Pattern Analysis and Applications (2021) 24:433–444
https://doi.org/10.1007/s10044-020-00911-7

ORIGINAL ARTICLE

An ensemble of fingerprint matching algorithms based on cylinder codes and mtriplets for latent fingerprint identification

Danilo Valdes-Ramirez[1,2] · Miguel A. Medina-Pérez[1] · Raúl Monroy[1]

Received: 5 February 2020 / Accepted: 5 September 2020 / Published online: 6 October 2020

Abstract

Automatic latent fingerprint identification is beneficial during forensic investigations. Usually, latent fingerprint identification algorithms are used to find a subset of similar fingerprints from those previously captured on databases, which are finally examined by latent examiners. Yet, the identification rate achieved by latent fingerprint identification algorithms is far from those obtained by latent examiners. One approach for improving identification rates is the fusion of the match scores computed with fingerprint matching algorithms using a supervised classification algorithm. This approach fuses the results provided by different lower-level algorithms to improve them. Thus, we propose a fusion of fingerprint matching algorithms using a supervised classifier. Our proposal starts with two different local matching algorithms. We substitute their global matching algorithms with another independent of the local matching, creating two lower-level algorithms for fingerprint matching. Then, we combine the output of these lower-level algorithms using a supervised classifier. Our proposal achieves higher identification rates than each lower-level algorithm and their fusion using traditional approaches for most of the rank values and reference databases. Moreover, our fusion algorithm reaches a Rank-1 identification rate of 74.03% and 71.32% matching the 258 samples in the NIST SD27 database against 29,257 and 100,000 references, the two largest reference databases employed in our experiments.

Keywords Latent fingerprint identification · Match-score fusion · Fingerprint matching algorithm

1 Introduction

Fingerprint matching algorithms are at the core of fingerprint verification and latent fingerprint identification. Fingerprint verification performs a one-to-one comparison to verify a claimed identity with a previously captured fingerprint [19, 24]. Contrarily, latent fingerprint identification performs a one-to-many comparison searching for the most similar fingerprints, enrolled in a reference database, to a latent fingerprint [17, 24]. Most of the fingerprint matching algorithms have been developed for fingerprint verification [36]. As a consequence, fingerprint verification algorithms have reported a high accuracy (see the results in the Fingerprint Verification Competition FVC-onGoing [7]), while latent fingerprint identification algorithms have considerable room for improvement.

Latent fingerprint identification has essential applications for forensic investigations. For example, missing persons can be traced through their last fingerprints found at some places as latent fingerprints, police departments conduct criminal investigations using latent fingerprints, and latent fingerprints are pieces of evidence during a trial. Due to the sensitivity of its applications, latent fingerprint identification requires high identification rates. Nevertheless, the results in Table 1 indicate that the identification rates reported in the literature are insufficient for meeting this requirement.

One infrequently explored approach for improving the identification rates of latent fingerprint identification algorithms involves ensembles of fingerprint matching algorithms using supervised learning. An ensemble in machine

✉ Danilo Valdes-Ramirez
a01746752@itesm.mx; dvramirez@gmail.com

Miguel A. Medina-Pérez
migue@tec.mx

Raúl Monroy
raulm@tec.mx

[1] Tecnologico de Monterrey, School of Science and Engineering, Carretera al Lago de Guadalupe Km. 3.5, 52926 Atizapán de Zaragoza, Estado deMéxico, Mexico

[2] Department of Computer Sciences, Universidad de Ciego de Ávila, 65100 Ciego de Ávila, Cuba

Table 1 Recent Rank-1 identification rates reported in the literature. These results were reported using reference databases with different sizes. Usually, forensic investigations use large reference databases. However, the higher the background database is, the lower the identification rate for any latent fingerprint identification algorithm

Proposal	Rank-1 identification rate (%)	Reference database size
Improving automated latent fingerprint identification using extended minutia types [18]	92.7	151
Latent fingerprint recognition: role of texture template [2]	78.2	10,000
Latent fingerprint matching using distinctive ridge points [6]	70.9	29,257
End-to-end latent fingerprint search [3]	65.7	100,000

learning is a general method that combines lower-level algorithms using a high-level algorithm to achieve higher accuracy than the lower-level algorithms [35]. In fingerprint matching research, this approach can be classified as a fusion scheme.

Although supervised classifiers have been used as fusion algorithms for fingerprint verification [4, 8, 13, 24], they are less frequent for latent fingerprint identification [30]. In the latent fingerprint identification problem, an algorithm must determine a list of references (fingerprints previously enrolled in a background database) with the highest match score (continuous value) to the sample (latent fingerprint) but supervised classifiers determine a class (nominal value) for each example [20]. However, there exist supervised classifiers that compute the probability of belonging to each class. In our proposal, we use the probability of belonging to the matching class as the fused match score.

In this work, we propose an ensemble of two fingerprint matching algorithms with the aim of improving the Rank-1 identification rate for latent fingerprint identification. We start by combining two different minutia descriptors, which have shown to be suitable for latent fingerprint identification [36]: Minutia cylinder codes (MCC) [5] and mtriplet [25]. We have substituted the global matchers of these algorithms with the algorithm deformable minutiae clustering (DMC) [26] to improve their performances. Moreover, we have tuned the parameters of both minutia descriptors for latent fingerprint identification because they were proposed for fingerprint verification. For fusing the match scores, we propose a multilayer perceptron, although we explored other fusion approaches in our experimentations. In summary, the contributions of this work are as follows:

1. We propose a novel match-score fusion scheme for latent fingerprint identification, which outperforms the Rank-1 identification rates output by the methods reported in the literature that match only minutiae. Such fusion scheme is based on an ensemble of two state-of-the-art fingerprint matching algorithms via a multilayer perceptron (MLP). We obtained the best fusion algorithm from the comparison between several supervised classifiers, various deep learning models, and two traditional fusion algorithms (a weighted sum and a product rule). This work explores 30 traditional supervised classifiers and 30 deep learning models as fusion algorithms for latent fingerprint identification.
2. We describe a subset of four attributes constructed from the comparison results between fingerprint pairs using the matching algorithms. These attributes characterize the similarities between each sample and reference.
3. We provide an experimental study comparing our results against those reached by the lower-level algorithms, other fusion algorithms, and the previous methods reported in the literature using three latent fingerprint databases: NIST SD27 [9], GCDB [18], and MOLF DB4 [32], and different references databases: NIST SD27 [9], NIST SD4 [38], NIST SD14 [37], GCDB [18], and MOLF DB1 [32]. Our proposal improves the identification rate of the lower-level algorithms for most of the rank values. Moreover, our fusion algorithm obtains 74.03% of Rank-1 identification rate matching 258 samples (NIST SD27 [9]) against 29,257 references (27,000 NIST SD14 [37], 2000 NIST SD4 [38], and 257 NIST SD27 [9]), and 71.32% against 100,000 references (27,000 NIST SD14 [37], 2000 NIST SD4 [38], 257 NIST SD27 [9], and a non-public database).

