

NISTIR 8382

NFIQ 2
NIST Fingerprint Image Quality

Elham Tabassi
Martin Olsen
Oliver Bausinger
Christoph Busch
Andrew Figlarz
Gregory Fiumara
Olaf Henniger
Johannes Merkle
Timo Ruhland
Christopher Schiel
Michael Schwaiger

This publication is available free of charge from:
<https://doi.org/10.6028/NIST.IR.8382>

NIST
National Institute of
Standards and Technology
U.S. Department of Commerce

NISTIR 8382

NFIQ 2

NIST Fingerprint Image Quality

Elham Tabassi
Andrew Figlarz
Gregory Fiumara
*Information Access Division
Information Technology Laboratory*

Timo Ruhland
Christopher Schiel
Bundeskriminalamt

Oliver Bausinger
Federal Office for Information Security

Olaf Henniger
Fraunhofer Institute for Computer Graphics Research IGD

Martin Olsen
Christoph Busch
Hochschule Darmstadt

Johannes Merkle
Michael Schwaiger
secunet Security Networks AG

This publication is available free of charge from:
<https://doi.org/10.6028/NIST.IR.8382>

July 2021



U.S. Department of Commerce
Gina M. Raimondo, Secretary

National Institute of Standards and Technology
*James K. Olthoff, Performing the Non-Exclusive Functions and Duties of the Under Secretary of Commerce
for Standards and Technology & Director, National Institute of Standards and Technology*

Certain commercial entities, equipment, or materials may be identified in this document in order to describe an experimental procedure or concept adequately. Such identification is not intended to imply recommendation or endorsement by the National Institute of Standards and Technology, nor is it intended to imply that the entities, materials, or equipment are necessarily the best available for the purpose.

National Institute of Standards and Technology Interagency or Internal Report 8382
Natl. Inst. Stand. Technol. Interag. Intern. Rep. 8382, 99 pages (July 2021)

This publication is available free of charge from:
<https://doi.org/10.6028/NIST.IR.8382>

Abstract

NIST Fingerprint Image Quality (NFIQ 2) is open source software that links image quality of optical and ink 500 pixel per inch fingerprints to operational recognition performance. This allows quality values to be tightly defined and then numerically calibrated, which in turn allows for the standardization needed to support a worldwide deployment of fingerprint sensors with universally interpretable image qualities. NFIQ 2 quality features are formally standardized as part of ISO/IEC 29794-4 and serve as the reference implementation of the standard.

Key words

biometrics; fingerprints; standards; quality.

Not Human Subjects Research

The National Institute of Standards and Technology Research Protection Office reviewed the protocol for this project and determined it is not human subjects research as defined in Department of Commerce Regulations, 15 CFR 27, also known as the Common Rule for the Protection of Human Subjects (45 CFR 46, Subpart A).

Acknowledgments

The authors would like to thank the sponsors of this project, Department of Homeland Security (DHS) Science and Technology Directorate, and the Federal Bureau of Investigation in the United States, and Federal Office for Information Security (BSI) and Federal Criminal Police Office (BKA) in Germany. Special thanks to MITRE Corporation for their assistance with the implementation of the code.

The authors would also like to thank all the organizations that have shared biometric data with NIST for this and other evaluations and studies. This data has proven invaluable for benchmarking and advancement of biometric matching technologies. Data used in development of NFIQ 2 came from the FBI, DHS, Los Angeles County Sheriff's Department (LACNTY), Arizona Department of Public Safety (AZDPS). Federal Criminal Police Office (BKA) in Germany performed testing of NFIQ 2 on their data.

Furthermore, the authors extend their gratitude to the fingerprint community and NFIQ 2 community in particular for their time and contribution to this project.

NFIQ 2

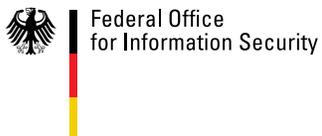
NIST Fingerprint Image Quality

NIST Interagency Report 8382

Elham Tabassi Martin Olsen

O. Bausinger C. Busch A. Figlarz G. Fiumara
O. Henniger J. Merkle T. Ruhland C. Schiel M. Schwaiger

NIST
**National Institute of
Standards and Technology**
U.S. Department of Commerce



secunet

12 July 2021

Contents

EXECUTIVE SUMMARY	vii
RELEASE NOTES	viii
CAVEATS	ix
1 INTRODUCTION	1
2 DATA	2
3 ALGORITHM SUBMISSION AND USE	2
4 QUALITY FEATURES	4
4.1 SEGMENTING THE FINGERPRINT IMAGE	5
4.2 FREQUENCY DOMAIN ANALYSIS	5
4.3 LOCAL CLARITY SCORE	7
4.4 ORIENTATION CERTAINTY LEVEL	8
4.5 ORIENTATION FLOW	10
4.6 RIDGE VALLEY UNIFORMITY	11
4.7 MU	11
4.8 MMB	12
4.9 MINUTIAE COUNT	12
4.10 MINUTIAE QUALITY	13
4.11 ROI-BASED FEATURES	13
4.12 QUALITY VECTORS FROM LOCAL QUALITIES	17
4.13 THE NFIQ 2 FEATURE VECTOR	18
4.14 PREDICTIVE POWER OF NFIQ 2 FEATURES	18
5 TRAINING	20
5.1 TRAINING DATA	20
5.2 TRAINING PARAMETERS AND RESULTS	20
5.3 MODEL VALIDATION	23
6 PREDICTIVE POWER OF NFIQ 2	24
6.1 REJECTION CURVES	24
6.2 RANKED DET	27
6.3 COMPUTATION EFFICIENCY	29
REFERENCES	31
APPENDIX A	32
NFIQ 2 CONFORMANCE TEST CRITERIA	32
A.1 PURPOSE	32
A.2 NFIQ 2 CONFORMANCE TEST SET	32
A.3 USE OF CONFORMANCE TESTING	35
APPENDIX B	36
RESULTS FOR PROVIDER 1F	36
APPENDIX C	41
RESULTS FOR PROVIDER 1O	41
APPENDIX D	46
RESULTS FOR PROVIDER 1T	46
APPENDIX E	51
RESULTS FOR PROVIDER 1Y	51

APPENDIX F	
RESULTS FOR PROVIDER 2B	56
APPENDIX G	
RESULTS FOR PROVIDER DERMALOG	61
APPENDIX H	
RESULTS FOR PROVIDER ID3	65
APPENDIX I	
RESULTS FOR PROVIDER PB	69
APPENDIX J	
RESULTS FOR PROVIDER R	73
APPENDIX K	
RESULTS FOR NFIQ 2 PARTICIPANTS	78
APPENDIX L	
IMPACT OF WSQ COMPRESSION ON THE NFIQ 2 VALUES	82
APPENDIX M	
SOFTWARE RELEASES	86
M.1 OPERATIONAL KIT	86
M.2 RESEARCH KIT	86

List of Figures

1	ILLUSTRATION OF BLOCK AND PIXEL INDEXING	4
2	EXAMPLE INPUT IMAGES FOR QUALITY COMPUTATIONS	4
3	PROCESSING STEPS OF FREQUENCY DOMAIN ANALYSIS (FDA)	7
4	PROCESSING STEPS OF LOCAL CLARITY SCORE (LCS)	9
5	PROCESSING STEPS OF ORIENTATION CERTAINTY LEVEL (OCL)	10
6	PROCESSING STEPS OF ORIENTATION FLOW (OFL)	11
7	PROCESSING STEPS OF RIDGE VALLEY UNIFORMITY (RVU)	12
8	PROCESSING STEPS OF THE REGION OF INTEREST (ROI) DETERMINATION ALGORITHM	15
9	NFIQ 2 FEATURE VALUES FOR IMAGES OF LOW AND HIGH UTILITY	19
10	TRAINING SET COMPOSITION	22
11	VALIDATION - RANDOM FOREST OUTPUT VS. COMPARISON SCORE	23
12	VALIDATION - CDF COMPARISON SCORE VS. QUALITY BIN	23
13	ERROR VS. REJECT CURVE - DATASET AZLA	25
14	ERROR VS. REJECT CURVE - DATASET AZLA	26
15	RANKED DET - PROVIDER 1O	28
16	COMPUTATION TIME	29
17	COMPUTATION TIME - LOGARITHMIC SCALE	30
18	EXAMPLE CONFORMANCE TEST SET IMAGES AND SCORES	35
19	RANKED DET - PROVIDER 1F	37
20	ERROR VS. REJECT CURVE - PROVIDER 1F	38
21	ERROR VS. REJECT CURVE $\leq 25\%$ - PROVIDER 1F	39
22	HEATMAP - PROVIDER 1F - DATASET AZLA	40
23	HEATMAP - PROVIDER 1F - DATASET DHS2	40
24	HEATMAP - PROVIDER 1F - DATASET POEBVA	40
25	RANKED DET - PROVIDER 1O	42
26	ERROR VS. REJECT CURVE - PROVIDER 1O	43
27	ERROR VS. REJECT CURVE $\leq 25\%$ - PROVIDER 1O	44
28	HEATMAP - PROVIDER 1O - DATASET AZLA	45
29	HEATMAP - PROVIDER 1O - DATASET DHS2	45

30	HEATMAP - PROVIDER 1O - DATASET POEBVA	45
31	RANKED DET - PROVIDER 1T	47
32	ERROR VS. REJECT CURVE - PROVIDER 1T	48
33	ERROR VS. REJECT CURVE $\leq 25\%$ - PROVIDER 1T	49
34	HEATMAP - PROVIDER 1T - DATASET AZLA	50
35	HEATMAP - PROVIDER 1T - DATASET DHS2	50
36	HEATMAP - PROVIDER 1T - DATASET POEBVA	50
37	RANKED DET - PROVIDER 1Y	52
38	ERROR VS. REJECT CURVE - PROVIDER 1Y	53
39	ERROR VS. REJECT CURVE $\leq 25\%$ - PROVIDER 1Y	54
40	HEATMAP - PROVIDER 1Y - DATASET AZLA	55
41	HEATMAP - PROVIDER 1Y - DATASET DHS2	55
42	HEATMAP - PROVIDER 1Y - DATASET POEBVA	55
43	RANKED DET - PROVIDER 2B	57
44	ERROR VS. REJECT CURVE - PROVIDER 2B	58
45	ERROR VS. REJECT CURVE $\leq 25\%$ - PROVIDER 2B	59
46	HEATMAP - PROVIDER 2B - DATASET AZLA	60
47	HEATMAP - PROVIDER 2B - DATASET DHS2	60
48	HEATMAP - PROVIDER 2B - DATASET POEBVA	60
49	RANKED DET - PROVIDER DERMALOG	62
50	ERROR VS. REJECT CURVE - PROVIDER DERMALOG	63
51	ERROR VS. REJECT CURVE $\leq 25\%$ - PROVIDER DERMALOG	64
52	RANKED DET - PROVIDER ID3	66
53	ERROR VS. REJECT CURVE - PROVIDER ID3	67
54	ERROR VS. REJECT CURVE $\leq 25\%$ - PROVIDER ID3	68
55	RANKED DET - PROVIDER PB	70
56	ERROR VS. REJECT CURVE - PROVIDER PB	71
57	ERROR VS. REJECT CURVE $\leq 25\%$ - PROVIDER PB	72
58	RANKED DET - PROVIDER R	74
59	ERROR VS. REJECT CURVE - PROVIDER R	75
60	ERROR VS. REJECT CURVE $\leq 25\%$ - PROVIDER R	76
61	HEATMAP - PROVIDER R - DATASET AZLA	77
62	HEATMAP - PROVIDER R - DATASET DHS2	77
63	HEATMAP - PROVIDER R - DATASET POEBVA	77
64	DET - DHS2	78
65	DET - POEBVA	79
66	DET - VISITIDF	80
67	DET - AZLA	81
68	WSQ COMPRESSION - FACTOR 4	82
69	WSQ COMPRESSION - FACTOR 8	83
70	WSQ COMPRESSION - AVERAGE RELATIVE ABSOLUTE DEVIATIONS	83
71	WSQ COMPRESSION ARTIFACTS - FACTOR 8	84
72	WSQ COMPRESSION ARTIFACTS - FACTOR 8 - GAMMA 0.1	85
73	WSQ COMPRESSION EFFECT - FACTOR 8	85
74	WSQ COMPRESSION EFFECT - FACTOR 8 - GAMMA 0.1	85
75	ARCHITECTURE OF THE NFIQ 2 FRAMEWORK	87

List of Tables

1	NFIQ 2 PROVIDERS	2
2	DATASETS USED TO TRAIN AND TEST NFIQ 2	3
3	CONFUSION MATRIX FROM TRAINING	20
4	RANDOM FOREST MEAN DECREASE GINI	21

5 NFIQ 2 CONFORMANCE TEST SET 32

Executive Summary

NFIQ 2 is a revision of the open source NIST Finger Image Quality NFIQ. In 2004 NIST developed the first publicly available fingerprint quality assessment tool NFIQ. The major innovation of NFIQ 2 was linking image quality to operational recognition performance. This had several immediate benefits; it allowed quality values to be tightly defined and then numerically calibrated. This, in turn, allowed for the standardization needed to support a worldwide deployment of fingerprint sensors with universally interpretable image qualities. Operationally, NFIQ 2 has increased the reliability, accuracy, and interoperability of fingerprint recognition systems by identifying the samples that are likely to cause recognition failure. Today, NFIQ is part of every worldwide large-scale biometric deployment, including the US-VISIT; the Federal Information Processing Standard (FIPS) 201, Personal Identity Verification (PIV) for Federal Employees and Contractors; the European Union-Schengen Visa Information System; Interpol; and the Unique Identification Authority of India.

