a similarity measure (distance) between the test fingerprint and several prototypes representing each class. Test fingerprints are assigned to the class of the most similar prototypes.
- **Support Vector Machines (SVMs)**: This family includes approaches based on SVMs [109].
- **Other**: This family includes all the approaches that are not represented by other families. It includes other statistical based techniques, such as linear discriminant analysis, Bayesian classifiers and so forth.
- **Fixed Classification**: A last family includes those approaches which do not follow a learning process to built the classifier. In contrast with the rest of techniques, the ones in this family define a fixed set of criteria (for example: *A fingerprint with no cores is an Arch*) to classify the fingerprints, regardless of the samples available for training.

As it can be observed in Fig. 16, there is a clear evolution from syntactic approaches to ANNs until the end of the past century. Since then, ANNs techniques have shared their dominant role with the rest of the families, from which it is possible to highlight NN-based approaches and SVMs. Finally, it is important to note the role of fixed classification techniques, which have shown a continuous development over these four decades. Through the rest of this section, the eight families chosen are described, pointing out the most relevant approaches of each one.

5.1. Syntactic approaches

Some of the earlier approaches for fingerprint classification are based on the use of grammars based on the symbols obtained with syntactic-coding FE methods. These symbols are then analyzed using a finite automaton or a grammar, whose final output is the class to which the fingerprint is classified.

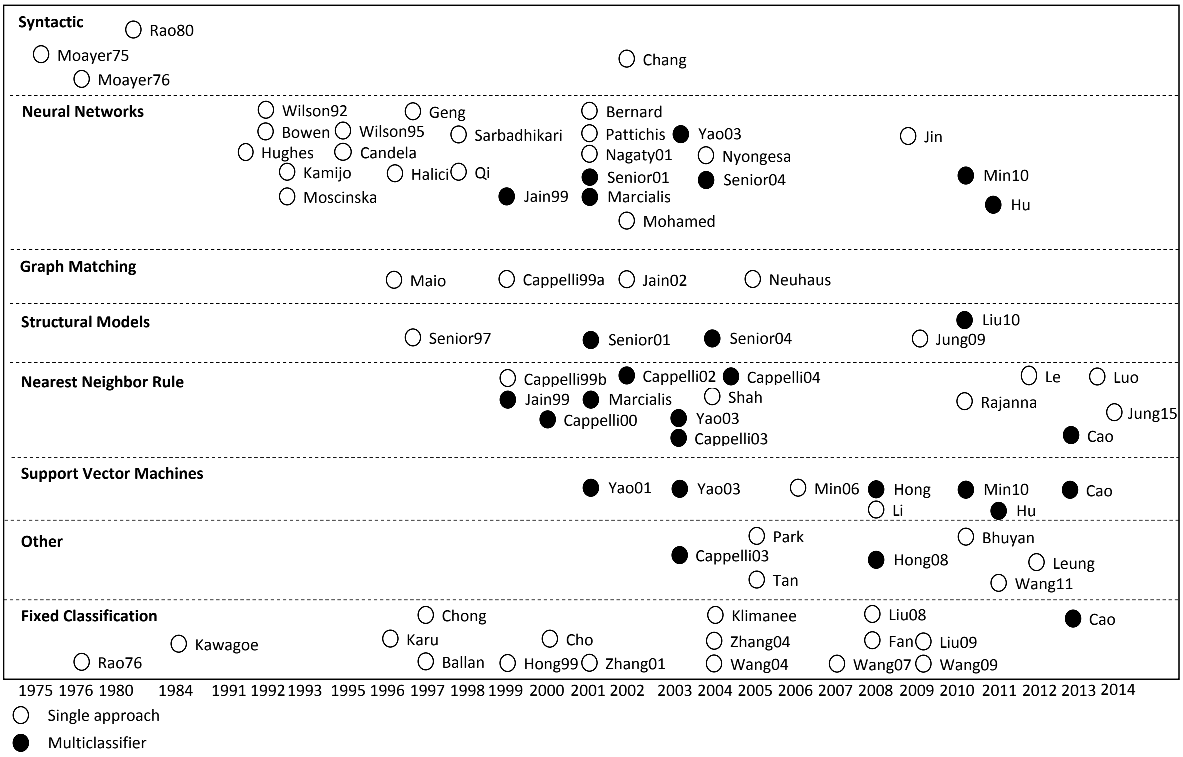


Figure 16: Map of fingerprint classification techniques.

Moayer and Fu [82] defined context-free grammars to describe the features extracted from the fingerprints, whereas stochastic grammars were used in [82]. A related approach, based on the analysis of the dominant direction of the ridges in each sector of the image was presented in [97].

There are very few recent syntactic approaches. One of them is [22], in which regular expressions are defined for 7 classes and are used to built a nondeterministic finite automata.

5.2. Neural networks approaches

A large number of the existing fingerprint classification approaches includes or are based on the use of ANNs, specially since the nineties. The initial approaches [43, 9, 122, 54, 121, 100, 86] proposed different architectures of multilayer perceptrons (MLPs) to tackle the problem. Moscinska and Tyma [85] introduced the use of Kohonen's Self Organizing Maps (SOMs) for this task, whose use has been continued in other works [36, 4]. In [10], where PCASYS was presented, a Probabilistic ANN combined with an auxiliary ridge-tracing module designed to detect whorl fingerprints was used. Since PCASYS has been a milestone in this area, its classifier has been also considered in further studies [120, 93, 102].

