Fingerprint Identification: Classification vs. Indexing

Xuejun Tan, Bir Bhanu and YingQiang Lin
Center for Research in Intelligence Systems, University of California, Riverside, CA 92521, USA
{xtan, bhanu,yqlin}@cris.ucr.edu

Abstract

In this paper, we present a comparison of two key approaches for fingerprint identification. These approaches are based on (a) classification followed by verification, and (b) indexing followed by verification. The fingerprint classification approach is based on a novel feature-learning algorithm. It learns to discover composite operators and features that are evolved from combinations of primitive image processing operations. These features are then used for classification of fingerprint into five classes. The indexing approach is based on novel triplets of minutiae. The verification algorithm based on Least Square Minimization over each of the possible triplets minutiae pair is used for identification in both cases. On the NIST-4 fingerprint database, the comparison shows that, although correct classification rate can be as high as 92.8% for 5-class problems, the indexing approach performs better based on size of search space and identification results.

1. Introduction

In fingerprint identification system, the input is only a query fingerprint, the system tries to answer the question: are there any fingerprints in the database, which resemble the query fingerprint? There are three kinds of approaches to solve the fingerprint identification problem:

- The first approach is based on verification only. However, if the size of the database is large, this approach will be time-consuming and it is not practical for real-world applications.
- The second approach is based on classification. Traditional classification techniques attempt to classify fingerprints into five classes: Right Loop (R), Left Loop (L), Whorl (W), Arch (A), and Tented Arch (T). Figure 1 shows the examples of each class. The most widely used approaches of fingerprint classification are based on the number and relations of the Singular Points (SPs), which are defined as the points where a fingerprint's orientation field is discontinuous. Using SPs as the reference points, Karu and Jain [10] present a classification approach based on the structure information around SPs. Other research works, which use SPs as reference points, include Candela et al. [11], Halici and Ongun [12], and Jain et al. [3]. The problem with this kind approach is that it is not easy to detect the SPs and some fingerprints do not have SPs. The worst thing is that the uncertainty of the locations of SPs is large, which has great effect on the classification results. Cappelli et al. present a structural analysis of a fingerprint's orientation field [1], which is unnecessary to find the SPs. Researchers also tried different methods to combine different classifiers to improve the classification performance, i.e. Senior [13] and Yao et al. [14].

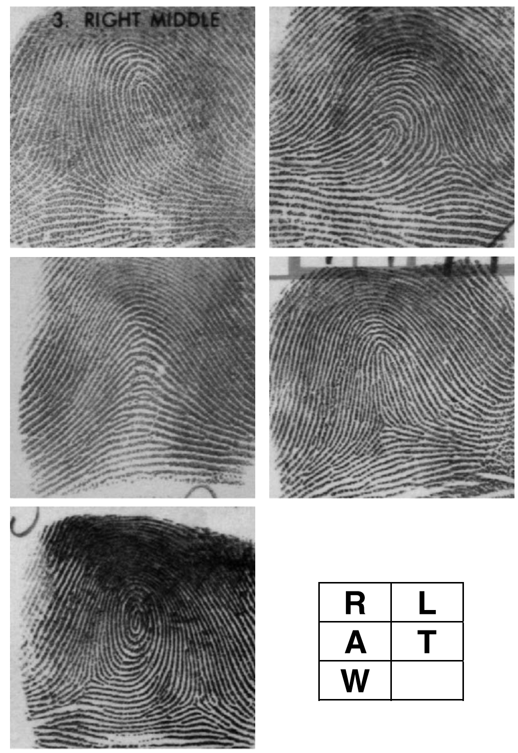
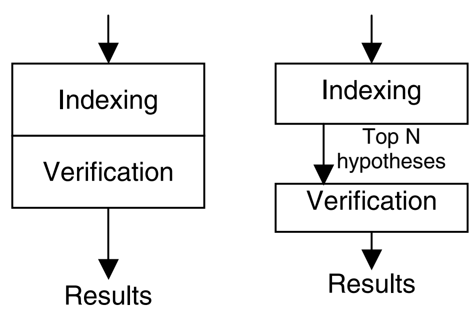


Figure 1. Examples of fingerprints from each class of *Henry System*.