Advances in fingerprint technology since 2004, necessitated an update to NFIQ. As such, development of NFIQ 2 was initiated in 2011 as collaboration between National Institute of Standards and Technology (NIST) and Federal Office for Information Security (BSI) and Federal Criminal Police Office (BKA) in Germany as well as research and development entities, MITRE, Fraunhofer IGD, Hochschule Darmstadt (h_da) and Secunet. NFIQ 2 provides a higher resolution quality score (in range of 0-100 according to the international biometric sample quality standard ISO/IEC 29794-1:2016 [8] as opposed to 1-5), lower computation complexity, as well as support for quality assessment in mobile platform. Furthermore, NFIQ 2 is the basis for a revision of the Technical Report ISO/IEC 29794-4 Biometric sample quality – Part 4: Finger image data [7] into an international standard. Specifically, NFIQ 2 quality features are being formally standardized as part of ISO/IEC 29794-4 Biometric sample quality – Part 4: Finger image data and NFIQ 2 source code serves as the reference implementation of the standard.

Release Notes

All NFIQ 2 related reports, drafts, announcements and news items may be found on the homepage <https://www.nist.gov/services-resources/software/nfiq-2>.

► Concept of Operations

NFIQ 2 is developed for images captured at 500 dpi and as such it shall not be used for images of different resolution, e.g. 1000 dpi.

NFIQ 2 is developed for plain impression captured using optical sensors or scanned from inked card. Therefore, it shall not be used for images captured using other capture technologies, e.g. capacitive.

NFIQ 2 employs FingerJet FX OSE minutia extractor which is an open source minutia extractor under the terms of the GNU Lesser General Public License as published by the Free Software Foundation, either version 3 of the License, or any later version, provided that the conditions specified in the COPYRIGHT.txt file provided with the software are met.

NFIQ 2 employs OpenCV's implementation of random forest under the terms of the Apache 2 license as specified in LICENSE file provided with the software. OpenCV version 4.5.2 or later is included in NFIQ 2 distribution.

► Source code

NFIQ 2 source code and user guide is publicly available from <https://github.com/usnistgov/NFIQ2>. The NFIQ 2 conformance test is available from <https://github.com/usnistgov/NFIQ2/tree/master/conformance> and documented in Appendix A.

► Supplemental reports

The following supplemental reports may follow:

Concept and baseline implementation for NFIQ lite.

NFIQ 2 Framework and Research kit.

Calibration of NFIQ 2 values including its mapping to NFIQ 1.0 values.

Performance of NFIQ 2 on rolled impressions.

Technical guidance on use of NFIQ 2 for quality assessment of slap impressions.

► Appendices

The main body of this document, describes the technical aspects of design, implementation and evaluation of NFIQ 2. The appendices consists of conformance test, provider specific results, and impact of wsq decoder on NFIQ 2 values.

► Algorithm identifiers

Throughout this report the implementations are identified by alphanumeric code. Table 1 shows the letters associated with the providers' names.

► Naming

The quality metrics and the methodology for combining these metrics into a unified quality score described in this report comprise the overall concept of NFIQ 2. The open source software released by NIST implementing the NFIQ 2 concept is given a software version number, starting with NFIQ v2.0. NIST first released NFIQ v2.0 in 2016 and has since released a major update to the codebase in June 2021, NFIQ v2.1. This report was written given the information available at the time of NFIQ v2.0.

► Contact: Correspondence regarding this report should be directed to TABASSI at NIST dot GOV.

Caveats

1. **Specific nature of the biometric data:** The absolute error rates quoted here were measured over a very large fixed corpus of operational finger images. The error rates measured here are realistic if the algorithms were applied to this kind of data. However, in other applications, the applicability of the results may differ due to a number of factors legitimately not reflected in the NFIQ 2 design. Among these are: images captured or scanned at resolutions other than 500 dpi, images captured using capture technologies other than live scan or scanned ink.

1 Introduction

Acquiring high quality finger images is a vital step to assure high recognition performance and user satisfaction. International Organization for Standardization/ International Electrotechnical Commission (ISO/IEC), subcommittee 37, working group 3 (ISO/IEC SC 37 WG 3) define quality as “degree to which a biometric sample fulfills specified requirements for a targeted application” [6]. It further discusses three components of quality:

Character An expression of quality based on the inherent properties of the source from which the biometric sample is derived. For example a scarred finger has a poor character,

Fidelity reflects the degree of the sample similarity to its source. Sample fidelity is comprised of fidelity components contributed by different processes, and

Utility An expression of quality based on utility reflects the predicted positive or negative contribution of an individual sample to the overall performance of a biometric system.

Utility-based quality is dependent on both the character and fidelity of a sample. Utility-based quality is intended to be more predictive of system performance, e.g. in terms of false match rate, false non-match rate, failure to enroll rate, and failure to acquire rate, than measures of quality based on character or fidelity alone. The term ‘quality’ should not be solely attributable to the acquisition settings of the sample, such as image resolution, dimensions in pixels, grey scale/colour bit depth, or number of features. Though such factors may affect sample utility and could contribute to the overall quality score.

Recognizing the negative impact of poor image quality on accuracy several large-scale biometric deployments, including the United States Visitor and Immigrant Status Indicator Technology (US-VISIT); the Federal Information Processing Standard (FIPS) 201, Personal Identity Verification (PIV) for Federal Employees and Contractors; the European Union-Schengen Visa Information System; Interpol; and the Unique Identification Authority of India mandates measuring and reporting finger image quality. Several finger image quality assessment algorithms have been documented in academic literatures, but their implementations are not readily available. Proprietary solutions are being offered by fingerprint technology providers. Proprietary solutions lacks uniform interpretation and therefore cannot guarantee interoperability.

NFIQ 2 fills this gap by providing a publicly available algorithm that quantifies the utility of a finger image. NFIQ 2 is a new and improved version of NFIQ 1.0 developed in consultation and collaboration with users and industry. NFIQ 2 is optimized for plain impressions captured at 500 dpi resolution using optical fingerprint scanners that use frustrated total internal reflection or scanned from inked-cards. It is developed in C/C++ and employs FingerJet FX OSE minutia extractor which is under GNU Lesser General Public License.

NFIQ 2 software reads a raw or WSQ compressed fingerprint image, computes a set of quality features from the image, and uses these features for predicting the utility of the image. The prediction is performed using a random forest classifier that is trained using comparison scores from several commercial fingerprint comparison algorithms from several operational fingerprint databases. NFIQ 2 software produces a quality score which is in conformance with the international biometric sample quality standard ISO/IEC 29794-1:2016 [8] is in [0-100], where 0 means no utility value and 100 is the highest utility value.

This document details the design, implementation and evaluation of NFIQ 2 “operational kit”. The source code can be obtained from <https://github.com/usnistgov/NFIQ2> as well as its documentation. The NFIQ 2 conformance test is available from <https://github.com/usnistgov/NFIQ2/tree/master/conformance> and documented in Appendix A.

Additionally, to facilitate and promote research in biometric sample quality, we may release NFIQ 2 “research kit” which includes source codes of the framework used during the development of the NFIQ 2 algorithm along with source code for computing features that were not selected for final NFIQ 2 “operational kit”.

Participant Name	Letter Code
Morpho	1F
NEC	1O
Neurotechnology	1T
L1 Identity	1Y
Morpho	2B
Cogent Systems	R
ID 3	id3
Precise Biometrics	pb
Dermalog	dermalog

Table 1: NFIQ 2 providers.

2 Data

Three sources of images were used in the development and evaluation of NFIQ 2: images from USG operational finger images, images from Federal Criminal Police Office (BKA) in Germany, and public datasets. All images were 8-bit grayscale. Images were previously compressed using Wavelet Scalar Quantization (WSQ) compression. Table 2 summarizes images used in training, validation and evaluation of NFIQ 2.

3 Algorithm submission and use

Participation in NFIQ 2 was open to any commercial, academic, or non-profit organization as well as individuals. The algorithm providers are listed in Table 1. The only necessary qualifications were those implied by the requirement to implement the interface given in the NFIQ 2 CALL FOR PARTICIPATION ¹. This necessitated only possession of fingerprint recognition algorithms and software engineering skills sufficient to implement specific C++ API calls and data structures. Naturally, the algorithms used for the study are dated by the time of development of the NFIQ 2.

Recognition accuracy of these implementations are reported in Appendix K.

¹<https://www.nist.gov/document/nfiq2callforparticipationv00pdf-0>

Name	Capture mode	Type	Number of subjects	Fingers	Number of comparisons per finger	Used for
AZLA	Scanned ink	Operational	240 000	Index and Thumb	120 000 mated. 120 000 non-mated	training + testing
FOEBVA	Live scan	Operational	180 000	Index	120 000 mated. 120 000 non-mated	training + testing
visiIDF	Live scan	Operational	220 000	Index and Thumb	95 000 mated. 12 000 non-mated	training + testing
DHS2	Live scan	Operational	180 000	Index	120 000 mated. 120 000 non-mated	training + testing
IQMI	Scanned ink	Operational	250 000	10 fingers	250 000 mated. 250 000 non-mated	testing
BKA	Live scan + Scanned ink	Operational	342 000 images	10 fingers	—	testing
SD 29	Scanned ink	Public	209	10 fingers	1 912 mated. 35 791 non-mated	testing
FVC 2000 Db1	Live scan	Public	110	8 fingers	—	conformance testing
FVC 2000 Db3	Live scan	Public	110	8 fingers	—	conformance testing
FVC 2002 Db1	Live scan	Public	110	8 fingers	—	conformance testing

Table 2: NFIQ 2 datasets.

4 Quality Features

This section describes NFIQ 2 features. We performed a comprehensive literature survey, and implemented a total of 155 of the quality features reported in the literature. The final fourteen NFIQ 2 features were selected as follows:

An initial list of candidate features was developed by removing features with low predictive power or computation time beyond NFIQ 2 computation time limitation. Then, we computed Spearman rank correlation between any pair of features to remove redundant features. Finally, variable importance parameter of random forest was used to further prune the list.

Source code and documentation for the features that were implemented and tested during the development of NFIQ 2 but did not make it to the final selection are included in the NFIQ 2 research kit.

Local vs. global features As it is conventional in fingerprint processing, some of these quality features analyze an image as a whole, or at global level, while others analyze an image at smaller region, or local level. Quality features operating on the local level may preserve spatial location and provide a quality map or histogram of locally computed values, thereby yielding a more nuanced assessment. The common approach to compute local level information is to subdivide the image into blocks as illustrated in fig. 1.

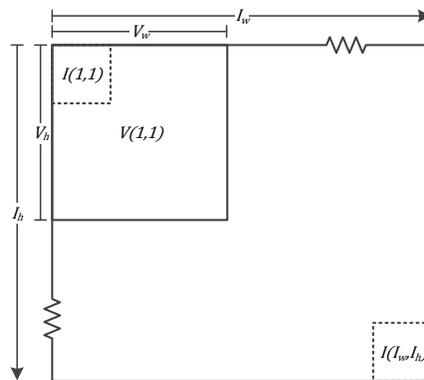


Figure 1: Illustration of block and pixel indexing within an image I with dimensions I_w, I_h . Shown is the pixel $I(1,1)$, the block $V(1,1)$, with dimensions V_w, V_h

For algorithms operating in a block-wise manner the input image is subdivided into blocks according to the overlay grid shown in fig. 2b. The block $V(10,7)$ is used as example in local processing and is marked up using a bold line. Figure 2c shows an enlarged view of $V(10,7)$ and fig. 2d shows $V(10,7)$ rotated according to ridge orientation its dominant ridge orientation as determined using eq. (10).

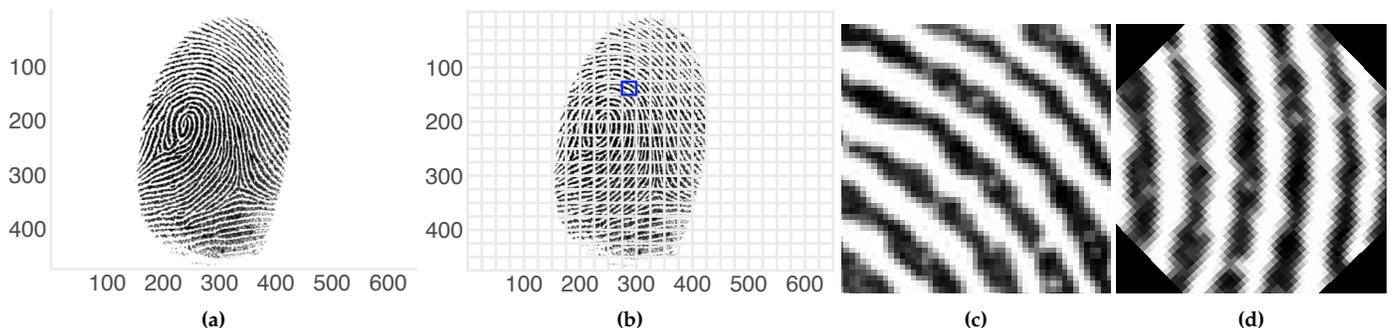


Figure 2: Input image used examples of the processing of quality. Input shown in (a) and division into blocks shown in (b). (c) shows an enlarged view of $V(10,7)$ and (d) shows $V(10,7)$ rotated according to ridge orientation its dominant ridge orientation as determined using eq. (10). Fingerprint image sourced from NFIQ 2 Conformance Test Set, subset from NIST Special Databases 300 and 302 [3, 4].

Computing the block orientation from gradients From a single block representing a local region of a fingerprint image, the dominant ridgeflow orientation is determined by computing the gradient information and then determining the orientation of the principal variation axis.

The numerical gradient of the block is determined using finite central difference for all interior pixels in x-direction eq. (1) and y-direction eq. (2).

$$\mathbf{f}_x = \frac{\mathbf{I}(x+1, y) - \mathbf{I}(x-1, y)}{2} \quad (1)$$

$$\mathbf{f}_y = \frac{\mathbf{I}(x, y+1) - \mathbf{I}(x, y-1)}{2}. \quad (2)$$

With \mathbf{f}_x and \mathbf{f}_y the principal axis of variation of \mathbf{V} is determined analytically using the sine and cosine doubled angles determined from the arithmetic means of the image gradient covariances eq. (10).

$$a = \overline{\mathbf{f}_x^2} \quad (3)$$

$$b = \overline{\mathbf{f}_y^2} \quad (4)$$

$$c = \overline{\mathbf{f}_x \cdot \mathbf{f}_y} \quad (5)$$

$$\mathbf{C} = \begin{bmatrix} a & c \\ c & b \end{bmatrix} \quad (6)$$

$$d = \sqrt{c^2 + (a-b)^2 + \epsilon} \quad (7)$$

$$\theta_{sin} = \frac{c}{d} \quad (8)$$

$$\theta_{cos} = \frac{a-b}{d} \quad (9)$$

$$angle(\mathbf{V}) = \frac{atan2(\theta_{sin}, \theta_{cos})}{2}. \quad (10)$$

Prior to computing features, fingerprint images are cropped to remove white pixels on the margins according to algorithm 1. A description of the process follows.