ANNs have also been considered as a component classifier in several multiclassifiers: [46] or [78] (MLPs + NN), [102, 103] (Probabilistic ANNs + Structural Models), [124] (recurrent ANNs + SMVs + NN), [81] (SOMs + SVMs) and [42] (MLPs + SVMs). Other advanced models of ANNs developed for fingerprint classification includes [35] and [84], describing the use of fuzzy ANNs; and [49], presenting radial basis function ANNs, whose comparative study was carried out in [91]. Another study comparing the performance of MLPs and other ANNs models (bidirectional associative memories, Hopfield networks and SOMs) was presented in [59].

5.3. Graph matching approaches

Graph Matching approaches are based on the comparison of graph-based patterns with the original fingerprints. These patterns can be compared with the full fingerprint or with a subregion of it. A dissimilarity measure is usually considered to determine how similar are the fingerprints to the patterns defined. The final classification is then obtained using the dissimilarities computed. Notice that this family is strongly related with segmentation-based orientation FE, and the matching phase could be considered either as a part of the FE or as a part of the classification.

Maio and Maltoni [73] defined an inexact graph matching approach based on the construction of a relational graph, built from the information extracted after the segmentation of the image. This approach was extended in [15], where 5 different costs were obtained related to each type of fingerprint that can be used to train another ML technique (for example, an ANN or a DTs).

Jain and Minut [45] presented a different approach based on the flow fields. Here, several kernel curves are used instead of masks. They are defined using splines, and fitted to the image until the kernel curve with the best fit is found. Another related approach is [87], where a graph edit distance is used to compute differences between a directional variance graph computed from the fingerprint and prototype graphs constructed by manually selecting promising candidates from each class.

5.4. Structural model approaches

Another kind of fingerprint classification methods includes algorithms for the construction of symbolic data structures, such as DTs or graphs, which can be termed as structural models.

In [101], HMMs are introduced to analyze the features obtained at intersection lines traced from the fingerprint. Later, in [102, 103], this model was enhanced with the introduction of multiple HMMs, DTs and Probabilistic ANNs, joined into a multiclassifier. In [52], HMMs were also considered.

DTs were used as basic pieces for an Adaboost [31] algorithm in [67], where features obtained at different scales of the image were considered to build an ensemble of ensembles.

5.5. Nearest neighbor rule approaches

Some approaches based on NN classification have been incorporated to this field. Most of them are founded on the computation of a distance measure between the fingerprints, to determine which ones are the most similar. A good example of its classic definition can be seen in [95], where NN classifier is used to evaluate the performance of some FE techniques (even though the comparative study is not exhaustive).

Cappelli et al. proposed the use of the Multi-space KL transform (MKL) [17] to preprocess and adapt the representation of the training data before applying a NN classifier. This approach was further developed with the inclusion of NN multiclassifiers and combinations of MKL transform and graph-based techniques [18, 20].

NN classifier has been also the base for two-stage classification schemes, such as in [46] and [78], where it is combined with ANNs to build a multiclassifier, or in [21], where the combination is performed with a clustering-based classifier.

Another related approach was presented in [104], where a simple NN classifier is considered, which used as training data the instances marked as support vectors by a SVM, instead of the full training set. Since support vectors are supposed to represent all the available information about the boundaries between classes, it is expected that they would offer a good performance.

Other recent approaches have also considered the usage of NN to classify fingerprints [11, 72, 53]. In the case of Jung and Lee [53] only five prototypes are used in the reference set, one for each class.

5.6. Support vector machines approaches

Several of the most recent approaches for fingerprint classification are based on SVMs, a high performance pattern classification technique for two-class problems. Since fingerprint classification is a multi-class problem, these approaches need to define different mechanisms to extend the two-class capabilities of SVMs to tackle multiple classes, for example using decomposition techniques [71, 32].

In [123], Yao et al. used SVMs that were combined using error correcting output codes. This approach was extended in [124], incorporating NN rule and recurrent ANNs to the multiclassifier. Another combined approach was proposed by Min et al. in [80, 81] using *one-versus-all* (OVA) strategy (the problem is divided into a subproblem per class, where each subproblem is defined as the classification of test samples into an specific class or to the rest). Different from the original OVA model, a SOM is then introduced to determine the final classification. OVA strategy was also used in [38], where Naive Bayes classifier is used to rank the different SVMs and choose which one will determine the class of the fingerprint. Other recent approaches using SVMs are [63] and [42], where Genetic Programming is used to refine the set of features that will be used in the classification phase. Otherwise, no details are given in [11] on how the multi-class problem is faced with SVMs.

5.7. Other approaches

Besides the rest of techniques mentioned before, there are still other classification techniques that do not match into any of these families. Among them, there are several recent approaches that are worth to be mentioned from the classification point of view.

One example is the approach presented in [106], where Genetic Programming is applied to the full OM to extract a set of complex features. After the evolutionary process, the features obtained are used to built a Bayesian classifier. In [92], Nonlinear Discriminant Analysis is introduced to reduce the dimensionality of very large feature vectors extracted from the images. The combination of several related techniques is presented in [61], where Fisher's Linear Discriminant Analysis and Quadratic Discriminant Analysis are introduced to improve the results of a Bayesian classifier.

Other relevant approach is [110], where evolutionary feature selection is introduced to improve the performance of a simple discriminant classifier using Mahalanobis distance.

5.8. Fixed classification approaches

Fixed classification approaches differ from the rest of techniques because the classification process is directly integrated within the FE process. Hence, these techniques do not require a training set to build a classification model. Instead, a fixed set of rules is defined to decide the class of the fingerprint.

Rao [98] proposed a method in which a binary version of the fingerprints is obtained. Depending on the number and position of 1's and 0's in the binary image, the fingerprint is classified into four different classes. Kawagoe and Tojo [57] developed a method based on ridge flow lines and its orientation. Later, Karu and Jain [55] proposed a method based on the number and position of the SPs (cores and deltas) detected in the images: Arch fingerprints have no cores or deltas, loops and tented arches contain one core and one delta, and and whorls and twin loops have two cores and two deltas. Several variations of this scheme can be found in [2, 58, 113, 66, 28, 112], and in [111], where a new type of SPs (*coredelta*) is defined.

