Experimental Results on Fusion of Multiple Fingerprint Matchers

Gian Luca Marcialis and Fabio Roli

Department of Electrical and Electronic Engineering – University of Cagliari
Piazza d'Armi – 09123 Cagliari – Italy
{marcialis, roli}@diee.unica.it

Abstract. Fingerprints are widely used in automatic identity verification systems. The core of such systems is the verification algorithm to match two fingerprints. So far, various method for fingerprint matching have been proposed, but few works investigated the fusion of two or more matching algorithms. In this paper, various methods for fusing such algorithms have been investigated. Experimental results showed that such fusion can outperform the best individual verification algorithm and increase the discrimination between genuine and impostor classes.

1 Introduction

Fingerprints are widely used in automatic person authentication systems. Fingerprints are different from person to person, cannot be forgotten, and it is very difficult to steal and reproduce them. The person that would access to a certain resource submits to the automatic verification system her/his identity and fingerprint. The system matches the given fingerprint with the one stored in its database and associated to the claimed identity. A degree of similarity, named «score», is computed. If such score is higher than a certain acceptance threshold, then the person is classified as a «genuine» (i.e., the claimed identity is accepted). Otherwise the person is classified as an «impostor» and the access to the required resource is denied.

So far many algorithms to match two fingerprints have been proposed [1, 2]. Some of these algorithms are based on different representations of fingerprint (e.g., representations based on minutiae-points or fingerprint texture). It is reasonable to hypothesise that verification scores of algorithms using different representations should contain "complementary" information; in particular, such algorithms should make different verification errors.

Although the fusion of multiple classifiers has been successfully applied to many pattern recognition problems [3], few works investigated the possibility of fusing different fingerprint matchers in order to improve verification performances [4, 5].

In this paper, various methods for fusing two different fingerprint verification algorithms are assessed and compared by experiments. Reported results show that such fusion can provide performances higher than the ones of the best individual fingerprint matcher. Moreover, such fusion provides an increase of the separation between genuine and impostor score distributions.

In Section 2, the selected matching algorithms and the fusion methodology are described. In Section 3, experimental results are reported. Conclusions are drawn in Section 5.

2 Fusion of Multiple Fingerprint Matchers

2.1 Methodology

Let us consider two fingerprint verification algorithms.
Given the input fingerprint image associated to the claimed identity *i*:

- For each algorithm, compute the matching score (a real value on the interval [0,1]) between the given fingerprint and the «template» fingerprint stored in the database and associated to the identity *i*. Let s_m and s_t be the matching scores provided by the two individual algorithms.
- Apply the following transformation to the above scores s_m and s_t:

$$s = f(s_m, s_t) \qquad \textbf{(1)}$$

All the fusion rules investigated in this paper can be regarded as the application of a particular transformation rule (Section 2.3) [3-5].

- Compare the obtained score value *s* with a threshold. The claimed identity is classified as «genuine» if:

$$s > th \qquad \textbf{(2)}$$

otherwise it is classified as «impostor».
It is easy to see that the above methodology can be also used for the case of more than two verification algorithms.

2.2 The selected fingerprint verification algorithms

In the literature, two main types of fingerprint verification algorithms have been proposed: the so called minutiae-based and texture-based algorithms [1-2, 4-5]. Each type uses representations of fingerprints that are substantially different. Therefore, one can expect that such two types of verification algorithms provide diverse and «complementary» matching scores. As it is well known that fusion of multiple pattern classifiers is effective if diverse and complementary algorithms are used [3], for our experiments, we selected one algorithm for each type.
The selected minutiae-based algorithm is commonly referred as «String» algorithm [1, 4]. The ridge bifurcations and endings, usually called «minutiae», are extracted from the input fingerprint image. Such «minutiae» set is compared with that of the template fingerprint. Such comparison is performed by considering each set as a «string». A «string» distance is computed and converted into a matching score. It is

worth noting that other minutiae-based algorithms have been proposed [4], but «String» shown the best performances [4].

The selected texture-based algorithm is also known as «Filter» algorithm [2, 5]. The input fingerprint image is «partitioned» around its «centre» (the so called «core» point) by a tessellation [2]. A feature vector (called «finger-code») is computed by evaluating the outcomes of a set of Gabor filters for each partition. The matching score is obtained by computing the Euclidean distance between the finger-code of the input image and the one of the template image. Other texture-based algorithms have been proposed, but their use is often limited to the fingerprint classification task [6], for which the finger-code approach shown the best performances [7]. In general, the characteristics of these approaches are also the main limitation, because the fingerprint description is less detailed with respect to the minutiae-based one. Consequently, the performance is inferior. However, we shown by experiments that the fusion of the minutiae-based and the texture-based algorithms can benefit from their "complementarity".

2.3 Fusion of String and Filter Algorithms

To fuse the above String and Filter algorithms, we used various score transformations according to eq. (1). In particular, the investigated score transformations were: maximum score value (Max), minimum score value (Min), mean of the scores (Mean), product of the scores (Product), logistic transformation.

