2014

Interoperability of fingerprint sensors and matching algorithms

Luca Lugini

Interoperability of Fingerprint Sensors and Matching Algorithms

by

Luca Lugini

Thesis submitted to the College of Engineering and Mineral Resources
at West Virginia University
in partial fulfillment of the requirements
for the degree of

Master of Science
in
Computer Science

Approved by

Bojan Cukic, Committee Chairperson
Tim Menzies
Thirimachos Bourlai

Lane Department of Computer Science and Electrical Engineering

Morgantown, West Virginia
2014

Keywords: Fingerprint interoperability, fingerprint recognition

ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 - 1346

Abstract

Interoperability of Fingerprint Sensors and Matching Algorithms

by Luca Lugini

Biometric systems are widely deployed in governmental, military and commercial/civilian applications. There are a multitude of sensors and matching algorithms available from different vendors. This creates a competitive market for these products, which is good for the consumers but emphasizes the importance of interoperability. In fingerprint recognition, interoperability is the ability of a system to work with a diverse set of fingerprint devices. Variations induced by fingerprint sensors include image resolution, scanning area, gray levels, etc. Such variations can impact the quality of the extracted features, and cross-device matching performance. This is true even when dealing with fingerprint sensors of the same sensing technology.

In this thesis, we perform a large-scale empirical study of the status of interoperability between fingerprint sensors and assess the performance consequence when interoperability is lacking. Additionally we develop a method to increase interoperability in fingerprint-based recognition systems deploying optical fingerprint sensors. A set of features to measure differences in fingerprint acquisition is designed and evaluated. Finally, different fusion schemes based on machine learning are tested end evaluated in order to exploit the designed set of features. Experimental results show that the proposed approach is able to reduce cross-device match error rates by a significant margin.

Acknowledgements

I would like to express my thanks to my advisor and committee chair Dr. Bojan Cukic. During the two years of my Master's program he has been a mentor and has provided me with priceless advices in research and beyond. His teachings always motivated me to do more than I thought I was capable of.

I wish to acknowledge the help provided by Dr. Tim Menzies and Dr. Thirimachos Bourlai for things I learned from them in class and during helpful discussions.

I am particularly grateful for the assistance given by Dr. Emanuela Marasco. Working together this past year made this work easier and more enjoyable. I can say with confidence that working on several different projects would have been impossible without her guidance.

Finally my most important acknowledgements are towards my family who have always motivated and supported me unconditionally. Without them I would not be where I am today.

Contents

1 Introduction **1**

1.1 Motivation 1

1.2 Contribution 3

1.3 Organization 4

2 Related Work **5**

2.1 Biometric System Evaluation 5

2.2 Fingerprint Interoperability 5

2.3 Machine Learning Algorithms 6

3 Interoperability Analysis and Enhancement Feature Set **11**

3.1 Data set 11

3.2 Interoperability Assessment 13

3.3 Interoperability Enhancement Feature Set 14

4 Integration of Features **34**

4.1 Feature Selection 38

4.2 Pairwise Models 50

4.3 Introduction of new sensors 51

4.4 Discussion 53

4.5 Threats to Validity 54

5 Conclusions and Future Work 60

5.1 Conclusions 60

5.2 Future Work 61

List of Figures

1.1 Examples of fingerprints captured using different devices. 2

1.2 Intra-device match score distributions. 3

1.3 Cross-device match score distributions. 4

3.1 Subjects information 12

3.2 DET curves for Identix matcher. 15

3.3 DET curves for Bozorth matcher. 16

3.4 DET curves for WebKey matcher. 17

3.5 NFIQ quality distributions for the different devices used in this study. 18

3.6 Mitre quality distributions for the different devices used in this study. 19

3.7 Frequency of low match scores across qualities 19

3.8 Box plots of minutiae count for the different devices. 21

3.9 Distributions of image gradients for the five sensors. 22

3.10 Grey level mean distributions for the different sensors. 23

3.11 Grey level standard deviation distributions for the different sensors. 24

3.12 PRNU distributions for the different sensors. 25

3.13 Energy distributions 26

3.14 Example of energy distributions 27

3.15 Entropy distributions 28

3.16 Maximum of LBP histograms for the different sensors. 30

3.17 Standard deviation of LBP histograms for the different sensors. 31

3.18 Maximum of LPQ histograms for the different sensors. 32

3.19 Standard deviation of LPQ histograms for the different sensors. 33

4.1 Architecture of the proposed approach. When building the model, we extract the interoperability features from a set of fingerprints acquired using different devices. We apply a feature selection technique, then train a machine learning algorithm. In the operational phase, we extract the interoperability features from the probe fingerprint, combine such 35

4.2 Random Forest performance on Identix data set. 38

4.3 Random Forest performance on Bozorth data set. 39

4.4 Random Forest performance on WebKey data set. 40

4.5 Random Forest performance on Identix data set after InfoGain feature selection. . . 43

4.6 Random Forest performance on Bozorth data set after InfoGain feature selection. . 44

4.7 Random Forest performance on WebKey data set after InfoGain feature selection. . 45

4.8 Neural Network performance on Identix data set after CFS feature selection. 46

4.9 Neural Network performance on Bozorth data set after CFS feature selection. . . . 47

4.10 Neural Network performance on WebKey data set after CFS feature selection. . . . 48

4.11 Random Forest performance on Identix data set after CFS feature selection. 49

4.12 Random Forest performance on Bozorth data set after CFS feature selection. 50

4.13 Random Forest performance on WebKey data set after CFS feature selection. . . . 51

4.14 DET curves for pairwise model pertaining to devices D0 and D2. 52

4.15 DET curves for pairwise model pertaining to devices D0 and D3. 53

4.16 DET curves for pairwise model pertaining to devices D2 and D4. 54

4.17 DET curves when introducing D0 as new device in the system. 55

4.18 DET curves when introducing D1 as new device in the system. 56

4.19 DET curves when introducing D2 as new device in the system. 57

4.20 DET curves when introducing D3 as new device in the system. 58

4.21 DET curves when introducing D4 as new device in the system. 59

List of Tables

3.1 Characteristics of the Live-scan devices used for fingerprint acquisition. 11

3.2 Statistical test on correlation of match scores. 13

3.3 Matching scenarios table. 14

4.1 Naive Bayes performance on Identix data set. 36

4.2 Naive Bayes performance on Bozorth data set. 36

4.3 Naive Bayes performance on WebKey data set. 36

4.4 Decision tree performance on Identix data set. 37

4.5 Decision tree performance on Bozorth data set. 37

4.6 Decision tree performance on WebKey data set. 37

4.7 Naive Bayes performance on Identix data set after InfoGain feature selection. . . . 40

4.8 Naive Bayes performance on Bozorth data set after InfoGain feature selection. . . . 41

4.9 Naive Bayes performance on WebKey data set after InfoGain feature selection. . . 41

4.10 Decision tree performance on Identix data set after InfoGain feature selection. . . . 41

4.11 Decision tree performance on Bozorth data set after InfoGain feature selection. . . 42

4.12 Decision tree performance on WebKey data set after InfoGain feature selection. . . 42

4.13 Naive Bayes performance on Identix data set after CFS feature selection. 43

4.14 Naive Bayes performance on Bozorth data set after CFS feature selection. 44

4.15 Naive Bayes performance on WebKey data set after CFS feature selection. 44

4.16 Decision tree performance on Identix data set after CFS feature selection. 46

4.17 Decision tree performance on Bozorth data set after CFS feature selection. 46

4.18 Decision tree performance on WebKey data set after CFS feature selection. 47

4.19 SVM performance on Identix data set after CFS feature selection. 47

4.20 SVM performance on Bozorth data set after CFS feature selection. 48

4.21 SVM performance on WebKey data set after CFS feature selection. 48

Chapter 1

Introduction

1.1 Motivation

Fingerprint-based user authentication is one of the most prolific commercial branches of biometrics. The low error rates and the extensive research in this area have made fingerprints the most widely adopted biometric traits in biometric-based user recognition systems.

A biometric system is typically developed in two distinct phases:

1. Enrollment: the user presents a sample of biometric data (called gallery o template) and is registered into the system.

2. Recognition: the user presents another biometric sample (called probe) whenever he/she requires access to the system.

A fingerprint matching operation therefore requires two fingerprint image samples. Most of the systems currently deployed do not require the fingerprint sensor used for enrollment to be the same as the sensor used at authentication time. Fingerprint sensors belong to three main categories:

- Optical: the finger is placed on the surface of a transparent prism which is typically illuminated through the left side and the image is taken using a camera. The light entering the prism is reflected at the valleys and absorbed at the ridges of a fingerprint.

- Solid-state: the finger is modeled as the upper electrode of the capacitor, while the metal plate is modeled as the lower electrode. The variation in capacity between valleys and ridges can be measured when the finger is placed on the sensor. In case of swipe solid-state sensors, impressions are obtained by swiping the finger on the surface of the sensor.
- Ultrasound: this type of sensors exploit difference of acoustic impedance between the skin of the ridges and the air of the valleys of a finger.

Even within a specific sensing technology the acquisition may vary across sensors. Different arrangements of the sensing elements in each device may introduce variations in the data. In particular, differences in resolution and scanning area impact the feature set extracted from the acquired images. In Figure 1.1 we can see the differences in the fingerprint of the same individual captured with the five sensors used in this study.

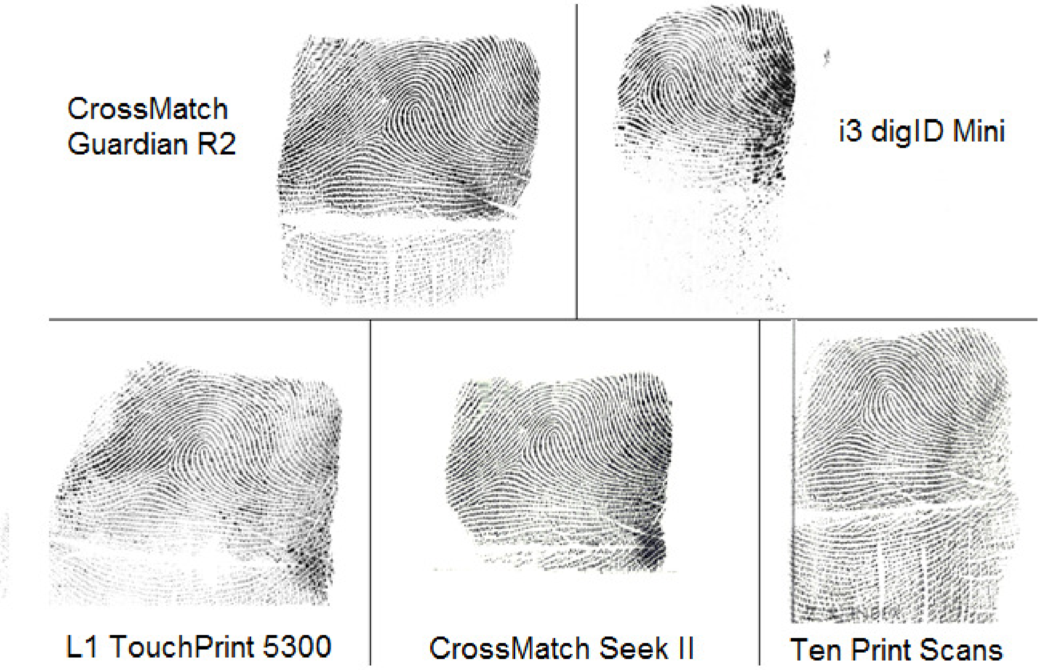

Figure 1.1: Fingerprints belonging to the same user, captured with five different sensors.

Interoperability is the ability of a fingerprint system to handle variations introduced by different sensors when acquiring fingerprints. We observed from our experiments that there is a lack of interoperability between optical fingerprint sensors. We can see from by comparing Figures 1.2 and 1.3 that for matches involving fingerprints acquired using two different devices, the value of genuine match scores decrease. This leads to more questionable matches and, ultimately, to increased error rates.

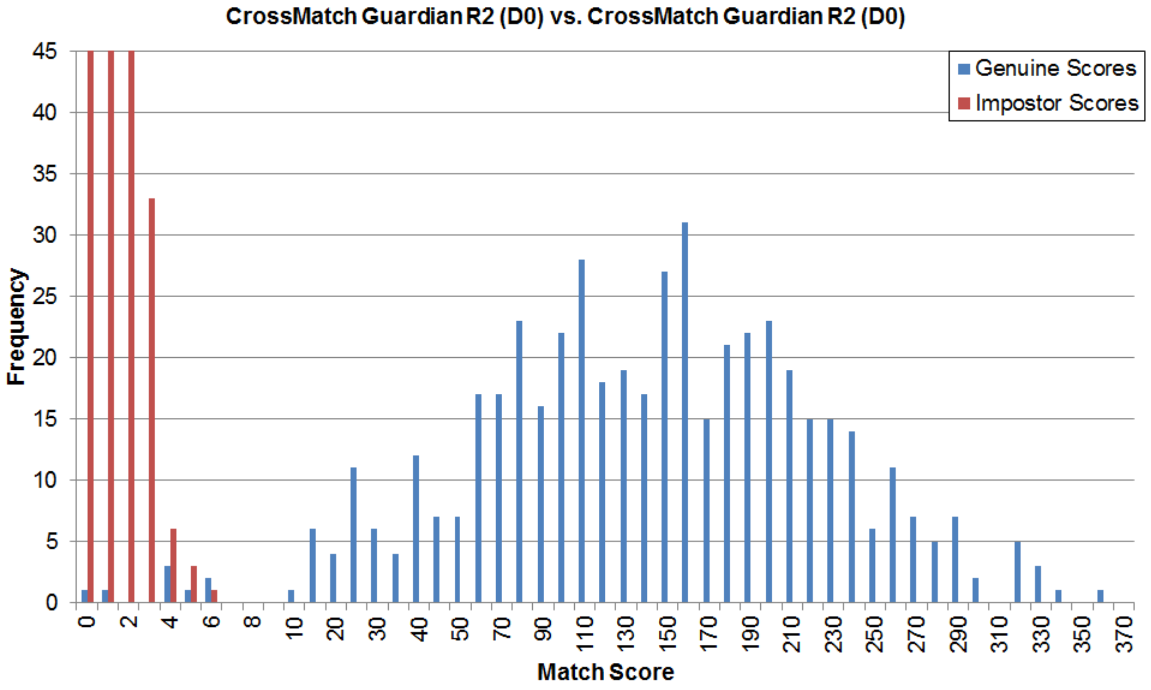

Figure 1.2: Genuine and impostor match score distributions an intra-device scenario. In this case the genuine score distribution and the impostor score distribution have little overlap, leading to good matching performance.

1.2 Contribution

The main contribution of this work is the development of a fingerprint-based recognition system which is able to compensate for the lack of interoperability between optical fingerprint devices. The proposed enhancement scheme will work in parallel to an existing fingerprint recognition system, exploiting additional information from fingerprints. In this work we define and evaluate a set of qualitative measures that can be extracted from a fingerprint image, and assess different fusion schemes in order to combine such measures.

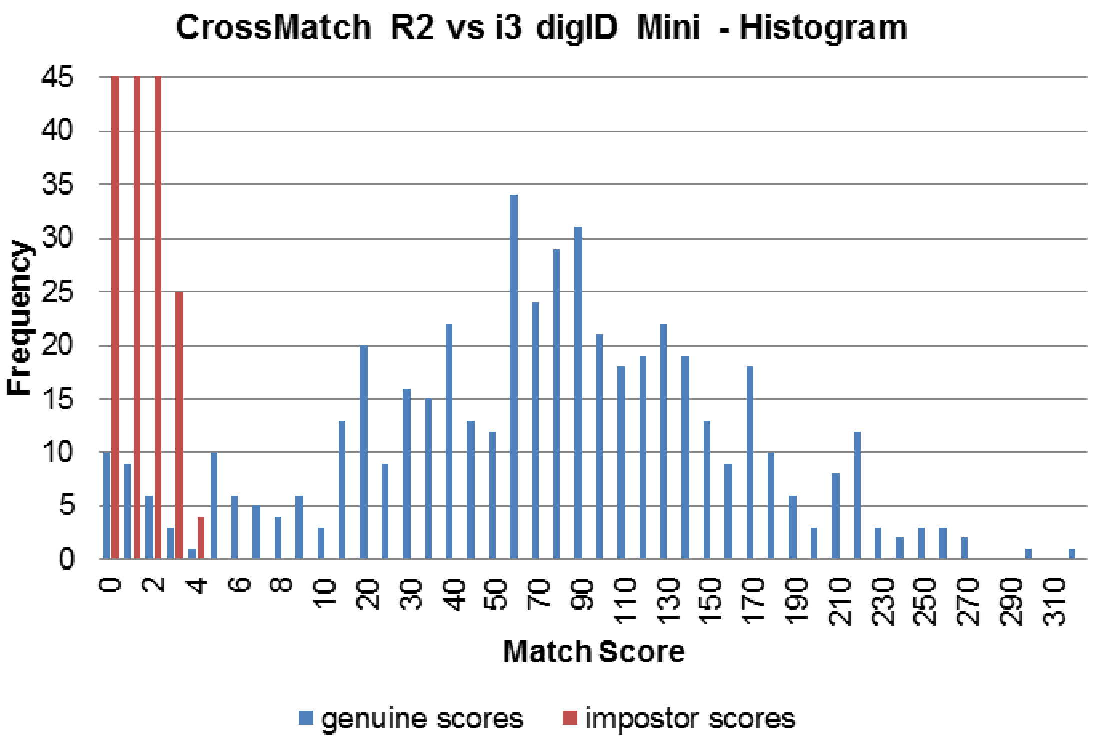

Figure 1.3: Genuine and impostor match score when matching fingerprint acquired using two different sensors. Compared to the intra-device matching scenario the genuine score distribution is shifted towards lower scores. More importantly, the tail of the distribution is more pronounced, leading to more overlap between genuine and impostor scores. This causes an increase of error rates.

1.3 Organization

The rest of this thesis is organized as follows. Chapter 2 presents a literature review of previous research in interoperability issues for fingerprint systems. Chapter 3 contains an analysis of match scores in two different matching scenarios, i.e. intra- and inter-device, and introduces the features used in our proposed approach. Chapter 4 outlines different score fusion techniques for the set of features described in Chapter 3, and presents results. Chapter 5 concludes this work and outlines future directions related to interoperability of fingerprint sensors and algorithms.

Chapter 2

Related Work

This section describes the literature on fingerprint interoperability, and the techniques used in the model we developed.

2.1 Biometric System Evaluation

In a biometric system that uses match score alone, thresholding on the match score is applied in order to decide whether a match is genuine or impostor. By changing the threshold and computing False Accept Rates (FAR) and False Reject Rates (FRR) we can construct a Detection Error Tradeoff (DET) curve. Usually the operating point on the DET curve is chosen by deciding a FAR required by the desired level of security in our system, and computing the correspondent threshold. Choosing an appropriate threshold is effectively compromising between FAR and FRR: lowering the FAR will result in an increase of the FRR, and vice versa.

2.2 Fingerprint Interoperability

Poh et al. proposed various methods to mitigate the impact of a mismatch between the acquisition devices of gallery and query samples [13] [19]. They investigated the problem of matching a gallery sample to a query which is acquired using an unknown device. This scenario was modeled using a Bayesian Belief Network (BBN) in several different configurations. A set of quality measures were extracted from the query samples and the acquisition device is inferred based on such measures. Clustering was applied to each device to explain hidden quality factors. Additionally, the BBN modeled the impact of the quality measures on match scores and class labels (i.e. genuine or impostor).

Jain and Ross analyzed the problem of the interoperability of a biometric system in terms of the variability introduced in the feature set by different sensor technologies [21]. The baseline Equal Error Rates (EER) when using only one device are 6.14% for the Digital Biometrics sensor, and 10.39% for Veridicom sensor. When moving to inter-device matching the EER increases to 23.13%.

Subsequently, Ross and Nadgir developed a model to compensate for distortions between fingerprint images acquired with different sensors [14]. They proposed a non-linear calibration scheme based on a thin-plate spline model in order to compensate for variations in minutiae distributions across multiple sensors. The model parameters are computed from a small set of images, based on manually established landmarks. The main disadvantage in this work is, therefore, the fact that the system is not completely automated.

2.3 Machine Learning Algorithms

We experimented with several algorithms: Naive Bayes, Decision Tree, Random Forest, Neural Network, Support Vector Machine (SVM).

Naive Bayes is a classification algorithm based on Bayes rule:

$$P(Y=y_i|X=x_k)=\frac{P(X=x_k|Y=y_i)P(Y=y_i)}{\sum_j P(X=x_k|Y=y_j)P(Y=y_j)} \tag{2.1}$$

where $X=(X_1,X_2,...,X_n)$ is the set of features and Y is the class variable. In our case Y is a binary variable: $Y=0$ if the match is an impostor, while $Y=1$ if the match is genuine. The training phase of the algorithm consists of estimating $P(X|Y)$ and $P(Y)$. While estimating $P(X)$ is not computationally expensive, in order to accurately estimate $P(X|Y)$ an unbiased Bayesian classifier would need to estimate approximately 2^{n+1} parameters. [1]

Obviously this is an impractical solution for any application. Naive Bayes greatly simplifies the probability estimation by assuming conditional independence between features. Naive Bayes therefore estimates $P(X|Y)|$ as follows

$$P(X_1...X_n|Y) = \prod_{i=1}^{n} P(X_i|Y) \tag{2.2}$$

Thus the Naive Bayes classification rule is given by

$$Y \leftarrow \arg\max_{y_k} \frac{P(Y = y_k)\prod_i P(X_i|Y = y_k)}{\sum_j P(Y = y_j)\prod_i P(X_i|Y = y_j)} \tag{2.3}$$

Artificial Neural Networks are learning algorithms inspired by the structure of human brain. Unlike computers the human brain consists of a large number of processing units (approximately 10^{11}) called *neurons*. Each neuron is connected to approximately 10^4 others through *synapses*. Furthermore, processing and memory are distributed over the network: while the memory is in the synapses, processing is carried out by neurons. Artificial neural networks model a neuron as a perceptron. Each perceptron has inputs that may be outputs of other perceptrons, or may come from the environment. Additionally, each input connection is associated with a *connection weight*, or *synaptic weight*. The perceptron is a very simple computational unit, which implements some kind of thresholding function. The most commonly used function is the sigmoid function:

$$S(x) = \frac{1}{1 + exp(-x)} \tag{2.4}$$

The output of a perceptron is given by:

$$y = \frac{1}{1 + exp(-\sum_{i=0}^{d} w_i x_i)} \tag{2.5}$$

[1] https://www.cs.cmu.edu/~tom/mlbook/NBayesLogReg.pdf

where $x_0, ..., x_d$ are the d inputs and $w_0, ..., w_d$ are the associated weights. Training is performed using the *Backpropagation* algorithm through these main steps:

- Propagate the each training sample through the network, applying the current weights, and compute the output value;
- Compute the prediction error, given by the difference between the output value computed at the previous step and the true value associated with the sample;
- Propagate the error backwards through the network, updating the set of weights. This process is repeated until a certain termination condition is met, i.e. until the network performs *acceptably well.*

A Support Vector Machine (SVM) is a classifier based on the idea of finding the optimal separating hyperplane: a hyperplane which correctly separates the instances of two classes and has the maximum margin to the respective nearest training samples [1]. SVM makes use of a linear model and its parameter, the weight vector, can be expressed in terms of a subset of the training set, i.e. the closest samples to the separating hyperplane, called *support vectors*. The training phase of a SVM consists of solving the optimization problem

$$\min_{w,b,\xi} \frac{1}{2} \parallel \mathbf{w} \parallel^2 + C \sum_t \xi^t \tag{2.6}$$

subject to

$$y_i(\mathbf{w}^T \phi(\mathbf{x}) + b) \geq 1 - \xi_i \tag{2.7}$$

where **w** is the weight vector, **x** are the samples in the training set, y is the class variable ($y \in \{-1, +1\}$), ξ is a slack variable, and C is a penalty factor associated with the regularization scheme. The transformation function ϕ is used if the training samples are not linearly separable. In this case, instead of trying to fit a nonlinear model, SVM maps the training samples in a higher (possibly infinite) dimensional space hoping that the samples mapped into the new space will be linearly separable.