Chong et al. [25] presented a model used to analyze fingerprint ridge lines through the use of splines. In this approach, the classification is performed by tracing the resulting curves, and analyzing changes in their direction. These changes are used to characterize the class to which each fingerprint belongs. Later, in [24], Cho et al. presented a different approach in which only the number of cores and the distance between them (if two cores are found) are considered.

Hong and Jain [39] developed a method using SPs and several characteristics extracted from the ridges. Similar approaches, combining information from SPs and ridges (or pseudo-ridges) can also be found in [125, 126]. Another recent approach, presented in [70], describes the use of a two-level set of rules, based on both SPs and ridge count. In [11], fixed rules are used to decide whether a fingerprint is a Whorl and whether a loop is a Right or Left one.

6. Discussion

As we have shown along this paper, a large number of different approaches have been considered for fingerprint classification, both in terms of FE methods and classification techniques. Such a wide variety has brought about an interesting collaboration between two different research areas: Image Processing and Machine Learning. However, even though many approaches have been proposed, what we have noticed when carrying out this review is that most of the papers on fingerprint classification leave too much to the imagination of the reader and hide (deliberately or not) many aspects that are important in order to be reusable by other researchers. This behavior may be due to the fact that companies supporting the research are not interested in completely revealing the method used, which does not help in the development of the area. There several points that stress this fact:

a. **Lack of details in the descriptions**. A great number of papers does not provide the sufficient details so as to be able to replicate the whole method. Some of them focus on the FE phase, giving few details of the classification model considered; on the contrary, others overlook the FE phase in favor of the latter. Even in well-known papers in which the methods are appropriately described [46, 15] there are steps which are not clarified. For example, in [46] the FingerCode method is thoroughly described but details about the OM extraction method, its smoothing or the covariance-based reference point detection remain unexplained. Similarly, Cappelli et al. [15] presented an exhaustive description of their method, but the masks defined for each type of fingerprint were only provided in a diagram, and hence measurements should be taken directly from the paper as well as how the rotation and translation affects them was not clearly explained. In the case of the works of Senior [101, 102], whose method is easily understandable, it does not detail how the features used are computed neither what type of thinning is carried out, which leaves too much decisions for the practitioner; the same occurs in, among others, [78, 124], in which the computation of the features and the final length of the feature vector is not reported. There are other works such as [63] using complex models based on previous works [62] to extract the features; such a complex models overlook too many aspects. This behavior can also be observed in the classifier learning phase [106, 92]. There are cases in which although they are based on the same original work (Li [63] and Liu [67], both uses complex-filters [88]) their approach differs even in that apparently common procedure. Even though this list is not exhaustive, it could be translated to almost all the papers reviewed, which puts out the difficulty in researching on this topic and developing new methods based on previous works. At last, there is another aspect that is mentioned in few papers [67, 61], the segmentation of the fingerprint image, that is, how the fingerprint is separated from the background [94]. This is not one of the hardest task, but in many cases it is not clear whether this task has been carried out or not and how it has been accomplished and thereafter used along the algorithm.

b. **Difficulties in their reimplementation**. As a consequence of the non-detailed description of the methods one cannot implement these methods as they are without making decisions based on their own knowledge. This problem is accentuated because the only code of fingerprint classification available for researchers is PCASYS, which despite its good documentation, is written in C language and it is rather difficult to work with, add code or modify it; in addition, it is a common practice not to share any source code with the rest of the papers. Besides these problems, there is another common obstacle, also present in many image processing and ML papers: the values used in the parameters of each method are not usually reported (there are cases where several parameter values are reported, but not all of them). Hence, although one could manage to implement the corresponding algorithm they should have to set the parameters arbitrarily or study their possible values.

c. **Non-standard evaluation procedures**. In addition to the above mentioned problems, there is another one which appears before them: one does not know which of the presented methods in the literature may be the best or could serve for their problem. This is a consequence of the way in which the performance evaluation is carried out. In this regard, the evaluation measure considered, i.e., the commonly used accuracy rate (percentage of correctly classified fingerprints), is the only point in which all the papers agree, whereas all the other elements of the experimental frameworks differ from one work to another. As it can be seen in Table 1 a variety of different databases have been used for testing purposes, even though NIST-4 database has been the most extended one. However, even in those papers considering the same database different evaluation modes have been considered: this database is composed of 2000 labeled fingerprints (400 of each class), having two captions of each one (making a total of 4000 images). It should be noticed that some of the fingerprints are labeled with two classes (350, that is, the 17.5%), which has produced different interpretations. Most of the authors have considered only the first label for training, but differences are found in the testing phase where only the first label is considered in [15] for example, whereas matching any of both labels is regarded as a correct classification in other works [55, 46, 126, 92, 38, 63]. Differences are also found regarding how the training and testing sets are formed. For example, in [55, 126] no training is carried out, and hence all the 4000 fingerprints are used for testing. Otherwise in [46, 63, 106, 92] the pair of captures from the first 1000 fingerprints (2000 examples in total) are considered for training and the remaining ones for testing; but exceptions such as [38] exists, where the first captures of the 2000 fingerprints are used for training and the second ones for testing. The procedure followed in [15] is even more different since some fingerprints are discarded in order to achieve a real distribution of fingerprints instead of a uniform one (1204 fingerprints are considered). Additionally, in many works the accuracy over the four class classification problem is also tested, which is easier since arch and tented-arch fingerprints are fused in a single class. Despite of such a large differences, it is usual to find comparisons among papers where accuracies obtained from other papers are presented to show how good the proposed method is. However, this comparison may be misleading due to the usage of different experimental frameworks (note that we only deal with NIST-4 database, but there are papers which do not consider this database but other). Therefore, even though all the papers present some type of results of the fingerprint classification problem, it is difficult to know which one would perform better. Moreover, it is also difficult to know whether a method would be implementable by following the description of the corresponding paper.