The Logistic transformation is as follows:

$$s = \frac{1}{1 + \exp\left[-\left(w_0 + w_1 s_m + w_2 s_t\right)\right]} \qquad \textbf{(3)}$$

The «weights» of the logistic transformation were computed by a gradient descent algorithm with a least-squares error function (Logistic-MSE transformation), and by a gradient descent algorithm with a cross-entropy error function (Logistic-ML transformation) [8].

3 Experimental results

3.1 The Data Set

For our experiments, we used the FVC-DB1 data base that was recently introduced as a benchmark data set for fingerprint verification algorithms [9]. This data set is made up of 800 fingerprint images acquired from a low-cost optical sensor. The image size is 300x300 pixels and the image resolution is 500 dpi. The number of identities is 100, and the number of images per identity is eight. In our experiments, 276 fingerprints were disregarded because the «core» point requested by the Filter algorithm could not be extracted in a reliable way.

3.2 Experimental planning

First of all, let us remark that, in fingerprint verification systems, the choice of the «acceptance» threshold (eq. (2)) is very critical. The rate of impostors accepted by the system (False Matching Rate, FMR) and the rate of genuine patterns rejected by the system (False Non Matching Rate, FNMR) strictly depend on such threshold. Usually, the criterion for selecting such threshold on the training set is to assess the score value that produces the same number of accepted impostors and rejected genuine patterns. This score value is called point of «Equal Error Rate» (EER).
In our experiments, the following evaluation protocol was used:

- For each verification algorithm, or any combination of the two selected algorithms, we computed two sets of scores. The first one is the so called «genuine-matching scores» set ***G***, made up of all matching among fingerprints of the same identity (all images were compared with all other images of a given identity). A score from the set ***G*** belongs to the «genuine pattern» class. The second one is the «impostor matching scores» set ***I***, made up of all matching among fingerprints of different identities. A score from the set ***I*** belongs to the «impostor» class. The total number of genuine matching scores was 1,440, and the total number of impostor matching scores was 135,586.
- We randomly subdivided the above sets in two parts, so that: ***G***=***G1***U***G2***, ***I***=***I1***U***I2***. ***G1*** and ***G2***, as well as ***I1*** and ***I2***, are disjoint sets.
- The training set ***Tr***={***G1***, ***I1***} was used to estimate the EER point and to compute the weights of the logistic transformation.
- The test set ***Tx***={***G2***, ***I2***} was used to evaluate the performances of algorithms, or fusions of algorithms, on novel patterns, using the acceptance threshold computed on the training set ***Tr***.

The individual algorithms and the different combinations of the two selected algorithms were assessed and compared in terms of EER on the training set, FMR and FNMR, Class Separation Statistic on the test set (Table 1), and Receiver Operating Characteristic curves (Figure 1).

3.3 Results

Table 1 shows the performances of the individual and the combined algorithms in terms of EER percentage values on the training set, FMR and FNMR percentage values on the test set. It should be remarked that EER=FMR=FNMR on the training set (Section 3.2), that is, the EER value completely characterises the performances on the training set. Table 1 also shows Class Separation Statistic values (CSS). Class Separation Statistic is a "distance" measure between the genuine and impostors score distributions [4]. Let us indicate the two score distributions with *p(s|genuine)* and *p(s|impostor)*. The CSS expression is as follows:

$$CSS = \int \left| p(s \mid genuine) - p(s \mid impostor) \right| ds \quad \textbf{(4)}$$

Table 1 shows that the performances of String and Filter algorithms are very different each other. In particular, String is much more accurate than Filter. In principle, this confirms that the fingerprint description by the minutiae is preferable to the one by the texture, as said in section 2. So, one could think that the fusion of such algorithms cannot outperform the best individual fingerprint matcher, especially when test set performances are considered. Differently, except for the Mean and Max transformations, the fusion of these two algorithms provided test-set performances higher than the ones of the best individual fingerprint matcher. Such results are very interesting for application purposes, as they point out that fusion of different algorithms can increase the verification performances on novel fingerprints (test set). In particular, the Product and Logistic-ML fusion rules exhibited the best performances on the test set.

Table 1. Performances of individual and combined algorithms in terms of EER percentage values on the training set, FMR and FNMR percentage values, and Class Separation Statistics (CSS) values on the test set.

	EER	**FMR**	**FNMR**	**CSS**
	Individual Algorithms			
String	2,0	1,9	1,9	1,92
Filter	6,1	6,0	8,3	1,72
	Combined Algorithms			
Product	1,3	1,2	1,7	1,95
Mean	2,6	2,6	2,4	1,92
Min	2,0	1,9	1,9	1,92
Max	6,1	6,0	7,8	1,73
Logistic-MSE	1,4	1,3	1,4	1,96
Logistic-ML	1,3	1,2	1,4	1,95

For application purposes, it is also worth noting that the reliability of a fingerprint verification system strongly depends on the difference between the training set and test set performances. A «robust» fingerprint verification system should exhibit test set performances very close to the training set performances, that is, to exhibit a low «generalisation» error. Table 1 shows that fusion of different matchers often provides low generalisation errors. For example, Logistic-MSE fusion rule outperform the best individual matcher (String), and also exhibits a low generalisation error, as pointed out by the similar values of EER, FMR, and FNMR..