Decision trees are hierarchical nonparametric models for supervised learning. A decision tree is composed of decision nodes, each implementing a test function, and terminal leaves. Given an input instance, a test function is applied at each node and depending on the outcome one of the branches is taken. Starting at the root this process is repeated until we reach a leaf node, and the corresponding leaf node contains the output classification. Therefore each leaf node defined a localized region in the input space, and the instances falling into this region have the same class labels. Every decision tree can be converted into a set of rules of the IF-THEN type, making the model easy to understand even for non experts in machine learning. The training phase of a decision tree consists of defining its structure and finding the test function for each node. At each node the tree effectively operates a split based on attribute values. The goodness of the split is quantified by an impurity measure. The measure most commonly used is *information gain*, i.e. the expected reduction in *entropy* when partitioning the data set according to the values of a considered attribute. For a k class problem the entropy of a data set S is defined as

$$Entropy(S) = -\sum_{i=1}^{k} p_i log_2 p_i \tag{2.8}$$

where p_i is given by the number of instances belonging to class i divided by the total number of instances. The information gain of an attribute A is defined as

$$InfoGain(S,A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v) \tag{2.9}$$

where S_v is the subset of samples in S having value v for attribute A. The tree is constructed top-down, with attributes having higher information gain placed in nodes towards the root of the tree. After learning the structure of the tree, *post-pruning* is applied in order to try enhance prediction performance. At the end of the training phase a decision tree can easily be converted into a set of rules for implementation.

Random Forest is an ensemble classifier consisting of a set of k tree-based classifiers $\{h(\Theta_1),...,h(\Theta_k)\}$, where Θ_i is the training set for the each of the decision trees h_i [2]. For each decision tree the training set Θ_i is chosen independently from those of the other classifiers, but with the same distribution. Additionally, when building each tree, at each node only a random subset of features is used. In the classification phase, when random forest receives an input sample, it will give such sample as input to each of the k decision trees and every tree will output a class label. The final class label will be decided by operating a majority vote among the k trees.

Chapter 3

Interoperability Analysis and Enhancement Feature Set

3.1 Data set

The data set used for this study was collected at West Virginia University. It consists of fingerprints for 500 users, acquired using four optical sensors (see Table 3.1) and one set of inked ten-print cards. All the sensors have the same resolution of 500 dpi and are FBI certified. Ten-print cards were scanned at a resolution of 500 dpi using an FBI certified flat-bed scan.

Device D0 is a livescan fingerprint sensor optimized for the acquisition of fingerprints in harsh environmental conditions. By using a proprietary silicone membrane technology this device is able to capture good quality fingerprints regardless of skin color and age. Device D1 employs an auto-calibration mechanism to ensure that every fingerprint captured complies with FBI standards.

	Manufacturer	**Model**	**Resolution (dpi)**	**Image size (pixels)**	**Capture area (mm)**
D0	Cross Match	Guardian R2	500	800 x 750	81 x 76
D1	i3	digID Mini	500	752 x 750	81 x 76
D2	L1 Identity Solutions	TouchPrint 5300	500	800 x 750	81 x 76
D3	Cross Match	Seek II	500	800 x 750	40.6 x 38.1
D4	Ten Print Scans	-	500	800 x 715	-

Table 3.1: Characteristics of the Live-scan devices used for fingerprint acquisition.

Device D2 adopts a *moisture discriminating optics* technology which is able to prevent a high moisture level from degrading image quality. Additionally, this device can capture fingerprints at a higher resolution of 1000 dpi. Device D3 is a mobile livescan sensor widely deployed in military applications. Because of this, the device has to accommodate for the acquisition of fingerprints under different illumination conditions. This device also adopts the silicone membrane technology used by device D0. D4 refers to inked fingerprint samples imprinted on ten print cards. This fingerprints are included in the study since many agencies still make an extensive use of them.

Some of the data pertaining to 6 users is missing, therefore the remaining 494 users were selected for this study. After providing biographical information, users provided two sets of fingerprints for each of the four optical sensors, followed by a set of inked fingerprints. Details regarding biographical information for people participating in this data collection are shown in Figure 3.1. The order in which the sensors were used to acquire fingerprints was the same for all the users. Inked fingerprints were acquired at the end in order to not affect the quality of live scans. Although each set of fingerprints consists of rolled individual prints for both hands, left slap, right slap, and thumbs slap, only the rolled print for the right index finger was used for this study.

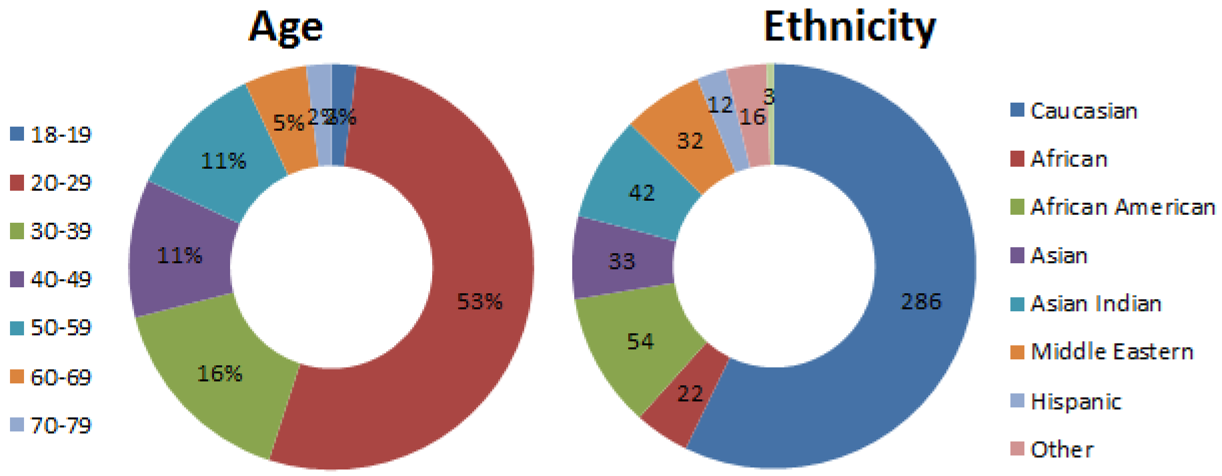

Figure 3.1: Information about age and ethnicity for the subjects participating in this study.

	DX-D0	DX-D1	DX-D2	DX-D3	DX-D4
D0	5.42 e-242	5.32 e-93	1.24 e-84	1.29 e-66	1.04 e-07
D1	2.72 e-68	6.19 e-242	2.99 e-65	2.35 e-59	2.59 e-06
D2	7.11 e-69	6.02 e-01	5.47 e-242	7.79 e-55	2.41 e-08
D3	2.14 e-76	6.28 e-01	5.62 e-01	5.47 e-242	3.03 e-08

Table 3.2: Statistical test on correlation of match scores.

3.2 Interoperability Assessment

As shown in Figures 1.2 and 1.3, the genuine match scores in cross-device matching are lower than those in intra-device matching. We performed a statistical test on genuine match scores in order to estimate the degree of change between intra- and inter-device matching. Table 3.2 shows the p-values from Kendall's rank correlation test. In this test the intra-device match scores are compared to match scores for which the gallery and probe fingerprints are acquired using two different devices. P-values close to zero indicate that the match scores in intra- and inter-device scenarios do not different significantly. Given the large difference in p-values from intra-device matching (i.e. the values on the diagonal) to cross-device matching, we can affirm that there is statistically significant difference in match scores when using two different devices for the acquisition of gallery and probe fingerprints.

Such variation of match scores ultimately reflects on the performance of the verification system. In Figure 3.2 we can see Detection Error Tradeoff (DET) curves for one of the matching algorithms used in this study. The DET curve for the inter-device matching scenario includes all the possible combinations of devices used in this study. It is clear that the performance of the system in cross-device matching configuration is much worse than those of the intra-device matching. The DET curves shown in Figure 3.2 refer to match scores extracted using the Identix matching algorithm from the BioEngine Software Development Kit. We only have one set of fingerprints for Ten Print cards, therefore there are no genuine intra-device match scores in this case. In the verification setup

Matching Scenarios	**Subjects**	**Devices**	**Match Scores**
Intra-device	494	4[a]	Gen: 1,976
	494	5	Imp: 120,855
Cross-device	494	5	Gen: 9,880
	494	5	Imp: 483,420

[a] In intra-device scenario, genuine scores for Ten Print cards are missing since we only have one set of ink-based prints.

Table 3.3: Matching scenarios table.

the number of genuine match score is

$$494 \times 5 \times 4 + 494 \times 4 = 11856 \tag{3.1}$$

The number of impostor score that could be extracted is over 6 million. In order to reduce the number we divided the users in groups of 100 (with the last group having 94 users) and computed impostor match scores only within each group, resulting in 604275 match scores. Details about different scenarios for genuine and impostor match scores are provided in Table 3.3.

We used two additional matching algorithms in this study: Bio-Key WebKey [1] and NIST Bozorth. In the case of WebKey 25% of the impostor match scores is missing due to license issues. Although the license expired before we could complete the extraction of impostor match scores, for the purpose of this study we have a sufficient number of them. Although there is a difference in performance between the three algorithms the trends observed are similar, with inter-device matching having the lowest performance in all cases, as shown in figures 3.3 and 3.4.

3.3 Interoperability Enhancement Feature Set

In order to enhance matching performance in cross-device matching, we exploit additional features which capture the variations in fingerprint images due to the specific acquisition devices.

[1] http://www.bio-key.com/products/overview-2/web-key

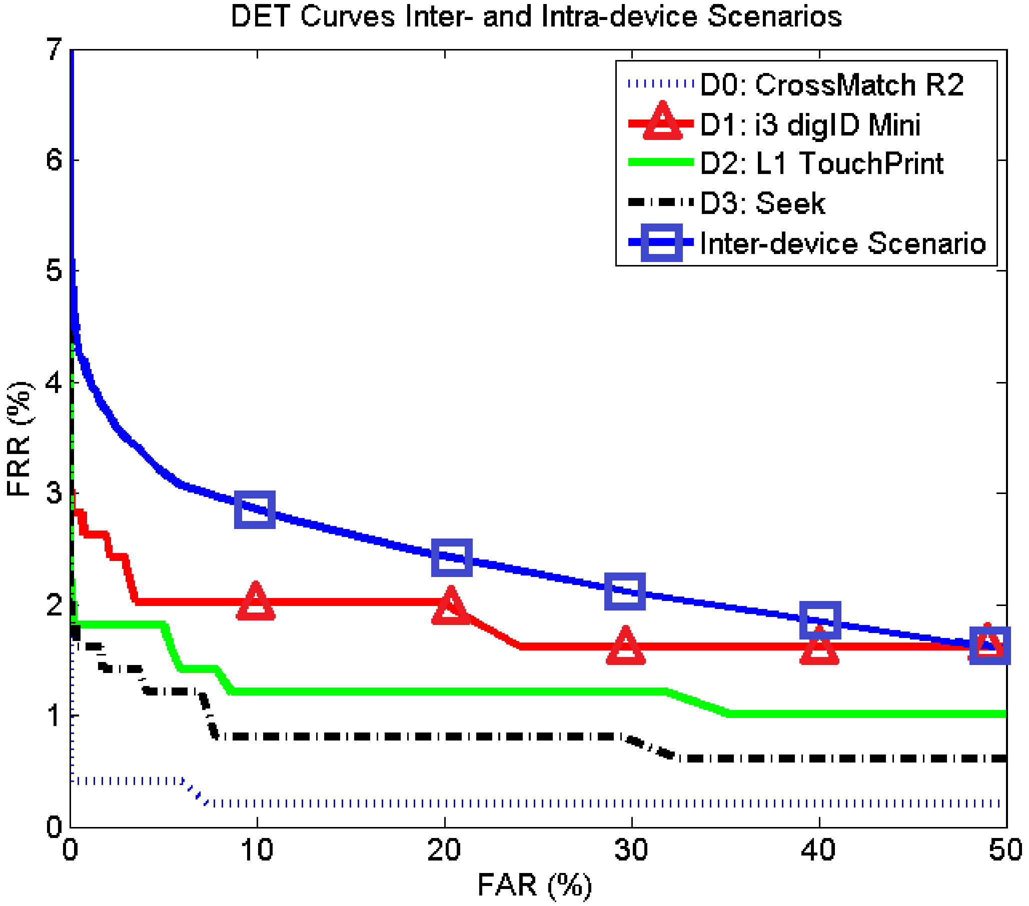

Figure 3.2: DET curves for the various intra- and inter-device matching scenarios. As we can see, the decrease in genuine match scores in the itra-device scenario reflects on the accuracy of the fingerprint system. In fact, the DET curve for the cross-device scenario is considerably higher than those for intra-device matching.

By analyzing features of different nature (i.e. image quality, texture), from different domains (e.g. spatial and frequency domains), and at different levels (i.e. local and global) we complement the match score with additional information to improve matching performance.

Image Quality is an indicator of the degree of usefulness of a biometric sample. The matching process is heavily affected by the quality of the biometric samples on which it is performed. A sample is of good quality if it is suitable for automated matching. Grother and Tabassi investigated a method for predicting the performance of a biometric system based on the quality of biometric samples [7]. The quality measure should not depend on human perception of quality for several

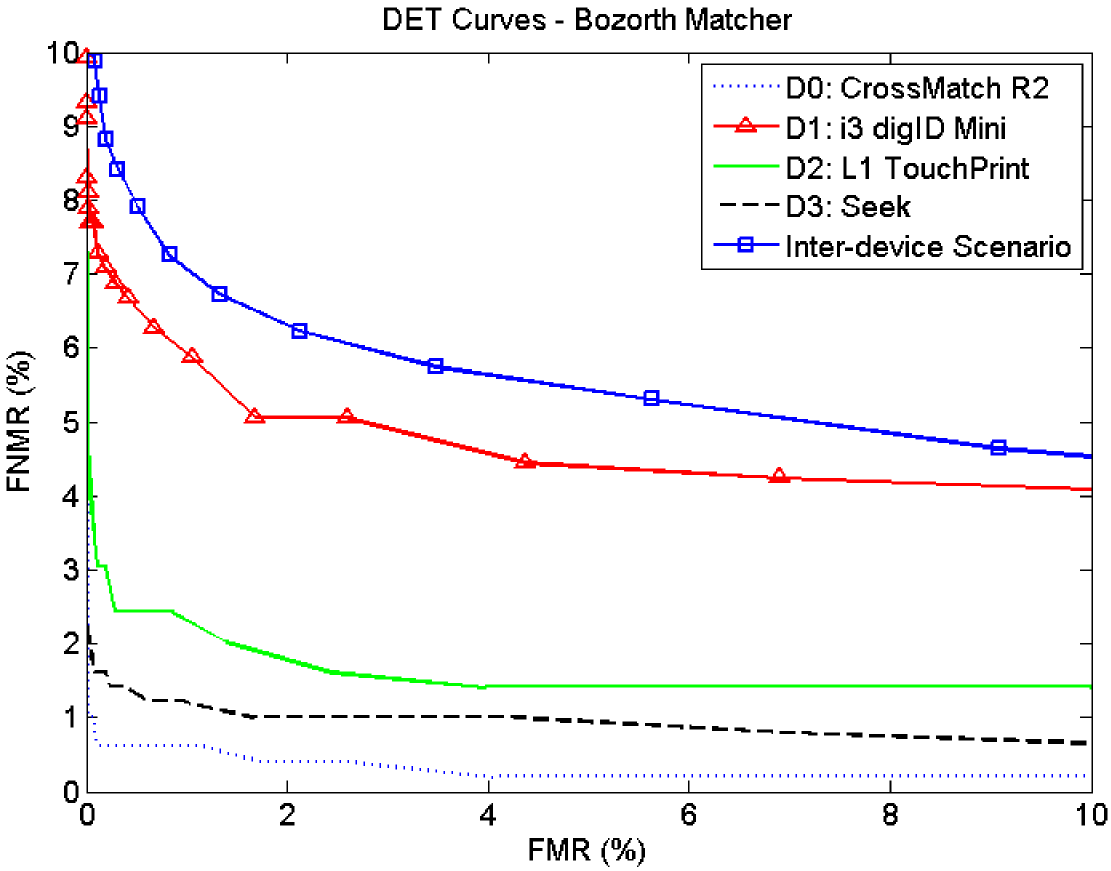

Figure 3.3: DET curves for the various intra- and inter-device matching scenarios when using the Bozorth matching algorithm.

reasons:

- Human perception of quality is very subjective. While a person might consider a fingerprint as being of ”high quality” others might disagree.

- The matching process is performed by an automated system so the human perception of quality might not agree with the matching system. Consider the case where an observer sees a fingerprint image with a considerable amount of blur and a relatively low contrast; the observer might reasonably say that the sample is of low quality, however if a matching system is able to extract enough minutiae from the image then it would perform well.

- An observer might not know how the matching process is performed, e.g. it would be wrong to assign an image a low quality score because it contains a low number of minutiae if the

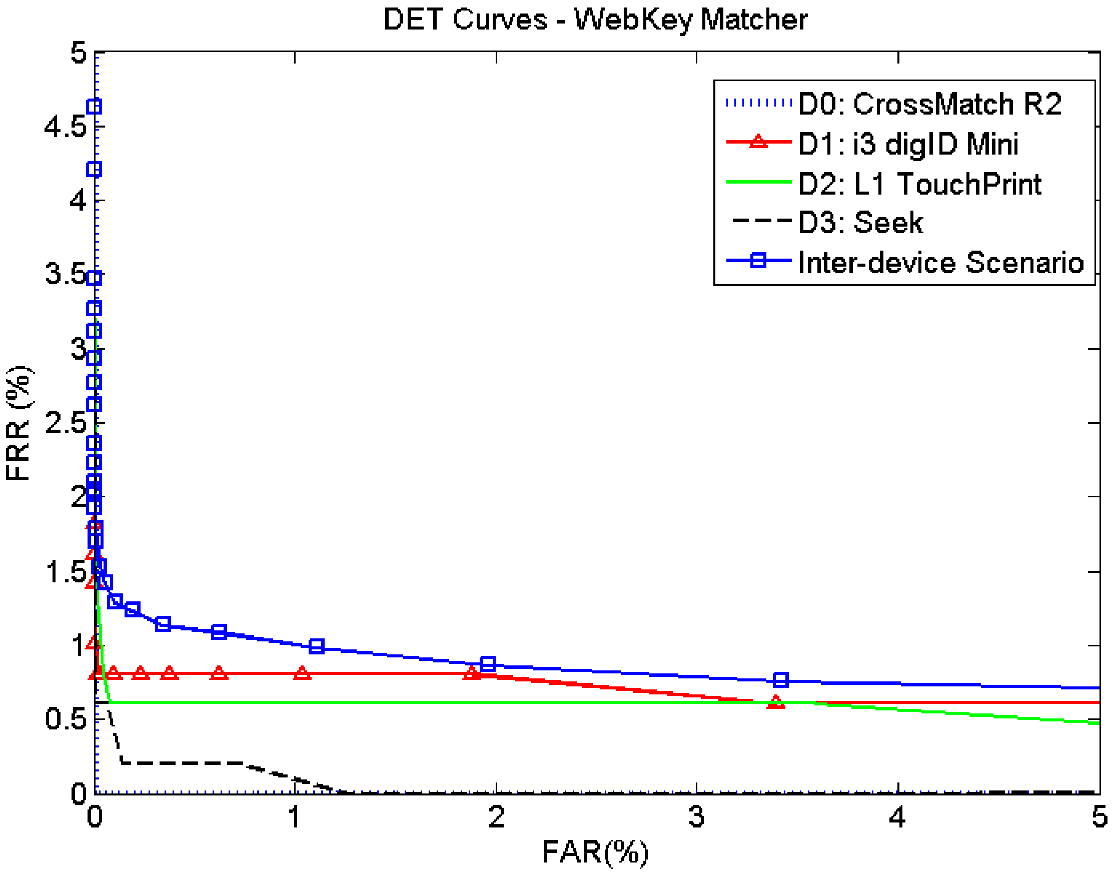

Figure 3.4: DET curves for the various intra- and inter-device matching scenarios when using the WebKey matching algorithm.

matcher is minutiae-based.

For this work, fingerprint image quality was assessed using two algorithms:

- NIST Fingerprint Image Quality algorithm (NFIQ), an open source tool in the NIST Biometric Image Software (NBIS) distribution [2] that has become the industry standard for fingerprint image quality assessment. The NFIQ quality score is an integer number in the range [1,5], where 1 is given to a fingerprint with highest quality and 5 represents fingerprints of the lowest quality. NFIQ predicts the impact that the fingerprint image will have on the system in terms of error rates.

[2] http://www.nist.gov/itl/iad/ig/nbis.cfm

- MITRE IQF [3] uses information derived from the power spectrum of the image. The process is carried out by analyzing the image in overlapping blocks and applying several normalizations based on total power of the image and on contrast. Additionally a Human Visual System (HSV) filter is applied. The output of this algorithm is an integer number in the range [1,99], where 99 is given to the images with highest quality. Unlike NFIQ, MITRE IQF gives an assessment of the visual quality of the fingerprint image.

Figure 3.5 shows the quality distribution, for each device, given by NFIQ quality scores, while Figure 3.6 the distributions for Mitre quality scores.