We organize the remainder of this paper as follows. We discuss some related work with fusion algorithms for latent fingerprint identification in Sect. 2. We continue describing our proposal and parameter tuning of the minutia descriptors for latent fingerprint identification in Sect. 3. Next, we analyze our experimental results and the computation time required to identify a latent fingerprint in Sect. 4. Finally, we state our conclusions and future work in Sect. 5.

2 Related work

Some authors [28, 31] have classified the fusion schemes for fingerprint matching [1, 11, 14, 27, 31] as feature level, match-score level, rank level, or data level. When fusing the match scores for latent fingerprint identification, every lower-level algorithm fused deals with the complexities of latent fingerprints, such as the partial information of latent fingerprints, background noise, nonlinear distortion, and brightness variations. As a consequence, match-score level fusion has gained popularity for latent fingerprint identification [31].

An early work published by Jain et al. [12] proposed a latent fingerprint identification algorithm combining the match scores computed using minutiae and orientation fields. The authors employed a weighted sum with weights empirically determined: 0.8 for the match score between minutiae and 0.2 for the match score between orientation fields. Their fusion algorithm reached a Rank-1 identification rate of 79.5% matching the samples in the database NIST SD27 [9] against the references in databases NIST SD27 [9] and NIST SD4 [38].

Another paper presented by Jain and Feng [11] combined several fingerprint features using successive weighted sums. The authors computed match scores between the minutiae, skeleton images, singular points, ridge quality maps, ridge frequency maps, and ridge wavelength maps. Successively, they fused the new match score with the accumulated match score using a weighted sum with weights empirically determined. Their proposal achieved a Rank-1 identification rate of 74% comparing the samples in the database NIST SD27 [9] against the references in the databases NIST SD27 [9], NIST SD4 [38], and NIST SD14 [37].

Sankaran et al. [31] developed a two-level fusion algorithm for simultaneous latent fingerprint identification. In the first level, they combined the match scores computed for each latent fingerprint extracted from the simultaneous image using a weighted sum and a product of the likelihood ratio. The weighted sum fusion surpassed the product of the likelihood ratio for all cases. In the second level, the authors fused the ranks output by each fusion algorithm using weighted Borda count. Although the rank-level fusion improved the Rank-10 identification rate, the match-score level fusion still reported the highest Rank-1 identification rate using a weighted sum.

Paulino et al. [27] proposed a fusion of the match scores computed between the minutiae and the orientation fields. The fusion approach was a weighted sum with weights empirically determined (0.6 for minutiae and 0.4 for orientation fields). The authors reported a 53.5% of Rank-1 identification rate comparing the samples in NIST SD27 [9] against 31,998 references in the databases NIST SD27 [9], WVU, and NIST SD14 [37].

Similarly, Jeyanti et al. [14] fused the match scores computed between minutiae and pores from the sample and the references. They used a weighted sum with weights empirically determined. Differently to the previous works, the authors assigned higher weights to the pores than the minutiae.

Krish et al. [18] combined some conventional fingerprint matching algorithms with a fitting error at the match-score level. The fitting error was computed using extended minutiae types. An extended minutia is a minutia different from the ridge ending and bifurcation, such as deviation, bridge, or fragment [18]. They fused the match scores with a weighted sum with weights equal to 0.5.

Cao and Jain [1] combined two minutia templates and a texture template at the match-score level. They used a weighted sum with weights empirically determined. Their proposal achieved a Rank-1 identification rate of 64.7% matching the samples in the database NIST SD27 [9] against 100,000 references in the NIST SD27 [9], NIST SD14 [37], and a forensic agency. Later, Cao et al. [3] improved that identification rate up to 65.7% using the same databases by including a third minutia template in the fusion. Additionally, the author explored a fusion at rank levels using Borda count. They found that the results using a weighted sum exceed the results of the fusion using Borda count.

These works corroborate that the match-score level has excellent popularity as a fusion scheme for latent fingerprint identification. Moreover, these papers show two conventional approaches for fusing match scores, the weighted sum [1, 3, 11, 12, 14, 18, 27, 31] and the product rule [31]. The weighted sum has reported the highest identification rates. However, both approaches lack of adaptability to changes in the reference database because their fusion formula remains the same. Moreover, some of them depend on fixed weights empirically determined. Conversely, match-score fusion using supervised classifiers can compute more complex functions than weighted sums or product rules. Besides, if the references database changes, the supervised classifiers can be retrained to update the fusion relation.

3 A fusion of fingerprint matching algorithms at match-score level

Latent fingerprint identification algorithms usually identify latent fingerprints via a hybrid approach [24], starting with a local matcher and following a global matcher. Global algorithms match fingerprints using features that describe the full fingerprint. Global matchers are also known as consolidation steps and they can follow different strategies, such as single or multiple transformations, a consensus of transformations, or an incremental consolidation [24]. Local algorithms match small regions of fingerprints, such as two minutia descriptors [24]. A minutia descriptor is a fingerprint feature representation with additional information to the minutia standard representation (ISO/IEC 19794-2:2005). This additional information can be, for example, the ridge orientations, the number of ridges between minutiae, or the relationships a minutia holds with others, either in a neighborhood or with a predefined number of minutiae.

Our proposal starts with two local matching algorithms MCC [5] and mtriplet [25], which has shown to be suitable for latent fingerprint identification [36]. These local algorithms match the minutiae in the sample against those in the reference and output the match scores between minutiae, which are input to the global matching algorithm (see step 1 in Fig. 1). A global matcher consolidates the output of each local matcher. Each hybridization between the global matching algorithm and each local matching algorithm is a lower-level algorithm in our ensemble. We will describe our selection for the lower-level algorithms in Sect. 3.1

We include a third step in our latent fingerprint identification algorithm. In this step, we fuse the output of the lower-level algorithms by configuring a vector of attributes (v) with their output. Then, a supervised classifier computes the fused match score with the vector of attributes as input. This supervised classifier is the high-level algorithm of our ensemble (see step 3 in Fig. 1 and Algorithm 1).