4.1 Segmenting the fingerprint image

Prior to computing features, fingerprint images are cropped to remove white pixels on the margins. Starting from the outer margins, rows and columns with average pixel intensity of 250 were removed.

The input fingerprint image is segmented to reduce the amount of necessary computations for individual quality features.

The first step in the segmentation is to remove areas of the input image which are near constant.

Pixel intensities take values $[0, 255]$ for an 8-bit gray scale image. As a first approximation of the region of interest, image columns and rows which are near constant white background are removed. Using the algorithm specified in algorithm 1 we set a fixed threshold for gray scale pixel intensity of $T_\mu = 250$ to obtain the image without near constant areas.

For quality features which require a foreground mask to indicate regions containing the fingerprint an algorithm using local standard deviation is adopted [10] (see algorithm 2).

4.2 Frequency Domain Analysis

The *fda* algorithm operates in a block-wise manner. A one-dimensional signature of the ridge-valley structure is extracted and the Discrete Fourier Transform (dft) is computed on the signature to determine the frequency of the sinusoid following

Algorithm 1: Removal of near constant white lines in image.

Input: Fingerprint image \mathbf{I} , Threshold T_μ
Output: Fingerprint image without near constant white areas $\widehat{\mathbf{I}}$

- 1 **for** each row R_i in \mathbf{I} , starting from the top **do**
- 2 Compute the row arithmetic mean μ_{row}
- 3 On the first occurrence where $\mu_{row} \leq T_\mu$ set $idx_t = i$
- 4 On the last occurrence where $\mu_{row} \leq T_\mu$ set $idx_b = i$
- 5 **end**
- 6 **for** each column C_i in \mathbf{I} , starting from the left **do**
- 7 Compute the column arithmetic mean μ_{col}
- 8 On the first occurrence where $\mu_{col} \leq T_\mu$ set $idx_l = i$
- 9 On the last occurrence where $\mu_{col} \leq T_\mu$ set $idx_r = i$
- 10 **end**
- 11 Extract the region of interest as $\widehat{\mathbf{I}} = \mathbf{I.roi}(idx_l, idx_t, idx_r, idx_b)$

Algorithm 2: Foreground segmentation based on local standard deviation.

Input: Fingerprint image \mathbf{I} , Threshold T
Output: Fingerprint foreground mask \mathbf{I}_{mask}

- 1 Normalize \mathbf{I} to zero mean, unit standard deviation to produce $\widehat{\mathbf{I}}$
- 2 **for** each block \mathbf{V} in $\widehat{\mathbf{I}}$ **do**
- 3 Compute the standard deviation of \mathbf{V} as σ_V
- 4 Mark the corresponding block in \mathbf{I}_{mask} as foreground if $\sigma_V > T$
- 5 **end**

the ridge-valley structure [11, 7].

A visual overview of the algorithm outputs are depicted in fig. 3.

Algorithm 3: fda algorithm

Input: Fingerprint image \mathbf{I}
Output: fda matrix of local quality scores \mathbf{Q}_{FDA}^{local}

- 1 **for** each block \mathbf{V} in \mathbf{I} **do**
- 2 pad \mathbf{V} with 2 pixel around border
- 3 rotate \mathbf{V} with nearest neighbour interpolation such that dominant ridge flow is perpendicular to x-axis
- 4 crop \mathbf{V} such that no invalid regions are included
- 5 with \mathbf{V} obtain the ridge-valley signature \mathbf{S} (eq. (11))
- 6 compute the dft of \mathbf{S} to obtain the magnitude representation \mathbf{A}
- 7 discard the first component of \mathbf{A}
- 8 determine F_{max} as the index of the largest magnitude in \mathbf{A}
- 9 compute \mathbf{Q}_{FDA}^{local} of \mathbf{V} using \mathbf{A} and F_{max} (eq. (12))
- 10 **end**

Ridge-valley signature The ridge-valley signature is a projection of the mean values of the local region along the y-axis onto a 1 dimensional vector. This effectively gives an approximated representation of the fundamental periodicity within the local region. The signature is computed as:

$$\mathbf{S} = \frac{1}{V_h} \sum_{k=1}^{V_h} \mathbf{V}(x, k), \quad (11)$$

where V_h is the height of \mathbf{V} .

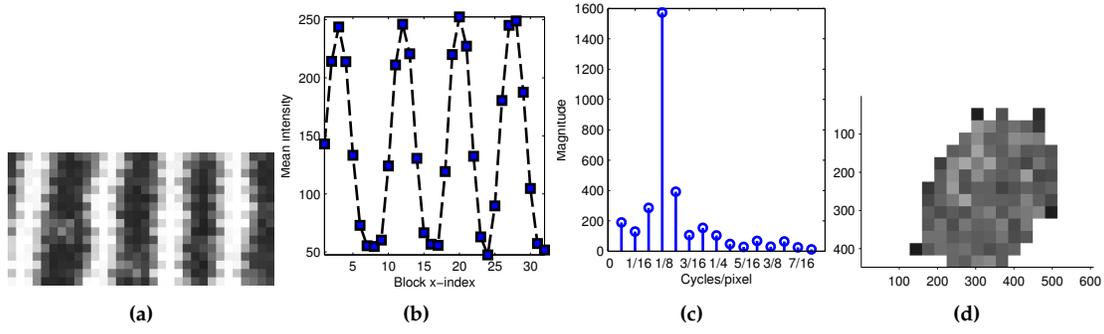


Figure 3: Processing steps of fda quality algorithm. (a) central area of input block; (b) ridge-valley profile; (c) dft of ridge-valley profile; (d) map of Q_{FDA}^{local} .

Computing the local Frequency Domain Analysis quality score The local fda quality score, Q_{FDA}^{local} is computed as:

$$Q_{FDA}^{local} = \begin{cases} \frac{\mathbf{A}_{F_{max}} + C(\mathbf{A}_{F_{max}-1} + \mathbf{A}_{F_{max}+1})}{\sum_{F=1}^{|A|/2} A_F} & , \text{if } F_{max} = \mathbf{A}_1 \text{ or } F_{max} = \mathbf{A}_{|A|} \\ 1 & , \text{otherwise} \end{cases} \quad (12)$$

where $C = 0.3$ according to the definition appearing in ISO/IEC TR 29794-4:2010. The effect of the constant is to retain an attenuated amplitude of the frequency bands immediately surrounding the band with the maximum amplitude $\mathbf{A}_{F_{max}}$.

The value of Q_{FDA}^{local} is set to 1 when maximum frequency amplitude bin is $F_{max} = \mathbf{A}_1$ or $F_{max} = \mathbf{A}_{|A|}$ as both positions will result in accessing out of bounds elements in \mathbf{A} due to the use of attenuated neighboring bins.

4.3 Local Clarity Score

lcs [2, 7] computes the block-wise clarity of ridge and valleys by applying linear regression to determine a gray-level threshold, classifying pixels as ridge or valley. A ratio of misclassified pixels is determined by comparing with the normalized ridge and valley width of that block.

Particular regions inherent in a fingerprint will negatively affect Q_{LCS} . For example, ridge endings and bifurcations or areas with high curvature such as those commonly found in the vicinity of core and delta points.

A visual overview of the algorithm outputs are depicted in fig. 4.

Algorithm 4: Local Clarity Score algorithm

Input: Fingerprint image \mathbf{I}

Output: lcs matrix of local quality scores Q_{LCS}^{local}

- 1 **for each block \mathbf{V} in \mathbf{I} do**
 - 2 rotate \mathbf{V} such that dominant ridge flow is perpendicular to x-axis
 - 3 crop \mathbf{V} such that no invalid regions are included
 - 4 with \mathbf{V} obtain the ridge-valley signature \mathbf{S} (eq. (11))
 - 5 Determine DT using linear regression on \mathbf{S}
 - 6 For each element $\mathbf{S}(x)$ set threshold $\mathbf{T}(x)$ of x being ridge or valley based on DT
 - 7 Classify columns in \mathbf{V} as ridge (1) or valley (0) with $\mathbf{P}(x) = \begin{cases} 1, & \text{if } \mathbf{S}(x) < \mathbf{T}(x) \\ 0, & \text{otherwise} \end{cases}$
 - 8 Determine ridge-valley transition vector \mathbf{C} from \mathbf{P}
 - 9 Compute the vector \mathbf{W} containing ridge and valley widths from \mathbf{C}
 - 10 Determine normalized ridge width and valley width \bar{W}_r and \bar{W}_v (eqs. (15) and (16))
 - 11 Determine proportion of misclassified pixel β and α in the ridge and valley regions (eqs. (13) and (14))
 - 12 Compute the local quality score Q_{LCS}^{local} (eq. (19))
 - 13 **end**
-

Determining the proportion of misclassified pixels For a block \mathbf{V} there are v_T pixels in the valley region and v_B pixels in the valley region with intensity lower than a threshold determined by DT . Similarly there are r_T pixels in the ridge region and r_B pixels in the ridge region with intensity lower than a threshold determined by DT . α and β are expressions of these ratios (eqs. (13) and (14)).

$$\alpha = \frac{v_B}{v_T} \quad (13)$$

$$\beta = \frac{r_B}{r_T} \quad (14)$$

Determining the normalized ridge and valley width The normalized valley width \bar{W}_v and the normalized ridge width \bar{W}_r are determined as

$$\bar{W}_v = \frac{W_v}{\left(\frac{res}{125}\right) W^{max}} \quad (15)$$

$$\bar{W}_r = \frac{W_r}{\left(\frac{res}{125}\right) W^{max}} \quad (16)$$

where res is the sensor resolution in dpi, W^{max} is the estimated ridge or valley width for an image with 125 dpi resolution, and W_v and W_r are the observed valley and ridge widths. According to [13] $W^{max} = 5$ is reasonable for 125 dpi resolution. For 500 dpi resolution, eqs. (15) and (16) may be expressed as

$$\hat{W}_v = \frac{W_v}{20} \quad (17)$$

$$\hat{W}_r = \frac{W_r}{20} \quad (18)$$

Computing the Local Clarity Score The local quality score \mathbf{Q}_{LCS}^{local} is computed using the average value of α and β in valid ridge and valley regions:

$$\mathbf{Q}_{LCS}^{local} = \begin{cases} 1 - \frac{\alpha + \beta}{2} & , \text{if } \left(W_v^{nmin} < \bar{W}_v < W_v^{nmax} \right) \left(W_r^{nmin} < \bar{W}_r < W_r^{nmax} \right) \\ 0 & , \text{otherwise} \end{cases} \quad (19)$$

where W_r^{nmin} and W_v^{nmin} are the minimum values for the normalized ridge and valley width, and W_r^{nmax} and W_v^{nmax} are the maximum values for the normalized ridge and valley width.

4.4 Orientation Certainty Level

ocl [12, 7] is a measure of the strength of the energy concentration along the dominant ridge flow orientation. The feature operates in a block-wise manner.

The computation of ocl presented here deviates from ISO/IEC 29794-4:2010 [7] in that we subtract the ratio between the eigen values from 1 such that $\mathbf{Q}_{OCL}^{local}(i, j) = 0$ reflects the lowest local orientation certainty and $\mathbf{Q}_{OCL}^{local}(i, j) = 1$ the highest local orientation certainty.

A visual overview of the algorithm outputs are depicted in fig. 5.

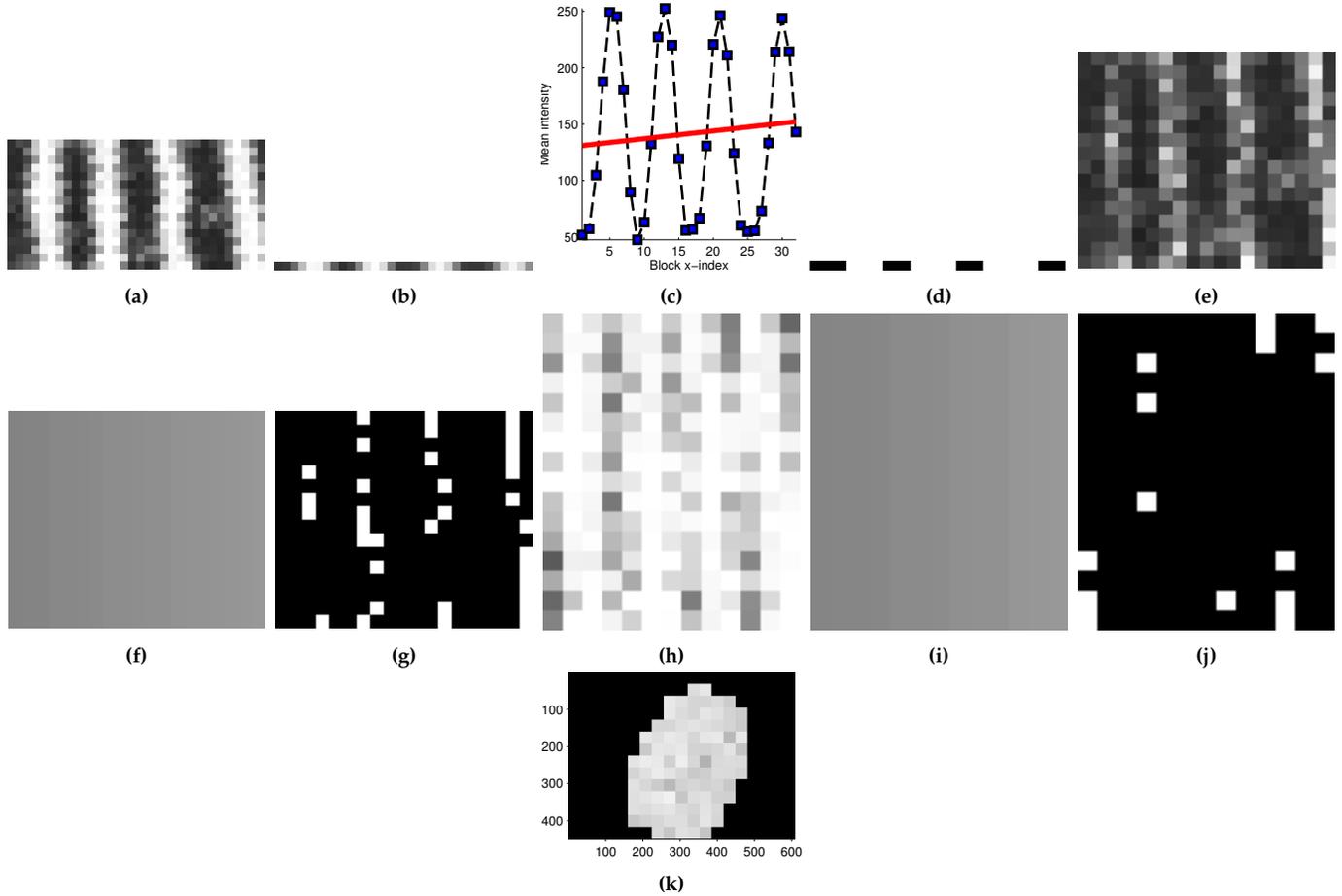


Figure 4: Processing steps of Local Clarity Score algorithm. (a) crop of current block; (b) average profile of block; (c) average block profile with linear regression line; (d) binarisation mask with ridge and valley regions based on regression line; (e) pixels determined to be ridge based on mask; (f) threshold based on DT across the block; (g) pixel misclassified as valley based on the threshold; (h), (i), (j) the same as (e), (f), (g) but for valley region; (k) local clarity scores.