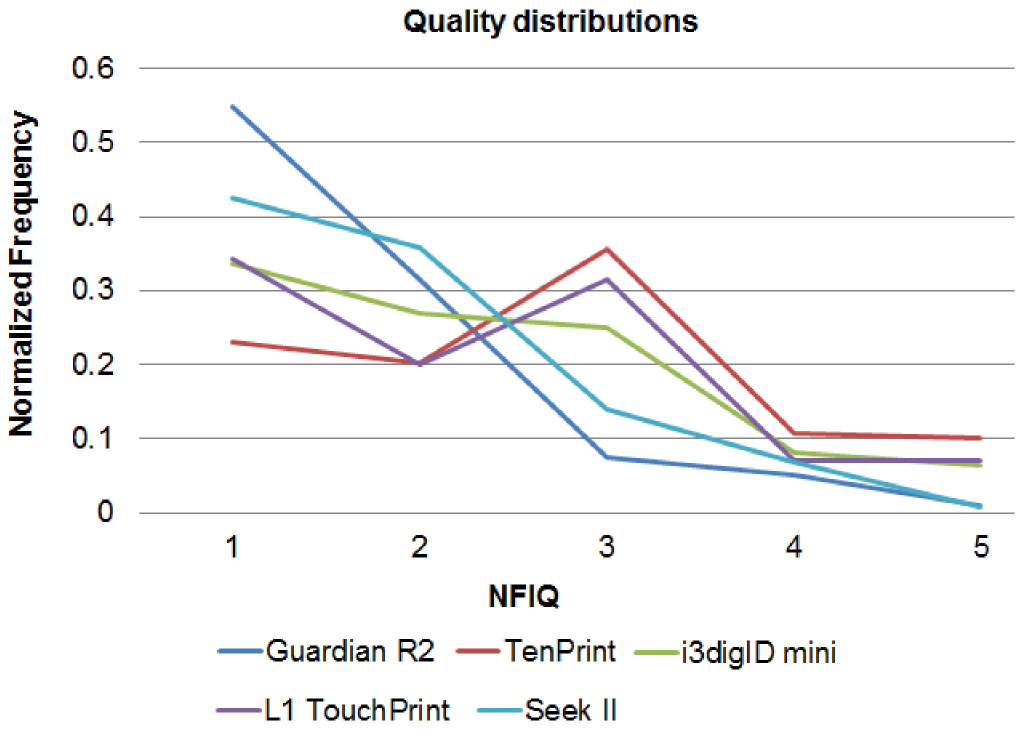

Figure 3.5: NFIQ quality distributions for the different devices used in this study.

We can see from the two figures that there is little correlation between the two quality measures, e.g. according to NFIQ quality scores device D0 has the highest quality, while according to Mitre scores device D3 shows the highest quality.

Figure 3.7 (a) shows the frequency of low genuine match scores, which represent challenging

[3] http://www2.mitre.org/tech/mtf/

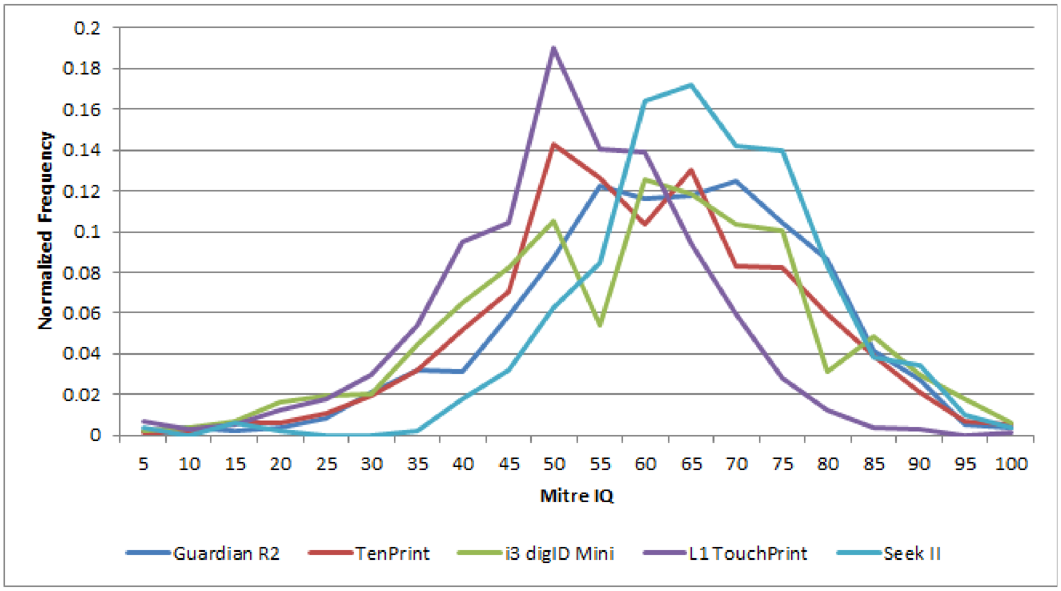

Figure 3.6: Mitre quality distributions for the different devices used in this study.

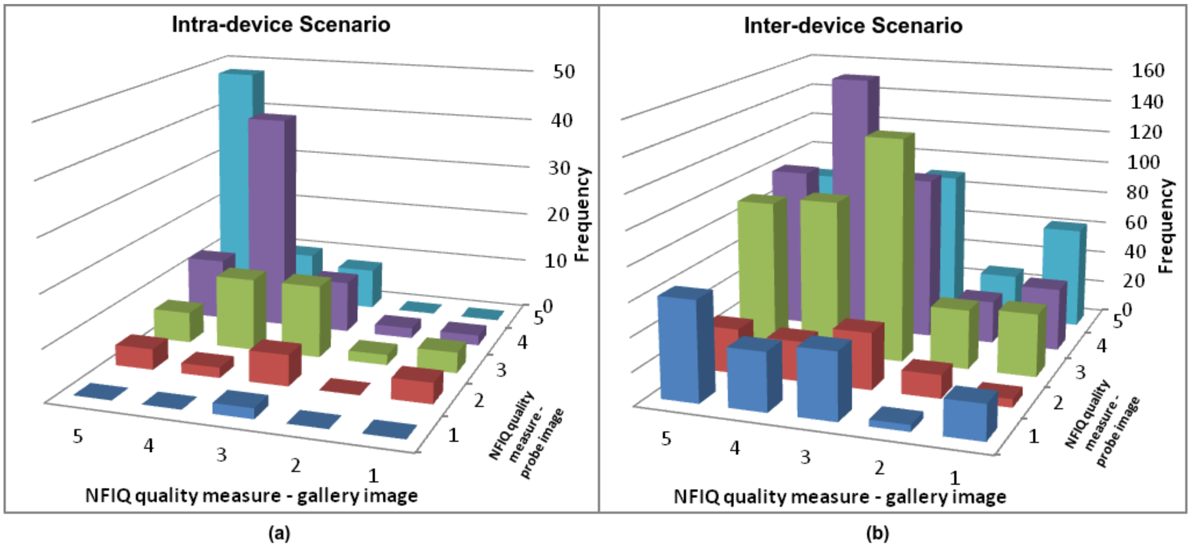

Figure 3.7: Frequency of low genuine match scores for different fingerprint quality combinations in (a) intra-device and (b) cross-device matching scenarios.

matches, in intra-device matching scenario. We considered a genuine match score low if its value is less than 10. In this case, there is a very small number of low match scores, as long as one of the two fingerprints involved in the matching is of high quality (NFIQ $<$ 4). NIST provides recommendations on quality control for the fingerprint acquisition process: fingerprints should be reacquired up to three times if the NFIQ score of thumbs and index fingers is greater than 3 [23].

While this recommendation works when using a single fingerprint scanner, it falls short when multiple devices are used in a fingerprint recognition system. Figure 3.7 (b) shows, in fact, that the number of questionable matches greatly increases in cross-device matching. In this scenario a more stringent quality control policy needs to be in place: both gallery and probe fingerprint images should have NFIQ quality score 1 or 2 in order to provide reliable match decisions.

Minutiae Count represents the number of minutiae in a fingerprint image. A minutiae point is a feature of fingerprints which typically consists of a ridge ending or a ridge bifurcation. A minutiae is typically represented by a triplet (x,y,r), where (x,y) is the location of the minutiae pixel in the image and r is the direction of the minutiae. Even if a fingerprint might appear to have low quality from a human perspective, if a sufficient number of minutiae can be extracted from it then a minutiae-based matcher might be able to match it effectively [5]. The number of minutiae extracted from a fingerprint image depends, among other factors, on human-sensor interaction [10]. In Figure 3.8 we can see how the minutiae count varies for different devices.

The MINDTCT program from the NBIS distribution was used for extracting the number of minutiae contained in each fingerprint image.

Image Gradient measures the rate of change in grey levels throughout the image. While the direction of the gradient tells us the direction in which the grey level is changing most rapidly, the magnitude of the gradient quantifies the amount of such change. The image gradient is computed as follows:

$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} \tag{3.2}$$

where G_x corresponds to $\frac{\partial f}{\partial x}$, the differences in x (horizontal) direction. G_y corresponds to $\frac{\partial f}{\partial y}$, the differences in y (vertical) direction [6]. Computing the gradient results in two matrices, a matrix for G_x and another for G_y, each with the same size as the original image. The magnitude of the image gradient is given by

$$\nabla \mathbf{f} = [G_x^2 + G_x^2]^{1/2} \tag{3.3}$$

and we use the average of the gradient magnitude in the final model.

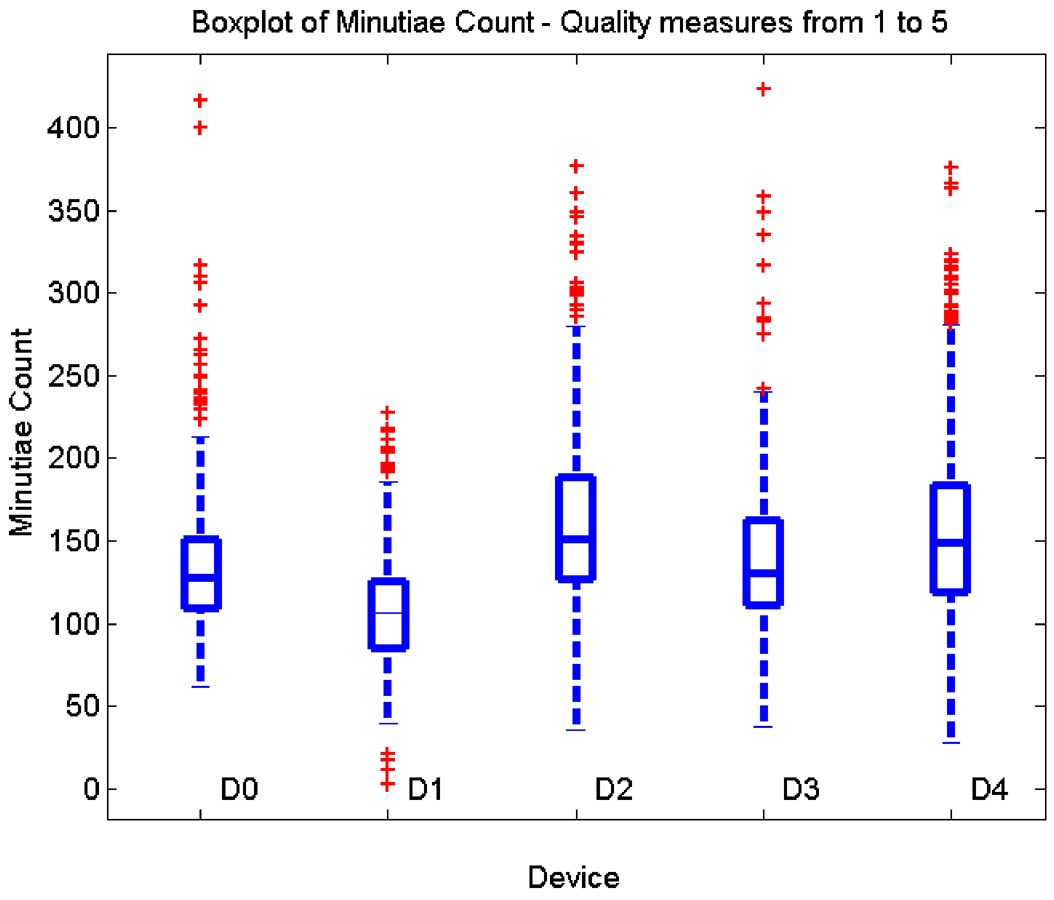

Figure 3.8: Box plots of minutiae count for the different devices.

The distributions of mean image gradients for the 5 devices are depicted in figure 3.9. Device D0 has the overall highest gradients, which translates in a very clear transition from fingerprint ridges to valleys and vice versa.

Coherence of Direction measures the extent to which gradients point in the same direction. This feature is a quality index which analyzes the image in the spatial domain in non-overlapping blocks of size 32 × 32 pixels, and for each block it computes how much the gradients are aligned [4] [11]. In regions with distinct ridge-valley orientation the coherence 1, while for low quality images the coherence is 0.

Grey Level Statistics are computed in the spatial domain, after segmenting the fingerprint image. For each image $I(x,y)$ two statistical measures were used:

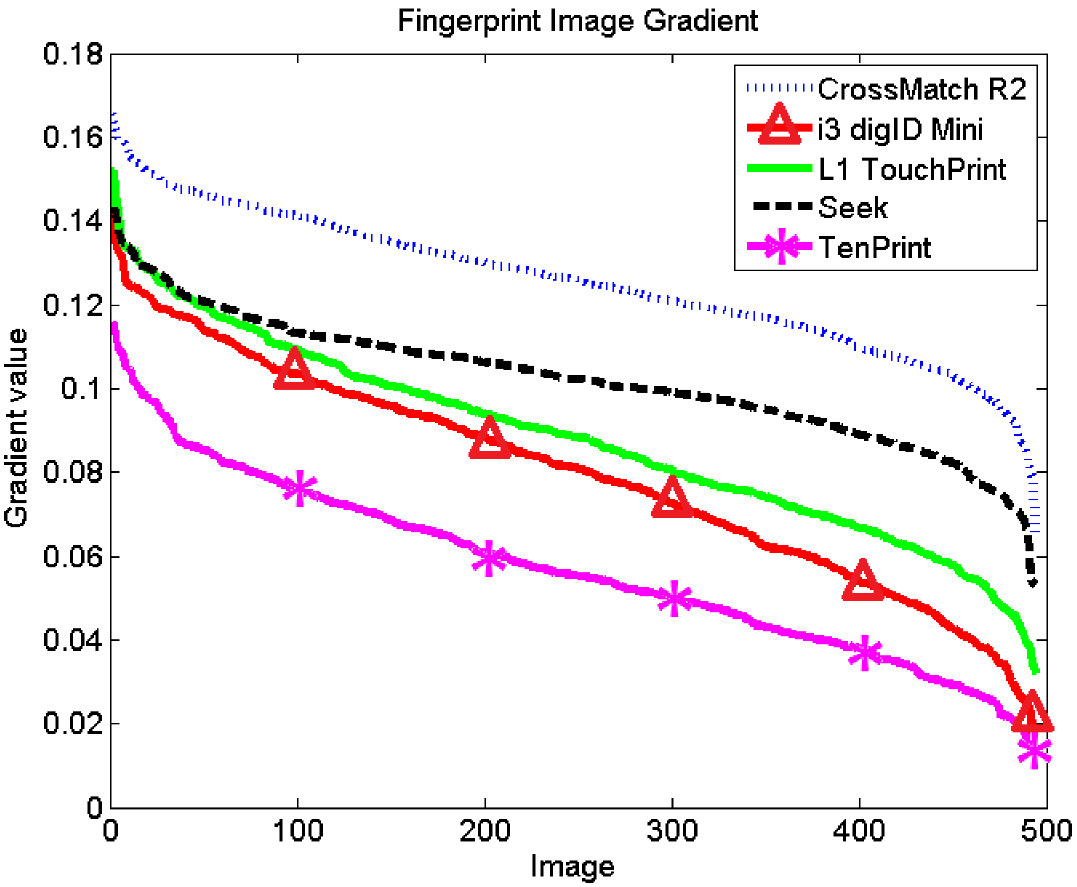

Figure 3.9: Distributions of image gradients for the five sensors.

- mean, computed as follows

$$\mu = \frac{1}{M \times N} \sum_{x=1}^{M} \sum_{y=1}^{N} I(x,y) \tag{3.4}$$

- standard deviation, computed as follows

$$\sigma = \sqrt{\frac{1}{M \times N} \sum_{x=1}^{M} \sum_{y=1}^{N} (I(x,y) - \mu)^2} \tag{3.5}$$

We can see from Figure 3.10 that the two devices from manufacturer Cross Match have significantly lower mean in grey levels compared to other devices. This results the in ridges being darker in fingerprint images acquired using Cross Match devices.

Figure 3.11 shows that Ten Print cards have significantly higher standard deviation in grey level

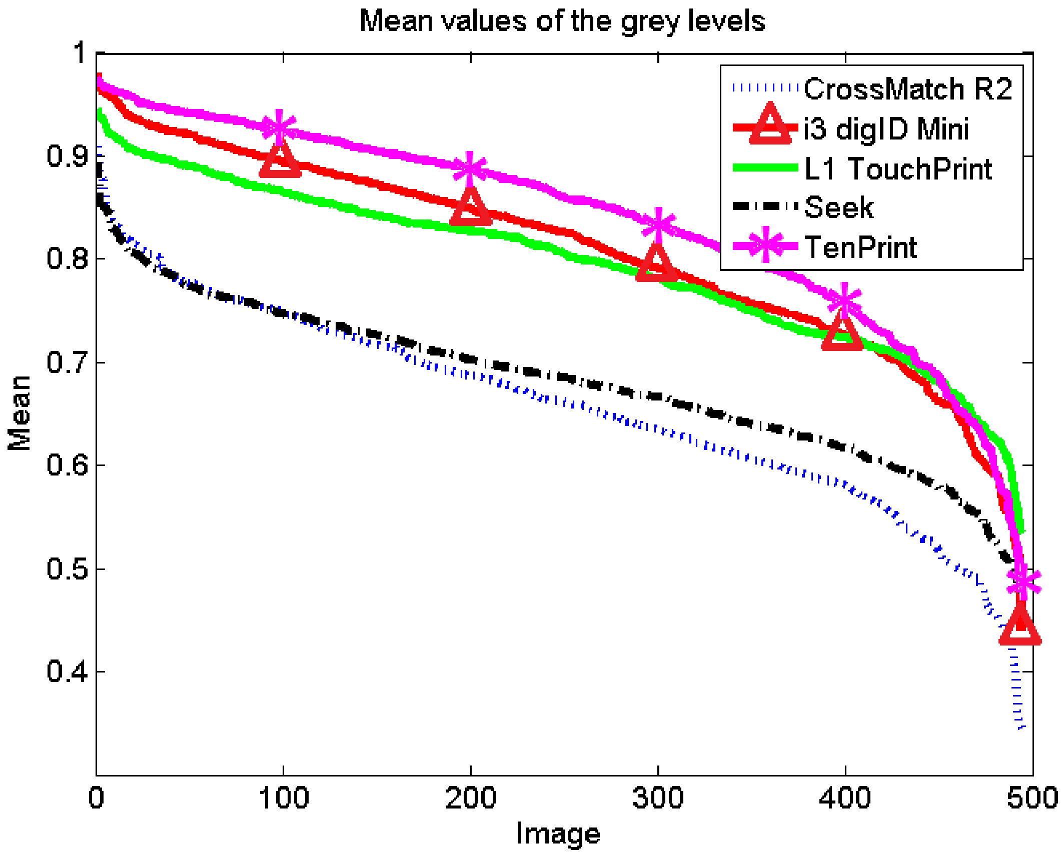

Figure 3.10: Grey level mean distributions for the different sensors.

than fingerprints acquired using optical devices.

Pattern Noise. During the acquisition of a fingerprint noise can be introduced. The main two components for such noise are a random component called *photonic noise*, and a deterministic component called *pattern noise*. The dominant part of the pattern noise is the Photo-Response Non-Uniformity noise (PRNU), which is caused by pixels having different sensitivity to light. Some of the factors that contribute to the PRNU are the refraction of light on optical surfaces and zoom settings. We first compute the approximated reference PRNU pattern for the sensor; then, for each image, we compute the correlation between the reference pattern and the noise pattern extracted from the image:

$$\rho = corr(\mathbf{n},\mathbf{r}) = \frac{(\mathbf{n}-\mu_n)(\mathbf{r}-\mu_r)}{\| \mathbf{n}-\mu_n \| \| \mathbf{r}-\mu_r \|} \qquad (3.6)$$

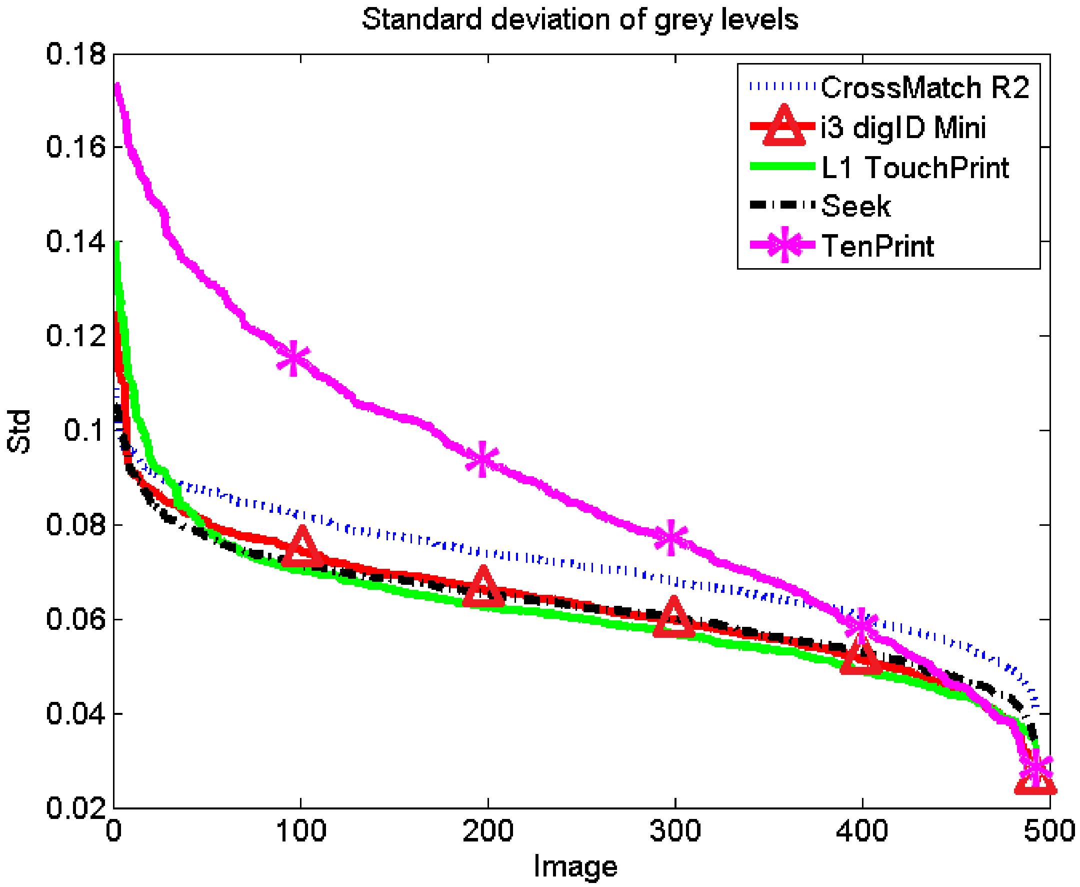

Figure 3.11: Grey level standard deviation distributions for the different sensors.

where **n** is the residual noise of the image and **r** is the reference PRNU pattern, computed as average of the residual PRNU of all the images for each sensor. Figure 3.12 shows the distributions of PRNU for different sensors.