Algorithm 1: Latent fingerprint identification algorithm with an ensemble of two fingerprint matching algorithms with parameters tuned for latent fingerprint identification.

```
function identifySample($l$, $T$)
Data:
  – $l$ is the sample (latent fingerprint)
  – $T = \{t_1, t_2, ..., t_m\}$, where $t_i$ is the i-th reference
    (impression)
  – $MCC(l, t_i)$ is the local matching algorithm for MCC,
    which computes the similarities between minutiae in the
    sample $l$ and each reference $t_i$
  – $mtriplet(l, t_i)$ is the local matching algorithm for
    mtriplet, which computes the similarities between
    minutiae in the sample $l$ and each reference $t_i$
  – $DMC(local\_matching\_algorithm\_output)$ is the global
    matching algorithm, which computes the match score
    and the matching minutia count between the sample $l$
    and each reference $t_i$ using the similarities between
    minutiae output by each local matching algorithm
  – $\delta$ is the match-score fusion algorithm, a Multilayer
    perceptron (MLP)
begin
    Let $S \longleftarrow \phi$ be the set of fused match scores
      between $l$ and each $t_i$
    foreach $t_i \in T$ do
        Let $v \longleftarrow \phi$ be a vector of matching features
        $v \longleftarrow DMC(MCC(l, t_i))$ concat
          $DMC(mtriplet(l, t_i))$: $v$ receives a quartet
          formed by the match score and the
          matching minutia count computed by each
          matching algorithm
        $s_i \longleftarrow \delta(v)$: $\delta$ computes the fused match
          score
    Return 20 sorted references ($t_i$) with the highest
      match scores ($s_i$)
```

3.1 Selection of lower-level algorithms

The salient decisions in an ensemble are the lower-level algorithms and the fusion algorithm. Latent fingerprint identification algorithms proposed in the literature tend to apply fingerprint matching algorithms developed for fingerprint verification [36]. However, algorithms with high

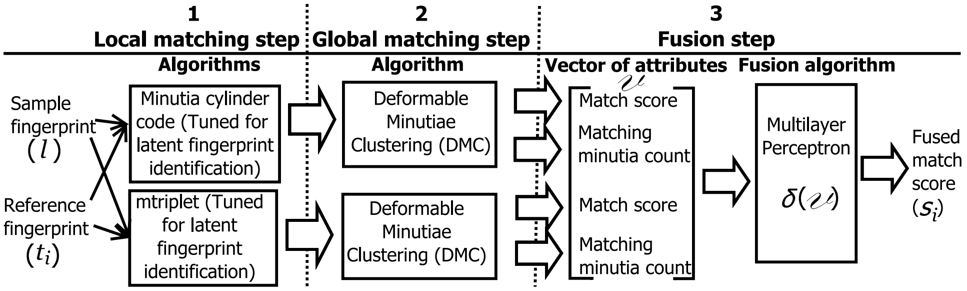
1
Local matching step
Algorithms
2
Global matching step
Algorithm
3
Fusion step
Vector of attributes
Fusion algorithm
Sample fingerprint (l)
Reference fingerprint (t_i)
Minutia cylinder code (Tuned for latent fingerprint identification)
mtriplet (Tuned for latent fingerprint identification)
Deformable Minutiae Clustering (DMC)
Deformable Minutiae Clustering (DMC)
v
Match score
Matching minutia count
Match score
Matching minutia count
Multilayer Perceptron
$\delta(v)$
Fused match score (s_i)

Fig. 1 Diagram of the match-score computation between a sample and a reference using our ensemble of fingerprint matching algorithms

performance for fingerprint verification do not necessarily do just as well on latent fingerprint identification [36]. Thus, we have avoided a straightforward fusion of fingerprint matching algorithms without knowing their worthiness for latent fingerprint identification. Instead, we have selected local and global matching algorithms with a higher performance identifying latent fingerprints.

Valdes-Ramirez et al. [36] compared the performance of various local matching algorithms of existing minutia descriptors for latent fingerprint identification. They found four local matching algorithms that outperform others: MCC [5], mtriplet [25], a combination of the orientations of ridges and minutiae in a spiral scheme [33], and the local structure proposed by Jiang and Yau [15]. Different performances of these matching algorithms on various subsets of latent fingerprints motivate the idea of ensembling such matching algorithms. Thus, we have evaluated the performance of these four local matching algorithms using their particular global matching algorithms and comparing subsets of the samples and references in the NIST SD27 [9] database. This latent fingerprint database has samples classified by latent examiners into good, bad, or ugly, depending on the quality of the fingerprint features.

Figure 2 depicts the performance measure used to compare identification algorithms: cumulative match characteristic curves (CMC) (ISO/IEC 19795-1). A CMC curve plots an identification rate for each rank number, *i.e.*, the fraction of genuine matching fingerprints with a computed match score higher than the match score in the *i*-th rank. According to the CMC curves, MCC [5] and mtriplet [25] always surpassed the other two. Besides, both output the highest identification rates for different rank values on diverse fingerprint qualities. Hence, we selected MCC [5] and mtriplet [25] for local matching.

To be sure about the prediction power of the selected lower-level matching algorithms, we had to validate that it does not degrade regardless of what global matching algorithm we choose to use. Therefore, we have compared the performance of the global matchers proposed within each local matching algorithm (mtriplet and MCC) against versions substituting their global matchers with the algorithm deformable minutiae clustering (DMC) [26]. The global matching algorithm proposed for mtriplet is M3gl [25], and for MCC [5], we have used the global matching algorithm provided by the authors in MCC SDK 1.4 [5].

DMC is a global matcher independent of the local matching algorithm, which has shown an improvement in the identification rates using different local matching algorithms [26]. Additionally, a new parallel version of DMC [26] increased its previous matching speed by 44.7x [29]. Figure 3 shows that in our experiments, DMC [26] surpasses the identification rates of the particular global matcher MCC