(a) indexing and verification integrated (b) indexing and verification separated

Figure 2. Block diagram of Indexing followed by verification to solve the identification problem.

- The goal of the third approach is to significantly reduce the number of candidate hypotheses to be considered by the verification algorithm. These approaches are called indexing techniques in the fingerprint recognition area. A prominent approach for fingerprint identification is by Germain et al. [2], which integrates the indexing and verification in their approach (Figure 2(a)).

The contributions of this paper are: 1) we present a learning algorithm to learn the composite operator of primitive features automatically. It helps us to find some useful unconventional features, which are beyond the imagination of humans. 2) we integrated the classification technique into the fingerprint identification system. 3) extensive comparison between classification and indexing techniques are performed. 4) all experiments are carried out on NIST-4, a standard fingerprint database.

2. Technical Approach

2.1. Classification

Genetic programming was first proposed by Koza in [5]. Applications used GP for classification can be found in Poli [6], Stanhope and Daida [7] and Howard et al. [8]. Figure 3 shows the block diagram of our classification approach. During training, GP is used to generate compositor operators, which are applied to the primitive features generated from the original orientation field. Feature vectors are generated by feature generation operators and used for classification. A Bayesian classifier is used for classification. Fitness value is computed according to the classification result and used for evolving GP. During testing, compositor operators are applied directly to generate feature vectors. The major design considerations are explained in the following:

- **The Set of Terminals:** For a fingerprint, we can estimate the orientation field. The set of terminals used in this paper are called primitive features, which are generated from the orientation field. Primitive features used in our experiments are: 1) original orientation image; 2) mean, standard deviation, min, max and median images obtained by applying 3×3 and 5×5 templates on orientation image; 3) edge images obtained by applying sobel filters along horizontal and vertical directions on orientation image; 4) binary image obtained by thresholding the orientation image with a threshold of 90; and 5) images obtained by applying *sin* and *cos* operations on the orientation image. These 16 images are input to the composite operators. GP determines which operations are applied on them and how to combine the results.
- **The Set of Primitive Operators:** A primitive operator takes one or two input images, performs a primitive operation on them and outputs a resultant image. Suppose 1) A and B are images of the same size and c is a constant; 2) for operators, which take two images as input, the operations are performed on the pixel-by-pixel basis. Currently, there are two kinds of primitive operators in our approach: computation operators and feature generation operators. Table 1 explains the meaning of these operators in detail. For computation operators, the output is an image, which is generated by applying the corresponding operations on the input image. However, for feature generation operators, the output includes an image and a real number or vector. The output image is the same as the input image and passed as the input image to the next node in the composite operator. The size of the feature vectors depends on the number of the feature generation operators.

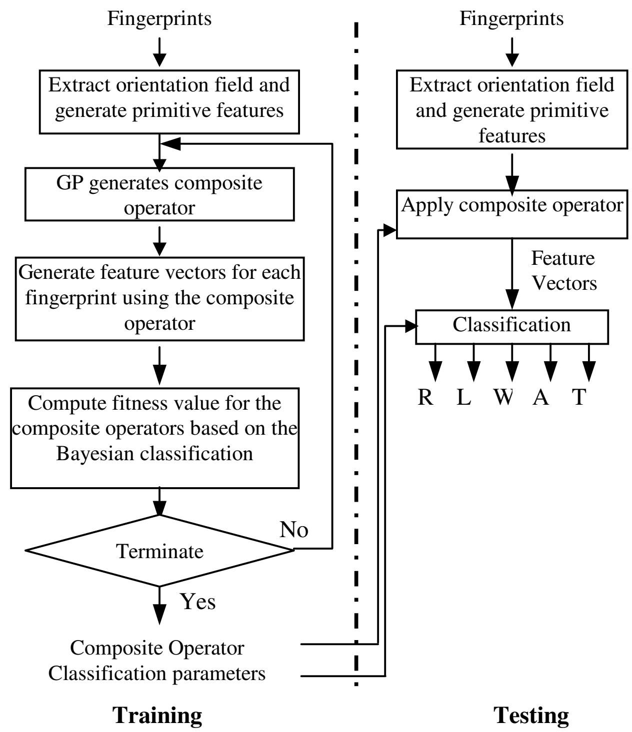


Figure 3. Block diagram of our classification approach.