Energy Concentration in a fingerprint is an indicator of the overall quality of the image. It has been observed that fingerprint images with energy concentrated in fewer bands have higher quality, while fingerprints with energy distributed across a vast spectrum exhibit lower quality [4].

Given an image $I(x,y)$, the Discrete Fourier Transform (DFT) of size $M \times N$ can be computed as follows:

$$F(u,v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} I(x,y)\; e^{-i2\pi(\frac{ux}{M}+\frac{vy}{N})} \tag{3.7}$$

where $i = \sqrt{-1}$, x and y are spatial variables, $I(x,y)$ represents the grey-level intensity value at

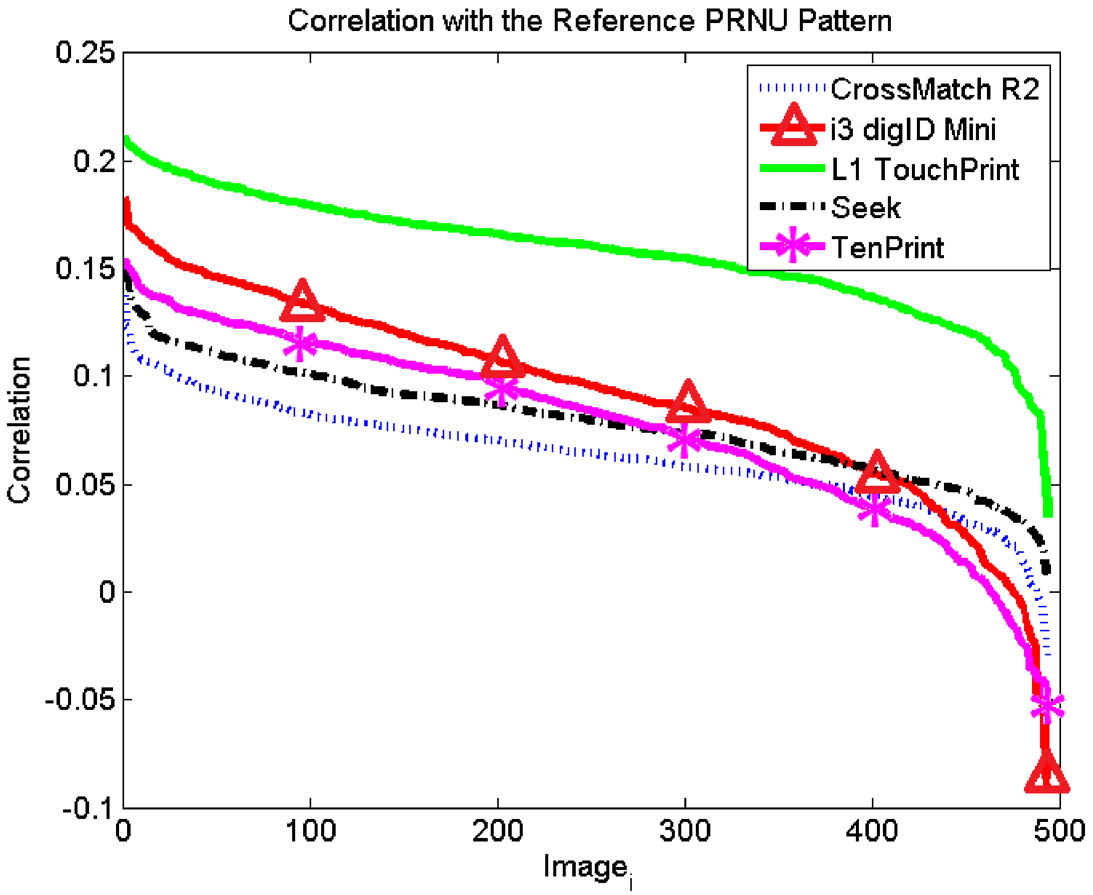

Figure 3.12: PRNU distributions for the different sensors.

pixel (x,y) of the image, u and v are frequency variables [6]. The relationship between the spatial domain and the Fourier domain is the following: higher frequencies represent patterns in the image corresponding to higher variations in the grey levels.

We established the Region of Interest (ROI) to be the interval between 50 and 250 Hz. In order to have a fine-grained distribution of energy for each fingerprint, we decided to analyze 40 equally-spaced bands of 5 Hz each in the ROI. The analysis on each band was carried out using a bandpass filter for each specific band. Each bandpass filter was constructed by computing the difference between two consecutive Butterworth low-pass filters. After applying the bandpass filter to the image, the energy concentration for a specific band was computed as the squared magnitude of the resulting spectrum. We can see the distributions of energy across sensors for a specific band in Figure 3.13 Once extracted, the set of 40 ring features $P = p_1,...,p_{40}$, we additionally computed

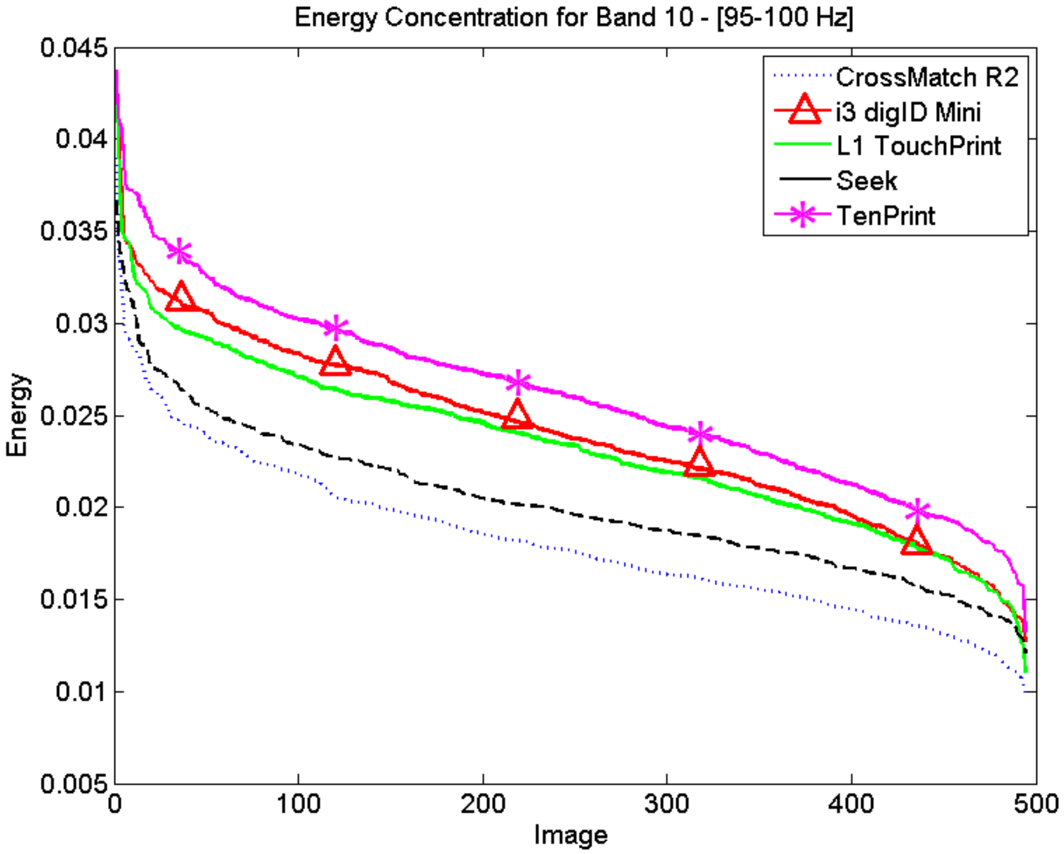

Figure 3.13: Energy distributions for the different devices. The energy in this plot belongs to the [95-100] Hz band.

the entropy of the energy distribution, as follows:

$$Entropy(I) = -\sum_{i=1}^{40} p_i log_{10} p_i \tag{3.8}$$

We can see an example of two fingerprints of different quality in Figure 3.14. In the first fingerprint the ridges are clearly defined and the image presents good contrast. Its energy distribution is mostly concentrated in bands 14-22. The fingerprint on the right has lower quality, poor contrast, and we can see that the energy is distributed across a large frequency spectrum.

The distributions for the entropy across devices are shown in Figure 3.15.

In our experiments, instead of using a set features for the energy concentration of each of the two fingerprints involved in matching, we considered the difference between the energy concentra-

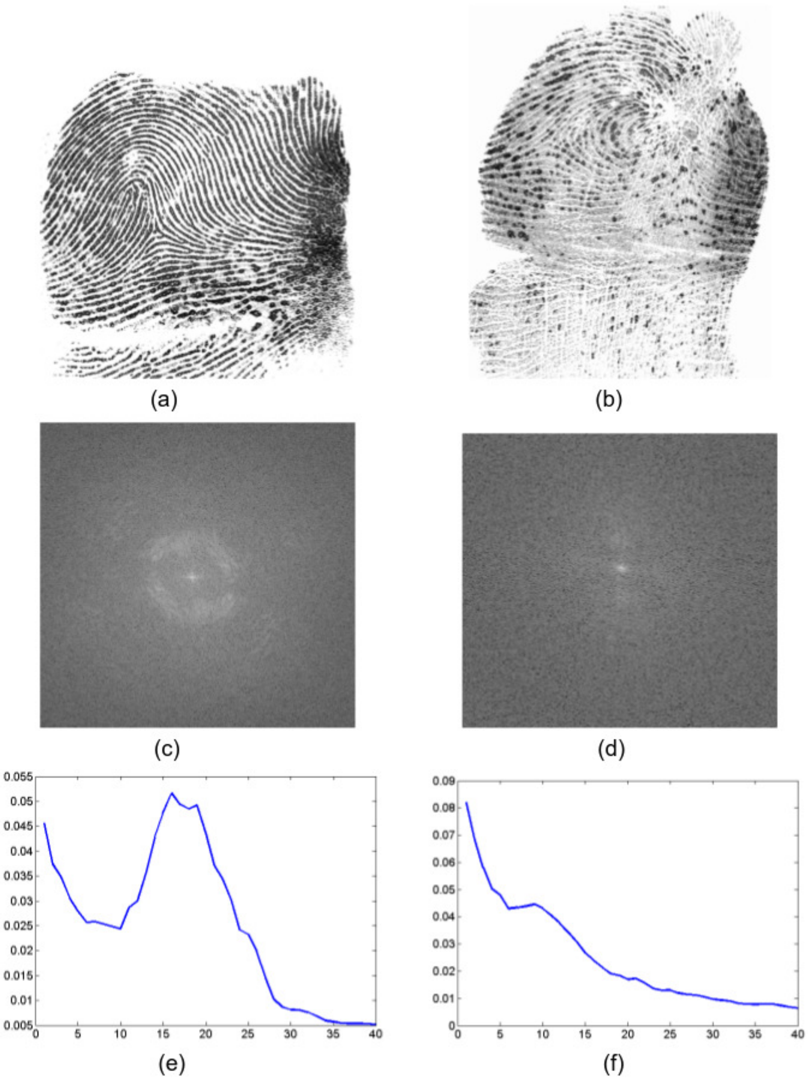

Figure 3.14: (a) a high quality fingerprint; (b) a low quality fingerprint; (c) and (d) show the corresponding Fourier transform, while (e) and (f) show the corresponding energy distributions. The two fingerprints belong to different individuals, and we can see how the different quality impacts the energy distribution.

tion of the gallery and that of the probe fingerprint. The difference in energy concentration between gallery and probe fingerprint proved to have more discriminative power than the two individual set of features. We observed that for each fingerprint, energy values for neighboring bands are highly correlated. This suggested that we might not need all the 40 ring features. We carried out further analysis based on two main aspects:

- the discriminative power of each energy band, as measured by Information Gain;

- the correlation of ring features, as measured by the correlation matrix;

Based on the fact that most of the ring feature i are highly correlated with the set of ring features $(i-5, i+5)$, we decided that 4 ring features are enough to capture the information needed. The goal was to choose the most discriminative and uncorrelated among the ring features. As result, bands number 10, 16, 25, and 40 were selected.

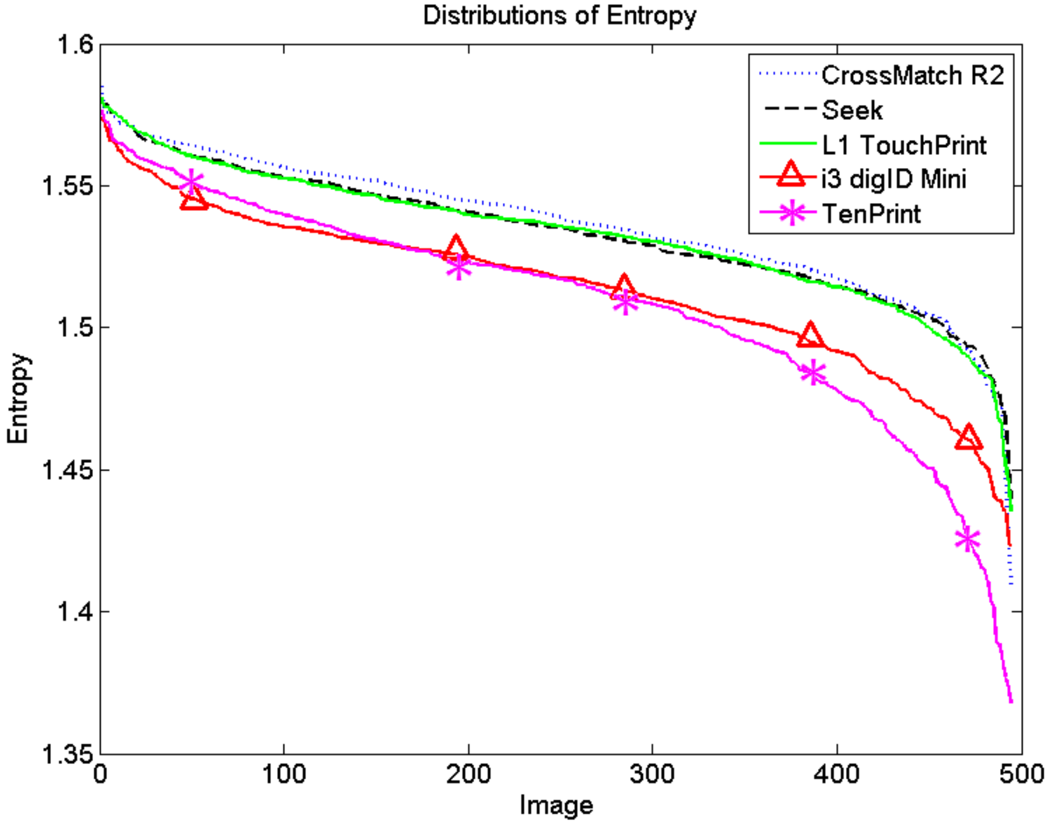

Figure 3.15: Entropy distributions for the different devices.

Local Binary Pattern (LBP) is a grey-scale invariant texture descriptor. LBP provides information about the spatial structure of texture by considering a neighborhood for each pixel [15]. The texture T, for each pixel, is defined as joint distribution of grey levels of the neighborhood consisting of P pixels ($P > 1$):

$$T = t(g_c, g_0, ..., g_{p-1}) \tag{3.9}$$

In our study we used the original LBP operator, $LBP_{P,R}$ with $P = 8$ and $R = 1$ [16] [12]. The neighborhood for each pixel is therefore composed of its 8 adjacent pixels, i.e. the 3×3 neighborhood. The next step consists of binarizing the P pixels by thresholding with respect to the central pixel g_c. Each of the thresholded values is multiplied by a weight factor, which is a power of 2, and all of the values are summed to obtain the LBP value for the considered pixel:

$$LBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c)2^p \tag{3.10}$$

where

$$s(x) = \begin{cases} 1, & x \geq 1 \\ 0, & x < 0 \end{cases} \tag{3.11}$$

Each LBP value is in the range 0-255. After applying the LBP operator to the whole fingerprint image, a frequency histogram composed of 256 bins is computed.

Given that each match considers features belonging to two fingerprints using the whole LBP histogram would add 512 features, which might be counterproductive. For the purpose of this study the features need to capture differences between the different fingerprint sensors, therefore we tried to summarize the LBP histograms using some statistical measures. Among the measures considered, we chose the maximum and the standard deviation computed from the 256-bin histogram. We can see from Figures 3.16 and 3.17 that there is a separation between the distributions of the devices used in this study.

Local Phase Quantization (LPQ) is a descriptor for texture classification which is robust to image blurring [17]. LPQ is based on the Fourier Transform being invariant to centrally symmetric blur [20] [18]. For each pixel the 2D Short-Term Fourier Transform (STFT) is computed in a $M \times M$ neighborhood N_x, as follows:

$$F(\mathbf{x},\mathbf{u}) = \sum_{\mathbf{y} \in N_x} f(x-y)e^{-j2\pi\mathbf{u}^T\mathbf{y}} \tag{3.12}$$

where $\mathbf{x}$ is the pair of coordinates (x,y) in the spatial domain and $\mathbf{u}$ is the pair of coordinates (u,v)

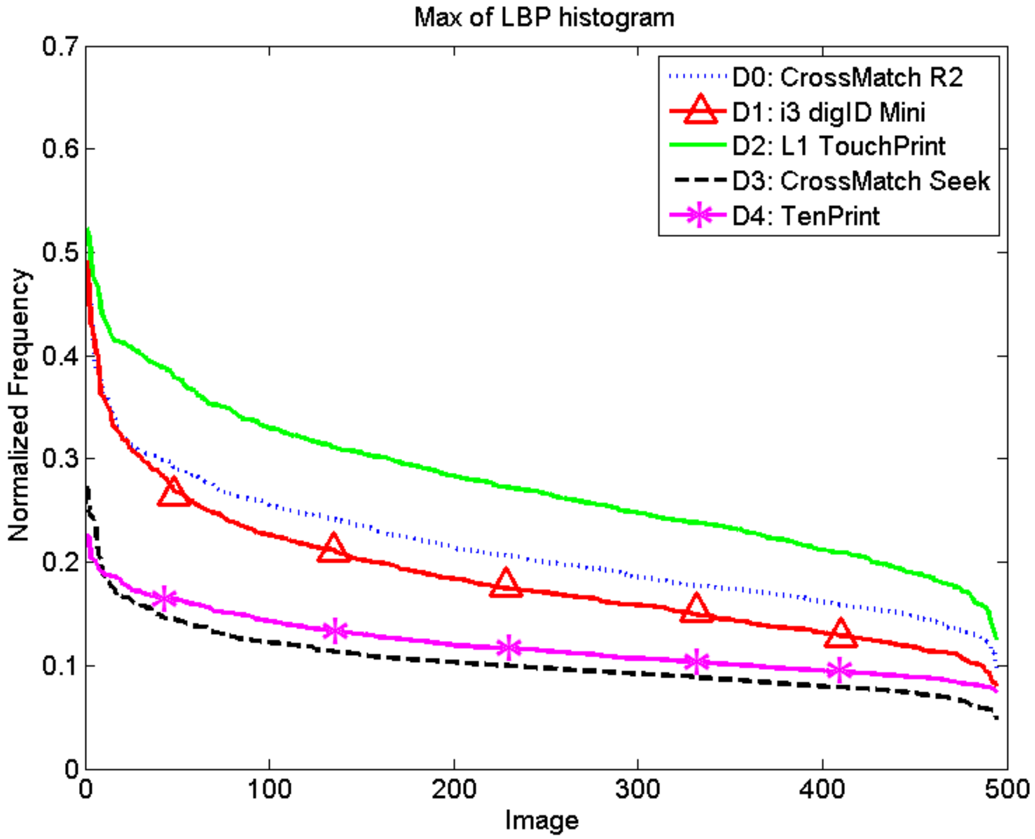

Figure 3.16: Maximum of LBP histograms for the different sensors.

in the frequency domain. From the frequency domain only four complex coefficients are selected at specific frequencies. After applying a whitening transform, binarization and quantization are applied in a similar fashion to that of LBP to form 8 bit features. The final step of LPQ involves computing the 256 bin histogram of the feature distribution.

As for LBP, we summarized the LPQ histogram for each fingerprint by computing the maximum and the standard deviation. Figures 3.18 and 3.19 show the distributions of these two measures across devices.

Alignment is the process through which two fingerprints are geometrically transformed into the same coordinate system [22]. This step is necessary since the location and orientation of minutiae points depend on the placement of the finger on the sensing device. Alignment is generally carried out using a linear transformation (i.e. translation and rotation) given by the Generalized Hough

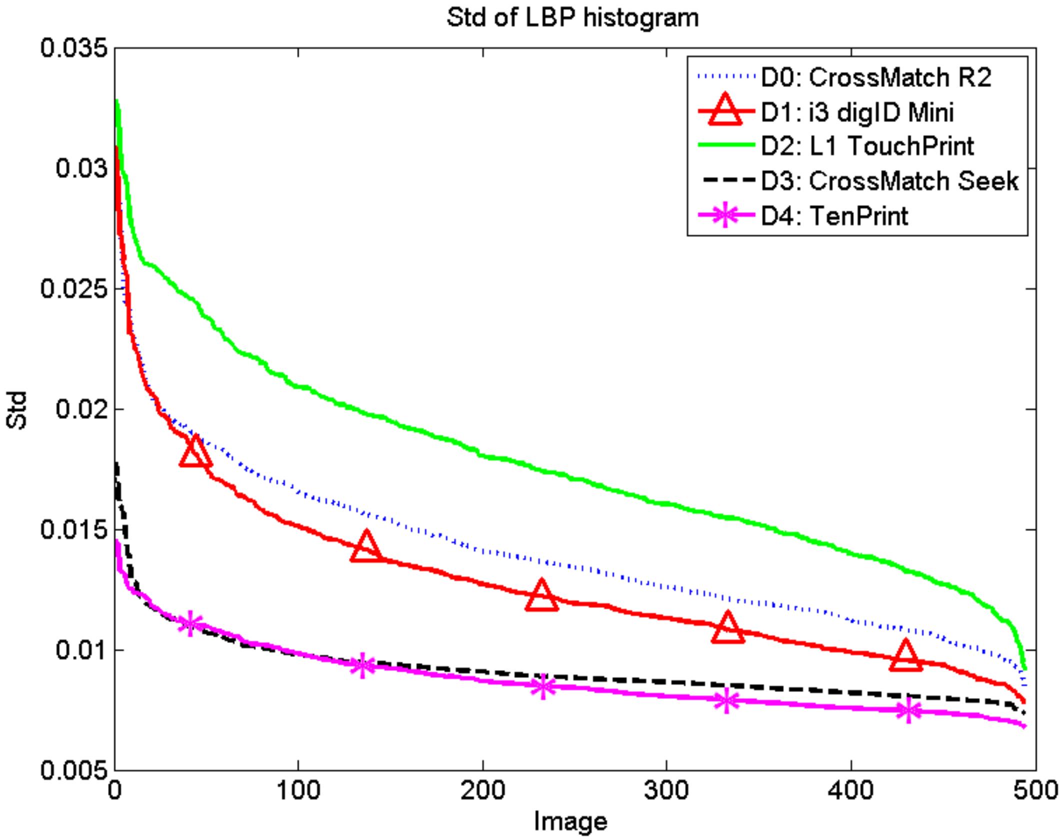

Figure 3.17: Standard deviation of LBP histograms for the different sensors.

transform, described in Algorithm 1.