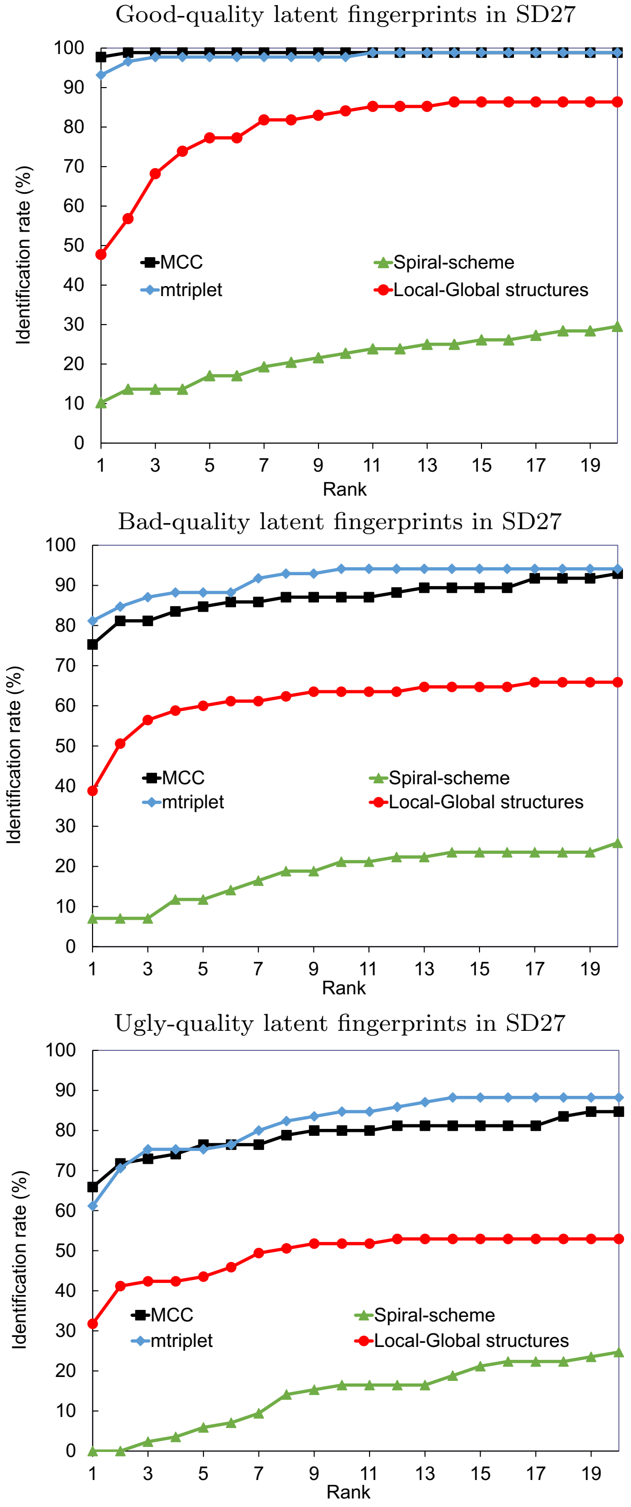

Fig. 2 Curves CMC Rank-20 output by lower-level algorithms with their global matchers for good, bad, and ugly subsets of latent fingerprints in the NIST SD27 [9] database. We can notice that MCC [5] and mtriplet [25] outperform the other two in all cases, but there is not an absolute winner between both. These results favor the idea of fusing these two local matching algorithms

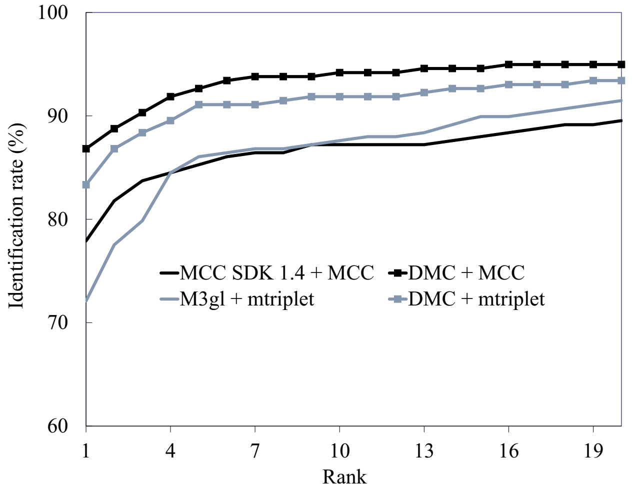

Fig. 3 CMC curves comparing the performance of the global matchers MCC SDK 1.4 [5] and M3gl [25] against the algorithm deformable minutiae clustering (DMC) [26]. The algorithm DMC [26] as the global matcher reaches higher identification rates than MCC SDK 1.4 and M3gl for both local matching algorithms MCC and mtriplet

SDK 1.4 [5] and M3gl [25] for all rank values comparing the fingerprint pairs in the database NIST SD27 [9]. As a consequence, we have selected the algorithm DMC [26] for global matching. Besides, we have modified DMC [26] for computing the matching minutia count in addition to the match score (see step 2 in Fig. 1).

3.2 Selection of the attributes and the fusion algorithm

To select the attributes and the fusion algorithm of our proposal, we have experimented with different setups. We explored three sets of attributes: using only the matching minutia count, using only the match score, and using both. Each set of attributes stores tuples of attributes with different sizes (two or four) that describe the matching relationship between a sample and a reference.

With these tuples of attributes, we set up three training datasets using the samples and references in a non-public database with 283 latent fingerprints matched with 243 impressions. The fingerprint pairs were acquired and manually marked by latent examiners from criminal investigations. Hence, they pose the same difficulty for successful matching as those in the database NIST SD27 [9]. Next, we associate with each tuple the class attribute: 1 for fingerprint pairs that are matched and 0 for fingerprint pairs that are not. As a result, the three datasets add 283 examples to the matched class (1) and 68,486 ($283 * 243 - 283$) examples to the not matched class (0). Such difference guides to class-imbalance datasets with an *imbalance ratio* $= \frac{68{,}486}{243}$.

Methods that tackle class imbalance problems are often divided into three categories: data level, algorithm level, and cost-sensitive [20–23]. In this work, we have used a data-level approach to address class imbalance. We have undersampled data using 0.01% of not matched fingerprints and 100% of the matched fingerprints, hence yielding a new imbalance ratio of $\frac{684}{243}$.

We compare five candidate algorithms (three supervised classifiers, a weighted sum, and a product rule) to select the fusion algorithm. The weighted sum and the product rule are fusion algorithms commonly found in the literature [1, 3, 11, 12, 14, 18, 27, 31]. We empirically set the weights for the weighted sum as follows: 0.4 and 0.1 for the match score and the matching minutia count computed with the local matcher MCC [5], respectively, and similar for those attributes computed with mtriplet [25]. The supervised classifiers are determined with two automatic tools, one for traditional machine learning algorithms and other for deep learning algorithms. For all supervised classifiers, we adopt the probability of each tuple belonging to the class 1 computed by the supervised classifier as the fused match score.

We employed Auto-WEKA [34] developed for WEKA 3.8.2 to determine the traditional machine learning algorithms with the best performances. Auto-WEKA evaluates 30 supervised classifiers with different parameter configurations optimizing a user-defined measure with a tenfold cross-validation test. We executed Auto-WEKA on a server with 48 cores, 1 TB of RAM, and a 1-TB hard drive for 48 hours.

Similarly, we employed Auto-KERAS [16] to determine the deep learning model with the best performance. Auto-KERAS is an automatic tool for deep learning based on KERAS, which tests the number of models predefined by the user. Also, Auto-KERAS outputs the best model according to the loss function defined by the user. We tested 30 different deep learning models for structured data with Auto-KERAS, optimizing the balanced accuracy.