- **Generation of New Composite Operator:** The initial population is randomly generated. The search of GP is done by performing reproduction, crossover and mutation operations. The reproduction operation used in our approach is the tournament selection. Crossover and mutation are standard, which could be found in [5]. And, we use *steady-state* GP in our experiments.
- **The Fitness Measure:** During training, in every generation for each composite operator proposed by GP, we estimate the Probability Distribution Function (PDF) of the feature vectors for each class using all the feature vectors obtained by applying special composite operators. Suppose the feature vectors for each class are normal distributed, $v_{i,j}$, where $i = 1,2,3,4,5$ and $j=1,2,\ldots n_i$, n_i is the number of feature vectors in the training for class i, ω_i . Then, for each i, we may estimate the mean μ_i, covariance matrix Σ_i, and PDF of ω_i by all $v_{i,j}$. According to Bayesian theory, we have:

Table 1. Primitive operators used in our approach.

Primitive Operator		Meaning
Computation Operators	ADD_OP, SUB_OP, MUL_OP and DIV_OP	A+B, A–B, A×B and A/B. If the pixel in B has value 0, the corresponding pixel in A/B takes the maximum pixel value in A.
	MAX2_OP and MIN2_OP	max(A,B) and min(A,B)
	ADD_CONST_OP, SUB_CONST_OP, MUL_CONST_OP and DIV_CONST_OP	A+c, A-c, A×c and A/c
	SQRT_OP and LOG_OP	$sign\ (A) \times \sqrt{\lvert A \rvert}$ and $sign\ (A) \times \log(\ \lvert A \rvert)$.
	MAX_OP, MIN_OP, MED_OP and STD_OP	max(A), min(A), med(A) and std(A), replace the pixel value by the maximum, minimum, median or standard deviation in a 3×3 block
	BINARY_ZERO_OP and BINARY_MEAN_OP	binarize A by zero or mean2(A)
	NEGATIVE_OP	-A
	LEFT_OP, RIGHT_OP, UP_OP and DOWN_OP	left(A), right(A), up(A) and down(A). Move A to the left, right, up or down by 1 pixel. The border is padded by zeros
	HF_DERIVATIVE_OP and VF_DERIVATIVE_OP	HF(A) and VF(A). Sobel filters along horizontal and vertical directions
Feature Generation Operators	SPE_MAX_OP, SPE_MIN_OP, SPE_MEAN_OP, SPE_ABS_MEAN_OP and SPE_STD_OP	max2(A), min2(A), mean2(A), mean2($\lvert$A$\rvert$) and std2(A)
	SPE_U3_OP and SPE_U4_OP	μ_3(A) and μ_4(A). Skewness and kurtosis of the histogram of A
	SPE_CENTER_MONMENT11_OP	μ_{11}(A). First order central moments of A
	SPE_ENTROPY_OP	H(A). Entropy of A
	SPE_MEAN_VECTOR_OP and SPE_STD_VECTOR_OP	mean_vector(A) and std_vector(A). A vector contains the mean or standard deviation value of each row/column of A

$$v \in \omega_k,\ iff.\ p(v|\omega_k) \cdot p(\omega_k) = \max_{i=1,2,3,4,5}(p(v|\omega_i) \cdot p(\omega_i)) \tag{1}$$

where v is a feature vector for classification.

During training, we estimate $p(x|\omega_i)$, then use the training set to do classification. The Percentage of Correct Classification (PCC) is taken as the fitness value of the composite operator:
$$Fitness\ Value = \frac{n_c}{n_s} \times 100\% \tag{2}$$
where n_c is the number of correctly classified fingerprints in the training set and n_s is the size of the training set.