On this account, we aim to experimentally show this problem in the second part of this paper [34]. We have considered a set of relevant algorithms covering most of the years, features and classification techniques as a representative models to carry out an objective experimental study. We have carefully implemented all these methods as accurately as possible following the papers mentioned (given that none of the authors provided us with the source code of their methods) and we aim to test them in a common framework where we can obtain meaningful conclusions. We should mention that we do not doubt on the results shown in the corresponding papers, but in the majority of the cases they are not comparable since different methodologies are considered. Furthermore, in this way we are not only comparing the quality of different methods but their reimplementability which is a very important point that too many times is overlooked. There are very interesting papers on fingerprint classification with good results that are nearly impossible to be used by other researchers due to a lack of details (parameters, explanations on some phases, etc.). In the second part of this paper [34] we also want to focus on this point so that the most practical approaches are recommended for future developments in the field. This way, their results will be objectively analyzed and their validity for their usage by other researchers will be shown.

In this manner, we will not only present a new experimental study on several FE and classification methods, but also a new experimental framework that can be used by researchers in the field so that researchers need not reimplement previous methods in order to test their approaches. Moreover, we provide a number of baseline methods implemented in the library available in the web-page associated with these papers (`http://sci2s.ugr.es/fingerprintClassification`).

Leaving all these problems in fingerprint classification aside, we have to stress out some future research lines.

- We have focused this review on the FE techniques that have been considered for fingerprint classification, leaving, for the sake of brevity, out of the scope of this work those models developed for other purposes in fingerprint identification. However, this is in fact one of the issues that new fingerprint classification developments should take into account. There are many preprocessing algorithms for fingerprint enhancement [50], OM extraction [114, 40, 69] or SP and reference point detection [1, 114] that have been developed in parallel with fingerprint classification literature. Most of them provide ways of obtaining more accurate features that should serve new FE methods being more robust against the quality of the fingerprints. Moreover, there are other features that have been used in other indexing schemes such as average ridge-distance [47, 68], which may be exploited in fingerprint classification.
- Even though it is impractical to analyze each work one by one, we should emphasize the fact that there are several works in which there is a clear lack of novelty. For example, there are many works that have considered FingerCode as FE method, introducing few variations in the algorithm. This does not mean that methods like FingerCode need to be left aside, but rather than introducing small

improvements it would be better to study their combination with other FE methods from different categories.

- The complexity added to the models presented should be better justified. There are works with an added complexity which does not result in a clear improvement with respect to previous approaches, and hence its contribution is not meaningful.
- More related to our previous discussion, authors should consider sharing the source codes of their methods (in addition to being clearer in their description) so that one need not reimplement every method that is going to be used. This way, much more complete models could be developed without adding more complexity.

7. Conclusions

In this paper we have reviewed the fingerprint classification literature which started in 1975. During these years, a variety of methods have been considered, both in terms of FE and classification, which is the double perspective we have considered. For this reason, we have classified the different FE methods in a new taxonomy and grouped the different classification approaches. Additionally, we have thoroughly reviewed two of the most important features used in fingerprint classification: orientations (OMs) and SPs, for which we have also presented two taxonomies where the methods considered in the fingerprint classification problem can be placed.

This review and the aim of finding which method performs better for the problem have led us to the discussion in Section 6, where we have outlined some of the key problems we have perceived in the specialized literature as well as possible future research lines. There is no way of knowing which method would perform better because each author tends to consider different experimental frameworks (different database, different usages of the same database, etc.); moreover, most of the papers lack of detailed descriptions of the models presented, which directly affects their reimplementability by other researchers.

In this context, we aim to carry out an experimental comparison over some of the most relevant methods, both from FE and classification. This way, we will be able to analyze which methods would be easier to be considered for future developments in the field as well as we will provide an objective evaluation of the different models in a common experimental framework, which can be used in future works. We think that such a deep study is necessary in order to continue the research of this mature problem. Due to the space needed in order to properly present these experiments, the second part of this paper [34] is devoted to it.

Acknowledgment

This work was supported by the Research Projects CAB(CDTI), TIN2011-28488, and TIN2013-40765-P. D. Peralta holds an FPU scholarship from the Spanish Ministry of Education and Science (FPU12/04902).

References

[1] V. Areekul, K. Suppasriwasuseth, S. Jirachawang, The new focal point localization algorithm for fingerprint registration, in: 18th International Conference on Pattern Recognition. ICPR 2006., vol. 4, 2006.

[2] M. Ballan, F. A. Sakarya, B. L. Evans, A fingerprint classification technique using directional images, in: Proceedings of the 31st Asilomar Conference on Signals, Systems & Computers, Pacific Grove, CA, November 2-5, 1997.

[3] A. M. Bazen, S. H. Gerez, Systematic methods for the computation of the directional fields and singular points of fingerprints, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (7) (2002) 905–919.

[4] S. Bernard, N. Boujemaa, D. Vitale, C. Bricot, Fingerprint classification using kohonen topologic map, in: Proceedings of the IEEE International Conference On Image Processing, Thessaloniki, Greece, October 7-10, 2001.

[5] J. C. Bezdek, Pattern Recognition With Fuzzy Objective Function Algorithms, New York: Plenum, 1981.

[6] B. Bhanu, X. Tan, Fingerprint indexing based on novel features of minutiae triplets, IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (5) (2003) 616–622.