From our experiment, we obtained three different supervised classifiers as candidates for the fusion algorithm. Auto-WEKA found with the undersampled training datasets that a Bayesian network maximized the balanced accuracy, and a multilayer perceptron (MLP) maximized the AUC-ROC. All nodes in the Bayesian network have only one parent, and the MLP has three layers: the input layer with four units, the hidden layer with two units, and the output layer with one unit. By contrast, Auto-KERAS outputs a deep learning model for structured data with two hidden layers with 32 and 16 units, respectively, using the full training datasets.

The CMC curves in Fig. 4 show that the fusion using the match scores and the matching minutia count with an MLP achieved the highest Rank-1 identification rate (87.98%), slightly higher than the fusion algorithm with the weighted sum (87.60%).

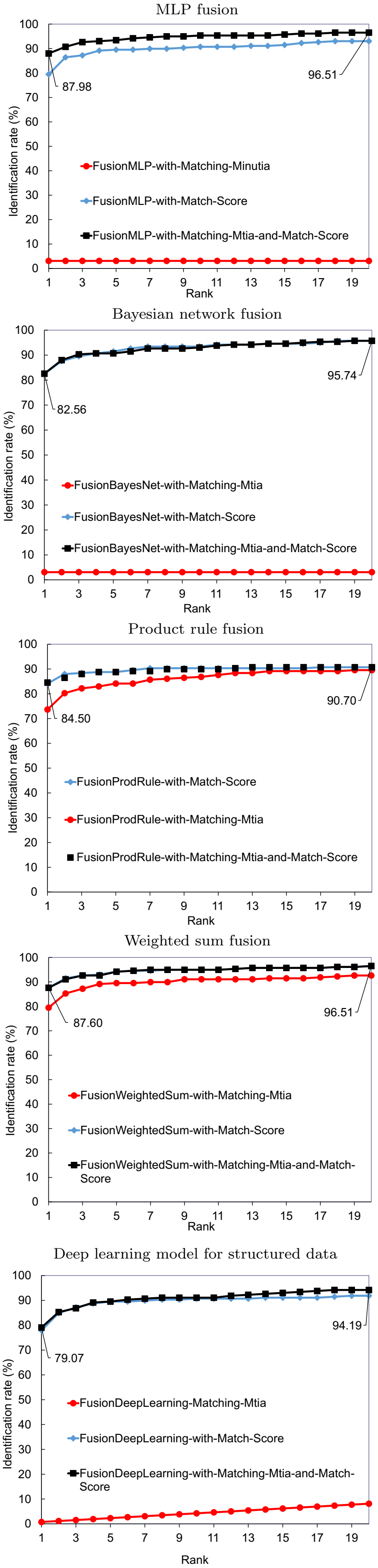

Fig. 4 Curves CMC Rank-20 of the five fusion algorithms computed ▶ with three different sets of attributes matching the samples and references in the database NIST SD27 [9]. Note: The MLP algorithm fusing the matching minutia count and the match score, computed by each lower-level algorithm, obtained the highest identification rates

The results in Figs. 2 and 4 lead us to develop an ensemble for latent fingerprint identification. Our proposal uses the local matching algorithms MCC [5] and mtriplet [25] with the global matching algorithm DMC [26] as the lower-level algorithms. Furthermore, we fuse both attributes computed by each lower-level matching algorithm with an MLP (as we described in Fig. 1 and Algorithm 1). Nonetheless, we can still improve the lower-level matching algorithms tuning the parameters of the local matching algorithms for latent fingerprint identification.

3.3 Parameter tuning of the local matching algorithms for latent fingerprints

The local matching algorithms selected for our stacking were proposed for fingerprint verification. Hence, the authors determined their parameters using databases with impressions, which are fingerprints with higher quality than the latent fingerprints. Since we are developing a latent fingerprint identification algorithm, we have tuned the parameters of both local matching algorithms MCC [5] and mtriplet [25] using a non-public database with 283 samples and their mated references.

Often, parameter tuning is a time-consuming task because it is necessary to explore a search space with as many dimensions as the number of parameters. To address this issue, we search the best parameter values only in a range around the value proposed by the authors. When we find the best parameter value at the extremes of an interval, we restart the search with a new interval around the found value.

For a finer parameter tuning, we could compute the CMC curve for each set of parameters. However, computing the CMC curve is time-consuming. Alternatively, we tuned the parameters using the metric ZeroFMR of the DET curve, which can be computed faster than the CMC curve.

3.3.1 Parameter tuning of the local matching algorithm Minutia cylinder code

We tuned the following parameters of the local matching algorithm MCC [5] for latent fingerprint identification by an exhaustive search:

- The length and width of the cuboid (Ns) with values in the range [14, 23], increasing the search with a step = 1.

- The height of the cuboid (Nd) with values in the range [4, 7], increasing the search with a step = 1.
- The radio of the cylinder (R) with values in the range [50, 76], increasing the search with a step = 2.
- The distance threshold with values in the range [2, 11], increasing the search with a step = 1.
- The angle threshold with values in the range [0.02, 0.74], increasing the search with a step = 0.1.

The best parameter values found regarding the ZeroFMR of the DET curve computed with the non-public database were (Ns = 21, Nd = 6, R = 60, distance threshold = 5, angle threshold = 0.12). The size of the cells increases from Ns = 16 and Nd = 5 to Ns = 21 and Nd = 6, due to the less presence of features in latent fingerprints. However, the radio and the thresholds diminish due to the nonlinear distortions in the latent fingerprints. Such distortions are higher for larger areas of the fingerprint. In particular, the radio goes from R = 70 to R = 60; the distance threshold goes from 7 to 5, and the angle threshold goes from 0.52 to 0.12. The CMC Rank-20 curve with these parameters exceeds the CMC curve with the original parameters for all rank values matching the minutiae of the fingerprints in the database NIST SD27 [9] (see Fig. 5 MCC).

3.3.2 Parameter tuning of the local matching algorithm mtriplet

Similarly, we tuned the following parameters of the local matching algorithm mtriplet [25] for latent fingerprint identification through an exhaustive search:

- The threshold for the distance between minutiae (dThr) with values in the range [10, 50] increasing the search with a step = 1.
- The angle threshold (aThr) with values in the range [10, 25] increasing the search with a step = 0.25.
- The number of neighboring minutiae revised to form the triplets (neighborCount) with values in the range [3, 11] increasing the search with a step = 1.