- **Parameters and Termination:** The key parameters are the population size, the number of generations, the crossover rate and the mutation rate. The GP stops whenever it finishes the pre-specified number of generations.

2.2. Indexing

Our previous work [9] of fingerprint indexing follows Germain et al. [2] in that we also use the triplets of minutiae and ridge counts. However, in identification, the indexing and verification in our approach are separated (Figure 2(b)). In our approach, totally we have two steps. During the offline processing, the features of each template fingerprint are computed and used to construct the indexing space $H(\alpha_{min}, \alpha_{med}, \phi, \eta, \lambda, \chi, \xi)$. During the online processing, we compute the features for the query fingerprint and use them to search the indexing space $H(\alpha_{min}, \alpha_{med}, \phi, \eta, \lambda, \chi, \xi)$. If the feature values of two triangles, which are from two different fingerprints, are within some error tolerance, then they are potential corresponding triangles. The output of indexing is a list of hypotheses for potential match between the query and template fingerprints, which are sorted in a descending order of the number of potential corresponding triangles. Only the top N hypotheses are input to the verification.

The features we use to find potential corresponding triangles are defined in the following:

- **Angles α_{min} and α_{med}:** Suppose α_i are three angles in the triangle, $i = 1, 2, 3$. Let $\alpha_{max} = max\{\alpha_i\}$, $\alpha_{min} = min\{\alpha_i\}$, $\alpha_{med} = 180° - \alpha_{max} - \alpha_{min}$. P_1; P_2; P_3 are the labels of vertex of α_{max}, α_{min} and α_{med}, respectively.
- **Triangle Orientation ϕ:** Let $Z_i = x_i + jy_i$ be the complex number ($j = \sqrt{-1}$) corresponding to the coordinates (x_i, y_i) of point P_i, $i = 1,2,3$. Define $Z_{21} = Z_2 - Z_1$, $Z_{32} = Z_3 - Z_2$, and $Z_{13} = Z_1 - Z_3$. Let $\phi = sign(Z_{21} \times Z_{32})$, where $sign$ is the signum function and $\times$ is the cross product of two complex numbers.
- **Triangle Direction η:** Search the minutia from top to bottom and left to right, if the minutia is the start point of a ridge or valley, then $\nu = 1$, else $\nu = 0$. η is the combination of ν_i, ν_i is the ν value of point P_i, $i = 1,2,3$.
- **Maximum Side λ:** Let $\lambda = max\{L_i\}$, where $L_1 = |Z_{21}|$, $L_2 = |Z_{32}|$, and $L_3 = |Z_{13}|$.

- **Minutiae Density χ:** In a local area centered at the minutiae P_i, if there exists n_χ minutiae, then minutiae density $\chi_i = n_\chi$. χ is a vector consisting of all χ_i*'s* .
- **Ridge Counts ξ:** ξ_1 is the ridge count of the side P_1P_2 , ξ_2 is the ridge count of the side P_2P_3 , and ξ_3 is the ridge count of the side P_3P_1 . ξ is a vector consisting of all ξ_i's.

If two triangles from two different fingerprints of the same finger satisfy the following criteria, then they are potential corresponding triangles. The criteria are:

$$\left|\alpha'_{min} - \alpha''_{min}\right| \leq T_{\alpha_{min}},\ \left|\alpha'_{med} - \alpha''_{med}\right| \leq T_{\alpha_{med}}$$

$$\phi' = \phi'',\ \eta' = \eta'',\ \left|\lambda' - \lambda''\right| \leq T_\lambda$$

$$\left|\chi'_i - \chi''_i\right| \leq T_\chi,\ \left|\xi'_i - \xi''_i\right| \leq T_\xi,\ i = 1,2,3 \qquad (3)$$

where $T_{\alpha_{min}}$, $T_{\alpha_{med}}$, T_λ, T_χ, and T_ζ are thresholds to deal with the local distortions.