A linear transformation is sufficient to align two fingerprints acquired using the same device. However, differences in capture area, resolution, and image size can introduce non-linear deformations using different devices for the gallery and probe fingerprints.

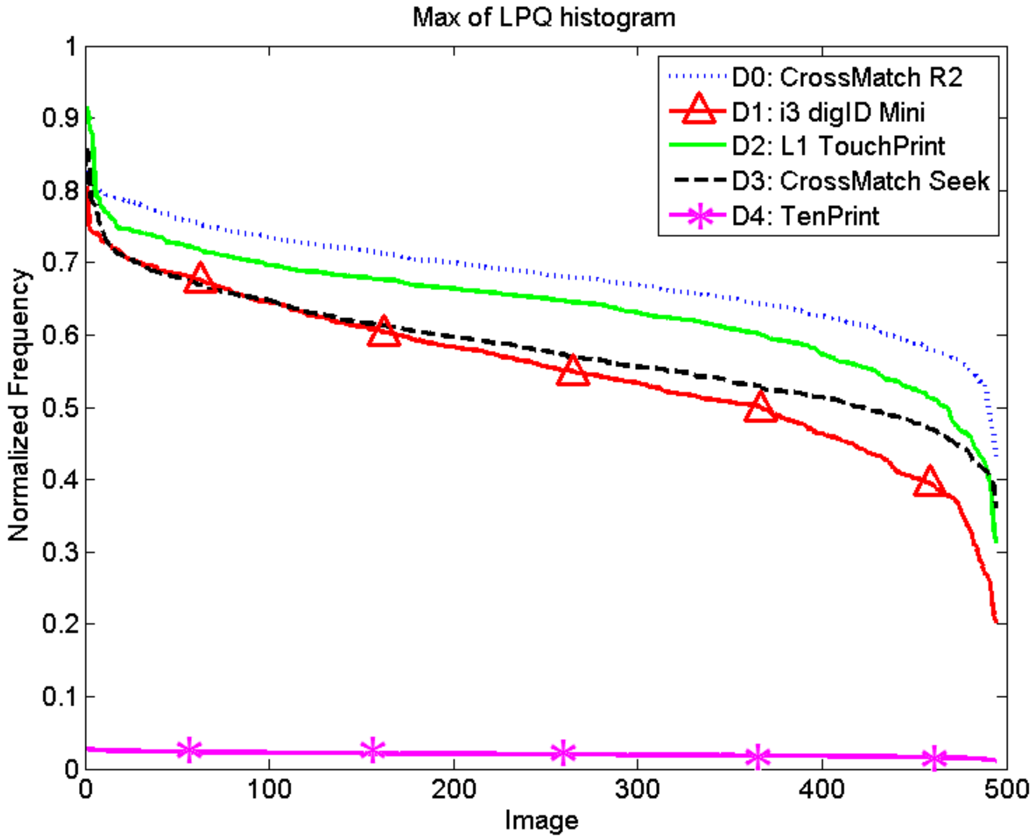

Figure 3.18: Maximum of LPQ histograms for the different sensors.

Input: Two minutiae sets
$m^g = (x_i^g, y_i^g, \Theta_i^g)_{i=1}^M$, and $m^p = (x_j^p, y_j^p, \Theta_j^p)_{j=1}^N$
Output: Transformation parameters Δx, Δy, ΔΘ.

```
for i = 1 to M do
    for j = 1 to N do
        $\Delta\Theta = \Theta_i^g - \Theta_j^p$
        $\Delta x = x_i^g - x_j^p cos(\Delta\Theta) - y_j^p sin(\Delta\Theta)$
        $\Delta y = y_i^g + x_j^p sin(\Delta\Theta) - y_j^p cos(\Delta\Theta)$
        $A[\Delta\Theta][\Delta x][\Delta y] = A[\Delta\Theta][\Delta x][\Delta y] + 1$
    end for
end for
return location of peak in A
```

Algorithm 1 Pseudo-code for the Generalized Hough transform for determining rotation parameters.

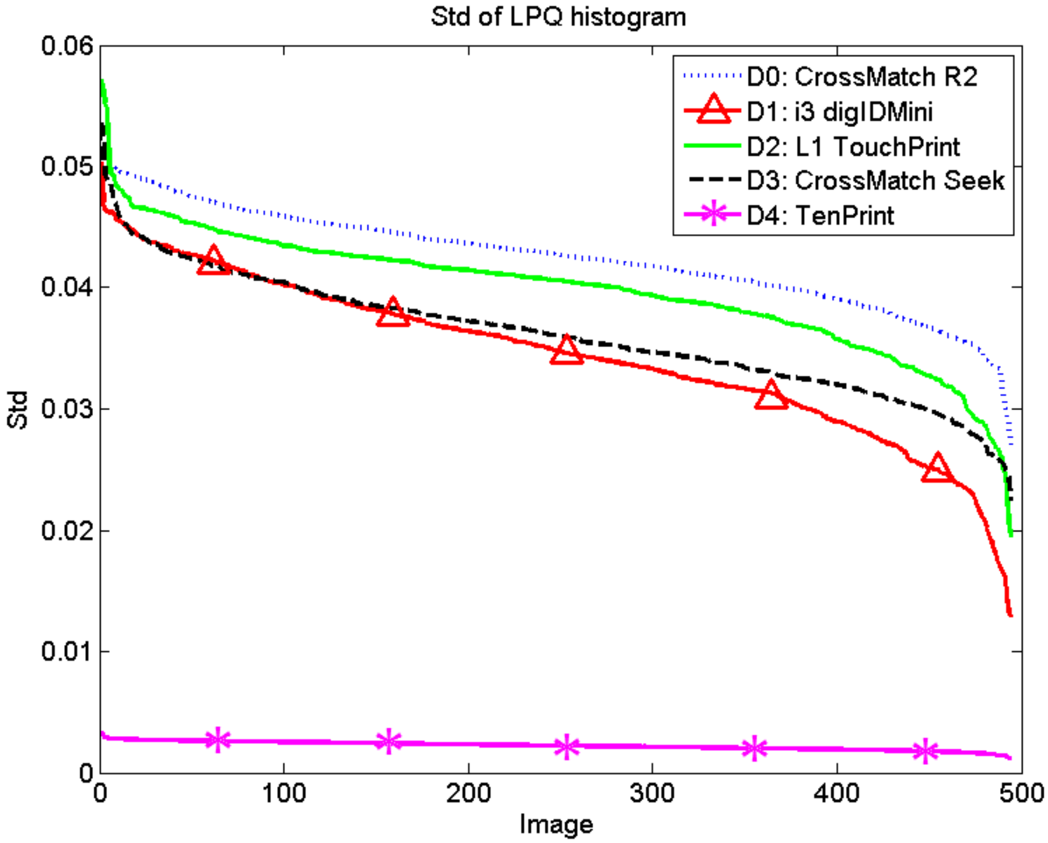

Figure 3.19: Standard deviation of LPQ histograms for the different sensors.

Chapter 4

Integration of Features

Once all the features defined in Chapter 3 are extracted we have multiple scores at our disposal for each match, therefore we need an efficient way of combining them in order to decide if a match should be declared genuine or impostor. One possible way is to develop a heuristic based on the range of features values and implement a set of ad-hoc rules accordingly. One of the main problems with this approach is that defining effective rules becomes extremely difficult given the number of features we have to deal with. We decided to adopt Machine Learning algorithms to efficiently combine all the features and make a final decision on the nature of the match. The feature vector is composed of:

- Features for a single fingerprint (a set for the gallery image and a set for the probe): NFIQ, Mitre IQ, minutiae count, image gradient, coherence of direction, grey level statistics, PRNU, LBP statistics, LPQ statistics;
- Features for a pair of fingerprint: match score, alignment parameters, energy concentration.

Figure 4.1 shows the building and deployment phases of the system. The interoperability enhancement is performed in parallel to the typical biometric system.

In order to evaluate our proposed approach with different fingerprint algorithms we built a data set for each matcher separately, given that most of the fingerprint recognition system use a single

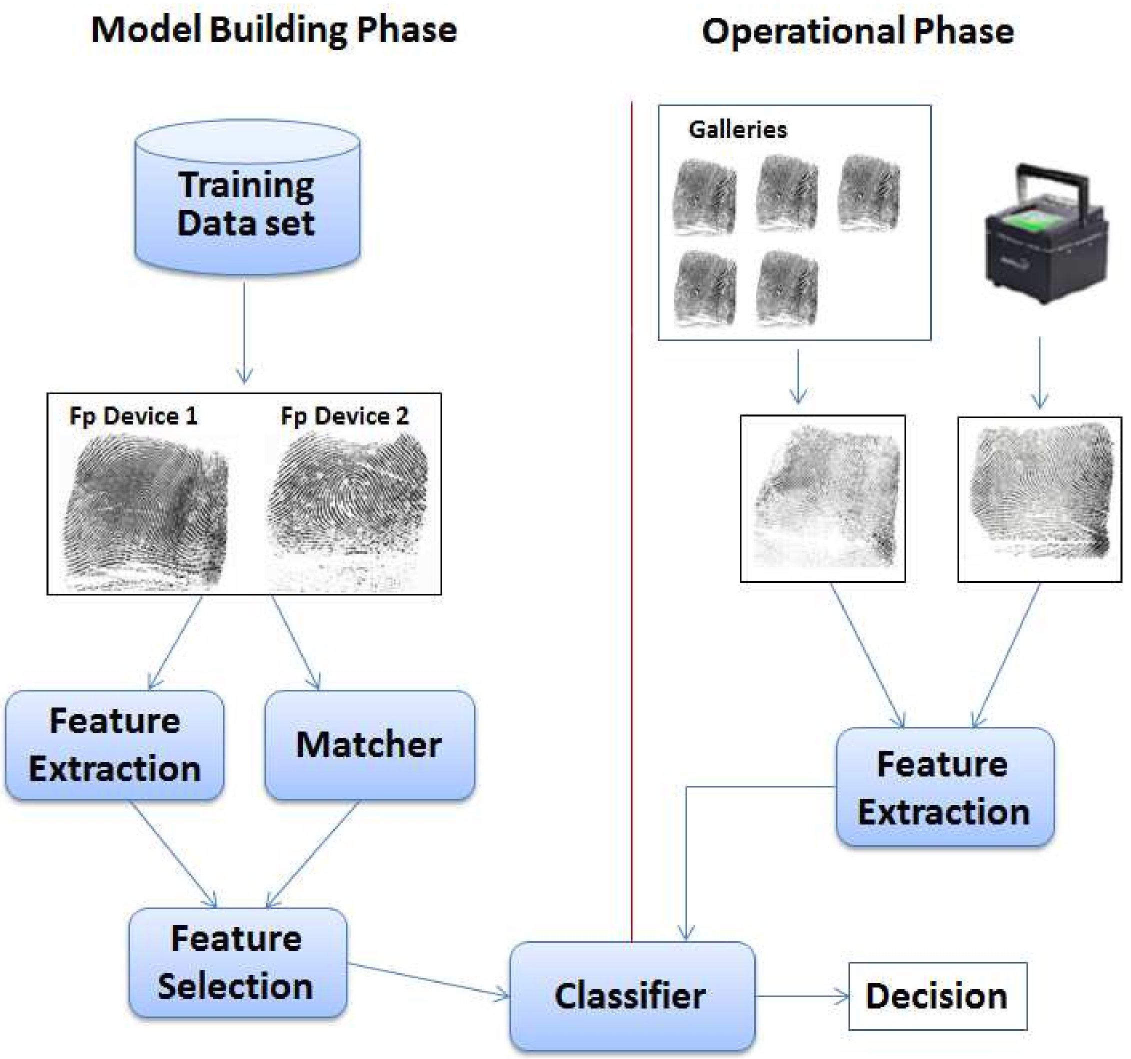

Figure 4.1: Architecture of the proposed approach. When building the model, we extract the interoperability features from a set of fingerprints acquired using different devices. We apply a feature selection technique, then train a machine learning algorithm. In the operational phase, we extract the interoperability features from the probe fingerprint, combine such

matcher. These data sets will be referred to as Identix data set, Bozorth dataset, and WebKey dataset.

All of the algorithm are evaluated as follows: we randomly select 25% of the data for training and the remaining 75% for test; this process is repeated 10 times and average results are reported.

Given the high correlation of the features we did not expect Naive Bayes to perform exceptionally well; Nevertheless, it is a good benchmark for comparing other algorithms and a lot less expensive than other algorithms from a computational point of view. The results for the Naive

	FMR (%)	FNMR (%)
Naive Bayes	0.17	4.75
Baseline	0.17	4.88
	0.22	4.75

Table 4.1: Naive Bayes performance on Identix data set.

	FMR (%)	FNMR (%)
Naive Bayes	0.09	8.74
Baseline	0.09	9.41
	0.16	8.74

Table 4.2: Naive Bayes performance on Bozorth data set.

Bayes classifier are shown in Tables 4.1, 4.2, and 4.3.

The small improvement in performance suggests that perhaps Naive Bayes is not the ideal classifier for this problem, given the high correlation between the features. Nevertheless, there is an error rate decrease in all three data sets.

Tables 4.4, 4.5, and 4.6 show the results of classification based on decision tree for the Identix data set, Bozorth data set, and WebKey data set respectively. Te Identix matcher benefits the most from this classifier, but the FMR is decreased for all three matching algorithms.

In our experiments we found that on all our data sets the best performance before applying feature selection techniques was obtained using a random forest classifier. Figures 4.2, 4.3, and 4.4 show the DET curves when using Random Forest classifiers on the complete set of features.

We tried several configurations and found that the best performance are achieved using 25 trees and 15 random features at each node. Increasing the number trees does not give sufficient reduction of error rates to justify the higher complexity of the model. Using this classification scheme we

	FMR (%)	FNMR (%)
Naive Bayes	0.03	1.48
Baseline	0.03	1.54
	0.04	1.48

Table 4.3: Naive Bayes performance on WebKey data set.

	FMR (%)	FNMR (%)
Decision Tree	0.011	3.044
Baseline	0.011	6.295
	6.6	3.044

Table 4.4: Decision tree performance on Identix data set.

	FMR (%)	FNMR (%)
Decision Tree	0.045	10.293
Baseline	0.045	10.5
	0.06	10.293

Table 4.5: Decision tree performance on Bozorth data set.

are able to improve the performance of all the three matching algorithms over their respective baselines, with the highest improvement for the Identix matcher.

Since training a Neural Network is extremely expensive in our case, we decided to use this classifier after applying feature selection techniques described in the next section.

Support vector machines are known to have issues with highly unbalanced data sets as ours (positive class to negative class ratio is 1 to 60), especially when dealing with many features. For this reason we will train SVMs after performing feature selection.

Various software was used for the machine learning algorithms:

- For Artificial Neural Networks, Matlab was used;
- For Naive Bayes, Decision Tree, Random Forest, Weka 3.6 was used [8];
- For Support Vector Machine, the C implementation of libsvm was used [3].

	FMR (%)	FNMR (%)
Decision Tree	0.004	1.921
Baseline	0.004	1.95
	0.05	1.921

Table 4.6: Decision tree performance on WebKey data set.

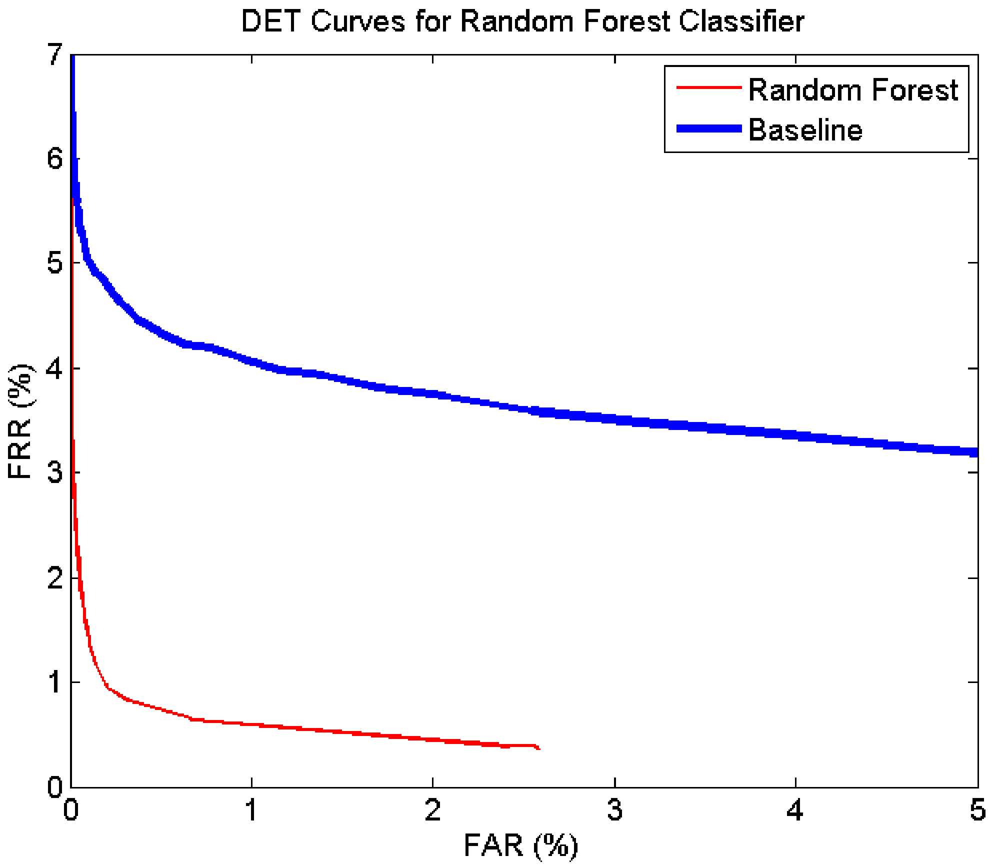

Figure 4.2: Random Forest performance on Identix data set.

4.1 Feature Selection

Principal Component Analysis (PCA) is a dimensionality reduction technique that analyzes linear combination of features. PCA projects the original feature vectors into a lower dimensional subspace determined by the principal components. The principal components are linear combinations of features and represent the direction of maximum variance in the data set.

When applying PCA to our data sets, the number of features decreases from 41 to 36. In this case every classifier we used on the reduced data set shows poor performance, with False Non Match Rates above 15% which is not an acceptable result. PCA seems to bias our data towards the impostor class even more. Given the high loss in performance when applying PCA, we can say that in this problem the correlation between features is non linear.

Information Gain, commonly referred to as InfoGain, is a feature ranking method based on

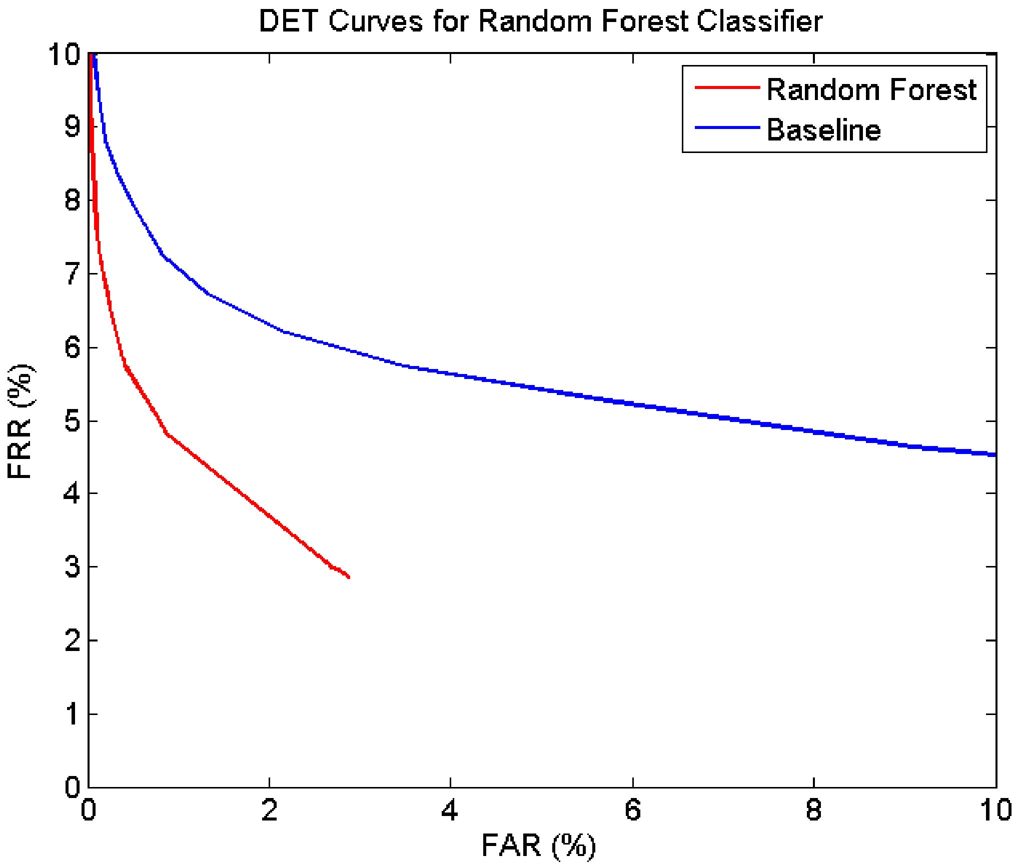

Figure 4.3: Random Forest performance on Bozorth data set.

entropy, as defined in the previous section. When applying InfoGain we selected the features ranked in the top 22. After the top 22 features the InfoGain dropped by 2 orders of magnitude, therefore we decided to ignore all the lower ranked features. Obviously, the match score was ranked first in all cases. While InfoGain feature selection works well with tree-based classifiers, it might not work as well for other types of algorithms.

When applying InfoGain to our data sets features were ranked similarly in all three cases. The most discriminant features appeared to be: match score, energy concentration, LBP and LPQ statistics, alignment parameters, and NFIQ quality score.

Tables 4.7, 4.8, and 4.9 show results for Naive Bayes classifier applied to the reduced data sets Identix, Bozorth, and WebKey respectively.