[7] J. Bigün, Pattern recognition in images by symmetries and coordinate transformations, Computer Vision and Image Understanding 68 (3) (1997) 290–307.

[8] C. M. Bishop, Pattern Recognition and Machine Learning (Information Science and Statistics), Springer, 2007.

[9] J. Bowen, The home office automatic fingerprint pattern classification project, in: IEE Colloquium on Neural Networks for Image processing Applications, London, UK, October 28, 1992.

[10] G. T. Candela, P. J. Grother, C. I. Watson, R. A. Wilkinson, C. L. Wilson, PCASYS - A pattern-level classification automation system for fingerprints, Tech. rep., UNIST Interagency/Internal Report (NISTIR) - 5647 (1995).

[11] K. Cao, L. Pang, J. Liang, J. Tian, Fingerprint classification by a hierarchical classifier, Pattern Recognition 46 (12) (2013) 3186–3197.

[12] R. Cappelli, M. Ferrara, A. Franco, D. Maltoni, Fingerprint verification competition 2006, Biometric Technology Today 15 (7-8) (2007) 7–9.

[13] R. Cappelli, M. Ferrara, D. Maltoni, Minutia cylinder-code: A new representation and matching technique for fingerprint recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 32 (12) (2010) 2128–2141.

[14] R. Cappelli, M. Ferrara, D. Maltoni, Fingerprint indexing based on minutia cylinder-code, IEEE Transactions on Pattern Analysis and Machine Intelligence 33 (5) (2011) 1051–1057.

[15] R. Cappelli, A. Lumini, D. Maio, D. Maltoni, Fingerprint classification by directional image partitioning, IEEE Transactions on Pattern Analysis and Machine Intelligence 21 (5) (1999) 402–421.

[16] R. Cappelli, D. Maio, The state of the art in fingerprint classification, in: N. Ratha, R. Bolle (eds.), Automatic Fingerprint Recognition Systems, Springer, New York, 2004, p. 183205.

[17] R. Cappelli, D. Maio, D. Maltoni, Fingerprint classification based on multi-space KL, in: Proceedings of the Workshop on Automatic Identification Advances Technologies, Summit, NJ, October, 1999.

[18] R. Cappelli, D. Maio, D. Maltoni, Combining fingerprint classifiers, in: Proceedings of the 1st International Workshop on Multiple Classifier Systems, Cagliari, Italy, June 21–23, 2000.

[19] R. Cappelli, D. Maio, D. Maltoni, Multispace kl for pattern representation and classification, IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (9) (2001) 977–996.

[20] R. Cappelli, D. Maio, D. Maltoni, A multi-classifier approach to fingerprint classification, Pattern Analysis & Applications 5 (2002) 136–144.

[21] R. Cappelli, D. Maio, D. Maltoni, L. Nanni, A two-stage fingerprint classification system, in: Proceedings of the ACM SIGMM workshop on Biometrics methods and applications (WBMA'03), Berkeley, CA, November 2-8, 2003.

[22] J.-H. Chang, K.-C. Fan, A new model for fingerprint classification by ridge distribution sequences, Pattern Recognition 35 (6) (2002) 1209–1223.

[23] X. Chen, J. Tian, X. Yang, A new algorithm for distorted fingerprints matching based on normalized fuzzy similarity measure, IEEE Transactions on Image Processing 15 (3) (2006) 767–776.

[24] B. H. Cho, J. S. Kim, J. H. Bae, I. G. Bae, K. Y. Yoo, Core-based fingerprint image classification, in: Proceedings of the 15th International Conference on Pattern Recognition (ICPR'00), Barcelona, Spain, September 3-7, 2000.

[25] M. M. S. Chong, T. H. Ngee, L. Jun, R. K. L. Gay, Geometric framework for fingerprint image classification, Pattern Recognition 30 (9) (1997) 1475–1488.

[26] R. O. Duda, P. E. Hart, D. G. Stork, Pattern Classification, 2nd ed., John Wiley, 2001.

[27] P. G. Espejo, S. Ventura, F. Herrera, A survey on the application of genetic programming to classification, IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 40 (2) (2010) 121–144.

[28] L. Fan, S. Wang, H. Wang, T. Guo, Singular points detection based on zero-pole model in fingerprint images, IEEE Transactions on Pattern Analysis and Machine Intelligence 30 (6) (2008) 929–940.

[29] J. Feng, A. K. Jain, Fingerprint reconstruction: From minutiae to phase, IEEE Transactions on Pattern Analysis and Machine Intelligence 33 (2) (2011) 209–223.

[30] R. A. Fisher, The use of multiple measurements in taxonomic problems, Annals of eugenics 7 (2) (1936) 179–188.

[31] Y. Freund, R. E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, Journal of Computer and System Sciences 55 (1) (1997) 119–139.

[32] M. Galar, A. Fernández, E. Barrenechea, H. Bustince, F. Herrera, An overview of ensemble methods for binary classifiers in multi-class problems: Experimental study on one-vs-one and one-vs-all schemes, Pattern Recognition 44 (8) (2011) 1761 – 1776.

[33] M. Galar, A. Fernández, E. Barrenechea, H. Bustince, F. Herrera, A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches, IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 42 (4) (2012) 463–484.

[34] M. Galar et al., A survey of fingerprint classification II: Experimental analysis and ensemble proposal, Submitted.

[35] Z. J. Geng, W. C. Shen, Fingerprint classification using fuzzy cerebellar model arithmetic computer neural networks, Journal of Electronic Imaging 6 (3) (1997) 311–318.

[36] U. Halici, G. Ongun, Fingerprint classification through self organizing feature maps modified to treat uncertainties, in: Proceedings of the IEEE, vol. 84, 1996.

[37] E. R. Henry, Classification and Uses of Finger Prints, London: Routledge, 1900.