2.3. Verification

Verification follows classification and indexing. For indexing, verification is simple, since after indexing, we know the potential corresponding triangles and we can use this information in the verification directly. However, for classification, we only know the class information. So, we have to find the potential corresponding triangles between the query fingerprint and each template fingerprint that belongs to the same class as the query fingerprint. Then we can apply the following verification approach.

Suppose the sets of minutiae in the template and the query fingerprints are $\{(t_{n,1}\ ,\ t_{n,2})\}$ and $\{(q_{m,1}\ ,\ q_{m,2})\}$ respectively, where $n = 1,2,3,\ldots N$, $m = 1,2,3,\ldots M$. The number of minutiae in the template and the query fingerprints are N and M, respectively. Let Δ_t and Δ_q be two potential corresponding triangles in the template and the query fingerprints, respectively. The coordinates of the vertices of Δ_t and Δ_q are $(x_{i,1}\ ,\ x_{i,2})$ and $(y_{i,1}\ ,\ y_{i,2})$, respectively, and $i = 1,2,3$. Suppose $X_i = [\ x_{i,1}\ ,\ x_{i,2}\ ]'$, $Y_i = [\ y_{i,1}\ ,\ y_{i,2}\ ]'$, and the transformation $Y_i = F(X_i)$ can be expressed as: $Y_i = s \cdot R \cdot X_i + T$ (4)

where s is the scaling factor, $R = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$, θ is the angle of rotation between two fingerprints, and $T = [t_1\ ,\ t_2]'$ is the vector of translation.

We estimate the transformation parameters by minimizing error ε^2, which is the sum of the squared distances between the transformed template points and their corresponding query points. The solution is:

$$\hat{\theta} = \arctan\left(\frac{B}{A}\right),\quad \hat{s} = \frac{\sum_{i=1}^{3}\{(X_i - \bar{X})'\hat{R}'(Y_i - \bar{Y})\}}{\sum_{i=1}^{3}\{(X_i - \bar{X})'(Y_i - \bar{Y})\}}$$

$$\hat{T} = \bar{Y} - \hat{s}\cdot\hat{R}\cdot\bar{X}$$

where

$$A = \sum_{i=1}^{3}\{(\bar{x}_1 - x_{i,1})(y_{i,1} - \bar{y}_1) + (\bar{x}_2 - x_{i,2})(y_{i,2} - \bar{y}_2)\}$$

$$B = \sum_{i=1}^{3}\{(\bar{x}_1 - x_{i,1})(y_{i,2} - \bar{y}_2) - (\bar{x}_2 - x_{i,2})(y_{i,1} - \bar{y}_1)\}$$

$$\bar{X} = \begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \end{bmatrix} = \sum_{i=1}^{3}\bar{X}_i,\quad \bar{Y} = \begin{bmatrix} \bar{y}_1 \\ \bar{y}_2 \end{bmatrix} = \sum_{i=1}^{3}\bar{Y}_i$$

$$\hat{R} = \begin{bmatrix} \cos\hat{\theta} & -\sin\hat{\theta} \\ \sin\hat{\theta} & \cos\hat{\theta} \end{bmatrix},\quad \hat{T} = \begin{bmatrix} \hat{t}_1 \\ \hat{t}_2 \end{bmatrix} \qquad (5)$$

If $\hat{s}$, $\hat{\theta}$, $\hat{t}_1$ and $\hat{t}_2$ are less than certain thresholds, then we take them as the parameters of the transformation between two potential corresponding triangles Δ_t and Δ_q . Based on the transformation $\hat{F}(\hat{s},\hat{\theta},\hat{t}_1,\hat{t}_2)$, $\forall\ j$, $j = 1,2,3,\ldots N$, we compute:

$$d = \underset{k}{\arg\min}\left\{\left\|\hat{F}\left(\begin{bmatrix} t_{j,1} \\ t_{j,2} \end{bmatrix}\right) - \begin{bmatrix} q_{k,1} \\ q_{k,2} \end{bmatrix}\right\|\right\} \qquad (6)$$

If d is less than a threshold T_d , then we define the points $[\ t_{j,1}\ ,\ t_{j,2}\ ]'$ and $[\ q_{k,1}\ ,\ q_{k,2}\ ]'$ are corresponding points. If the number of corresponding points based on $\hat{F}(\hat{s},\hat{\theta},\hat{t}_1,\hat{t}_2)$ is greater than a threshold T_n , then we define Δ_t and Δ_q as the *corresponding triangles* between the template and the query fingerprints. The identification score is the number of corresponding triangles between the query and template fingerprints.