Algorithm 5: ocl algorithm

Input: Fingerprint image I
Output: ocl matrix of local quality scores Q_{OCL}^{local}

- 1 **for** each block V in I **do**
- 2 Compute the intensity gradient with centered differences method
- 3 Compute the covariance matrix C
- 4 Compute the eigenvalues of C to obtain Q_{OCL}^{local} (eqs. (20) and (22))
- 5 **end**

Computing the eigenvalues and local orientation certainty From the covariance matrix C the eigenvalues λ_{min} and λ_{max} are computed as

$$\lambda_{min} = \frac{a + b - \sqrt{(a - b)^2 + 4c^2}}{2} \quad (20)$$

$$\lambda_{max} = \frac{a + b + \sqrt{(a - b)^2 + 4c^2}}{2} \quad (21)$$

this yields a local orientation certainty level Q_{OCL}^{local} :

$$Q_{OCL}^{local} = \begin{cases} 1 - \frac{\lambda_{min}}{\lambda_{max}} & , \text{if } \lambda_{max} > 0 \\ 0 & , \text{otherwise} \end{cases} \quad (22)$$

which is a ratio in the range 0 to 1 where 1 and 0 respectively is highest and lowest orientation certainty level.

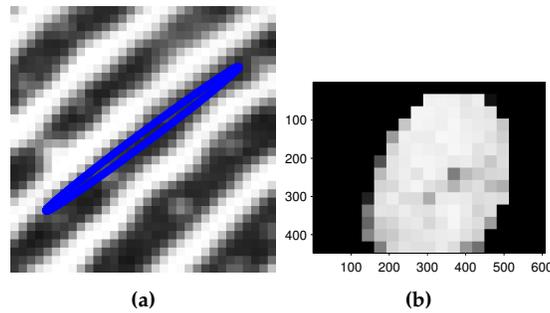


Figure 5: Processing steps of Orientation Certainty Level quality algorithm. (a) current block with ration between eigen values marked as ellipse; (b) local quality scores.

4.5 Orientation Flow

ofl [2, 7] is a measure of ridge flow continuity which is based on the absolute orientation difference between a block and its 8-neighborhood.

In ISO/IEC TR 29794-4:2010 [7] the parameter θ_{min} is a constant for angular tolerance between dominant orientation of neighboring blocks. We set ($\theta_{min} = 4$).

A visual overview of the algorithm outputs are depicted in fig. 6.

Absolute orientation difference The ridge flow is determined as a measure of the absolute difference between a block and its neighboring blocks. The absolute difference for block $V(i, j)$ is:

Algorithm 6: ofl algorithm

Input: Fingerprint image I
Output: ofl matrix of local quality scores Q_{OFL}^{local}

- 1 Determine the dominant ridgeflow orientation O of blocks V in I
- 2 **for each block V in I do**
- 3 Compute the absolute orientation difference $D(i, j)$ using orientations O (eq. (23))
- 4 Compute the local orientation quality score $Q_{OFL}^{local}(i, j)$ (eq. (24))
- 5 **end**

$$D(i, j) = \frac{\sum_{m=-1}^1 \sum_{n=-1}^1 |O(i, j) - O(i - m, j - n)|}{8}. \quad (23)$$

Local orientation quality score The local orientation quality score $Q_{OFL}^{local}(i, j)$ for the block orientation difference $D(i, j)$ is determined as:

$$Q_{OFL}^{local}(i, j) = \begin{cases} \frac{D(i, j) - \theta_{min}}{90 \text{ deg} - \theta_{min}} & , \text{ if } D(i, j) > \theta_{min} \\ 0 & , \text{ otherwise} \end{cases} \quad (24)$$

where θ_{min} is the threshold for the minimum angle difference to be considered as significant.

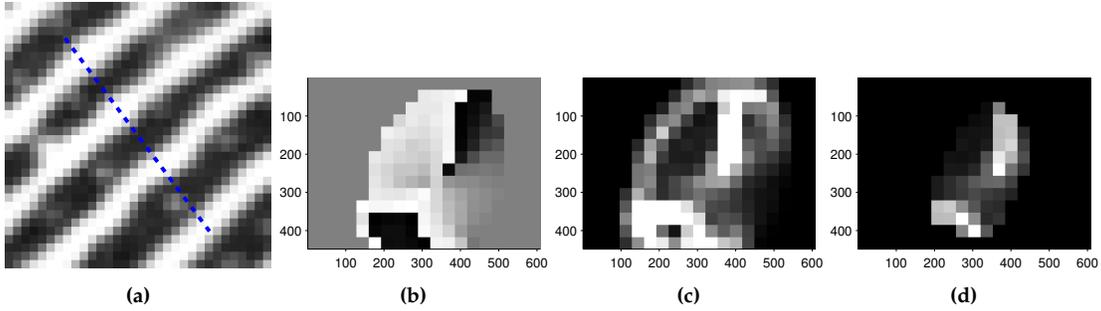


Figure 6: Processing steps of Orientation Flow quality algorithm. (a) line marking the normal to the ridgeline orientation; (b) local orientations; (c) orientation differences; (d) local quality scores.

4.6 Ridge Valley Uniformity

Ridge Valley Uniformity is a measure of the consistency of the ridge and valley widths [12, 7]. The expectation for a finger image with clear ridge and valley separation is that the ratio between ridge and valley widths remains fairly constant and thus the standard deviation of ratios is used as an indication of the sample quality. The Ridge Valley Uniformity quality feature is resolution dependent.

A visual overview of the algorithm outputs are depicted in fig. 7.

4.7 MU

The MU quality feature is the arithmetic mean of the gray scale input image.

Algorithm 7: rvu algorithm

Input: Fingerprint image I
Output: rvu matrix of local quality scores Q_{RVU}^{local}

- 1 **for** each block V in I **do**
- 2 Determine dominant ridgeflow orientation $\theta(V)$
- 3 Rotate V such that $\theta(V)$ is perpendicular to x -axis
- 4 crop V such that no invalid regions are included
- 5 with V obtain the ridge-valley signature S (eq. (11))
- 6 Determine DT using linear regression on S
- 7 For each $S(x)$ compute threshold $T(x) = x \times DT(1) + DT(0)$
- 8 Binarize S using T
- 9 Classify ridge and valley in S as $P(x) = \begin{cases} 1 & , \text{if } S(x) < T(x) \\ 0 & , \text{otherwise} \end{cases}$
- 10 Compute ridge-valley transition vector as $C(x) = \begin{cases} 1 & , \text{if } P(x-1) \neq P(x) \\ 0 & , \text{otherwise} \end{cases}$
- 11 Drop first and last transition from S using C to remove incomplete ridges or valleys and obtain S'
- 12 Compute Q_{RVU}^{local} as the ratio between widths of ridge and valleys in S'
- 13 **end**

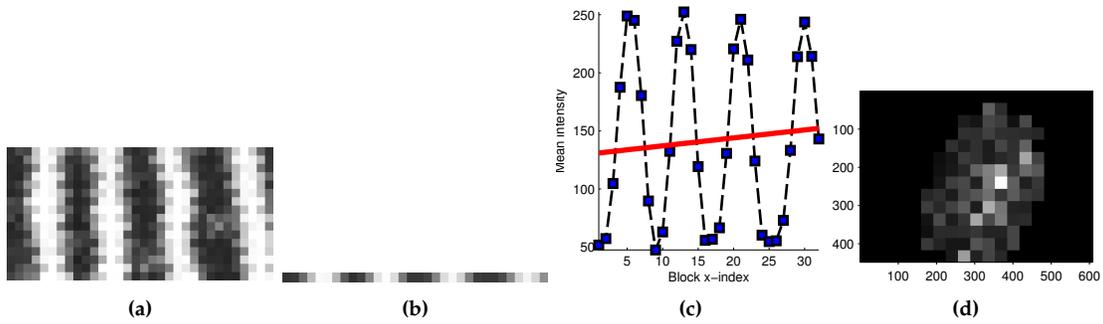


Figure 7: Processing steps of Ridge Valley Uniformity quality algorithm. (a) crop of current block; (b) average profile of block; (c) average profile with regression line; (d) local quality score as the standard deviation of local ridge to valley ratios.

4.8 MMB

The MMB quality feature is the arithmetic mean of per block computed arithmetic mean in the gray scale input image.

4.9 Minutiae Count

Using the (modified) FingerJet FX OSE (fjfx) minutiae extractor, we used two measures of minutia count:

- ▶ FingerJet FX OSE Total Minutiae Count (FingerJetFX_MinutiaeCount) counts the number of minutiae detected in the whole image.
- ▶ FingerJet FX OSE COM Minutiae Count (FingerJetFX_MinCount_COMMinRect200x200) counts the number of minutiae lying in a rectangle of 200×200 pixels centered at the center of mass (COM) of the locations of all detected minutiae.

The fjfx minutiae extractor was modified to remove the limitation of the number of output minutiae. The fjfx is an open source minutia extractor under the terms of the GNU Lesser General Public License as published by the Free Software Foundation, either version 3 of the License, or any later version, provided that the conditions specified in the COPYRIGHT.txt file provided with this software are met.

Algorithm 8: MU algorithm

Input: Fingerprint image I **Output:** MU quality score Q_{MU}

- 1 Compute Q_{MU} as the arithmetic mean of pixel values in I ;
-

Algorithm 9: MMB algorithm

Input: Fingerprint image I **Output:** MMB quality score Q_{MMB}

- 1 **for** each block V in I **do**
 - 2 | compute the arithmetic mean of pixels in V as Q_{MMB}^{local} ,
 - 3 **end**
 - 4 Compute Q_{MMB} as the arithmetic mean of set of Q_{MMB}^{local} ;
-

4.10 Minutiae Quality

This feature computes minutiae quality values for the minutiae detected by the (modified) `fjfx` minutiae extractor and outputs the arithmetic mean of all minutiae quality values. Two different methods for computing the minutiae quality are implemented resulting in two distinct features.

- Q_{MIN}^{mu} is computed as

$$Q_{MIN}^{mu} = (\mu_{Image}(I) - \mu_{Block}(I)) / \sigma_{Image}(I) \quad (25)$$

where I is the pixel intensity, μ and σ are the arithmetic mean and standard deviation, respectively, and the Block is of size 32×32 pixels and centered at the minutia location. The returned feature value is the percentage of these minutiae quality values between 0 and 0.5.

- Q_{MIN}^{ocl} computes the minutiae quality as the Orientation Certainty Level of the block of size 32×32 pixels centered at the minutia location. The returned feature value is the percentage of these minutiae quality values greater than 80.

4.11 ROI-based features

These features are based on a determination of a Region of Interest that is supposed to determine the foreground area of the fingerprint image, i.e., the image area, where the fingerprint is visible. The ROI is computed using algorithm 10.

Algorithm 10: ROI determination algorithm

Input: Grayscale fingerprint image I **Output:** Binarized image R specifying the ROI

- 1 Erode the fingerprint image I using a square of size 5×5 as structuring element
 - 2 Apply to the image resulting from the previous step a Gaussian blur filter with a kernel having height and width of 41 and standard deviation of 0
 - 3 Binarize the image resulting from the previous step using the threshold method of Otsu [15] (see description below)
 - 4 Apply to the image resulting from the previous step a Gaussian blur filter with a kernel having height and width of 91 and standard deviation of 0
 - 5 One again, binarize the image resulting from the previous step using the threshold method of Otsu [15]
 - 6 Find contours in the image, e.g., using the algorithm of Suzuki [16]
 - 7 Fill all white holes, i.e., regions completely surrounded by black pixels, by setting the pixels in that area to black
 - 8 Remove all black areas that reach the image border and are not the largest black area, by setting the pixels in that area to white
 - 9 The resulting binarized image R contains the ROI specified by all black pixels.
-

The processing of the ROI determination algorithm is illustrated with two example input images in fig. 8.