[38] J.-H. Hong, J.-K. Min, U.-K. Cho, S.-B. Cho, C. H. Leung, Fingerprint classification using one-vs-all support vector machines dynamically ordered with naive bayes classifiers, Pattern Recognition 41 (2008) 662–671.

[39] L. Hong, A. K. Jain, Classification of fingerprint images, in: Proceedings of the 11th Scandinavian Conference on Image Analysis, Kangerlussuaq, Greenland, June 7-11, 1999.

[40] Z. Hou, H.-K. Lam, W.-Y. Yau, Y. Wang, A variational formulation for fingerprint orientation modeling, Pattern Recognition 45 (5) (2012) 1915–1926.

[41] Z. Hou, W.-Y. Yau, Y. Wang, A review on fingerprint orientation estimation, Security and Communication Networks 4 (5) (2011) 591–599.

[42] J. Hu, M. Xie, Fingerprint classification based on genetic programming, in: International Conference on Computer Engineering and Technology(ICCET), Chengdu, China, April 16-18, 2010.

[43] P. Hughes, A. Green, The use of neural networks for fingerprint classification, in: Proceedings of the 2nd International

Conference on Artificial Neural Networks, Bournemouth, UK, November 18-20, 1991.

[44] A. K. Jain, Fundamentals of digital image processing, vol. 3, Prentice-Hall Englewood Cliffs, 1989.

[45] A. K. Jain, S. Minut, Hierarchical kernel fitting for fingerprint classification and alignment, in: Proceedings of the 16th International Conference on Pattern Recognition (ICPR'02), Quebec, Canada, August 11-15, 2002.

[46] A. K. Jain, S. Prabhakar, L. Hong, A multichannel approach to fingerprint classification, IEEE Transactions on Pattern Analysis and Machine Intelligence 21 (4) (1999) 348–359.

[47] X. Jiang, M. Liu, A. C. Kot, Fingerprint retrieval for identification, IEEE Transactions on Information Forensics and Security 1 (4) (2006) 532–542.

[48] X. Jiang, W. Yau, Fingerprint minutiae matching based on the local and global structures, in: Proceedings of the 15th International Conference on Pattern Recognition, vol. 2, IEEE, 2000.

[49] C. Jin, P. Jin, Fingerprint classification in DCT domain using RBF neural networks, Journal of Information Science and Engineering 25 (6) (2009) 1955–1962.

[50] S. Jirachaweng, Z. Hou, W.-Y. Yau, V. Areekul, Residual orientation modeling for fingerprint enhancement and singular point detection, Pattern Recognition 44 (2) (2011) 431–442.

[51] I. Jolliffe, Principal component analysis, Wiley Online Library, 2005.

[52] H.-W. Jung, J.-H. Lee, Fingerprint classification using the stochastic approach of ridge direction information, in: IEEE International Conference on Fuzzy Systems(FUZZIEEE'09), Jeju Island, South Korea, August 20-24, 2009.

[53] H.-W. Jung, J.-H. Lee, Noisy and incomplete fingerprint classification using local ridge distribution models, Pattern Recognition 48 (2) (2015) 473–484.

[54] M. Kamijo, Classifying fingerprint images using neural network: Deriving the classification state, in: Proceedings of the EEE International Conference on Neural Networks, San Francisco, CA, March 28 – April 1, 1993.

[55] K. Karu, A. K. Jain, Fingerprint classification, Pattern Recognition 29 (3) (1996) 389–404.

[56] M. Kass, A. Witkin, Analyzing oriented patterns, Computer vision, graphics, and image processing 37 (3) (1987) 362–385.

[57] M. Kawagoe, A. Tojo, Fingerprint pattern classification, Pattern Recognition 17 (1984) 295–303.

[58] C. Klimanee, D. T. Nguyen, Classification of fingerprints using singular points and their principal axes, in: Proceedings of the International Conference on Image Processing (ICIP'04), Singapore, October 24-27, 2004.

[59] T. Kristensen, J. Borthen, K. Fyllingsnes, Comparison of neural network based fingerprint classification techniques, in: International Joint Conference on Neural Networks (IJCNN'07), Orlando, FL, August 12-17, 2007.

[60] T. H. Le, H. T. Van, Fingerprint reference point detection for image retrieval based on symmetry and variation, Pattern Recognition 45 (9) (2012) 3360 – 3372.

[61] K. Leung, C. H. Leung, Improvement of fingerprint retrieval by a statistical classifier, IEEE Transactions on Information Forensics and Security 6 (1) (2011) 59–69.

[62] J. Li, W. Y. Yau, H. Wang, Constrained nonlinear models of fingerprint orientations with prediction, Pattern Recognition 39 (1) (2006) 102–114.

[63] J. Li, W.-Y. Yau, H. Wang, Combining singular points and orientation image information for fingerprint classification, Pattern Recognition 41 (1) (2008) 353–366.

[64] X. Liang, A. Bishnu, T. Asano, A robust fingerprint indexing scheme using minutia neighborhood structure and low-order delaunay triangles, IEEE Transactions on Information Forensics and Security 2 (4) (2007) 721–733.

[65] T. Lindeberg, Edge detection and ridge detection with automatic scale selection, International Journal of Computer Vision 30 (2) (1998) 117–156.

[66] L.-M. Liu, C.-Y. Huang, D. C. D. Hung, A directional approach to fingerprint classification, International Journal of Pattern Recognition and Artificial Intelligence 22 (2) (2008) 347–365.

[67] M. Liu, Fingerprint classification based on adaboost learning from singularity features, Pattern Recognition 43 (2010) 1062–1070.

[68] M. Liu, X. Jiang, A. C. Kot, Efficient fingerprint search based on database clustering, Pattern Recognition 40 (6) (2007) 1793–1803.