3. Experimental Results

NIST Special Database 4 (NIST-4) [4] is used in our experiments, which contains a large portion of poor quality images. Totally, there are 2000 pairs of fingerprints in NIST-4. Some sample fingerprints are shown in Figure 1.

Table 2. Confusion matrix for 5-class classifications.

	R	L	W	A	T
R	**180**	1	6	2	5
L	5	**188**	6	1	10
W	1	3	**187**	0	2
A	1	3	0	**208**	6
T	14	2	1	4	**172**

3.1. Classification Results

We use the first 1000 pairs of fingerprints for training. In order to reduce the effect of overfitting, we use only the first 500 pairs to estimate the parameters for each class and use the entire training set to validate the training results. Since we want to compare the results of classification and indexing, we only test the second impression of the second 1000 pairs of fingerprints. The first impressions of the second 1000 pairs of fingerprints are used as templates in verification. The parameters in

Table 3. Classification results on NIST-4.

Approaches	Class #	Error rate %	Reject rate %	Dataset	Comments
Karu and Jain 1996 [10]	5	14.6	zero	4000 images, no training	Decision based on topological information
Jain et al. 1999 [3]	5	14.6	**1.8**	Training: First 1000 pairs of images Testing: Second 1000 pairs of images	KNN
	5	13.6			Neural Network
	5	**10.0**			KNN+NN, two stage classifier
Senior 2001 [13]	4	Averagly 8.5	zero	Same as above	Neural Network fusion with priors
Yao et al. [14]	5	10.0	1.8	Same as above	SVM+RNN
This approach	5	**7.2**	**zero**	Training: First 1000 pairs of images Testing: Second impression of the second 1000 pairs of images	GP based learned features + Bayesian classifier

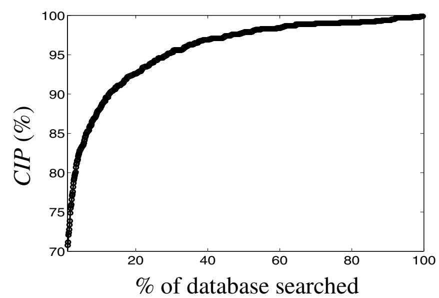


Figure 4. Indexing performance.

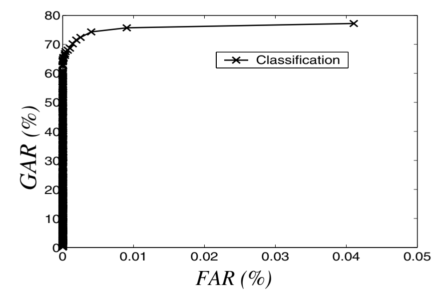


Figure 5. Identification results using classification based approach.

