Classifier DECOC based Minutiae Extraction

Mossaad Ben Ayed 1, Faouzi Bouchhima 2 and Mohamed Abid 3
National Engineering School of Sfax
University of Sfax, Tunisia

Abstract. *Human fingerprints are rich in details denoted minutiae. In this paper a method of minutiae extraction from fingerprint skeletons is described. To identify the different shapes and types of minutia a Data-driven Error Correcting Output Coding (DECOC) is trained to work as a classifier. The proposed classifier is applied throughout the fingerprint skeleton to locate various minutiae. Extracted minutiae can then be used as identification marks for automatic fingerprint matching.*

keywords. *Biometrics, Fingerprint, recognition, Minutiae extraction, DECOC.*

1. Introduction

A modern society is challenged by the need to identify individuals. Among all the biometrics, fingerprint matching is one of the most popular, mature, and advanced technologies. In 1888 Sir Francis Galton found that fingerprints are rich on details in form of discontinuities in ridges.The uniqueness of an individual fingerprint is exclusively determined by the local ridge characteristics and their relationships. There are various types of local characteristics, called minutiae, in a fingerprint but widely used fingerprint features are restricted to two types of minutiae. The first is a ridge termination defined as the point where a ridge ends abruptly. The second is a bifurcation defined as the point where a ridge merges or splits into branch ridges. He also discovered that such features are permanent during lifespan [1].

Due to the varying quality of fingerprints, some preprocessing is usually required. Consequently an enhancement algorithm is applied on gray-scale images to improve and separate fingerprints from the background. The first preprocessing step is denoted binarization (Fig. 1). Some of the most frequent methods are directional filters [2].

Minutiae are determined only by discontinuities in ridges, these are totally independent of ridges thickness.

By minimizing data that represents minutiae without corrupting it, a more effective and fast minutiae extraction can be achieved. Thinning the ridges to only 1-pixel wide lines preserves minutiae with minimum data usage. This process is denoted skeletonization and follows the binarization. It is usually an iterative method, either sequential or parallel [3].

The next step is the extraction of the minutiae from the skeletonized fingerprint (Fig.1). A method that handles this simply examines the nearest neighbor pixels around a pixel that belongs to a 1-pixel wide line [4].

Another method [5] studies the relationship between the thinned ridges and depending on the flow. It detects and extracts the various minutiae. The risk of this method is that some important details of a fingerprint might be removed with the binarization and skeletonization. Therefore, there are some algorithms [6], [7] that are used for extracting the minutiae directly from the gray-scale image through ridge line.

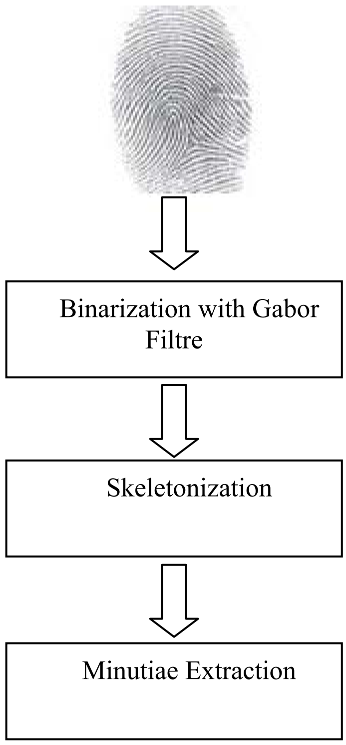


Figure 1. Minutiae extraction with the preprocessing steps.

These algorithms have different rules or ad-hoc methods to handle the various situations that arise when extracting minutiae. This makes them more or less difficult to cover all possible situations.

The method proposed in this paper is based on Data-driven Error Correcting Output Coding (DECOC) to recognize minutiae patterns in the skeletonized fingerprint images. A well defined training set of patterns with a suitably chosen size yields that no additional ad-hoc rules are required. However, methods extracting minutiae from the skeletonized fingerprints are heavily dependent on the preprocessing stage. Producing high quality skeleton fingerprints relies on properly performed binarization and skeletonization. The method used to produce skeleton fingerprints is based on thinning process; we consider a (3X3) pixel neighborhood of a black pixel with its 8 neighbors [3].

2. Minutia Extraction

2.1. Error Correcting Output Coding

Error correcting output codes (ECOC) have been used in the fields of network communication and information theory for the purpose of enhancing the reliability of transmitting binary signals and maintaining information integrity. It adds the redundant parity bits to an information word. The result is called a code word, which is a binary code string. Distances between two code words are described using Hamming distance, which is the count of the different bits in the two patterns.