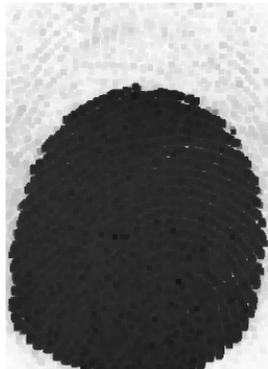
The binarization by the threshold method of Otsu [15] tries to find a threshold value t which minimizes the weighted



(a) Input image A



(b) Input image B



(c) Result of step 1 for input image A



(d) Result of step 1 for input image B



(e) Result of step 2 for input image A



(f) Result of step 2 for input image B



(g) Result of step 3 for input image A



(h) Result of step 3 for input image B

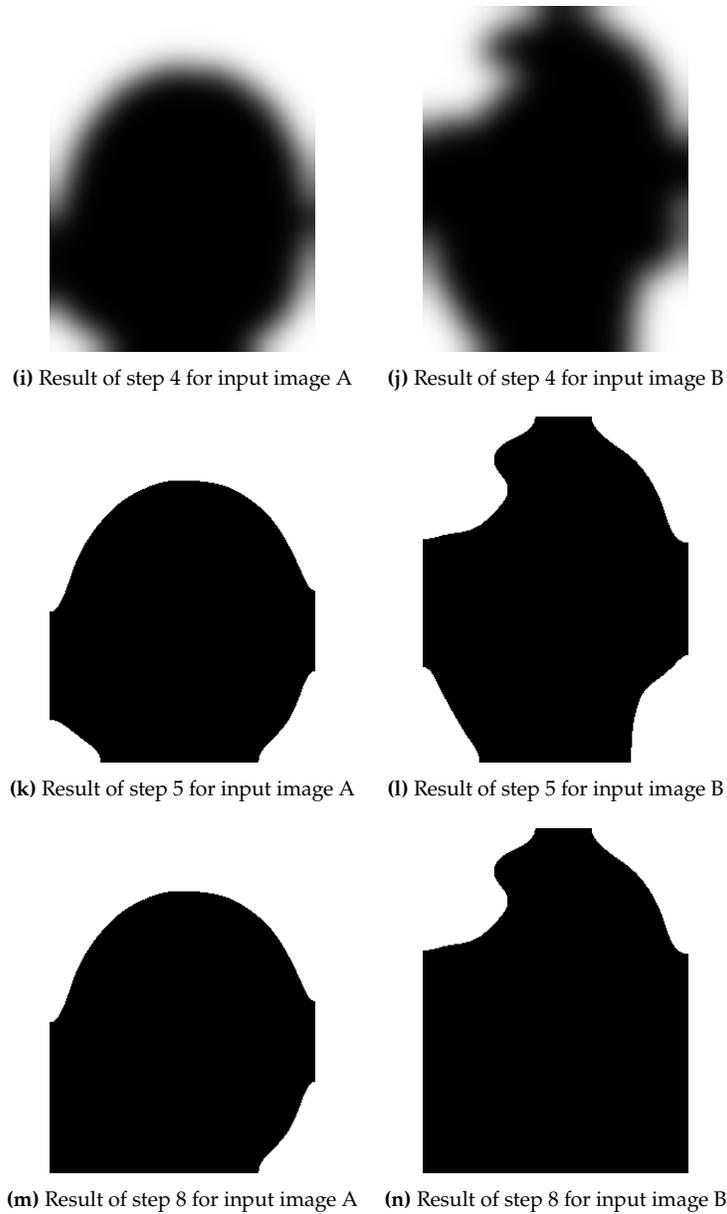


Figure 8: Processing steps of the ROI determination algorithm. Fingerprint images sourced from NFIQ 2 Conformance Test Set, subset from NIST Special Databases 300 and 302 [3, 4].

within-class variance of the grayscale intensity histogram ($P(0), \dots, P(255)$) given by

$$\sigma_w^2(t) = q_1(t)\sigma_1^2(t) + q_2(t)\sigma_2^2(t),$$

where $q_1, q_2, \sigma_1, \sigma_2$ are defined as

$$\begin{aligned} q_1(t) &= \sum_{i=1}^t P(i), & \text{and} & & q_2(t) &= \sum_{i=t+1}^I P(i), \\ \mu_1(t) &= \sum_{i=1}^t \frac{iP(i)}{q_1(t)}, & \text{and} & & \mu_2(t) &= \sum_{i=t+1}^I \frac{iP(i)}{q_2(t)}, \\ \sigma_1^2(t) &= \sum_{i=1}^t [i - \mu_1(t)]^2 \frac{P(i)}{q_1(t)}, & \text{and} & & \sigma_2^2(t) &= \sum_{i=t+1}^I [i - \mu_2(t)]^2 \frac{P(i)}{q_2(t)}. \end{aligned}$$

4.11.1 ROI Area Mean

The feature Q_{AREA}^{μ} applies the ROI determination algorithm described in section 4.11 to determine the ROI, determines the ROI blocks as those image blocks of size 32×32 having at least one pixel in the ROI, and computes the quality value as the arithmetic mean of the grayscale intensity value of the pixels of all these ROI blocks.

4.11.2 ROI Orientation Map Coherence Sum

The feature $Q_{\text{COH}}^{\text{sum}}$ computes the coherence map of the orientation field estimation as specified in [9], and returns the sum of coherence values over all image blocks of size 16×16 in the ROI. The feature value is computed with algorithm 11, where the gradient field $\mathbf{g} = (g_x, g_y)^T$ of the pixel intensity $I(i, j)$ of \mathbf{I} is defined by

$$\begin{aligned} g_x(i, j) &= (I(i+1, j) - I(i-1, j))/2 & \text{for} & & 1 \leq i \leq \mathbf{I}_w - 1, 0 \leq j \leq \mathbf{I}_h \\ g_x(0, j) &= I(1, j) - I(0, j) & \text{for} & & 0 \leq j \leq \mathbf{I}_h \\ g_x(\mathbf{I}_w, j) &= I(\mathbf{I}_w, j) - I(\mathbf{I}_w - 1, j) & \text{for} & & 0 \leq j \leq \mathbf{I}_h \end{aligned} \quad (26)$$

and

$$\begin{aligned} g_y(i, j) &= (I(i, j+1) - I(i, j-1))/2 & \text{for} & & 0 \leq i \leq \mathbf{I}_w, 1 \leq j \leq \mathbf{I}_h - 1 \\ g_y(i, 0) &= I(i, 1) - I(i, 0) & \text{for} & & 0 \leq i \leq \mathbf{I}_w \\ g_y(i, \mathbf{I}_h) &= I(i, \mathbf{I}_h) - I(i, \mathbf{I}_h - 1) & \text{for} & & 0 \leq i \leq \mathbf{I}_w \end{aligned} \quad (27)$$

4.11.3 ROI Relative Orientation Map Coherence Sum

The feature $Q_{\text{COH}}^{\text{rel}}$ computes the coherence map of the orientation field estimation as specified in [9], and returns the average coherence values over all image blocks of size 16×16 in the ROI. The returned quality value $Q_{\text{COH}}^{\text{rel}}$ is computed as $Q_{\text{COH}}^{\text{rel}} = Q_{\text{COH}}^{\text{sum}}/n$, where n is the number of image blocks which contain at least one pixel in the ROI.

Algorithm 11: Algorithm for computing the ROI orientation map coherence sum feature value

Input: Grayscale fingerprint image \mathbf{I}
Output: ROI orientation map coherence sum feature value $Q_{\text{COH}}^{\text{sum}}$

- 1 Compute the gradient field $\mathbf{g} = (g_x, g_y)^T$ of the pixel intensity of \mathbf{I} as defined in equations (26) and (27)
- 2 Compute the (so-called) *squared gradient field* $\mathbf{g}_s = (g_x^2 - g_y^2, 2g_x g_y)^T$
- 3 Determine the ROI of \mathbf{I} using algorithm 10
- 4 **for** each 16×16 block \mathbf{V} in \mathbf{I} **do**
- 5 **if** Block \mathbf{V} contains at least one pixel of the ROI **then**
- 6 Compute the coherence of the orientation map of block \mathbf{V} as $\text{Coh}(\mathbf{V}) = |\sum g_s(i, j)| / \sum |g_s(i, j)|$, where the sums are taken over all pixels (i, j) in \mathbf{V} and $|\cdot|$ denotes the Euclidean norm
- 7 **else**
- 8 Set $\text{Coh}(\mathbf{V}) = 0$
- 9 **end**
- 10 **end**
- 11 Compute $Q_{\text{COH}}^{\text{sum}}$ as the sum of the coherence $\text{Coh}(\mathbf{V})$ over all blocks \mathbf{V}

4.12 Quality vectors from local qualities

Mean of local quality values The mean quality value Q_{QNAME}^{μ} over a $N \times M$ matrix of local quality values $\mathbf{Q}_{\text{QNAME}}^{\text{local}}$ is computed with eq. (28).

$$Q_{\text{QNAME}}^{\mu} = \frac{1}{N * M} \sum_{i=1}^N \sum_{j=1}^M \mathbf{Q}_{\text{QNAME}}^{\text{local}}, \quad (28)$$

where $qname$ is one of FDA, LCS, OCL, OFL, RVU.

Standard deviation of local quality values The standard deviation $Q_{\text{QNAME}}^{\sigma}$ over a $N \times M$ matrix of local quality values $\mathbf{Q}_{\text{QNAME}}^{\text{local}}$ is computed with eq. (29).

$$Q_{\text{QNAME}}^{\sigma} = \left(\frac{1}{N * M - 1} \sum_{i=1}^N \sum_{j=1}^M \left(\mathbf{Q}_{\text{QNAME}}^{\text{local}}(i, j) - Q_{\text{QNAME}}^{\mu} \right)^2 \right)^{\frac{1}{2}}, \quad (29)$$

where $qname$ is one of FDA, LCS, OCL, OFL, RVU.

Histogram of local quality For the quality algorithms FDA, LCS, OCL, OFL, RVU the local quality values are represented as a histogram in addition to the mean and standard deviations. The purpose of representing the local quality values as a histogram is to obtain a fuller representation of the distribution of local qualities while retaining a fixed-length feature

vector. The bin boundaries are determined empirically and specified in eqs. (30) to (34).

$$B_{\text{FDA}} = \{ -\infty, 0.26800, 0.30400, 0.33000, 0.35500, \\ 0.38000, 0.40700, 0.44000, 0.50000, 1.00000, \infty \}. \quad (30)$$

$$B_{\text{LCS}} = \{ -\infty, 0.00000, 0.70000, 0.74000, 0.77000, \\ 0.79000, 0.81000, 0.83000, 0.85000, 0.87000, \infty \}. \quad (31)$$

$$B_{\text{OCL}} = \{ -\infty, 0.33700, 0.47900, 0.57900, 0.65500, \\ 0.71600, 0.76600, 0.81000, 0.85200, 0.89800, \infty \}. \quad (32)$$

$$B_{\text{OFL}} = \{ -\infty, 0.01715, 0.03500, 0.05570, 0.08100, \\ 0.11500, 0.17180, 0.25690, 0.47580, 0.74800, \infty \}. \quad (33)$$

$$B_{\text{RVU}} = \{ -\infty, 0.50000, 0.66700, 0.80000, 1.00000, \\ 1.25000, 1.50000, 2.00000, 24.0000, 30.0000, \infty \}. \quad (34)$$

For each of FDA, LCS, OCL, OFL, RVU a histogram is computed using the specified bin boundaries where the i th bin in the histogram is given by the interval

$$[B_{\text{Q}}^i, B_{\text{Q}}^{i+1}), \quad 1 \leq i \leq |B_{\text{Q}}|. \quad (35)$$

The histograms of local qualities are specified according to their bin boundaries as defined in eqs. (36) to (40) where the i th bin in the histogram contains the cardinality of the multiset that contains values bounded by the histogram boundaries.

$$Q_{\text{FDA}}^i = |\{(x, y) \mid B_{\text{FDA}}^i \leq \mathbf{Q}_{\text{FDA}}^{\text{local}} < B_{\text{FDA}}^{i+1}\}|, \text{ for } 1 \leq i \leq |B_{\text{FDA}}|, \quad (36)$$

$$Q_{\text{LCS}}^i = |\{(x, y) \mid B_{\text{LCS}}^i \leq \mathbf{Q}_{\text{LCS}}^{\text{local}} < B_{\text{LCS}}^{i+1}\}|, \text{ for } 1 \leq i \leq |B_{\text{LCS}}|, \quad (37)$$

$$Q_{\text{OCL}}^i = |\{(x, y) \mid B_{\text{OCL}}^i \leq \mathbf{Q}_{\text{OCL}}^{\text{local}} < B_{\text{OCL}}^{i+1}\}|, \text{ for } 1 \leq i \leq |B_{\text{OCL}}|, \quad (38)$$

$$Q_{\text{OFL}}^i = |\{(x, y) \mid B_{\text{OFL}}^i \leq \mathbf{Q}_{\text{OFL}}^{\text{local}} < B_{\text{OFL}}^{i+1}\}|, \text{ for } 1 \leq i \leq |B_{\text{OFL}}|, \quad (39)$$

$$Q_{\text{RVU}}^i = |\{(x, y) \mid B_{\text{RVU}}^i \leq \mathbf{Q}_{\text{RVU}}^{\text{local}} < B_{\text{RVU}}^{i+1}\}|, \text{ for } 1 \leq i \leq |B_{\text{RVU}}|. \quad (40)$$

4.13 The NFIQ 2 feature vector

The NFIQ 2 feature vector is produced as a concatenation of individual quality features:

$$\mathbf{Q}_{\text{NFIQ 2}} = \left(Q_{\text{FDA}}^{\mu}, Q_{\text{LCS}}^{\mu}, Q_{\text{OCL}}^{\mu}, Q_{\text{OFL}}^{\mu}, Q_{\text{RVU}}^{\mu}, \right. \\ Q_{\text{FDA}}^{\sigma}, Q_{\text{LCS}}^{\sigma}, Q_{\text{OCL}}^{\sigma}, Q_{\text{OFL}}^{\sigma}, Q_{\text{RVU}}^{\sigma}, \\ \mathbf{Q}_{\text{FDA}}, \mathbf{Q}_{\text{LCS}}, \mathbf{Q}_{\text{OCL}}, \mathbf{Q}_{\text{OFL}}, \mathbf{Q}_{\text{RVU}}, \\ Q_{\text{MU}}, Q_{\text{MMB}}, Q_{\text{COH}}^{\text{rel}}, Q_{\text{COH}}^{\text{sum}}, Q_{\text{AREA}}^{\mu}, \\ \left. Q_{\text{MIN}}^{\text{cnt}}, Q_{\text{MIN}}^{\text{com}}, Q_{\text{MIN}}^{\text{mu}}, Q_{\text{MIN}}^{\text{ocl}} \right). \quad (41)$$

4.14 Predictive power of NFIQ 2 features

A useful feature will give statistically different values for images with low or high utility. Figure 9 shows NFIQ 2 feature values for images of low and high utility as defined in section 5.1. When the notches of the two boxes do not overlap this is strong evidence that their medians differ (Chambers et al., 1983, p. 62).