[69] M. Liu, P.-T. Yap, Invariant representation of orientation fields for fingerprint indexing, Pattern Recognition 45 (7) (2012) 2532–2542.

[70] W. Liu, Z. Ye, H. Chen, H. Li, Ridgeline based 2-layer classifier in fingerprint classification, in: International Workshop on Intelligent Systems and Applications, Wuhan, China, May 23-24, 2009.

[71] A. Lorena, A. Carvalho, J. Gama, A review on the combination of binary classifiers in multiclass problems, Artificial Intelligence Review 30 (1-4) (2008) 19–37.

[72] J. Luo, D. Song, C. Xiu, S. Geng, T. Dong, Fingerprint classification combining curvelet transform and gray-level cooccurrence matrix, Mathematical Problems in Engineering 2014 (Article ID 592928) (2014) 15 pages.

[73] D. Maio, D. Maltoni, A structural approach to fingerprint classification, in: Proceedings of the 13th International Conference on Pattern Recognition (ICPR'96), Vienna, Austria, August 25-29, 1996.

[74] D. Maio, D. Maltoni, R. Cappelli, J. L. Wayman, A. K. Jain, FVC2000: Fingerprint verification competition, IEEE Transactions on Pattern Analysis Machine Intelligence 24 (3) (2002) 402–412.

[75] D. Maio, D. Maltoni, R. Cappelli, J. L. Wayman, A. K. Jain, FVC2002: Second fingerprint verification competition, in: Proceedings of the 16th International Conference on Pattern Recognition (ICPR'02), Quebec City, Canada, August 11-15, 2002.

[76] D. Maio, D. Maltoni, R. Cappelli, J. L. Wayman, A. K. Jain, FVC2004: Third fingerprint verification competition, in: Proceedings of the First International Conference on Biometric Authentication (ICBA'04), Hong Kong, China, July 15-17, 2004.

[77] D. Maltoni, D. Maio, A. Jain, S. Prabhakar, Handbook of fingerprint recognition, Springer-Verlag New York Inc, 2009.

[78] G. L. Marcialis, F. Roli, P. Frasconi, Fingerprint classification by combination of flat and structural approaches, in: Proceedings of the 3rd International Conference on Audio- and Video-Based Biometric Person Authentication, Halmstad, Sweden, June 6–8, 2001.

[79] B. M. Mehtre, N. N. Murthy, S. Kapoor, B. Chatterjee, Segmentation of fingerprint images using the directional image, Pattern Recognition 20 (4) (1987) 429–435.

[80] J.-K. Min, J.-H. Hong, S.-B. Cho, Effective fingerprint classification by localized models of support vector machines, in: Proceedings of the International Conference on Advances in Biometrics (ICB'06), Hong Kong, China, January 5-7, 2006.

[81] J.-K. Min, J.-H. Hong, S.-B. Cho, Fingerprint classification based on subclass analysis using multiple templates of support vector machines, Intelligent Data Analysis 14 (3) (2010) 369–384.

[82] B. Moayer, K. Fu, A syntactic approach to fingerprint pattern recognition, Pattern Recognition 7 (1975) 1–23.

[83] B. Moayer, K. Fu, An application of stochastic languages to fingerprint pattern recognition, Pattern Recognition 8 (1976) 173–179.

[84] S. M. Mohamed, H. O. Nyongesa, Automatic fingerprint classification system using fuzzy neural techniques, in: Proceedings of the IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'02), Honolulu, HI, May 12-17, 2002.

[85] K. Moscinska, G. Tyma, Neural network based fingerprint classification, in: Proceedings of the Third International Conference on Artificial Neural Networks, Brighton, UK, May 25-27, 1993.

[86] K. A. Nagaty, Fingerprints classification using artificial neural networks: a combined structural and statistical approach, Neural Networks 14 (9) (2001) 1293–1305.

[87] M. Neuhaus, H. Bunke, A graph matching based approach to fingerprint classification using directional variance, in: Proceedings of the 5th International Conference on Audio- and Video-Based Biometric Person Authentication, Hilton Rye Town, NY, July 20-22, 2005.

[88] K. Nilsson, J. Bigun, Localization of corresponding points in fingerprints by complex filtering, Pattern Recognition Letters 24 (13) (2003) 2135–2144.

[89] M. Nixon, A. S. Aguado, Feature extraction & image processing, Elsevier, 2008.

[90] S. O. Novikov, V. S. Kot, Singular feature detection and classification of fingerprints using hough transform, in: SPIE 3346, Sixth International Workshop on Digital Image Processing and Computer Graphics: Applications in Humanities and Natural Sciences, 1998.

[91] H. O. Nyongesa, S. Al-Khayatt, S. M. Mohamed, M. Mahmoud, Fast robust fingerprint feature extraction and classification, Journal of Intelligent and Robotic Systems 40 (1) (2004) 103–112.

[92] C. H. Park, H. Park, Fingerprint classification using fast fourier transform and nonlinear discriminant analysis, Pattern Recognition 38 (4) (2005) 495–503.

[93] M. S. Pattichis, G. Panayi, A. C. Bovik, S.-P. Hsu, Fingerprint classification using an am-fm model, IEEE Transactions on Image Processing 10 (6) (2001) 951–954.

[94] D. Peralta, M. Galar, I. Triguero, O. Miguel-Hurtado, J. M. Benitez, F. Herrera, Minutiae filtering to improve both efficacy and efficiency of fingerprint matching algorithms, Engineering Applications of Artificial Intelligence 32 (2014) 37–53.

[95] U. Rajanna, A. Erol, G. Bebis, A comparative study on feature extraction for fingerprint classification and performance improvements using rank-level fusion, Pattern Analysis & Applications 13 (3) (2010) 263–272.