The fifth column of Table 1 shows the CSS values between the genuine and impostor distributions of the individual and combined algorithms. The CSS takes values in [0,2]. The maximum value indicates that the distributions are completely separated, while the minimum value indicates the total distributions overlapping. It is worth noting that the degree of distributions overlapping is correlated with the FMR and the FNMR values. Reported values show that the fusion can increase the distance between the distributions, that is, the separation between the genuine and impostors classes.

Finally, Figure 1 shows the ROC curves on the test set. For sake of clarity, we reported only the ROC curves of the best individual algorithm (String), and the best fusion rules (Product and Logistic-ML). Product and Logistic-ML fusion rules exhibited similar ROC curves, and outperformed the ROC curve of the String algorithm. Figure 1 shows that fusion can improve the performances of the individual fingerprint matchers for any couple of FMR and FNMR values, that is, for any «operational» point.

4 Conclusions

So far, many algorithms to match two fingerprints have been proposed [1, 2], but few works investigated the possibility of fusing them in order to improve verification performances [4, 5].

In this paper, various methods for fusing two different fingerprint verification algorithms were assessed and compared by experiments. Reported results showed that such fusion can provide performances higher than the ones of the best individual fingerprint matcher, and can exhibit a lower generalisation error. Further works will be aimed to investigate the performances of such fusion in larger data sets and with other fusion rules (e.g. fusion rules more powerful than the simple perceptron).

In this work, we also showed that fusion can increase the discrimination between genuine and impostor classes. In our opinion, future investigations should take into account such effect in order to design fusion rules explicitly aimed to maximise the "separation" between genuine and impostor classes.

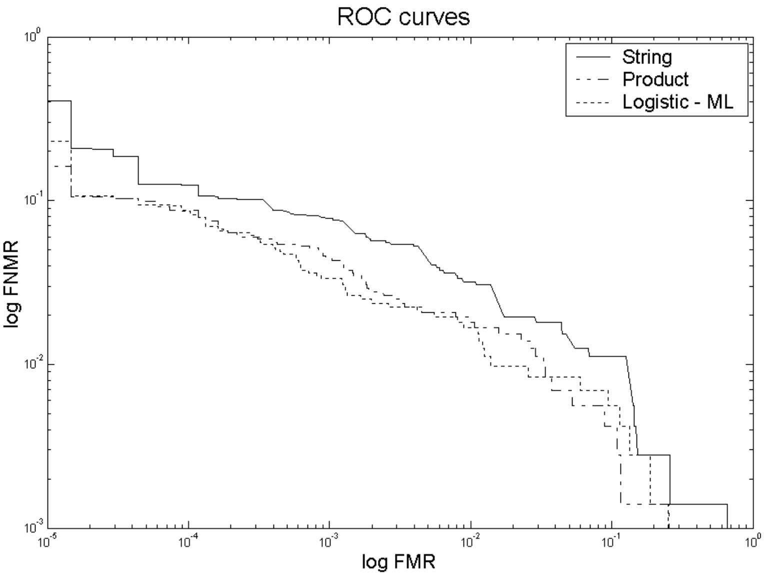

Fig. 1. ROC curves of the String algorithm and the best fusion rules (Product and Logistic-ML).

References

[1] A.K. Jain, L. Hong, and R. Bolle, On-line Fingerprint Verification, IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(4) pp.302-314, 1997.

[2] A.K. Jain, S. Prabhakar, L. Hong, and S. Pankanti, Filterbank-based Fingerprint Matching, IEEE Transactions on Image Processing, 9(5) pp.846-859, 2000.

[3] F. Roli and J. Kittler Eds., Multiple Classifier Systems, Springer-Verlag, Lecture Notes in Computer Science, Vol. 2364, 2002.

[4] S. Prabhakar and A.K. Jain, Decision-level Fusion in Fingerprint Verification, Pattern Recognition, 35(4) pp.861-874, 2002.

[5] A.K. Jain, S. Prabhakar, and S. Chen, Combining Multiple Matchers for a High Security Fingerprint Verification System, Pattern Recognition Letters, 20 (11-13) pp.1371-1379, 1999.

[6] G.T. Candela, P.J. Grother, C.I. Watson, R.A. Wilkinson and C.L. Wilson, PCASYS - A Pattern-Level Classification Automation System for Fingerprints, NIST tech. Report NISTIR 5647, 1995.

[7] A.K. Jain, S. Prabhakar, L. Hong, A Multichannel Approach to Fingerprint Classification, IEEE Transactions on PAMI, vol.21, no.4, pp.348-358, 1999.

[8] C.M. Bishop, Neural Networks for Pattern Recognition, Oxford Univ. Press 1995.

[9] D. Maio, D. Maltoni, R. Cappelli, J.L. Wayman, and A.K. Jain, FVC-2000: Fingerprint Verification Competition, IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(3) pp. 402-412, 2002.