The best parameter values found regarding the ZeroFMR of the DET curve computed with the non-public database were (dThr = 13, aThr = 15, neighborCount = 10). The neighboring minutiae revised to create the triplets increases from neighborCount = 7 to neighborCount = 10 due to the higher presence of low quality and spurious minutiae in latent fingerprints. The thresholds decrease from dThr = 15 to dThr = 13, and aThr = 21.25 to aThr = 15 due to the nonlinear distortions in the latent fingerprints, which are higher for larger areas of the fingerprint.

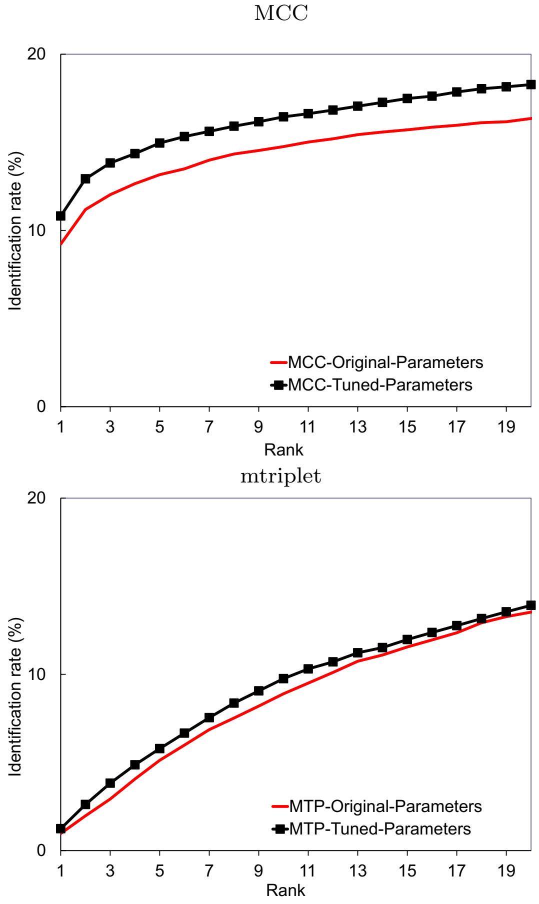

Fig. 5 Curves CMC Rank-20 output by the local matching algorithms MCC [5] and mtriplet [25] with the original parameters and the parameters tuned for latent fingerprint identification comparing the fingerprints pair in the database NIST SD27 [9]. We tuned the parameters with a non-public database regarding the zero false matching rate (ZeroFMR) of the DET curve. We have magnified the *y*-axis for visualization purposes

The CMC Rank-20 curve with these parameters is higher or equal to the CMC curve with the original parameters for all rank values matching the minutiae of the fingerprints in the database NIST SD27 [9] (see Fig. 5 mtriplet).

4 Results and discussion

Intending to compare the ensemble of fingerprint matching algorithms against the previous algorithms reported in the literature, we have configured seven testing datasets. Table 2 describes such datasets in terms of the input fingerprint databases, the number of samples, the number of references, and the minutia extraction method.

Testing for latent fingerprint identification in a closed scenario involves computing the CMC curve with the testing datasets, which have not been previously “seen” by the

Table 2 Description of the datasets employed to test our proposal. To provide comparisons with previous work, we have included DS_2 with a subset of the references in DS_3, and DS_6 with the subset of samples and references with extended minutiae in the database GCDB [18]

Dataset	Databases	Samples	References	Latent features extraction
DS_1	NIST SD27 [9] SD4 [38]	258	2257	Manual
DS_2	NIST SD27 [9] SD14 [37]	258	10,000	Manual
DS_3	NIST SD27 [9] SD14 [37]	258	27,257	Manual
DS_4	NIST SD27 [9] SD4 [38] SD14 [37]	258	29,257	Manual
DS_5	MOLF [32] DB1 DB4 (subset)	859	1000	Automatic (NIST-NBIS)
DS_6	GCDB [18]	151	151	Manual
DS_7	NIST SD27 [9] SD4 [38] SD14 [37] non-public database	258	100,000	Manual