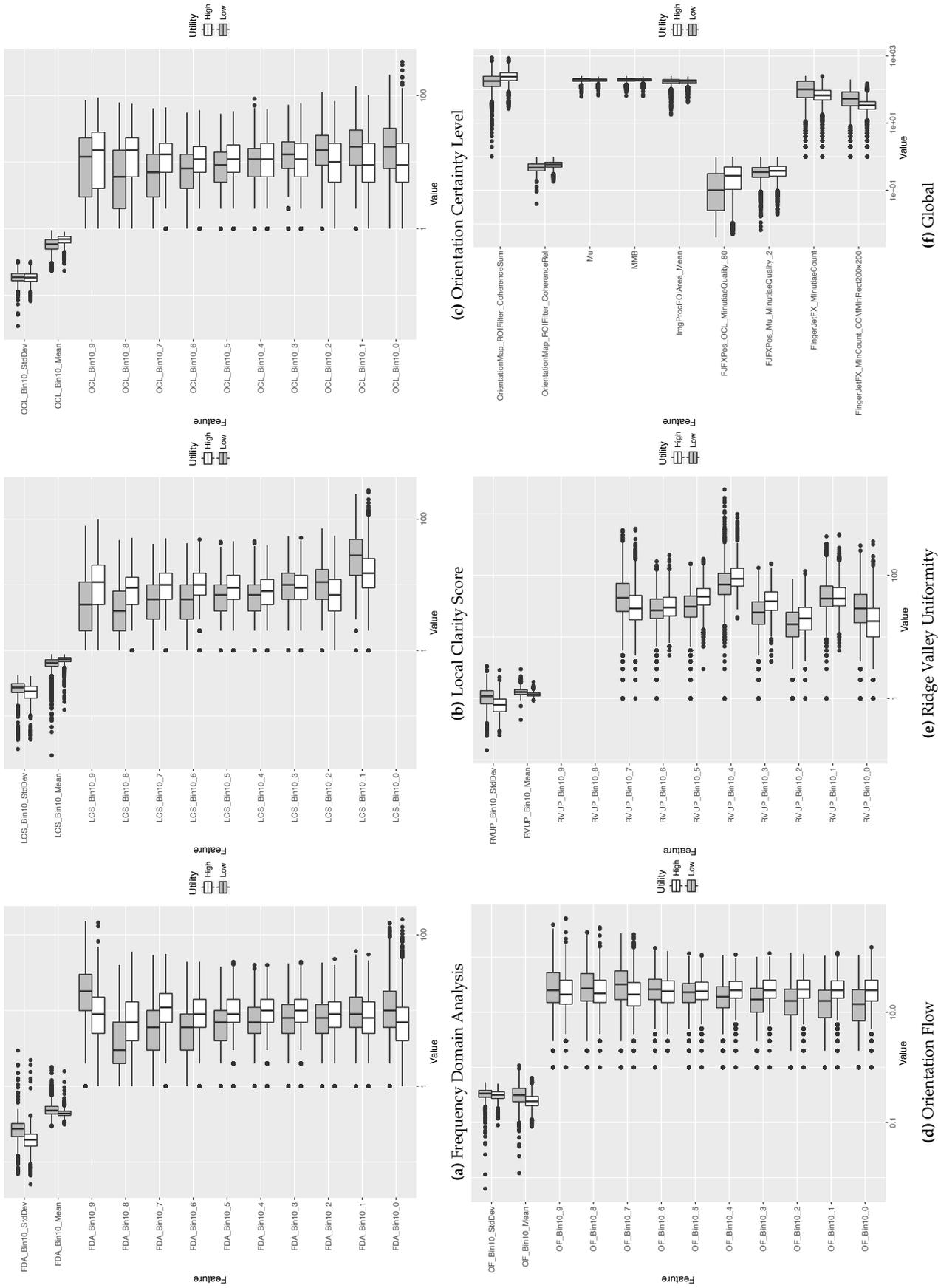


Figure 9: NFIQ 2 feature values for images of low and high utility. When the notches of the two boxes do not overlap this is strong evidence their medians differ (Chambers et al., 1983, p. 62).

5 Training

Training uses random forest for a binary classification: Class 0 represents images of very low utility and Class 1 represents images of very high utility. The trained random forest outputs class membership along with its probability. NFIQ 2 score is the probability that a given image belongs to class 1 multiplied by 100 and rounded to its closest integer.

Feature selection was done based on the “variable importance” parameter generated by random forest during the training. Starting with a very large number of the features, list of candidate features was pruned by iterative runs using only the most important variables from the previous run until the training stabilized.

5.1 Training data

Training set consist of 6 629 images (3 295 in Class 0 and 3 334 in Class 1) which are carefully selected from datasets AZLA, POEBVA and DHS2. Figure 10 shows the composition of training set. The selection rule was as follows:

- ▶ **Class 1 (or high utility)** consists of images with NFIQ 1.0 value of 1 (with activation score > 0.7) and genuine score in the 90th percentile for each of the NFIQ 2 providers.
- ▶ **Class 0 (or low utility)** consists of images with NFIQ 1.0 value of 5 (with activation score > 0.9) and genuine score smaller than a threshold value that corresponds to false match rate of 1 in 10 000, i.e. false reject at false match rate of 0.0001.

Furthermore, 99 797 images were randomly selected for model validation.

5.2 Training parameters and results

The two training parameters m (number of variables randomly sampled as candidates at each split), and N (number of trees to grow) were optimized by surveying over all feasible values. $N = 100$ and $m = 10$ gave the lowest out of bag error and therefore were selected as optimal values. The trained random forest has 100 trees and out of bag error of 0.24.

Training error and variable importance are shown in Tables 3 and 4 respectively.

	0	1	class error
0	3 295	0	0.000
1	0	3 334	0.000

Table 3: Confusion matrix from training. Row is predicted value, column is reference value.

	Name	MeanDreaseGini
Q_{FDA}^{σ}	Frequency Domain Analysis_Standard Deviation	140.760
Q_{MIN}^{COM}	FingerJet FX OSE COM Minutiae Count	92.089
Q_{MIN}^{ocl}	FingerJet FX OSE OCL MinutiaeQuality	83.027
Q_{RVU}^{μ}	Ridge Valley Uniformity_Mean	69.517
Q_{FDA}^{μ}	Frequency Domain Analysis_Mean	62.229
Q_{MIN}^{SUM}	FingerJet FX OSE Total Minutiae Count	57.565
Q_{RVU}^{σ}	Ridge Valley Uniformity_Standard Deviation	50.946
Q_{LCS}^7	Local Clarity Score_Bin_7	50.688
Q_{LCS}^8	Local Clarity Score_Bin_8	50.100
Q_{FDA}^9	Frequency Domain Analysis_Bin_9	47.844
Q_{COH}^{SUM}	ROI Orientation Map Coherence Sum	38.104
Q_{OFL}^2	Orientation Flow_Bin_2	37.172
Q_{LCS}^{μ}	Local Clarity Score_Mean	36.483
Q_{RVU}^5	Ridge Valley Uniformity_Bin_5	35.617
Q_{RVU}^3	Ridge Valley Uniformity_Bin_3	35.139
Q_{AREA}^{μ}	ROI Area Mean	34.932

Q_{OFL}^1	Orientation Flow_Bin_1	33.751
Q_{OFL}^0	Orientation Flow_Bin_0	33.513
Q_{MU}	MU	32.914
Q_{MMB}	MMB	32.625
Q_{MIN}^{mu}	FingerJet FX OSE Mu MinutiaeQuality	32.316
Q_{FDA}^8	Frequency Domain Analysis_Bin_8	31.428
Q_{FDA}^7	Frequency Domain Analysis_Bin_7	31.236
Q_{OFL}^{μ}	Orientation Flow_Mean	31.172
Q_{RVU}^4	Ridge Valley Uniformity_Bin_4	30.801
Q_{OCL}^{μ}	Orientation Certainty Level_Mean	30.035
Q_{OFL}^3	Orientation Flow_Bin_3	29.721
Q_{LCS}^{σ}	Local Clarity Score_Standard Deviation	28.777
Q_{ROI}^{rel}	ROI Relative Orientation Map Coherence Sum	28.700
Q_{OCL}^{σ}	Orientation Certainty Level_Standard Deviation	28.429
Q_{OFL}^{σ}	Orientation Flow_Standard Deviation	27.556
Q_{OCL}^8	Orientation Certainty Level_Bin_8	26.425
Q_{FDA}^6	Frequency Domain Analysis_Bin_6	25.161
Q_{LCS}^6	Local Clarity Score_Bin_6	23.837
Q_{OFL}^5	Orientation Flow_Bin_5	23.431
Q_{LCS}^9	Local Clarity Score_Bin_9	23.283
Q_{OFL}^4	Orientation Flow_Bin_4	22.883
Q_{RVU}^2	Ridge Valley Uniformity_Bin_2	22.843
Q_{RVU}^1	Ridge Valley Uniformity_Bin_1	22.380
Q_{OFL}^9	Orientation Flow_Bin_9	21.413
Q_{RVU}^6	Ridge Valley Uniformity_Bin_6	21.306
Q_{OFL}^7	Orientation Flow_Bin_7	21.296
Q_{OFL}^6	Orientation Flow_Bin_6	20.939
Q_{LCS}^1	Local Clarity Score_Bin_1	20.910
Q_{OFL}^8	Orientation Flow_Bin_8	20.867
Q_{RVU}^7	Ridge Valley Uniformity_Bin_7	20.798
Q_{RVU}^0	Ridge Valley Uniformity_Bin_0	20.470
Q_{OCL}^0	Orientation Certainty Level_Bin_0	20.234
Q_{OCL}^7	Orientation Certainty Level_Bin_7	19.887
Q_{OCL}^1	Orientation Certainty Level_Bin_1	19.434
Q_{OCL}^6	Orientation Certainty Level_Bin_6	19.188
Q_{LCS}^5	Local Clarity Score_Bin_5	19.073
Q_{OCL}^4	Orientation Certainty Level_Bin_4	18.865
Q_{LCS}^3	Local Clarity Score_Bin_3	18.861
Q_{OCL}^2	Orientation Certainty Level_Bin_2	18.663
Q_{FDA}^0	Frequency Domain Analysis_Bin_0	18.657
Q_{FDA}^5	Frequency Domain Analysis_Bin_5	18.438
Q_{OCL}^5	Orientation Certainty Level_Bin_5	18.262
Q_{OCL}^9	Orientation Certainty Level_Bin_9	18.164
Q_{LCS}^2	Local Clarity Score_Bin_2	18.136
Q_{FDA}^3	Frequency Domain Analysis_Bin_3	17.847
Q_{OCL}^3	Orientation Certainty Level_Bin_3	17.834
Q_{FDA}^1	Frequency Domain Analysis_Bin_1	17.710
Q_{LCS}^4	Local Clarity Score_Bin_4	17.598
Q_{FDA}^4	Frequency Domain Analysis_Bin_4	17.283
Q_{FDA}^2	Frequency Domain Analysis_Bin_2	17.145
Q_{RVU}^8	Ridge Valley Uniformity_Bin_8	0.000
Q_{RVU}^9	Ridge Valley Uniformity_Bin_9	0.000
Q_{LCS}^0	Local Clarity Score_Bin_0	0.000

Table 4: Random Forest Mean Decrease Gini. A low Gini (i.e. higher decrease in Gini) means that a particular predictor variable plays a greater role in partitioning the data into the defined classes.

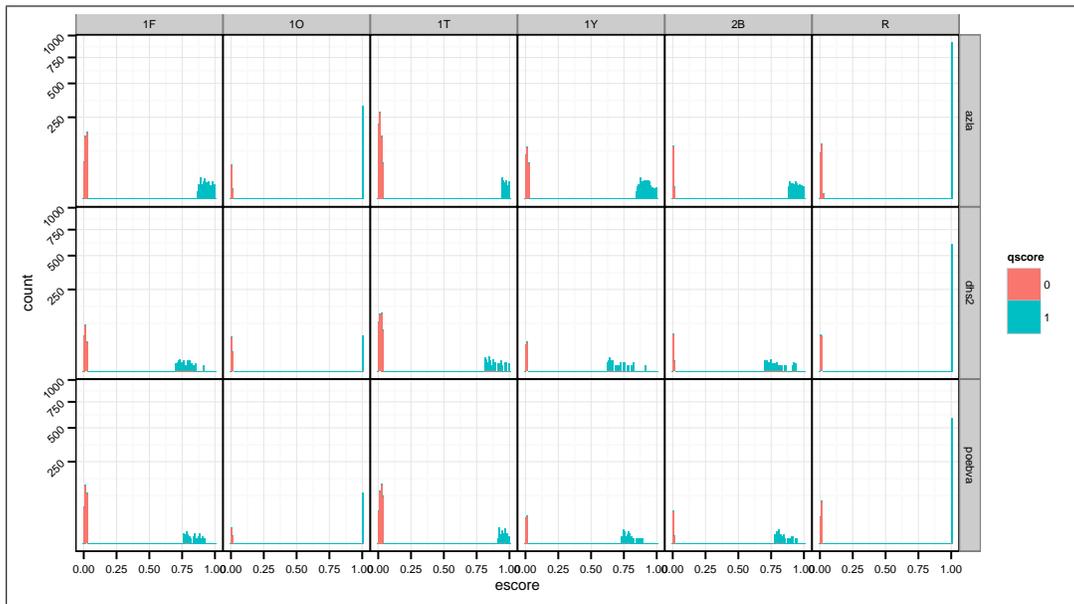


Figure 10: Training set composition: Histograms of sample counts according to provider and dataset, colored by ground truth class

5.3 Model Validation

Validation set comprised of 99 797 randomly selected images from the three datasets used for training. Validation set and training set are mutually exclusive. Figures 11 and 12 show the relationship of comparison scores and the output of the trained random forest for several comparison score providers and datasets. As expected, images with higher class 1 probability (i.e., higher NFIQ 2 scores), produce higher genuine comparison scores (see figure 11) and vice versa, images involved in higher genuine comparison scores have higher class 1 probability (i.e., higher NFIQ 2 scores).