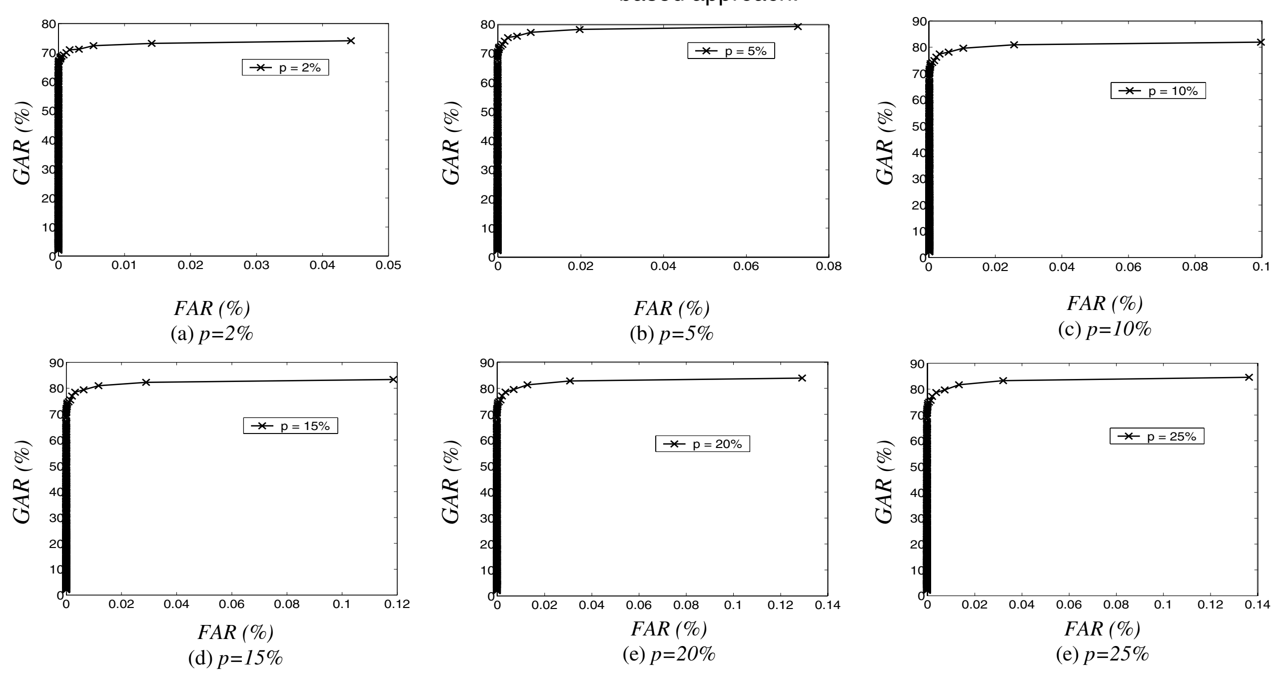


Figure 6. Identification results using indexing based approach.

our experiments are: Maximum size of composite operator 150, population size 100, mutation rate 0.05, crossover rate 0.6 and number of generation 100.

We did the experiments 10 times and took the best result as the learned composite operator. Table 2 shows the confusion matrix of our testing results of the second 1000 pairs of fingerprint in NIST-4. Note that, because of bad quality, the ground truths of some fingerprints provided by NIST-4 contain 2 classes, e.g. the ground truth labels of f0008_10 include class T and L. We only use the first ground truth label to estimate the parameters of classifiers. However, in testing, we use all the ground truth labels and consider it as a correct classification, if the output of the composite operator matches to one of the ground truth labels. The PCC is 92.8% for 5-class classifications. Table 3 shows the results on NIST-4 database reported in the literatures. Considering that we have not rejected any fingerprints from NIST-4, our result is one of the best.

3.2. Indexing Results

In order to compare the results between indexing and classification, we only do indexing experiments on the second impressions of the second 1000 pairs of fingerprints. Figure 4 shows the Correct Indexing Power (CIP), which is defined as the percentage of correctly indexed queries based on the percentage of hypotheses that need to be searched in the verification step. We observe that CIP increases as p, the percentage of the database searched, increases. The CIP are 83.3%, 88.1%, 91.1%, and 92.6% for p are 5%, 10%, 15%, and 20%, respectively. As p reached about 60%, the relation between CIP and p becomes linear.

3.3. Identification Results

For classification, since the number of classes in fingerprint is small, we have to check more hypotheses in verification. For example, the classification result of our approach is one of the best results reported in published papers, however, we can only classify fingerprints into 5 classes. Since each class is uniformly distributed in NIST-4, after classification, about 200 hypotheses need to be considered in verification. And, this number can not be tuned. As for indexing, since CIP varies according to the size of the search space, we have different performances of identification by indexing approach depending on the percentage of the database that is searched. Conceptually, each fingerprint as a query is verified against all the stored fingerprint templates. That is 1,000,000 verifications. Among them, 999,000 verifications are estimating False Acceptance Rate (FAR) and 1,000 verifications are for estimating Genuine Acceptance Rate (GAR). The Receiver Operating Characteristic (ROC) curve is defined as the plot of GAR against FAR. Based on different CIP, we have different ROCs for identification results for indexing based approach and only one ROC for classification based approach.