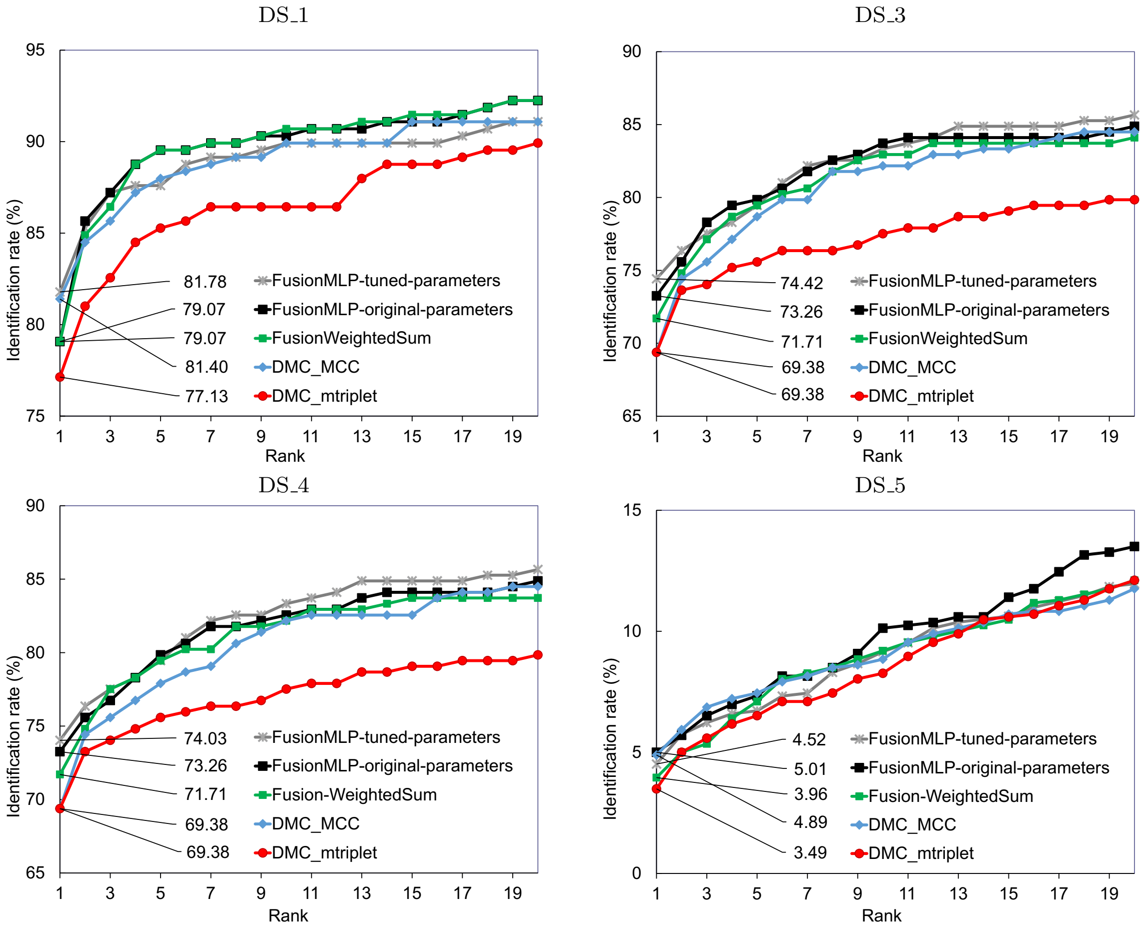

Fig. 6 Curves CMC Rank-20 output by each evaluated algorithm using datasets DS_1, DS_3, DS_4, and DS_5. The evaluated algorithms are the lower-level matching algorithms, their fusion using a weighted sum, their fusion using an MLP, and their fusion using an MLP with the parameters of the local matching algorithms tuned for latent fingerprint identification. We have pointed out the Rank-1 identification rates. We have adjusted the *y-axis* for each graph to improve the visualization

algorithm. A closed scenario indicates that each sample has a mated reference in the reference database [24], which is our case.

Figure 6 depicts the curves CMC Rank-20 for the lower-level matching algorithms, their fusion using a weighted sum, their fusion using an MLP with the original parameters of the local matching algorithms, and their fusion using an MLP with the parameters tuned for latent fingerprint identification. The curves corroborate that the fusion using an MLP improves the identification rates of the lower-level algorithms using all datasets and for most of the rank values. Additionally, we can observe that the fusion using an MLP exceeds the fusion using the weighted sum for larger databases (see the differences in datasets DS_3, DS_4, and

DS_5). We can see that the parameter tuning using the ZeroFMR improves the Rank-1 identification rate for DS_1, DS_3, and DS_4, but it worsens for the DS_5. Unlike the other datasets, DS_5 stores minutiae automatically extracted from latent fingerprints. Although we used the minutiae extractor NIST-NBIS suggested by the authors of the database MOLF [32], this automatic extractor detects more spurious minutiae than the latent examiners. Considering that we tuned the parameters using a database with manually marked minutiae, we conclude that the set of parameters proposed is inefficient for samples with minutiae automatically extracted. Nevertheless, the fusion using an MLP but with the original parameters outperforms the lower-level matching algorithms and the fusion using a weighted sum for most of the rank values also using DS_5.

Table 3 compares the identification rates achieved by the ensemble of fingerprint matching algorithms against those reported in the literature. Our algorithm with the parameters of the local matching algorithms tuned for latent fingerprint identification reached a higher Rank-1 identification rate than other methods, except when comparing the samples and references in the dataset DS_2. Nevertheless, our proposed mechanism outperforms the work of Cao and Jain 2018 [2] using only minutiae. Identifying latent fingerprints using only minutiae reduces the size of the reference databases and the computational time.

The average computational time of our proposal was six milliseconds matching a sample against a reference. Although we performed our training and validation experiments on a high-performance server, we measured the computational time on a device with mobility considering remote crime scenes. Thus, we used a laptop with a Core i7 processor, 16 GB of RAM, and a solid-state hard drive. We measured the average computational time to identify the samples against the references in the NIST SD27 [9] database. Since this database is considered representative of latent fingerprints, we measured the average computational time as a more reliable measure than the computational time of identifying a randomly selected latent fingerprint.

5 Conclusions

In this research, we have studied the latent fingerprint identification problem as a machine learning problem, fusing two lower-level algorithms for fingerprint matching. We have selected as attributes the matching score and the matching minutia count computed by each lower-level matching

Table 3 Comparison between the ensemble of fingerprint matching algorithms and previous works regarding the Rank-1 identification rate

Algorithm	Rank-1 identification rate (%)	Fingerprint features used	Reference database
Krish et al. 2019 [18]	92.72	Minutiae and extended minutiae	DS_6 (151 references)
Our proposal with parameters of the lower-level algorithms tuned for latent fingerprint identification	96.71	Minutiae	
Cao and Jain 2018 [2]	78.20	Minutiae and texture template	DS_2 (10,000 references)
*Cao and Jain 2018 [2] using only minutiae	75.60	Minutiae	
Our proposal with parameters of the lower-level algorithms tuned for latent fingerprint identification	76.74	Minutiae	
Jain and Feng 2011 [11]	74.00	Minutiae, skeleton, singular point, region of interest, ridge quality map, ridge frequency map, ridge wavelength map	DS_4 (29,257 references)
*Jain and Feng 2011 [11] using only minutiae	≈ 35.00	Minutiae	
Hernández-Palancar et al. 2014 [10]	58.13	Minutiae, singular point	
Medina-Pérez et al. 2016 [26]	68.60	Minutiae	
Castillo and Hernández-Palancar 2019 [6]	70.9	Minutiae and ridges	
Our proposal with parameters of the lower-level algorithms tuned for latent fingerprint identification	74.03	Minutiae	
Cao et al. 2020	65.7	Minutiae and texture template	DS_7 (100,000 references)
Our proposal with parameters of the lower-level algorithms tuned for latent fingerprint identification	71.32	Minutiae	

Italic face type indicates our proposal

*indicates the results achieved by the previous algorithm when using only minutiae algorithm. We have chosen two local matching algorithms from those suitable for latent fingerprint identification, tuning their parameters for latent fingerprint identification. Moreover, we have substituted their global matching algorithms by the algorithm DMC, which has shown to improve their performance. Finally, we have computed the fused match score between a sample and a reference using a function learned with a multilayer perceptron.

From our experiments, we found that the fusion of lower-level matching algorithms using a supervised classifier improves the identification rate of the lower-level matching algorithms. We also observed that a supervised classifier reaches higher identification rates than a weighted sum as the fusion algorithm because a supervised classifier can find more complex functions than the weighted sum. Moreover, we noticed that the minutia extraction method affects the parameters for the local matching algorithms.

Other attributes extracted from the local matching algorithms will be analyzed as future work to determine a new set of attributes that describes better the matching relationship between a latent fingerprint and an impression. Furthermore, ongoing work also involves exploring the use of other supervised classifiers that are known to be robust to class imbalance [23].

Acknowledgements This research is partially supported by the National Council of Science and Technology of Mexico (CONACyT) under Grant PN720 and scholarship Grant 638948.