[96] A. R. Rao, A Taxonomy for Texture Description and Identification, Springer Verlag, 1990.

[97] K. Rao, K. Balck, Type classification of fingerprints: A syntactic approach, IEEE Transactions on Pattern Analysis and Machine Intelligence 2 (3) (1980) 223–231.

[98] M. Rao, Feature extraction for fingerprint classification, Pattern Recognition 8 (1976) 599–605.

[99] N. Ratha, R. Bolle (eds.), Automatic fingerprint recognition systems, Springer, 2004.

[100] S. N. Sarbadhikari, J. Basak, S. K. Pal, M. Kundu, Noisy fingerprints classification with directional FFT based features using MLP, Neural Computing and Applications 7 (2) (1998) 180–191.

[101] A. W. Senior, A hidden markov model fingerprint classifier, in: Proceedings of the 31st Asilomar Conference on Signals, Systems & Computers, Pacific Grove, CA, November 2-5, 1997.

[102] A. W. Senior, A combination fingerprint classifier, IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (10) (2001) 1165–1174.

[103] A. W. Senior, R. Bolle, Fingerprint classification by decision fusion, in: N. Ratha, R. Bolle (eds.), Automatic Fingerprint Recognition Systems, Springer, New York, 2004, pp. 207–227.

[104] S. Shah, P. S. Sastry, Fingerprint classification using a feedback-based line detector, IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 34 (1) (2004) 85–94.

[105] R. M. Stock, C. W. Swonger, Development and evaluation of a reader of fingerprint minutiae, Tech. rep., Cornell Aeronautical Laboratory, Technical Report CAL no. XM-2478-X-1 (1969).

[106] X. Tan, B. Bhanu, Y. Lin, Fingerprint classification based on learned features, IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 35 (3) (2005) 287–300.

[107] V. Torre, T. A. Poggio, On edge detection, IEEE Transactions on Pattern Analysis and Machine Intelligence (2) (1986) 147–163.

[108] F. Turroni, D. Maltoni, R. Cappelli, D. Maio, Improving fingerprint orientation extraction, IEEE Transactions on Information Forensics and Security 6 (3) (2011) 1002–1013.

[109] V. Vapnik, Statistical Learning Theory, New York: Wiley, 1998.

[110] J.-W. Wang, Evolutionary optimization approach for fingerprint classifications, in: Proceedings of the World Congress on Engineering, London, UK, July 6-8, 2011.

[111] L. Wang, M. Dai, Application of a new type of singular points in fingerprint classification, Pattern Recognition Letters

28 (13) (2007) 1640–1650.

[112] X. Wang, F. Wang, J. Fan, J. Wang, Fingerprint classification based on continuous orientation field and singular points, in: IEEE International Conference on Intelligent Computing and Intelligent Systems(ICIS'09), Shanghai, China, November 20-22, 2009.

[113] X. Wang, M. Xie, Fingerprint classification: An approach based on singularities and analysis of fingerprint structure, in: Proceedings of the First International Conference on Biometric Authentication (ICBA'04), Hong Kong, China, July 15-17, 2004.

[114] Y. Wang, J. Hu, D. Phillips, A fingerprint orientation model based on 2D fourier expansion (FOMFE) and its application to singular-point detection and fingerprint indexing, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (4) (2007) 573–585.

[115] C. I. Watson, NIST Special Database 14, Fingerprint Database, Tech. rep., U.S. National Institute of Standards and Technology (1993).

[116] C. I. Watson, M. D. Garris, E. Tabassi, C. L. Wilson, R. M. McCabe, S. Janet, K. Ko, User's Guide to NIST Biometric Image Software (NBIS), National Institute of Standards and Technology.

[117] C. I. Watson, C. L. Wilson, NIST Special Database 4, Fingerprint Database, Tech. rep., U.S. National Institute of Standards and Technology (1992).

[118] C. I. Watson, C. L. Wilson, NIST Special Database 9, Fingerprint Database, Tech. rep., U.S. National Institute of Standards and Technology (1992).

[119] J. H. Wegstein, An automated fingerprint identification system, NBS Special Publication 500-89.

[120] D. Wei, Q. Yuan, T. Jie, Fingerprint classification system with feedback mechanism based on genetic algorithm, in: Proceedings of the 14th International Conference on Pattern Recognition (ICPR'98), Brisbane, Qld, August 16-20, 1998.

[121] C. Wilson, J. Blue, O. Omidvar, Improving neural network performance for character and fingerprint classification by altering network dynamics, in: Proceedings of the World Congress on Neural Networks (WCNN'95), Washington, USA, July 17-21, 1995.

[122] C. L. Wilson, Massively parallel neural network fingerprint classification system, in: International Joint Conference on Neural Networks (IJCNN'92), Baltimore, MD, June 7–1, 1992.

[123] Y. Yao, P. Frasconi, M. Pontil, Fingerprint classification with combination of support vector machines, in: Proceedings of the 3rd International Conference on Audio- and Video-Based Biometric Person Authentication, Halmstad, Sweden, June 6–8, 2001.

[124] Y. Yao, G. L. Marcialis, M. Pontil, P. Frasconi, F. Roli, Combining flat and structured prepresentations for fingerprint classification with recursive neural networks and support vector machines, Pattern Recognition 36 (2) (2003) 397–406.

[125] Q. Zhang, K. Huang, H. Yan, Fingerprint classification based on extraction and analysis of singularities and pseudoridges, in: Proceedings of the Pan-Sydney Area Workshop on Visual Information Processing, Sydney, Australia, December, 2001.

[126] Q. Zhang, H. Yan, Fingerprint classification based on extraction and analysis of singularities and pseudo ridges, Pattern Recognition 37 (11) (2004) 2233–2243.