On the receiving end, a decoding process examines the Hamming distances between the received binary message and all the valid code words to detect and recover errors.

Table 1 gives an illustrative example of the ECOC code matrix of a 5-class classification problem, decomposed into 6 binary classification problems.

Table 1. Example of an ECOC code matrix.

	BL1	BL2	BL3	BL4	BL5	BL6
Classe1	1	1	0	1	1	1
Classe2	0	0	0	1	0	0
Classe3	0	0	1	1	1	1
Classe4	1	0	1	0	1	0
Classe5	1	1	1	0	1	0

The code matrix is used to guide the training and testing processes of ECOC classifiers. In the training, 6 binary base learners are trained. For the binary base learner f_i ($i = 1,\ldots, 6$), if an element $b_{k,i}$ ($k = 1,\ldots,5$ and $i = 1,\ldots,6$) in the code matrix is 1, then all samples of class k will be considered positive. The remaining samples are considered negative for f_i (can be labeled as -1 or 0. Here we use 0 for analogy of binary coding). The testing process determines the class label y of a testing sample $\boldsymbol{x}$ by first applying all base learners to the unknown sample, yielding a codeword $w(\boldsymbol{x})$ (a bitstring of 1s and 0s) . A decision on the label is then made based on the shortest Hamming distance:

$$y = \text{argmin}_k\, H(w_k,w(x)),\ k = 1, \ldots , K,$$

where w_k is the ideal code word for class k, i.e., the k[th] row of the code matrix. $H(w_k, w(x))$ is the decoding function which computes the Hamming distance between w_k and $w(x)$.

We assign the class label of the closest codeword, i.e., with the shortest Hamming distance, to the testing sample. In the case of the code matrix given in Table 1, the ideal code word for class 1 is [1 1 0 1 1 1], and the ideal code word for class 5 is [1 1 1 0 1 0]. If a testing sample yields a code word [1 1 0 1 1 1], it will be determined as class 1 which corresponds to the shortest Hamming distance.

Motivated by providing new solutions to the problem of multi-class decomposition and extending the applications of ECOC, we propose a new data-driven decomposition approach. Different from current methods, it is a mechanism that adaptively designs the code matrix of ECOC based on the inherent structure of the training data. The method does not limit itself to any particular base learner. We then apply DECOC to two multi-class pattern recognition problems that have not yet been addressed by the ECOC approach so far [8].

2.2. Methodology: Data-driven ECOC

A Data-driven ECOC (DECOC) is proposed to design the code matrix for ECOC by choosing the code words utilizing the intrinsic information from the training data. In a preset decomposition mechanism, for a K-class problem such as pairwise coupling, K*(K-1)/2 base learners are always needed which can be a large number of base learners when K gets big. The key idea of DECOC is to selectively include some of the binary learners into the code matrix based on a confidence score defined for a binary base learner.

This measure is helpful to determine how likely a learner in the ensemble will be included.

Before introducing the confidence score, these parameters are difined:

- *separability criterion*

$$S(G)=\begin{cases} \frac{2}{|G|^2-|G|} \sum_{j\neq k, c_j c_k \in G} d(c_j, c_k) & |G|\neq 1 \ and \ |G|\neq K-1 \\ 0 & \end{cases}$$

where $d(c_j, c_k)$ is the distance between two classes c_j and c_k, which is the Euclidean distance of the mean or median vectors of the two classes; |*G*| is the size of the set G, i.e., number of classes in the group of samples associated with classes of *G ; 2/(|G|*^{2} *–|G|)* is the normalization factor. If there is only one class or there are *K-1* classes in *G*, then *S(G)* is set to 0 since both situations correspond to the 1vo partition of classes which is a special case to be considered separately. *S(G)* also indirectly describes the inherent homogeneity for a group of samples: the smaller *S(G)* is, the more homogenous the group of samples is. Note that sample groups associated with *G* are drawn from the training samples.