Figure 5 and Figure 6 show identification results based on classification and indexing, respectively. Using the classification based approach, the GAR is 77.2% when FAR is 4.1×10^{-2}%, while using the indexing based approach with p=5%, the GAR is 77.2% and FAR is 8.0×10^{-3}%. It shows that in order to achieve similar GAR in identification, we only need to search 5% of the database by indexing based approach for identification, while classification based approach for identification may need to search 20% of the entire search space. FAR for indexing based approach is much less than that for classification based approach.

4. Conclusions

In this paper, we compared the performance of two approaches for identification. One is the traditional approach that first classifies a fingerprint into one of the five classes (L,R,W,A,T) and then perform verification. The alternative approach is based on indexing followed by verification. Using state of the art highly competitive approach for classification, indexing and verification, we compared the performance of the two approaches for identification using NIST-4 fingerprint database. We find that indexing technique performs better considering the size of search space (5% vs. 20%) that need to be searched. Also for the same GAR, the FAR performance (8.0×10^{-3}% vs. 4.1×10^{-2}%) of indexing based approach is lower. Thus, the indexing based approach provides a potential alternative to the traditional classification based approach commercially used for fingerprint identification.

Acknowledgments: This work is supported in part by a grant from SONY, DiMI, I/O Software and F49620-02-1-0315. The contents and information do not necessarily reflect the positions or policies of the sponsors.

References

1. R. Cappelli, A. Lumini, D. Maio and D. Maltoni, Fingerprint classification by directional image partitioning, IEEE Trans. PAMI 21(5), pp. 402-421, 1999.
2. R.S. Germain, A. Califano and S. Colville, Fingerprint matching using transformation parameter clustering, IEEE Computational Science and Engineering, 4(4), pp. 42-49, 1997.
3. A.K. Jain, S. Prabhakar, L. Hong, A multichannel approach to fingerprint classification, IEEE Trans. PAMI 21(4), pp. 348-359, 1999.
4. C.I. Watson and C.L. Wilson, NIST special database 4, fingerprint database, U.S. National Institute of Standards and Technology, 1992.
5. J.R. Koza, Genetic Programming II: Automatic Discovery of Reusable Programs, MIT Press, 1994.
6. R. Poli, Genetic programming for feature detection and image segmentation, Evolutionary Computation, T.C. Forgarty Ed., pp. 110-125, 1996.
7. S.A. Stanhope and J.M. Daida, Genetic programming for automatic target classification and recognition in synthetic aperture radar imagery, Proc. Evolutionary Programming VII, pp. 735-744, 1998.
8. D. Howard, S.C. Roberts, and R. Brankin, Target detection in SAR imagery by genetic programming, Advances in Eng. Software, 30(5), pp. 303-311, May 1999.
9. B. Bhanu and X. Tan, Fingerprint indexing based on novel features of minutiae triplets, IEEE Trans. PAMI, May 2003.
10. K. Karu and A.K. Jain, Fingerprint classification, Pattern Recognition, vol. 29, no. 3, pp. 389-404, 1996.
11. 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, Technical Report NISTIR 5647, NIST, Apr. 1995.
12. U. Halici and G. Ongun, Fingerprint classification through self-organizing feature maps modified to treat uncertainties, Proc. IEEE, vol. 84, no. 10, pp. 1497-1512, Oct. 1996.
13. A. Senior, A combination fingerprint classifier, IEEE Trans. on PAMI, vol. 23, no. 10, pp. 1165-1174, Oct. 2001.
14. Y. Yao, G.L. Marcialis, M. Pontil, P. Frasconi and F. Roli, Combining flat and structured representations for fingerprint classification with recursive neural networks and support vector machines, Pattern Recognition, vol. 36, no. 2, pp. 397-406, Feb. 2003.