Comparing these results to the ones obtained when using the complete feature set, we can see that there is a slight decrease in error rates for the Identix and Bozorth data sets. For the WebKey

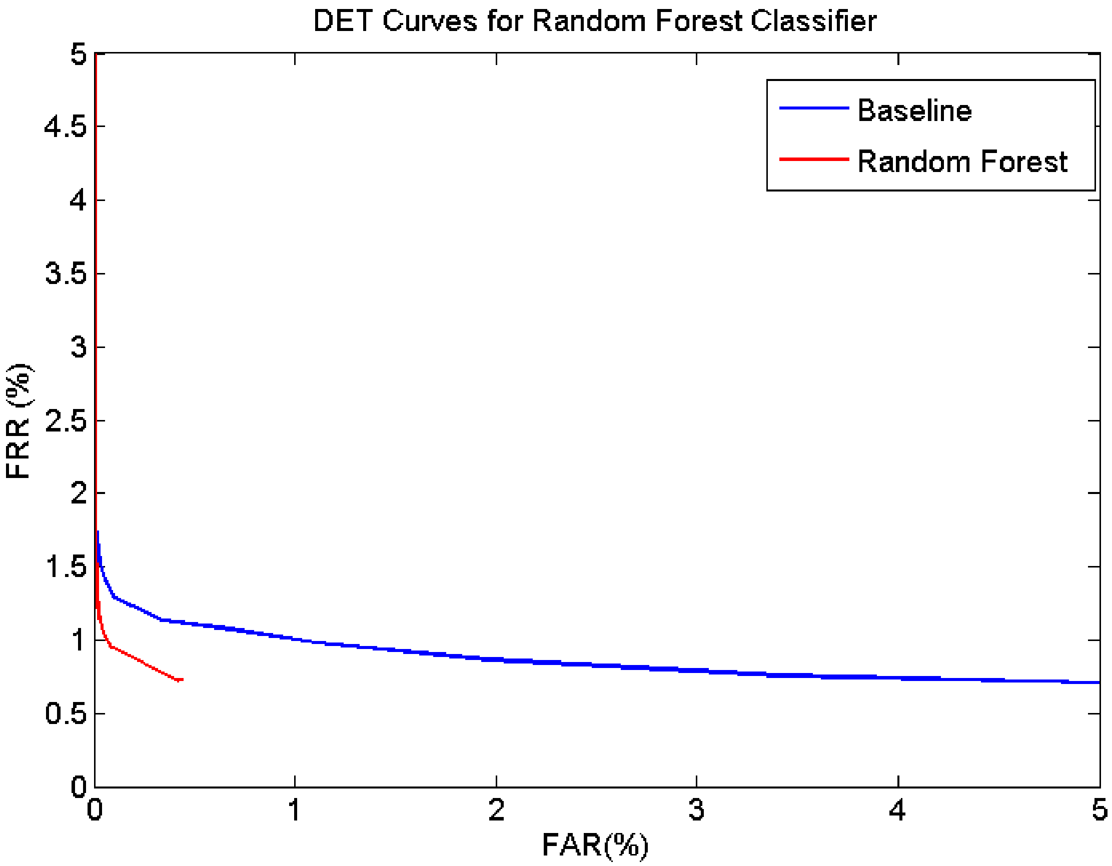

Figure 4.4: Random Forest performance on WebKey data set.

data set the error rates now show that the baseline has slightly higher performance, therefore, the feature selection method to be used is also affected by the matching algorithm.

We trained neural network classifiers using different configurations with various number of hidden layers and number of perceptrons for each layer. In all cases the results showed that the performance of these classifiers was inferior to the baseline case when considering only match scores, for all three data sets. As shown in the next section, when using another feature selection

	FMR (%)	**FNMR (%)**
Naive Bayes	0.09	4.75
Baseline	0.09	5.08
	0.22	4.75

Table 4.7: Naive Bayes performance on Identix data set after InfoGain feature selection.

	FMR (%)	FNMR (%)
Naive Bayes	0.11	8.58
Baseline	0.11	9.59
	0.27	8.58

Table 4.8: Naive Bayes performance on Bozorth data set after InfoGain feature selection.

	FMR (%)	FNMR (%)
Naive Bayes	0.04	1.66
Baseline	0.04	1.48
	0.02	1.66

Table 4.9: Naive Bayes performance on WebKey data set after InfoGain feature selection.

method neural networks classifiers are able to improve performance compared to the baseline. Therefore we think that for this particular problem, InfoGain is not an effective feature selection method when used in conjunction with neural network classifiers.

The same scenario was observed when using support vector machine classifiers. We experimented with different kernel types, i.e. Radial Basis Function (RBF), linear, polynomial, sigmoid. In all cases, only a few hundred out of over 8000 genuine match scores were correctly classified. As for neural networks, by using another feature selection method SVM are able to reduce error rates compared to the baseline case. This leads us to believe that InfoGain is not able to capture the features that SVM deems most discriminative.

Tables 4.10, 4.11, and 4.12 show results for decision tree classifier used on the reduced data sets.

These results show that InfoGain is a good feature selection technique when combined with a tree-based classifier. In all three data sets the error rates are reduced, with an increased improve-

	FMR (%)	FNMR (%)
Decision Tree	0.01	2.6
Baseline	0.01	6.42
	15.8	2.6

Table 4.10: Decision tree performance on Identix data set after InfoGain feature selection.

	FMR (%)	**FNMR (%)**
Decision Tree	0.02	10.94
Baseline	0.02	11.29
	0.03	10.94

Table 4.11: Decision tree performance on Bozorth data set after InfoGain feature selection.

	FMR (%)	**FNMR (%)**
Decision Tree	0.001	2.023
Baseline	0.001	2.29
	0.003	2.023

Table 4.12: Decision tree performance on WebKey data set after InfoGain feature selection.

ment over the baseline compared to using all the features. The matching algorithm which benefits the most from this feature selection is Identix: at the baseline operating point correspondent to the same FNMR, the FMR is reduced by three orders of magnitude.

Figures 4.5, 4.6, and 4.7 show DET curves derived from using a random forest classifier on the reduced feature set.

The performance remains stable after performing feature selection for the Identix and Bozorth data sets. For the WebKey dataset there is a slight loss in performance compared to the case of using the entire feature set. Even in this case the matcher with highest improvement over the baseline is Identix.

Correlation-based Feature Selection (CFS) is a feature selection technique developed on the idea that a good feature set contains features which are highly correlated with the class variable while being uncorrelated with each other [9]. The core of CFS, the subset evaluation function, is defined as:

$$M_S = \frac{k\overline{r_{cf}}}{\sqrt{k+(k-1)\overline{r_{ff}}}} \tag{4.1}$$

where M_S is the ”merit” value of a subset S containing f features, $\overline{r_{ff}}$ is the average inter-feature correlation, and $\overline{r_{cf}}$ is the average feature-class correlation.

As for InfoGain, CFS also seems to select similar features for all the datasets: match score,

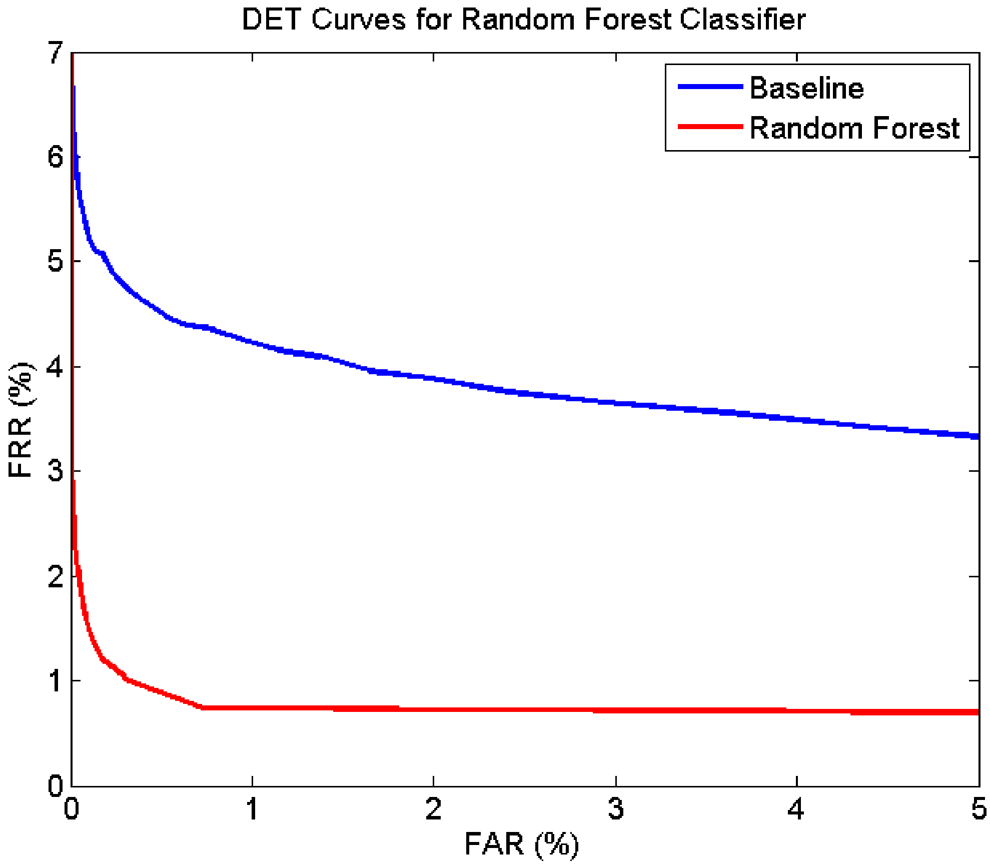

Figure 4.5: Random Forest performance on Identix data set after InfoGain feature selection.

minutiae count for the probe fingerprint, one of the energy bands, and one of the LBP statistics, rotation parameter from the alignment process. Additionally, for the WebKey dataset the PRNU and the gradient of the gallery fingerprint are retained by CFS.

Tables 4.13, 4.14, and 4.15 show FMR and FNMR originating from Naive Bayes classification on the reduced data sets.

	FMR (%)	**FNMR (%)**
Naive Bayes	0.11	4.83
Baseline	0.11	5.02
	0.19	4.83

Table 4.13: Naive Bayes performance on Identix data set after CFS feature selection.

For the Identix and Bozorth datasets the respective Naive Bayes classification performance are in line with the performance of the same classifier before performing CFS. The WebKey data set,

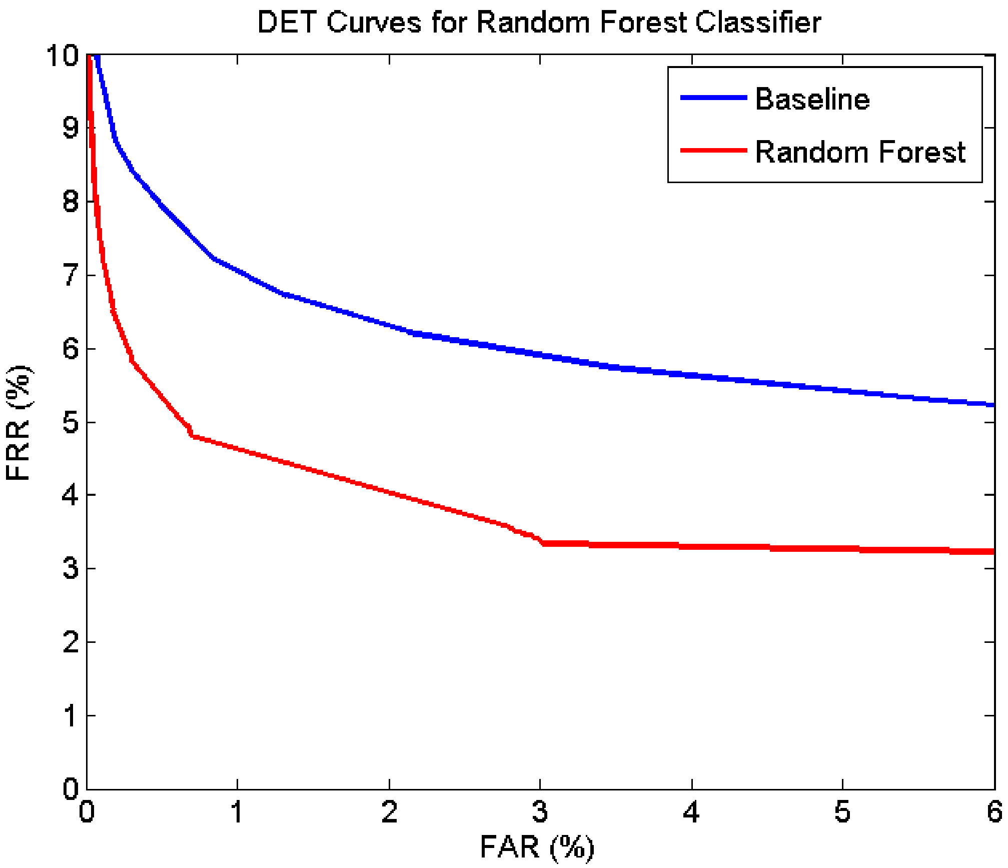

Figure 4.6: Random Forest performance on Bozorth data set after InfoGain feature selection.

	FMR (%)	FNMR (%)
Naive Bayes	0.11	8.89
Baseline	0.11	9.21
	0.15	8.89

Table 4.14: Naive Bayes performance on Bozorth data set after CFS feature selection.

	FMR (%)	FNMR (%)
Naive Bayes	0.04	1.62
Baseline	0.04	1.48
	0.02	1.62

Table 4.15: Naive Bayes performance on WebKey data set after CFS feature selection.

however, shows a loss of performance, with error rates higher than the baseline.

We trained several neural network classifiers, each with different number of hidden layers and

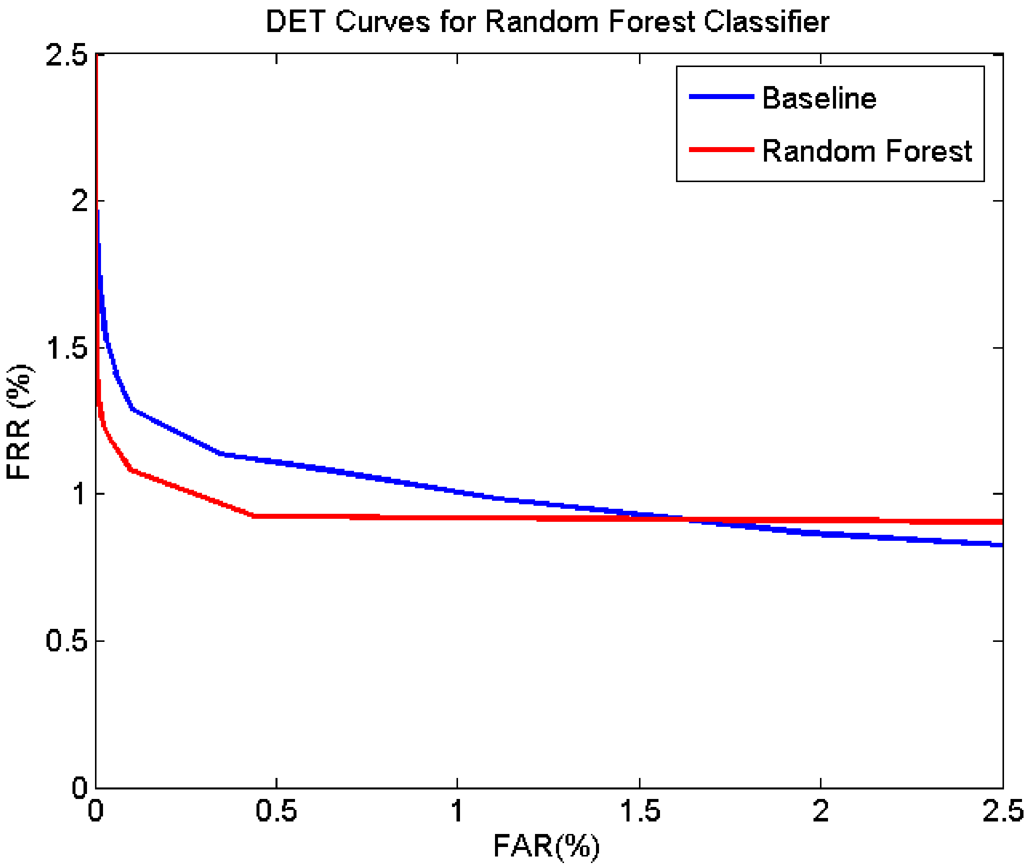

Figure 4.7: Random Forest performance on WebKey data set after InfoGain feature selection.

number of perceptron at each layer. Among all the configurations, the one that worked the best was the simplest: 2 hidden layers of 5 perceptrons each. The results on the tree data sets are reported in Figures 4.8, 4.9, and 4.10.

While the performance for matcher WebKey are about the same as the baseline, there is a marginal improvement of error rates for Identix and Bozorth. Identix is again the matcher which benefits the most from this scheme.

The performance of neural networks on our data sets are by far inferior to that of other classifiers, specifically decision tree and random forest.

The results of classification using a decision tree classifier are reported in Tables 4.16, 4.17, and 4.18.

When comparing these results with the ones obtained using the same classifier on the complete feature set, we see that the error rates increase for the Bozorth and WebKey data sets. For the

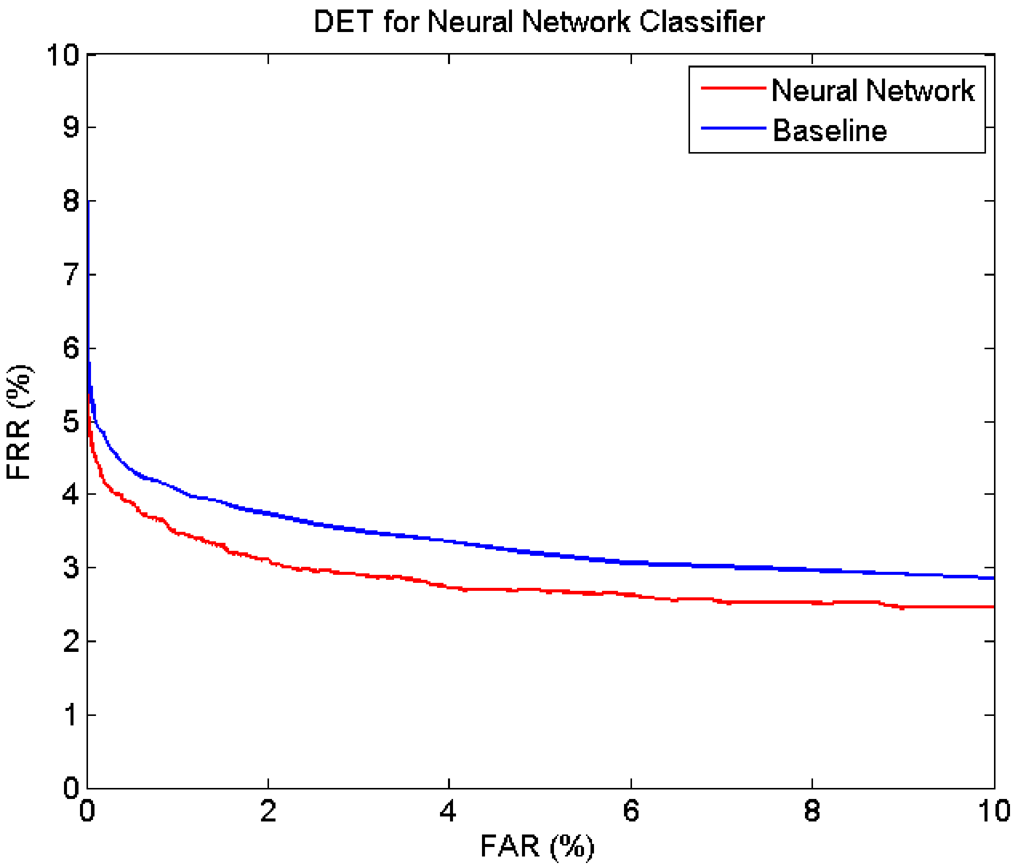

Figure 4.8: Neural Network performance on Identix data set after CFS feature selection.

	FMR (%)	FNMR (%)
Decision Tree	0.007	2.91
Baseline	0.007	6.6
	9.18	2.91

Table 4.16: Decision tree performance on Identix data set after CFS feature selection.

	FMR (%)	FNMR (%)
Decision Tree	0.02	11.53
Baseline	0.02	11.29
	0.01	11.53

Table 4.17: Decision tree performance on Bozorth data set after CFS feature selection.

Identix data set, this classifier is still able to greatly decrease error rates. Although the improvement over the baseline is slightly smaller than the one obtained using the complete feature set, one might consider using this model for the reduced complexity: by performing CFS the set of feature is

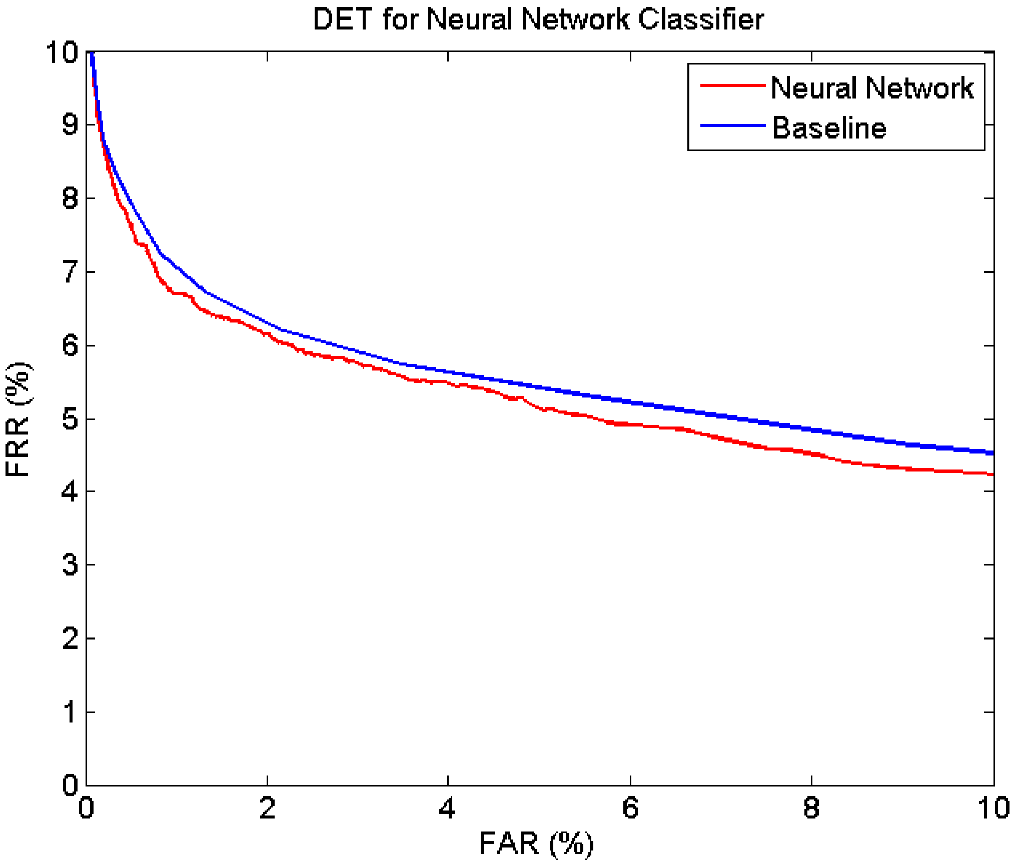

Figure 4.9: Neural Network performance on Bozorth data set after CFS feature selection.