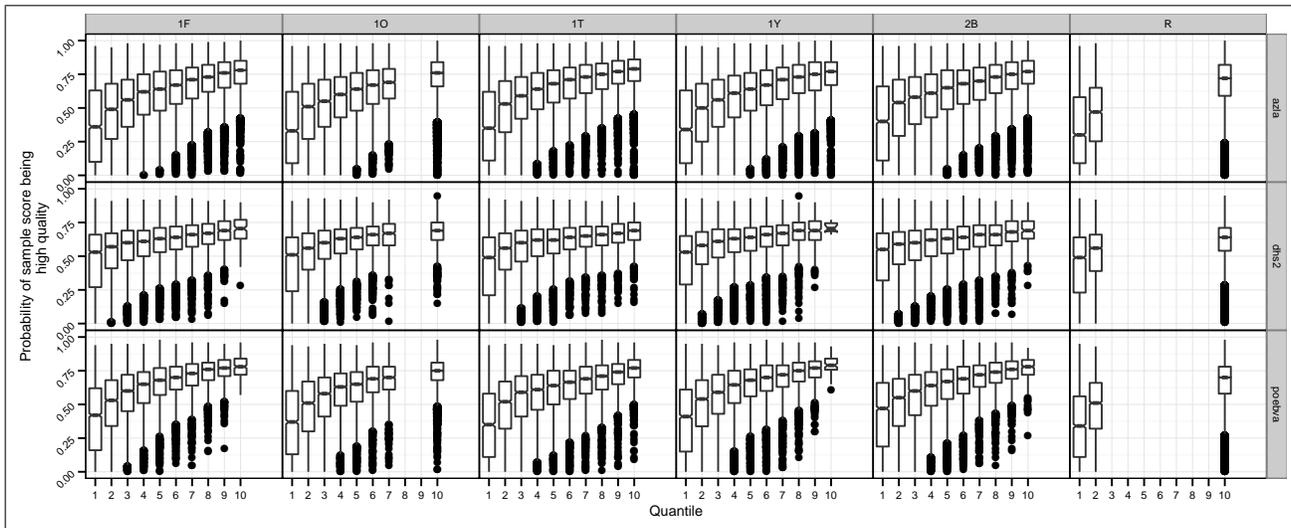


Figure 11: Boxplot of the output of random forest per quantized comparison score bin for the validation set.

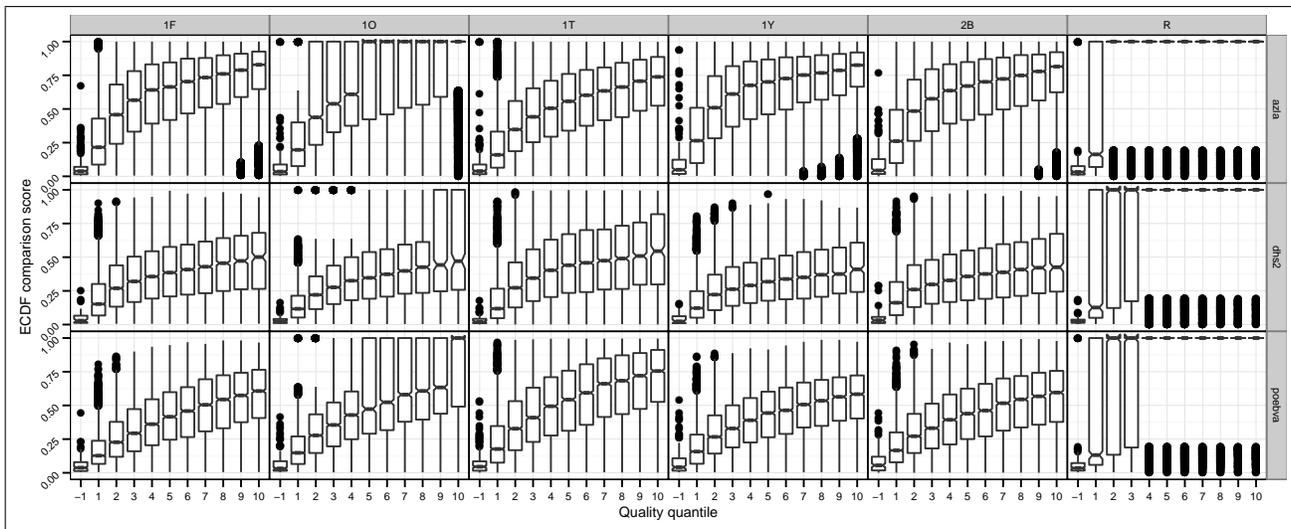


Figure 12: Boxplot of the CDF comparison score per quantile quality score bin for the validation set.

6 Predictive power of NFIQ 2

6.1 Rejection curves

One metric for comparative evaluation of quality assessment algorithms is the error versus reject curves [5]. The goal is to demonstrate how efficiently rejection of low-quality samples results in improved performance. This models the operational case in which quality is maintained by reacquisition after a low-quality sample is detected. Consider that a pair of samples (from the same subject) with qualities $q_i^{(1)}$ and $q_i^{(2)}$ are compared to produce a genuine score, and this is repeated for N such pairs.

We introduce thresholds u and v that define levels of acceptable quality and define the set of low-quality entries as

$$R(u, v) = \left\{ j : q_j^{(1)} < u, \quad q_j^{(2)} < v \right\} \quad (42)$$

We compute false non-match rate (FNMR) as the fraction of genuine scores above threshold (τ) computed for the samples *not* in this set

$$\text{FNMR}(\tau) = \frac{M_\psi(\tau)}{M_\psi(-\infty)} \quad (43)$$

where

$$M_\psi(\tau, u, v) = \sum_{s \in \mathcal{G} \cap R^C} H(s - \tau) \quad (44)$$

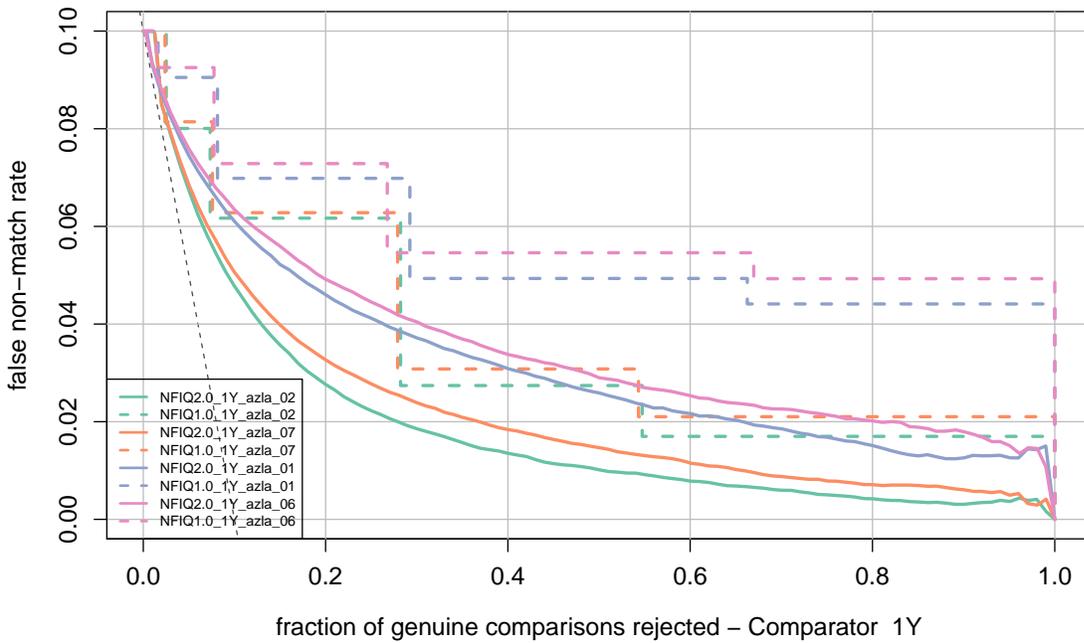
where R^C is the complement of R .

If the quality values are perfectly correlated with the genuine comparison scores, setting threshold τ to give an overall false non-match rate of x and then rejecting x percent with the lowest qualities should result in false non-match rate of zero after recomputing FNMR.

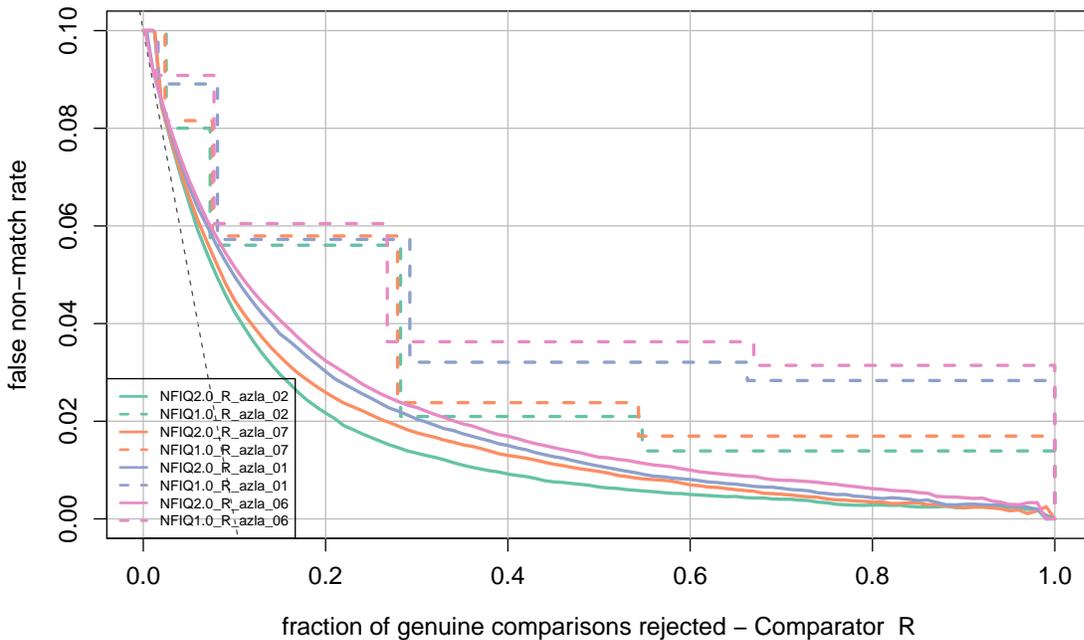
Figure 13 shows the error vs. reject curves for NFIQ 2 (solid lines) and NFIQ 1.0 (dotted lines) for images from AZLA dataset and comparison algorithms R and 1Y. Results for other datasets and comparison algorithms are in provider specific appendices.

We set the value of τ to give a false non-match rate of ten percent. u and v are varied to show the dependence of false non-match rate on NFIQ values. Pairwise quality is computed using the minimum quality of the two images being compared. For an effective quality assessment algorithm false non-match rate should decrease quickly with the fraction of comparisons rejected. Behavior of a perfect quality assessment algorithm is displayed by the gray dotted line where the rejection of the lowest 10% quality would result in an false non-match rate of zero. Both NFIQ 1.0 and NFIQ 2 trend in the correct direction (false non-match rate improves as more low-quality samples are rejected). NFIQ 2 is a better predictive of performance than NFIQ 1.0 because its error vs. reject curves are closer to the gray dotted line.

The most operationally relevant part of the error vs. reject curves is usually on the left side where a small fraction, x , of low-quality rejections would be tolerable from the perspective of forcing a second enrollment attempt. Figure 14 shows the error vs. reject curves for rejection rates of up to 25%.



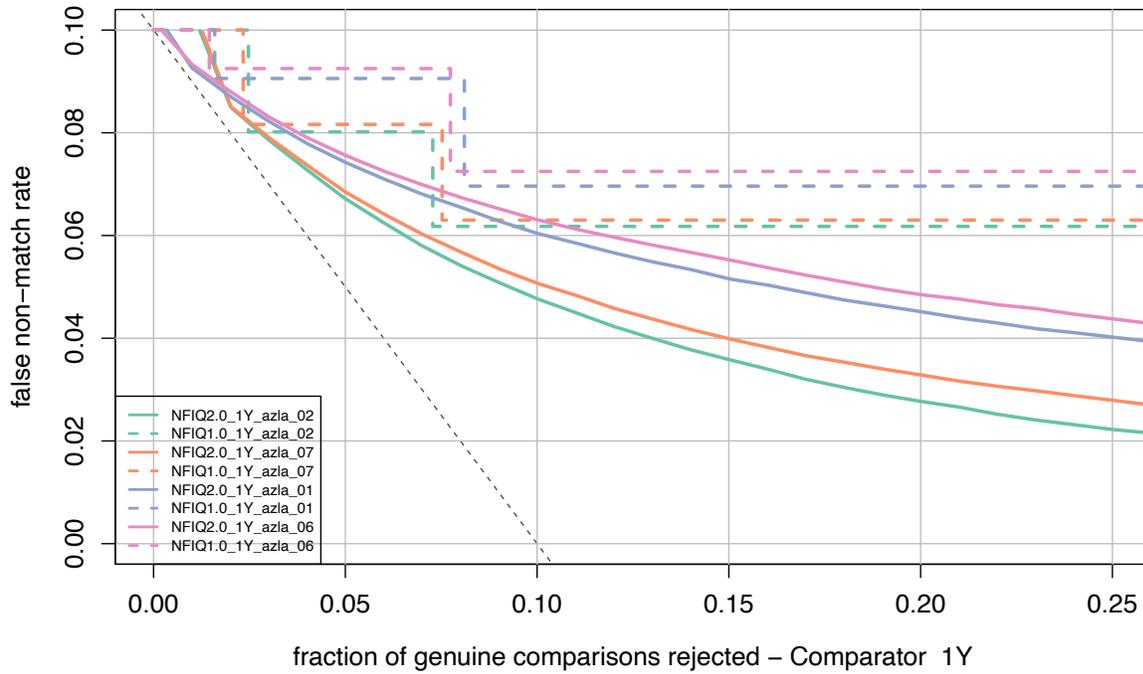
(a) Error vs. reject curve - Provider 1Y - Dataset AZLA