- *Confidence score*

$$C(f)=\begin{cases} \frac{S(G_{+/-}(f))}{S(G_+(f))+S(G_-(6))} & |G_+|\neq 1 \ \ and \ \ |G_+|\neq K-1 \\ 0 & \end{cases}$$

where G+(f) is the set of classes whose samples are considered as positive by the base learner f, G-(f) is the set of remaining classes whose samples are considered negative. For example, for the base learner 4 in Table 1, classes 1, 2 and 3 among 5 classes are considered as positive, so G+(f) = {1, 2, 3} and S(G+(f)) = 2/(32-3)*Σc_j,c_k{1,2,3} d(c_j,c_k). S(G+/-(f)) denotes the separability by viewing the data set as two groups of positive and negative samples separated by the base learner f. S(G+/-(f)) equals the distance between the two groups: S(G+/-(f)) = 2/(22-2)*d(c+, c-)= d(c+, c-), where c+ represents the group of all the samples considered as positive by the base learner f, and c- represents the group of all the samples considered as negative by f.

Figure 2 describes the flow of calculating the confidence scores and selecting the base learners, which is the core of the DECOC algorithm. DECOC is a data-driven approach of designing code matrix: instead of having a preset matrix, DECOC adaptively generates the code matrix based on the structure of the given training data.

Training data is divided into three different pattern classes; termination, bifurcation and non-minutiae. The

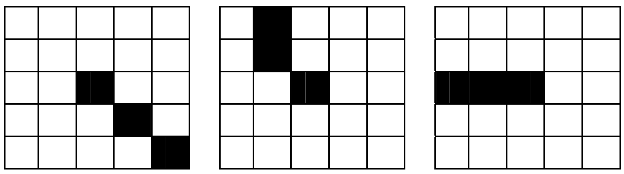
3 × 3 pixel window does not regard much information and the 7×7 pixel window shows too much information.

Therefore training data size is chosen to a 5 × 5 pixel window. The size of the window is deliberately an odd number to have a single pixel in the center.

A total of 136 different patterns have been gathered. The different classes have 32 termination patterns, 104 bifurcation patterns and 216 no minutia patterns. Patterns were carefully selected so the detection occurs in the center of the minutia. Patterns with minutiae off center are classified as non minutiae to avoid overlap detection. Figure 3 illustrates some used patterns for each class: Termination (a) and Bifurcation (b).

3. Experimental Results

The method of extracting minutiae from skeletonized fingerprints presented in this paper was evaluated by implementing it into the whole fingerprint recognition system. A database was assembled from pre-stored fingerprints in FVC2004 Db3. This database was chosen randomly with different qualities. Figure 4 shows the best and the worst fingerprint quality. The proposed system extracts minutiae from the skeletonized images.



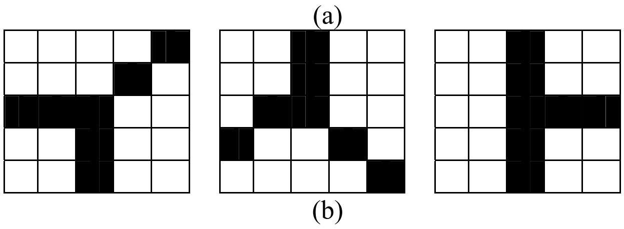


Figure 3. Sample of different patterns

(a) (b)

Figure 4. Different qualities of finger image: (a) good quality (b) bad quality

The following definitions are needed for the purpose of comparing the experimental results.

True minutia: A minutia point detected by an expert, $\mathbf{m_t}$.

False minutia: A minutia $\mathbf{m_a}$ which does not coincide with $\mathbf{m_t}$ is said to be a false minutia.

Dropped minutia: When a minutia $\mathbf{m_t}$ is not detected in $\mathbf{m_a}$, $\mathbf{m_t}$ is said to be a dropped minutia.

Exchanged minutia: A minutia $\mathbf{m_a}$ which corresponds to $\mathbf{m_t}$ with their types exchanged.

So, True Minutiae Ratio (TMR), False Minutiae Ratio (FMR), Dropped Minutiae Ratio (DMR), Exchanged Minutiae Ratio (EMR) and Average Computation Time were defined. Table 2 gives a comparaison of performance and computational time between Hwang methods [9], Jain's method [10], Ray's method [10] and our method.