	FMR (%)	FNMR (%)
Decision Tree	0.004	2.1
Baseline	0.004	1.95
	0.002	2.1

Table 4.18: Decision tree performance on WebKey data set after CFS feature selection.

reduced from 41 to 5.

	FMR (%)	FNMR (%)
SVM	0.002	5.53
Baseline	0.002	7.62
	0.43	5.53

Table 4.19: SVM performance on Identix data set after CFS feature selection.

SVM performance are reported in Tables 4.19, 4.20, and 4.21. For the Bozorth and Identix

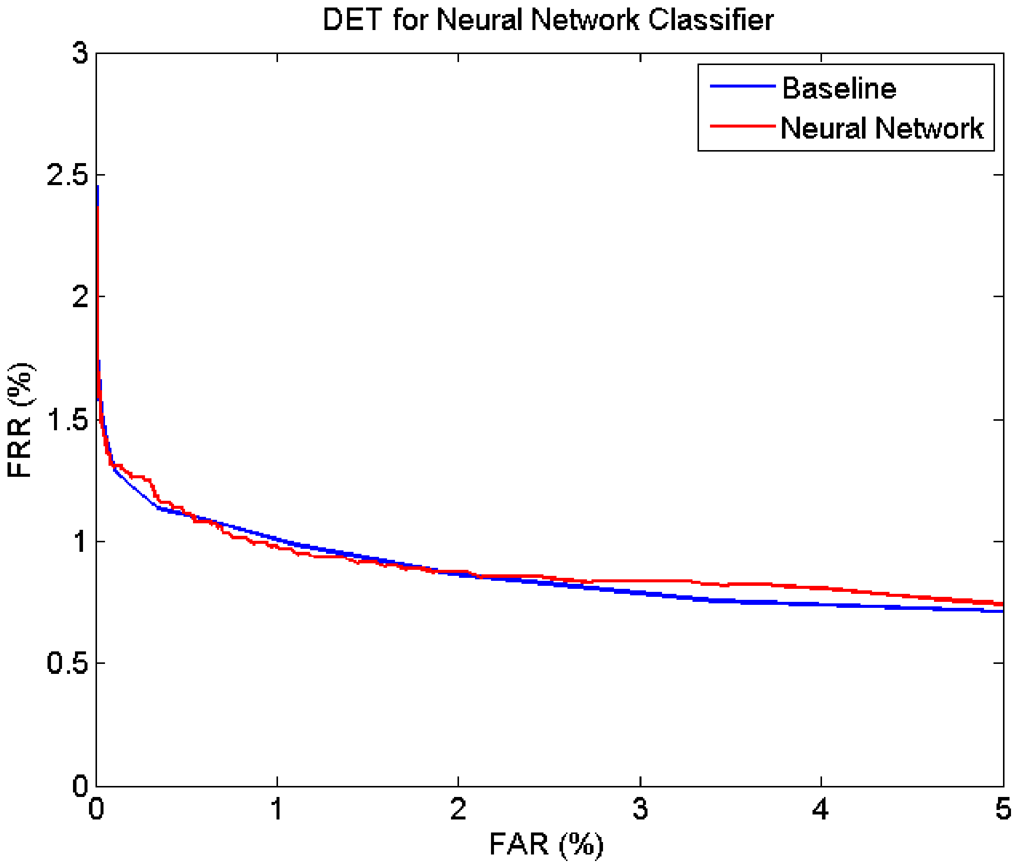

Figure 4.10: Neural Network performance on WebKey data set after CFS feature selection.

	FMR (%)	FNMR (%)
SVM	0.0004	18.28
Baseline	0.0004	17.25
	0.0001	18.28

Table 4.20: SVM performance on Bozorth data set after CFS feature selection.

	FMR (%)	FNMR (%)
SVM	0.001	2.068
Baseline	0.001	2.29
	0.002	2.068

Table 4.21: SVM performance on WebKey data set after CFS feature selection.

dataset, SVM generates error rates higher than the baseline, while for the Identix dataset it is able to reduce the FMR by an order of magnitude compared to the baseline corresponding operating point. We believe this is due to the fact that the data sets are highly unbalanced. In order to improve these results we applied sampling for the training sets with different rates for the genuine and impostor. By undersampling the impostor class we tried to restore balance in the data sets hoping this would improve error rates. Different undersampling rates were tested, however this method did not yield better result in any case.

Figures 4.11, 4.12, and 4.13 show the DET curves for Random Forest classifier evaluated on the data sets after performing CFS feature selection.

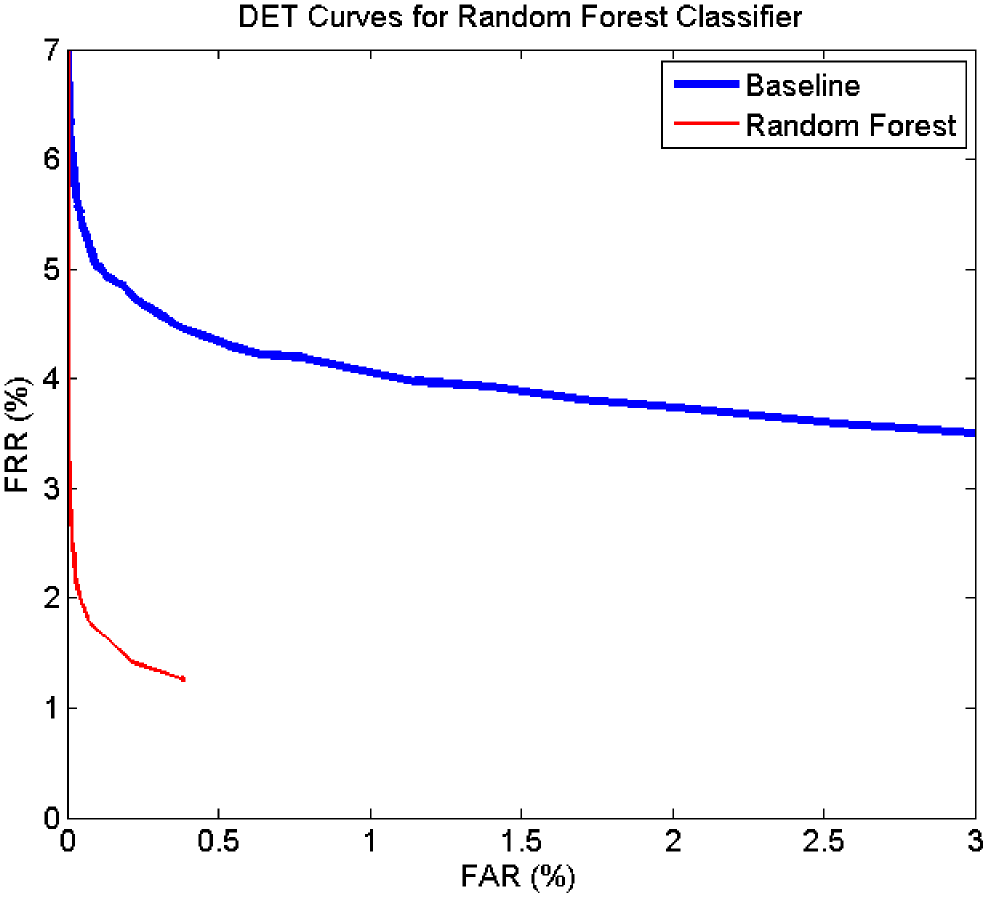

Figure 4.11: Random Forest performance on Identix data set after CFS feature selection.

It is surprising to see that while for the Bozorth and WebKey data sets there is a limited improvement in error rates, for the Identix data set the performance improves significantly compared to the baseline.

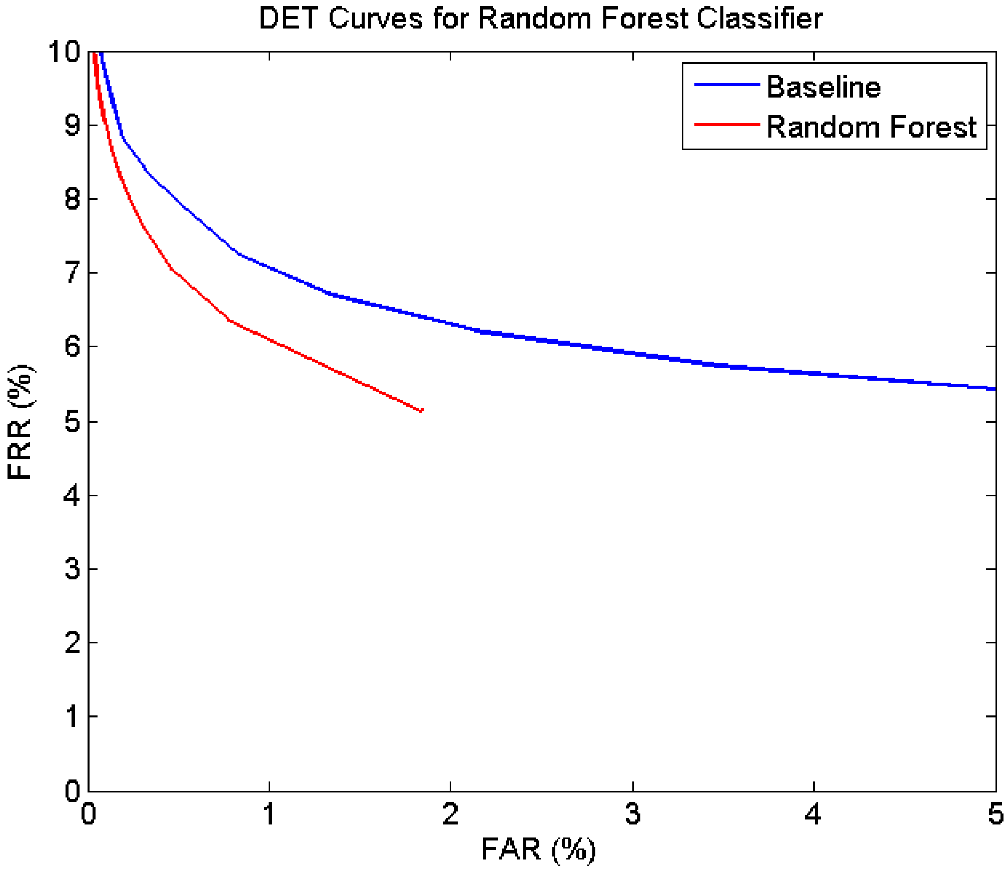

Figure 4.12: Random Forest performance on Bozorth data set after CFS feature selection.

4.2 Pairwise Models

The model discussed so far is a global model which takes into account all the possible combination of devices. Alternatively, we can implement a model for each pair of devices we decide do deploy in our system. In our case, having 5 devices would result in 10 models. The set of features is the same as the one adopted in the global model after performing feature selection. The main difference is in the training phase: each pairwise model will be trained only on fingerprint samples acquired using two devices. The idea behind these pairwise models is that each one specializes in the combination of 2 devices, possibly reducing error rates further than the global model.

We can see the results for three out the 10 models in Figures 4.14, 4.15, and 4.16.

Although it's not always the case, the pairwise models are able to further reduce error rates for some devices.

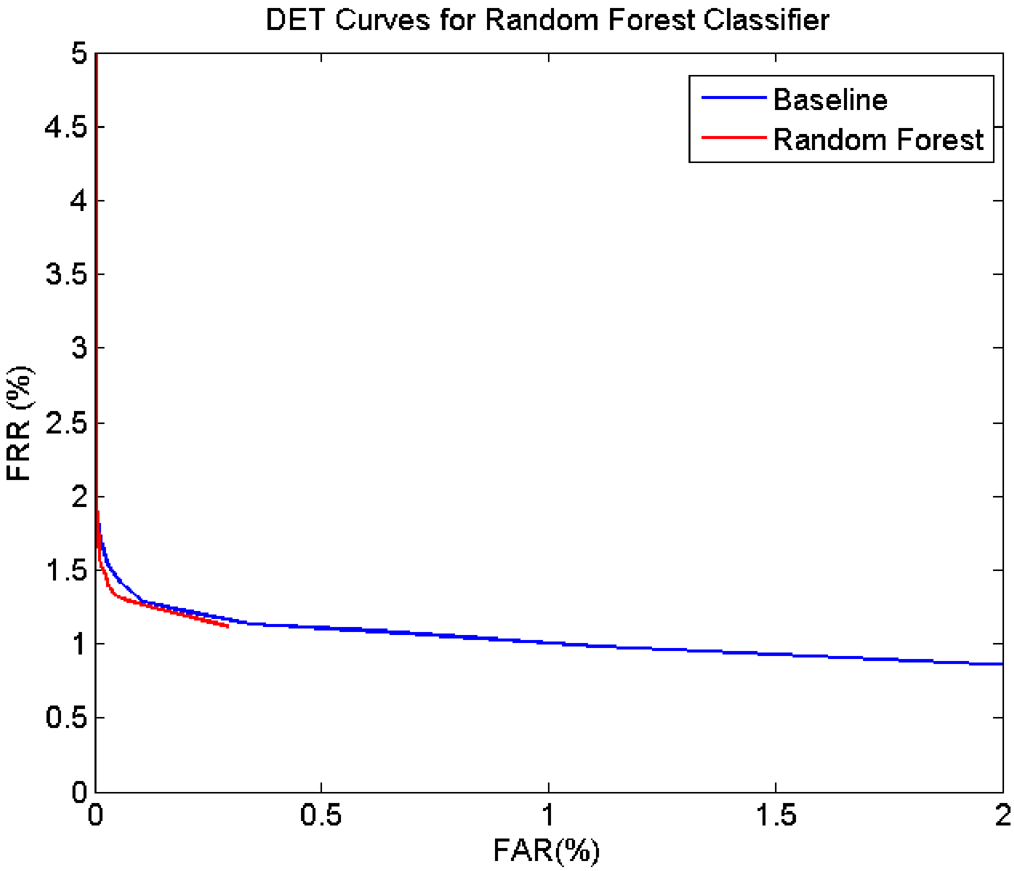

Figure 4.13: Random Forest performance on WebKey data set after CFS feature selection.

4.3 Introduction of new sensors

An interesting scenario to analyze is the introduction of a new fingerprint sensor in a system which implemented our proposed approach for interoperability enhancement. Ideally, when introducing a new fingerprint sensor one should acquire fingerprints using the new device and retrain the model in order to fully take advantage of the interoperability enhancement scheme. There are cases for which this is not possible. In order to analyze the behavior of our proposed approach in such scenarios, we performed additional experiments on our set of 5 sensors. In this case the Random Forest classifier was used in conjunction with InfoGain feature selection. For each of the 5 sensors the evaluation was carried out as follows:

- Train a random forest classifier on samples of the remaining 4 devices (referred to as partial training model), with the same parameters used for the global model;

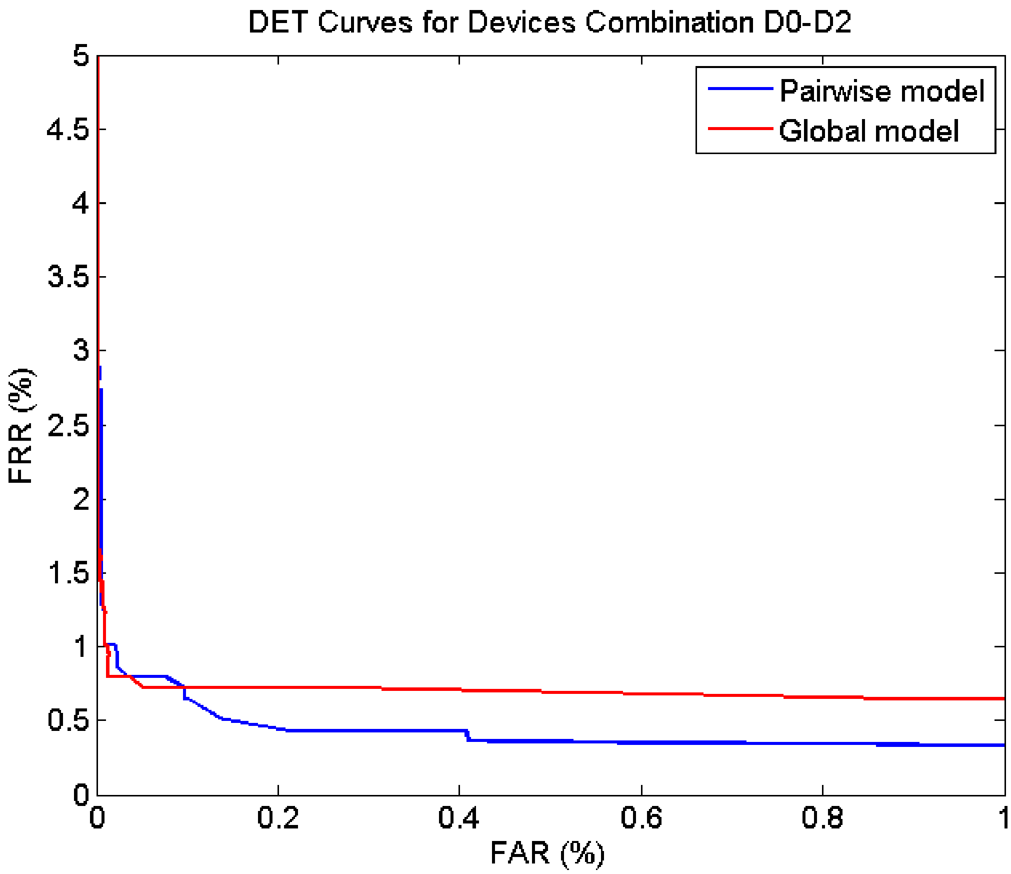

Figure 4.14: DET curves for pairwise model pertaining to devices D0 and D2.

- Test the model on matches for which the device used for the probe fingerprint is the one excluded from the training, while the device used for the probe fingerprint can be any of the remaining 4.

- Compare the performance of the partial training model to the global model (which is trained using fingerprints acquired from all the devices) and the baseline case of using match scores only.

Figures 4.17, 4.18, 4.19, 4.20, and 4.21 show the DET curves for assessing performance in this scenario when using the Identix matching algorithm, although the same trends were observed for the other two matchers.

When introducing devices D0, D2, and D3 as new device, the performance of the random forest classified without performing the training phase again is not too far off the performance of

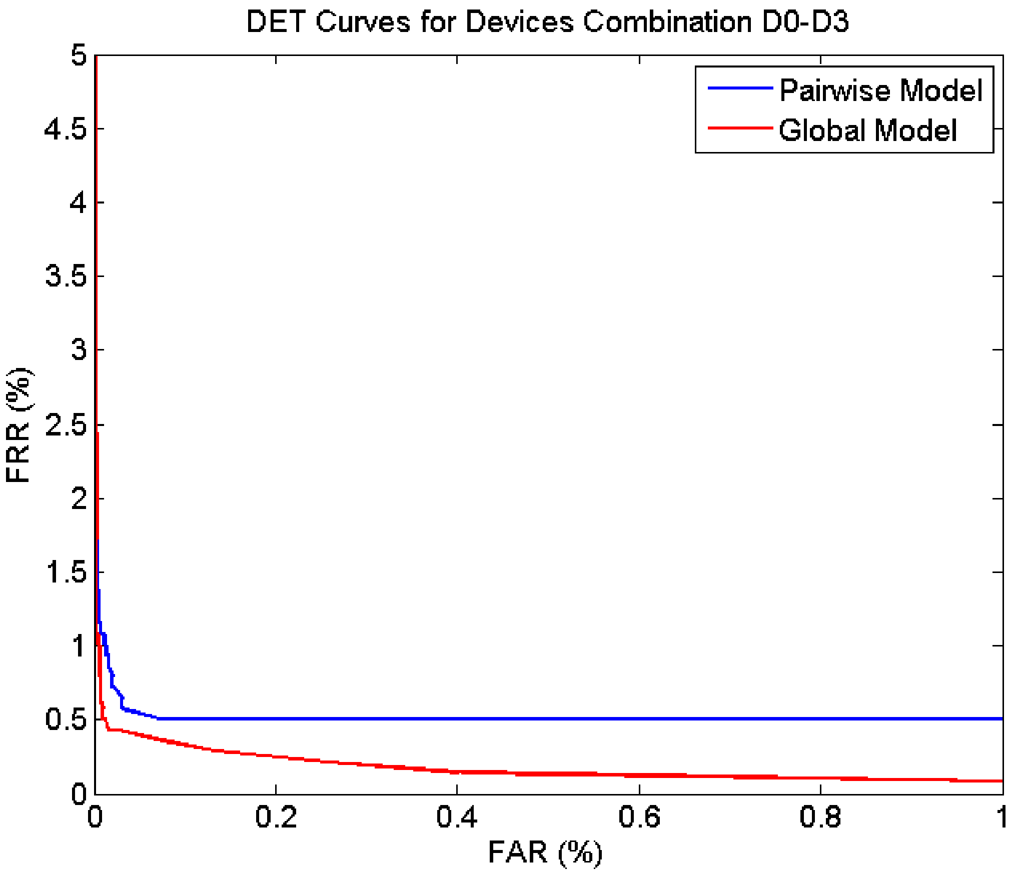

Figure 4.15: DET curves for pairwise model pertaining to devices D0 and D3.

the global model. When considering devices D1 and D4 as new devices there is a clear increase in error rates, although the DET curve is below that of the baseline in all cases. We can infer from this results that as long as we introduce a device *as good as or better than* the ones we currently have in the system, we can still use the current model without retraining, until we get a sufficient fingerprint samples from the new device. Introducing a new device which generates lower quality fingerprints on the other hand will become problematic without retraining the current classifier.