References

1. Cao K, Jain AK (2019) Automated latent fingerprint recognition. IEEE Trans Pattern Anal Mach Intell 41(4):788–800
2. Cao K, Jain AK (2019) Latent fingerprint recognition: role of texture template. In: 9th International conference on biometrics theory, applications and systems (BTAS). IEEE, pp 1–9
3. Cao K, Nguyen DL, Tymoszek C, Jain AK (2020) End-to-end latent fingerprint search. IEEE Trans Inf Forensics Secur 15:880–894
4. Cao K, Yang X, Tian J, Zhang Y, Li P, Tao X (2009) Fingerprint matching based on neighboring information and penalized logistic regression. In: International conference on biometrics. Springer, pp 617–626
5. Cappelli R, Ferrara M, Maltoni D (2010) Minutia cylinder-code: a new representation and matching technique for fingerprint recognition. IEEE Trans Pattern Anal Mach Intell 32(12):2128–2141
6. Castillo-Rosado K, Hernández-Palancar J (2019) Latent fingerprint matching using distinctive ridge points. Informatica 30(3):431–454
7. Dorizzi B, Cappelli R, Ferrara M, Maio D, Maltoni D, Houmani N, Garcia-Salicetti S, Mayoue A (2009) Fingerprint and on-line signature verification competitions at icb 2009. In: International conference on biometrics. Springer, pp 725–732
8. Feng J (2008) Combining minutiae descriptors for fingerprint matching. Pattern Recognit 41(1):342–352
9. Garris MD (2000) NIST special database 27: fingerprint minutiae from latent and matching tenprint images. US Department of Commerce, National Institute of Standards and Technology
10. Hernández-Palancar J, Munoz-Briseno A, Gago-Alonso A (2014) Using a triangular matching approach for latent fingerprint and palmprint identification. Int J Pattern Recognit Artif Intell 28(07):1460004
11. Jain AK, Feng J (2011) Latent fingerprint matching. IEEE Trans Pattern Anal Mach Intell 33(1):88–100
12. Jain AK, Feng J, Nagar A, Nandakumar K (2008) On matching latent fingerprints. In: 2008 IEEE Computer Society conference on computer vision and pattern recognition workshops. IEEE, pp 1–8
13. Jea TY, Govindaraju V (2005) A minutia-based partial fingerprint recognition system. Pattern Recognit 38(10):1672–1684
14. Jeyanthi S, Uma Maheswari N, Venkatesh R (2015) Neural network based automatic fingerprint recognition system for overlapped latent images. J Intell Fuzzy Syst 28(6):2889–2899
15. Jiang X, Yau WY (2000) Fingerprint minutiae matching based on the local and global structures. In: Proceedings 15th international conference on pattern recognition, pp 1038–1041
16. Jin H, Song Q, Hu X (2019) Auto-keras: an efficient neural architecture search system. In: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining. ACM, pp 1946–1956
17. Komarinski P (2005) Automated fingerprint identification systems (AFIS). Elsevier, Amsterdam
18. Krish RP, Fierrez J, Ramos D, Alonso-Fernandez F, Bigun J (2019) Improving automated latent fingerprint identification using extended minutia types. Inf Fusion 50:9–19
19. Lee HC, Ramotowski R, Gaensslen RE (2001) Advances in fingerprint technology. CRC Press, Boca Raton
20. Loyola-González O (2019) Black-box vs. white-box: understanding their advantages and weaknesses from a practical point of view. IEEE Access 7:154096–154113
21. Loyola-Gonzlez O, Martnez-Trinidad JF, Carrasco-Ochoa JA, Garca-Borroto M (2016) Study of the impact of resampling methods for contrast pattern based classifiers in imbalanced databases. Neurocomputing 175:935–947
22. Loyola-Gonzlez O, Martnez-Trinidad JFCO, Carrasco-Ochoa JA, Garca-Borroto M (2019) Cost-sensitive pattern-based classification for class imbalance problems. IEEE Access 7:60411–60427
23. Loyola-Gonzlez O, Medina-Pérez MA, Martnez-Trinidad JF, Carrasco-Ochoa JA, Monroy R, Garca-Borroto M (2017) Pbc-4cip: a new contrast pattern-based classifier for class imbalance problems. Knowl Based Syst 115:100–109
24. Maltoni D, Maio D, Jain AK, Prabhakar S (2009) Handbook of fingerprint recognition. Springer, Berlin
25. Medina-Pérez MA, García-Borroto M, Gutierrez-Rodríguez AE, Altamirano-Robles L (2012) Improving fingerprint verification using minutiae triplets. Sensors 12(3):3418–3437
26. Medina-Pérez MA, Moreno AM, Ballester MAF, García-Borroto M, Loyola-González O, Altamirano-Robles L (2016) Latent fingerprint identification using deformable minutiae clustering. Neurocomputing 175:851–865
27. Paulino AA, Feng J, Jain AK (2013) Latent fingerprint matching using descriptor-based hough transform. IEEE Trans Inf Forensics Secur 8(1):31–45
28. Ross AA, Jain AK, Nandakumar K (2006) Handbook of multibiometrics. International series on biometrics, vol 6. Springer, Berlin
29. Sanchez AJ, Romero LF, Peralta D, Medina-Pérez MA, Saeys Y, Herrera F, Tabik S (2020) Asynchronous processing for latent fingerprint identification on heterogeneous CPU-GPU systems. IEEE Access 8:124236–124253

30. Sankaran A, Dhamecha TI, Vatsa M, Singh R (2011) On matching latent to latent fingerprints. In: International joint conference on biometrics (IJCB), pp 1–6
31. Sankaran A, Vatsa M, Singh R (2012) Hierarchical fusion for matching simultaneous latent fingerprint. In: IEEE fifth international conference on biometrics: theory, applications and systems (BTAS). IEEE, pp 377–382
32. Sankaran A, Vatsa M, Singh R (2015) Multisensor optical and latent fingerprint database. IEEE Access 3:653–665
33. Shi Z, Govindaraju V (2009) Robust fingerprint matching using spiral partitioning scheme. In: Advances in biometrics, pp 647–655
34. Thornton C, Hutter F, Hoos HH, Leyton-Brown K (2013) Auto-WEKA: combined selection and hyperparameter optimization of classification algorithms. In: Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, pp 847–855
35. Ting KM, Witten IH (1999) Issues in stacked generalization. J Artif Intell Res 10:271–289
36. Valdes-Ramirez D, Medina-Pérez MA, Monroy R, Loyola-González O, Rodríguez J, Morales A, Herrera F (2019) A review of fingerprint feature representations and their applications for latent fingerprint identification: trends and evaluation. IEEE Access 7:48484–48499
37. Watson CI (2001) NIST special database 14: mated fingerprint cards pairs 2 version 2. Technical report, National Institute of Standards and Technology, Gaithersburg, MD, USA
38. Watson CI, Wilson C (1992) NIST special database 4. Fingerprint database, vol 17. National Institute of Standards and Technology, Gaithersburg, p 77