Table 2. Comparison of performance and computational time

Factors	TMR %	FMR %	DMR %	EMR %	Average classification Time
Hwang without skele-tonization	75.32	22.5	10.18	14.5	35.9
Hwang with skele-tonization	79.20	48.60	6.20	14.60	105
Jain's	74.10	22.20	Not indicated		
Ray's	63.40	20.40	Not indicated		
DECOC	**78.23**	**15.07**	15.23	**3.08**	**93.75**

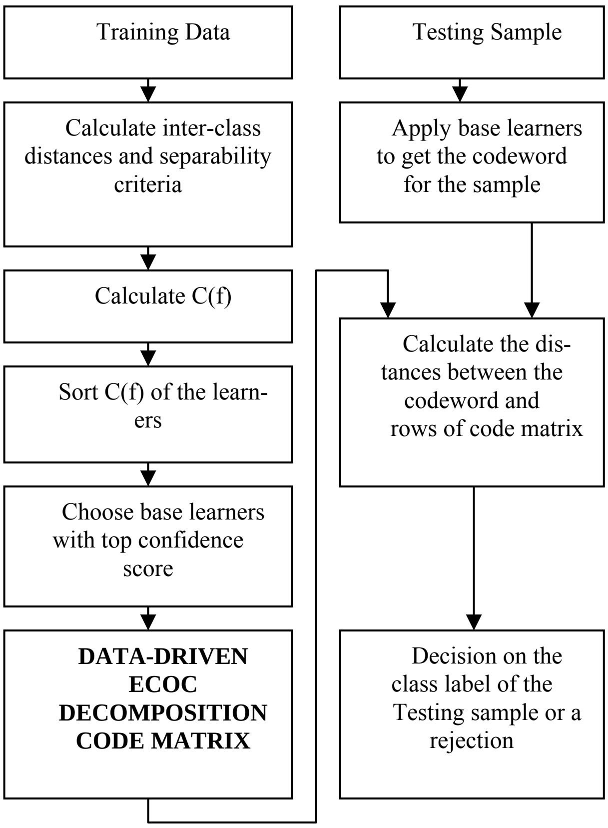


Figure 2. The flow of the training and testing algorithms for DECOC

4. Conclusions

A novel method for reliable and fast feature extraction based on classifier DECOC from skeleton fingerprint images have been proposed. This method classifies a bloc of 5x5 pixels into bifurcation, termination and no minutie. The experimental results show that the proposed method has the best TMR, the best EMR and a good average classification time.

References

1. D. Maltoni, D. Maio, A.K. Jain, and S. Prabhakar, Handbook of Fingerprint Recognition. Springer; 1st edition, 2003, ISBN 0- 387-95431-7.

2. B.G. Sherlock, D.M. Monro, and K. Millard, "Fingerprint enhancement by directional fourier filtering," Vision, Image and Signal Processing, IEE Proceedings-, vol. 141, no. 2, pp. 87–94, 1994.

3. Patil, P.M.; Suralkar, S.R.; Sheikh, F.B."Rotation invariant thinning algorithm to detect ridge bifurcations for fingerprint identification" Tools with Artificial Intelligence, 2005. ICTAI 05. 17th IEEE International Conference on Volume , Issue , 16-16 Nov. 2005 Page(s):8 pp. – 641 Digital Object Identifier 10.1109/ICTAI.2005.112.

4. N. Ratha, S. Chen, and A.K. Jain, "Adaptive flow orientation based feature extraction in fingerprint images," *Pattern Recognition*, vol. 28, no. 11, pp. 1657–1672, 1995.

5. A. Farina, Z.M. Kovacs-Vajna, and A. Leone, "Fingerprint minutiae extraction from skeletonized binary images," *Pattern Recognition*, vol. 32, no. 5, pp. 877–889, 1999.

6. D. Maio and D. Maltoni, "Direct gray-scale minutiae detection in fingerprints," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 19, no. 1, pp. 27–41, 1997.

7. X. Jiang, W.Y. Yau, and W. Ser, "Detecting the fingerprint minutiae by adaptive tracing the gray-level ridge," *Pattern ecognition*, vol. 34, no. 5, pp. 999–1013, 2001.

8. Jie Zhou, Hanchuan Peng, Ching Y Suen , "Data Driven Decomposition for Multiclass Classification", published on Pattern Recognition, 2007.

9. Hui-Yeoun Hwang, Jung-Hwan Shin, Sung-I1 chien, “Run Representation Based Minutiae Extraction in Fingerprint Image”, *MVA2OOZ* lAPR Workshop on Machine Vision Applications, Dec. 1 1 13, 2002, Nara- ken New Public Hall. Nara, Japan.

10. Qijun Zhao, Lei Zhang, David Zhang, Nan Luo, Adaptive Pore Model for Fingerprint Pore Extraction, 2008.
a check (mail orders only).