4.4 Discussion

There is a large difference in the performance of the different machine learning in combination with several feature selection techniques, as shown in the previous section. Given the different baselines, different accuracy across the three data sets is to be expected. However, we are able to

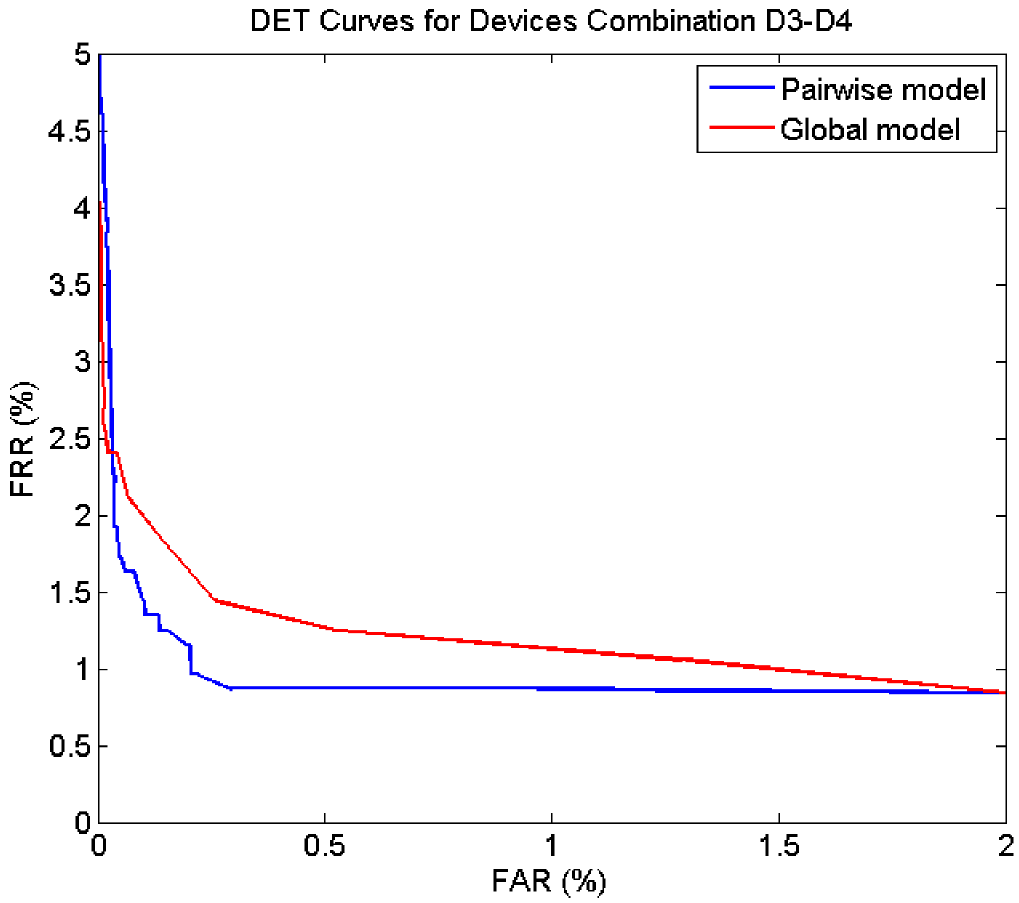

Figure 4.16: DET curves for pairwise model pertaining to devices D2 and D4.

improve error rates for all the fingerprint matchers considered in this study. Specifically, Random Forest was the classifier which consistently outperformed the others. Additionally, in order to reduce the computational complexity of the model, we can operate InfoGain feature selection and discard almost half of the features while the Random Forest error rates remain stable.

4.5 Threats to Validity

We understand that, as with every experimental study, there might be threats to the validity of results presented in this work. For example, the results of our interoperability enhancement scheme are valid for the considered devices; however we cannot make more general statements about other devices outside the five we used. Additionally, we evaluated the proposed approach only on a

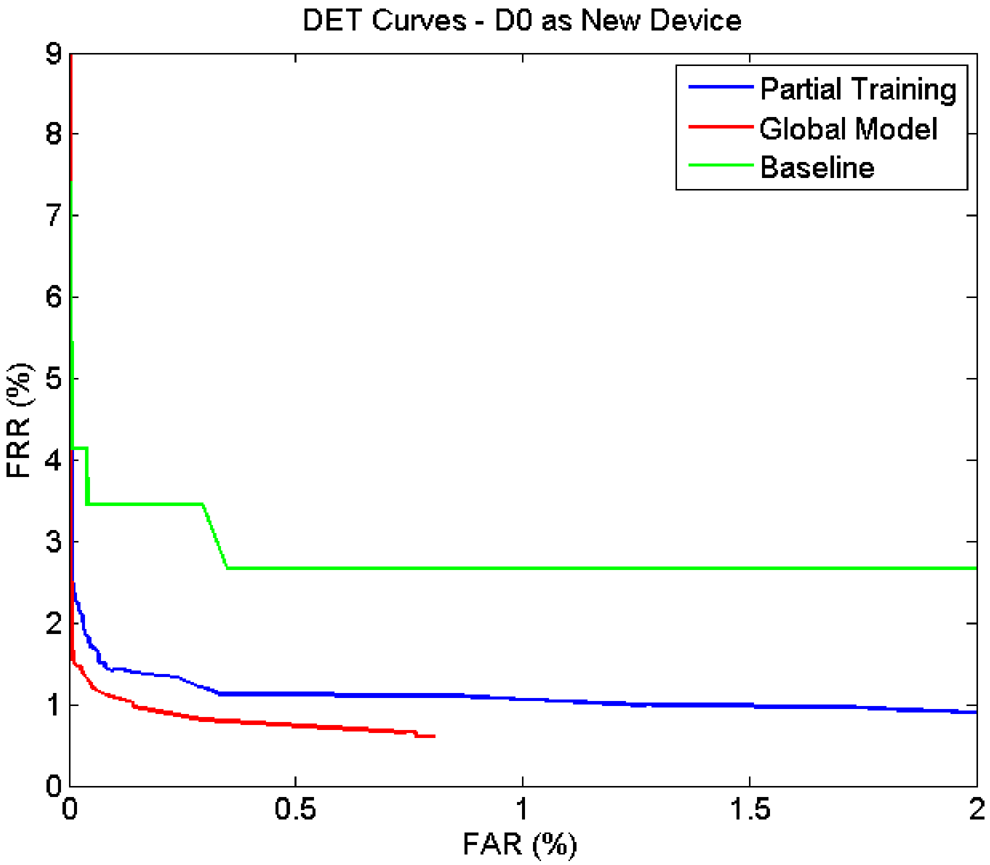

Figure 4.17: DET curves when introducing D0 as new device in the system.

private data set having a certain population distribution (e.g. gender, age, ethnicity, etc.). We are unable to test our enhancement scheme on other data due the absence of publicly available data sets for fingerprint interoperability. Another note of concern is how the proposed approach will scale when dealing with tens or hundreds of fingerprint sensors, and possibly millions of users.

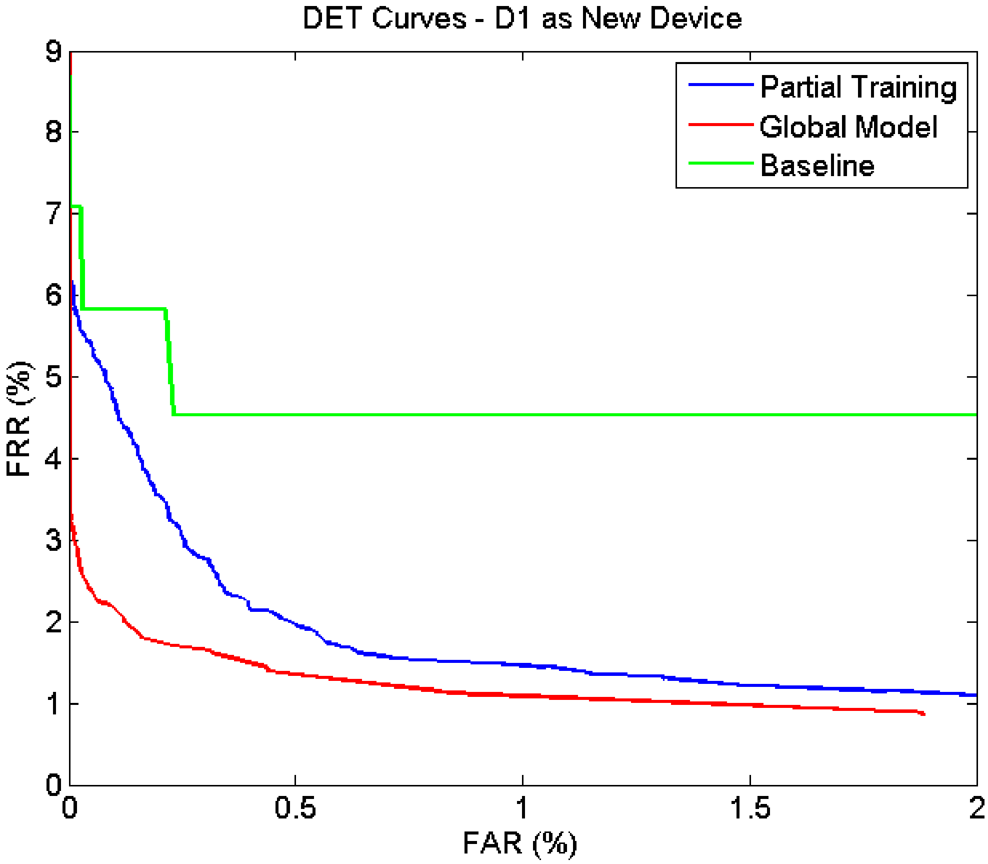

Figure 4.18: DET curves when introducing D1 as new device in the system.

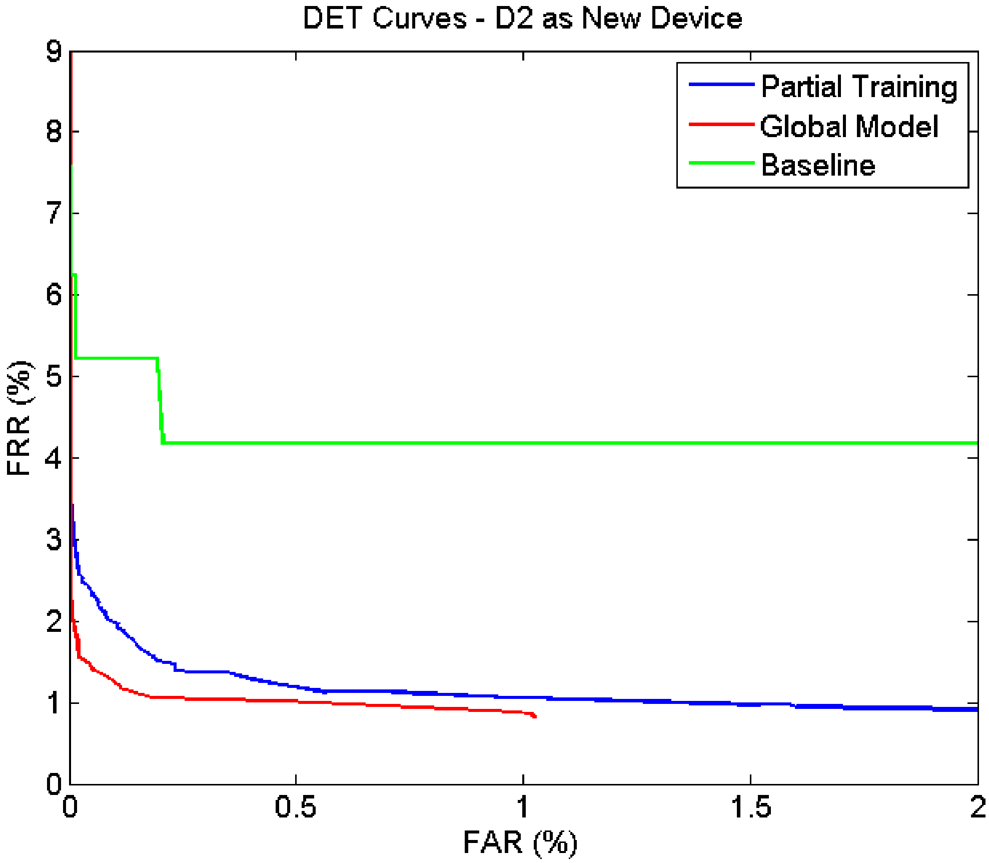

Figure 4.19: DET curves when introducing D2 as new device in the system.

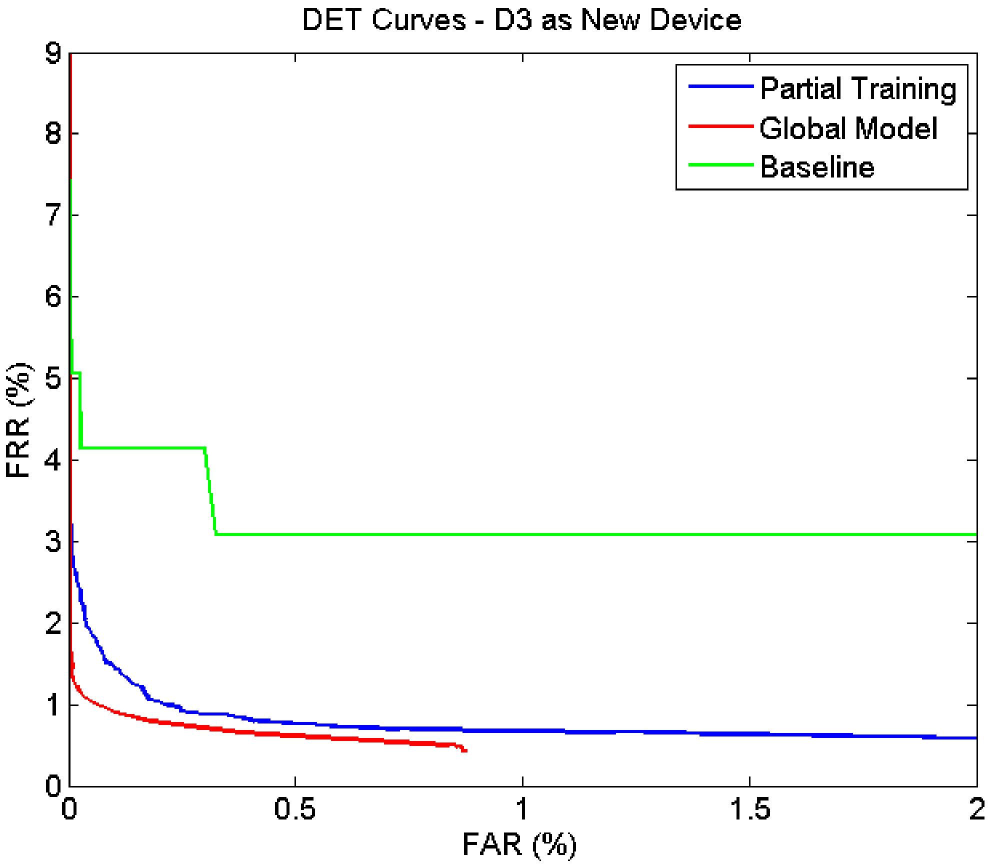

Figure 4.20: DET curves when introducing D3 as new device in the system.

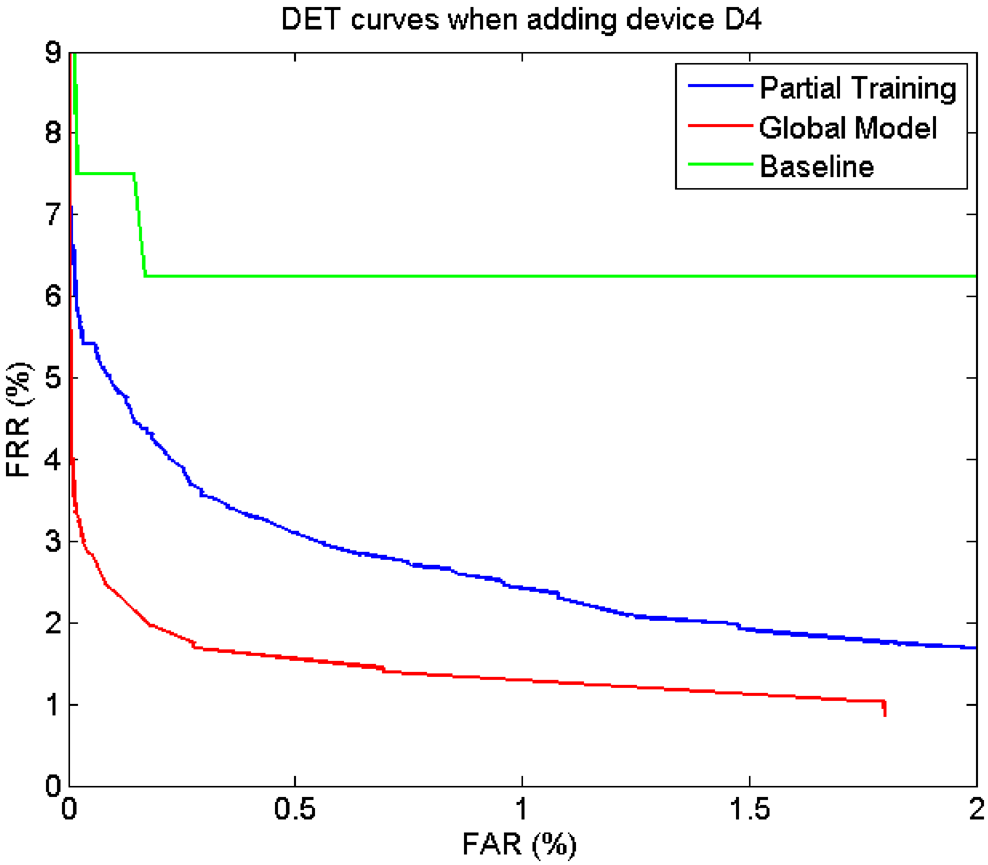

Figure 4.21: DET curves when introducing D4 as new device in the system.

Chapter 5

Conclusions and Future Work

5.1 Conclusions

Interoperability between different fingerprint sensors is a very important aspect of a fingerprint recognition system. In this work we analyzed how deploying different fingerprint sensors in a system impacts the quality of fingerprint samples and, consequently, error rates. We performed match score analysis to show that interoperability is lacking when considering the set of fingerprint sensors and matching algorithms used in this study. Although the matching algorithms exhibit different baseline performance, they all share the same trend of increased error rates in cross-device matching scenarios. If we want to increase accuracy when deploying different fingerprint sensors in the same system, we must exploit additional information from fingerprints that matchers don't normally take into account.

We defined a set of features which are able to capture device-specific characteristics and fuse them together in order to render matching more accurate. The interoperability enhancement feature set consists of features of different nature, extracted from domains. This ensures that we use information which is discriminative yet not tightly correlated, therefore avoiding redundant features. Our model does not interfere with the typical matching operation, but rather works in parallel to the typical biometric system. Several Machine Learning algorithms were used in the fusion scheme, in conjunction with different feature selection techniques. As result, we show that using a Random Forest classifier, used in conjunction with InfoGain feature selection, yields to improvements in error rates for all three matching algorithms. Specifically, for all three matching algorithms, our interoperability enhancement scheme is able to achieve cross-device matching performance comparable to that of intra-device scenario. There are a few cases for which our approach slightly increases the error rates over the baseline, e.g. decision tree classifier used in conjunction with CFS feature selection for Bozorth and WebKey data sets. The behavior of our proposed approach differs across the three data sets. Specifically, Identix is the matching algorithm which by far benefits the most from our approach. Identix benefits from our model even in cases where applying the same model to another data set increases its error rates.

In contrast to constructing a global model which takes into account all the possible combination of devices, we can build pairwise models which act only on a specific pair of devices. For some device pairs the pairwise models are able to further increase matching accuracy beyond that of the global model. A note of concern when using pairwise models is the introduction of a new device into the system. In this case we might need to train and add a large number of new pairwise models, though the existing models do not need retraining.

When deploying our proposed global model in a fingerprint recognition system a problem can arise if we decide to add a new fingerprint sensor. We performed experiments by excluding each of the available devices from the training set and introducing it at the test phase. Results showed that, at least until we acquire a sufficient number of fingerprints using the new device, the model does not need to be retrained right away.

5.2 Future Work

Some possible future directions for this study are:

- Expand the set of sensors, possibly including other sensing technologies as well;
- Extend experiments to other data sets;

- Extend the set of features;
- Evaluate additional classifiers and additional feature selection techniques, in particular non-linear feature reduction methods;
- Analyze the computational complexity and running time required by the implementation of this model;
- Evaluate the performance of the proposed scheme when the biometric system is operating in identification mode.

Bibliography

[1] B. E. Boser, I. Guyon, and V. Vapnik. A training algorithm for optimal margin classifiers. *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, pages 144–152, 1992.

[2] L. Breiman. Random forest. *Machine Learning*, pages 5–32, 2001.

[3] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[4] Y. Chen, S.C. Dass, and A.K. Jain. Fingerprint quality indices for presicting authentication performance. *Audio- and Video-Based Biometric Person Authentication*, pages 160–170, 2005.

[5] S. Elliott, S. Modi, L. Maccarone, M. Young, J. Changlong, and H. Kim. Image quality and minutiae count comparison for genuine and artificial fingerprints. *41st Annual IEEE International Carnahan Conference on Security Technology*, pages 30–36, 2007.

[6] R. Gonzalez and R. Woods. Digital image processing. *ed: Prentice Hall Press*, 2002.

[7] P. Grother and E. Tabassi. Performance of biometric quality measures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(4):531–543, 2007.

[8] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and Ian H. Witten. The weka data mining software: An update. *SIGKDD Explorations*, 11, 2009.

[9] M. A. Hall. *Correlation-based Feature Selection for Machine Learning*. PhD thesis, The University of Waikato, 1999.

[10] E. Kukula, C. Blomeke, S. Modi, and S. Elliott. Effect of human-biometric sensor interaction on fingerprint matching performance, image quality and minutiae count. *International Journal of Computer Applications in Technology*, 34(4):270–277, 2009.

[11] Eyung Lim, Xudong Jiang, and Weiyun Yau. Fingerprint quality and validity analysis. *International Conference on Image Processing. Proceedings*, 1:469–472, 2002.

[12] T. Maenpaa. *The Local binary pattern approach to texture analysis: Extensions and applications*. Oulun yliopisto, 2003.

[13] Norman N. Poh, J. Kittler, and T. Bourlai. Improving biometric device interoperability by likelihood ratio-based quality dependent score normalization. *First IEEE International Conference on Biometrics: Theory, Applications, and Systems BTAS 2007*, pages 1–5, 2007.

[14] R. Nadgir. Facilitating sensor interoperability and incorporating quality in fingerprint matching systems. 2006.

[15] T. Ojala, M. Pietikainen, and D. Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern recognition*, 29(1):51–59, 1996.

[16] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(7):971–987, 2002.

[17] V. Ojansivu and J. Heikkila. Blur insensitive texture classification using local phase quantization. *Image and Signal Processing*, pages 236–243, 2008.

[18] V. Ojansivu, E. Rahtu, J. Heikkila, E. Rahtu, and J. Heikkila. Rotation invariant blur insensitive texture analysis using local phase quantization. *19th International Conference on Pattern Recognition*, 4, 2008.

[19] N. Poh, J. Kittler, and T. Bourlai. Quality-based score normalization with device qualitative information for multimodal biometric fusion. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 40(3):539–554, 2010.

[20] E. Rahtu, J. Heikkila, V. Ojansivu, and T. Ahonen. Local phase quantization for blur-insensitive image analysis. *Image and Vision Computing*, 30(8):501–512, 2012.

[21] A. Ross and A. Jain. Biometric sensor interoperability: A case study in fingerprints. *Proc. of International ECCV Workshop on Biometric Authentication*, pages 134–145, 2004.

[22] A. Ross, A. Jain, and K. Nandakumar. *Introduction to Biometrics: A Textbook*. Springer, 2011.

[23] C. Wilson, P. Grother, and R. Chandramouli. Biometric data specification for personal identity verification. *NIST Special Publication 800-76-1*, 2007.