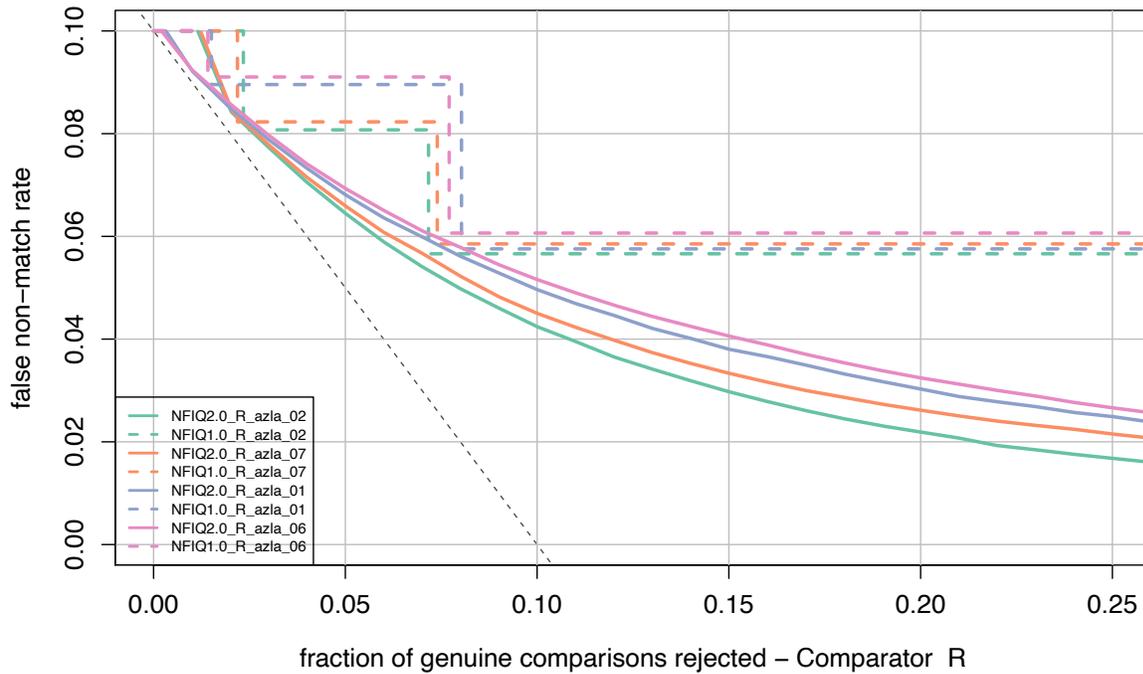
(b) Error vs. reject curve - Provider R - Dataset AZLA

Figure 13:

Error vs. reject curve - Dataset AZLA, Finger positions right index (02), left index (07), right thumb (01) and left thumb (06). The threshold is set to give an initial false non-match rate of 0.1. NFIQ 2 results are shown in solid lines and NFIQ 1.0 in dotted steps. The gray dotted line shows the ideal case where the rejection of the ten percent lowest quality results in zero false non-match rate. NFIQ 2 is a better predictive of performance than NFIQ 1.0 because its error vs. reject curves are closer to the gray dotted line.



(a) Error vs. reject curve $\leq 25\%$ - Provider 1Y - Dataset AZLA



(b) Error vs. reject curve $\leq 25\%$ - Provider R - Dataset AZLA

Figure 14:

Error vs. reject curve - Dataset AZLA, for rejection rate of smaller than 25%. Finger positions right index (02), left index (07), right thumb (01) and left thumb (06). The threshold is set to give an initial false non-match rate of 0.1. NFIQ 2 results are shown in solid lines and NFIQ 1.0 in dotted steps. The gray dotted line shows the ideal case where the rejection of the ten percent lowest quality results in zero false non-match rate. NFIQ 2 is a better predictive of performance than NFIQ 1.0 because its error vs. reject curves are closer to the gray dotted line.

6.2 Ranked DET

Detection error trade-off (DET) characteristic curves are the primary performance metric for offline testing of biometric recognition algorithms [14], [1]. Each point on a DET curve exhibits the false match and false non-match rates associated with a certain threshold value. The DET curve spans the whole range of possible threshold values, which is normally the range of the comparison scores. A quality assessment algorithm is useful if it can at least give an ordered indication of an eventual performance. For example, for L distinct quality levels, there should notionally be L DET characteristics that do not cross.

Using the minimum of the two samples ($\min(q_1, q_2)$) as their pairwise quality, we divide each comparator's comparison scores into three groups based on the pairwise quality of the images being compared. The set of the lowest quality contains comparison scores with pairwise qualities in the lowest 15 percentile. Comparisons with pairwise quality in the middle 70 percent comprise the second or medium quality set. Finally, comparison scores of images whose pairwise quality are in the highest 15 percentile make up the third or best quality set. Three DET characteristic curves, one for each set above, are generated, as shown in the following Figures. To reveal the dependence of false non-match and false match rates on quality at a fixed threshold, τ , the DET curves of each cell are connected at false non-match and false match rates that are observed at the "same threshold" values.

Figure 15 shows ranked DET results for provider 1o. Ranked DET for other providers are in provider specific appendices. The ranking and the separation of the DET curves in figure 15, as explained above, show that NFIQ 2 is predictive of the false non-match rates. Effect of quality on false match rate is demonstrated by the lines connecting the DET curves which are the brown lines in the figures. Assuming the correct ranking, a positive slope is expected meaning high-quality images produce low false match rate. A negative slope means that high-quality images produce higher false match rate than the low-quality images, which is not desired.

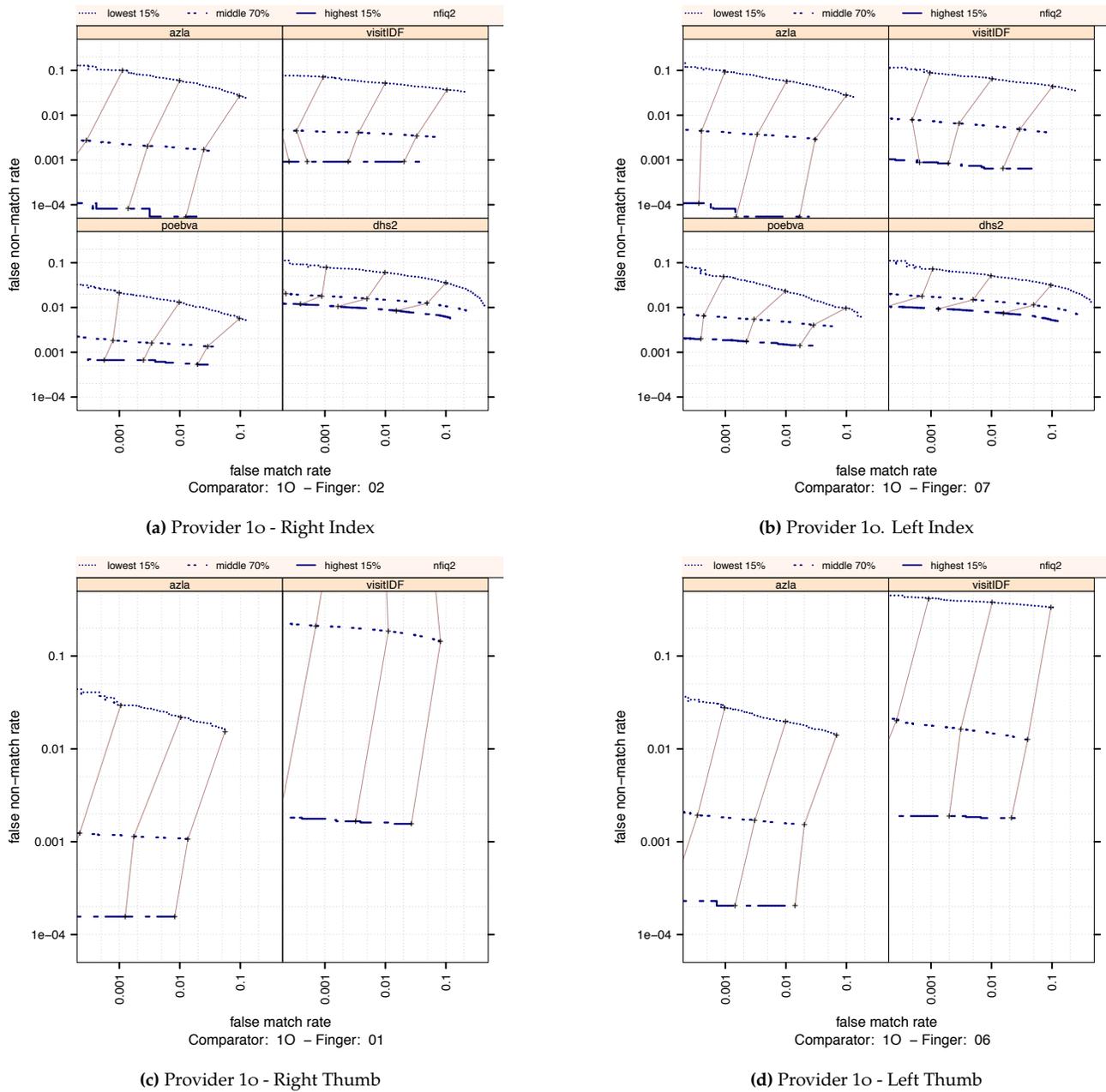


Figure 15:

Ranked DET - Provider 1o. The set of all comparisons are partitioned into three groups based on the pair-wise NFIQ 2 scores of the images being compared. The lowest quality set contains comparisons with pairwise quality in the lowest 15 percentile. The highest quality set contains comparisons with pairwise quality in the highest 15 percentile. The rest of the comparisons, namely the middle 70%, make up the third set. The DETs are connected at the same score threshold values (brown lines). Lower false non-match rate and false match rates are expected for higher quality images. That means well separated curves in each cell, with the DET curve corresponding to the lowest NFIQ 2 values appearing above, and the DET curve of highest NFIQ 2 values below all the other curves.

6.3 Computation efficiency

NFIQ 2 computation time depends on the size of the image area. The FVC2000/DB1 images, for which computation takes around 65 milliseconds, have an image size of only 300×300 pixels. NFIQ 2 computation time for FVC2000/DB3 is around 150 milliseconds, and their image sizes are 448×47 pixels. The reported measurements are for Windows 7 Enterprise, 64-bit OS platform. On Linux system (CentOS 6.2 64-bit), the average computation time for FVC2000/DB3 was 118 milliseconds.

Figure 16 shows computation time for images in a public dataset. Figure 17 shows the same computation time for the same images, but displayed on a logarithmic scale.

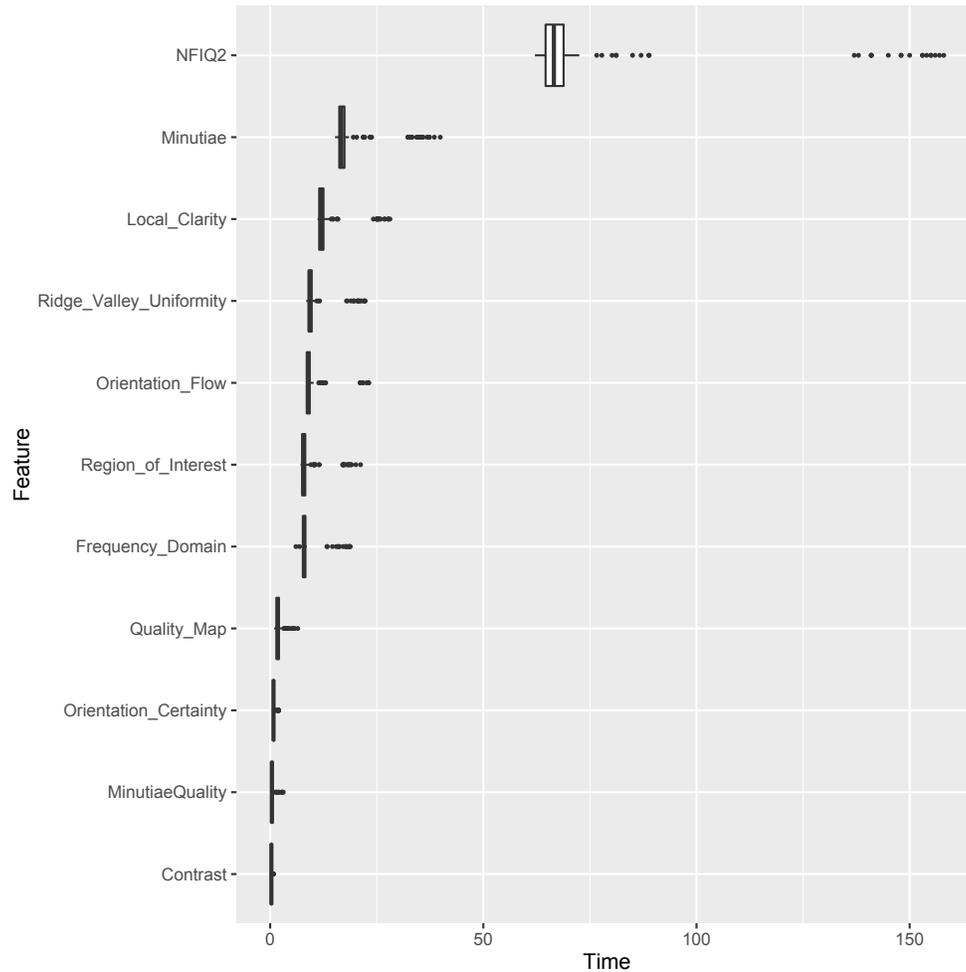


Figure 16: Computation time for NFIQ 2 and its features for FVC2000/DB1 and FVC2000/DB3 on Windows 7 Enterprise, 64-bit OS platform.

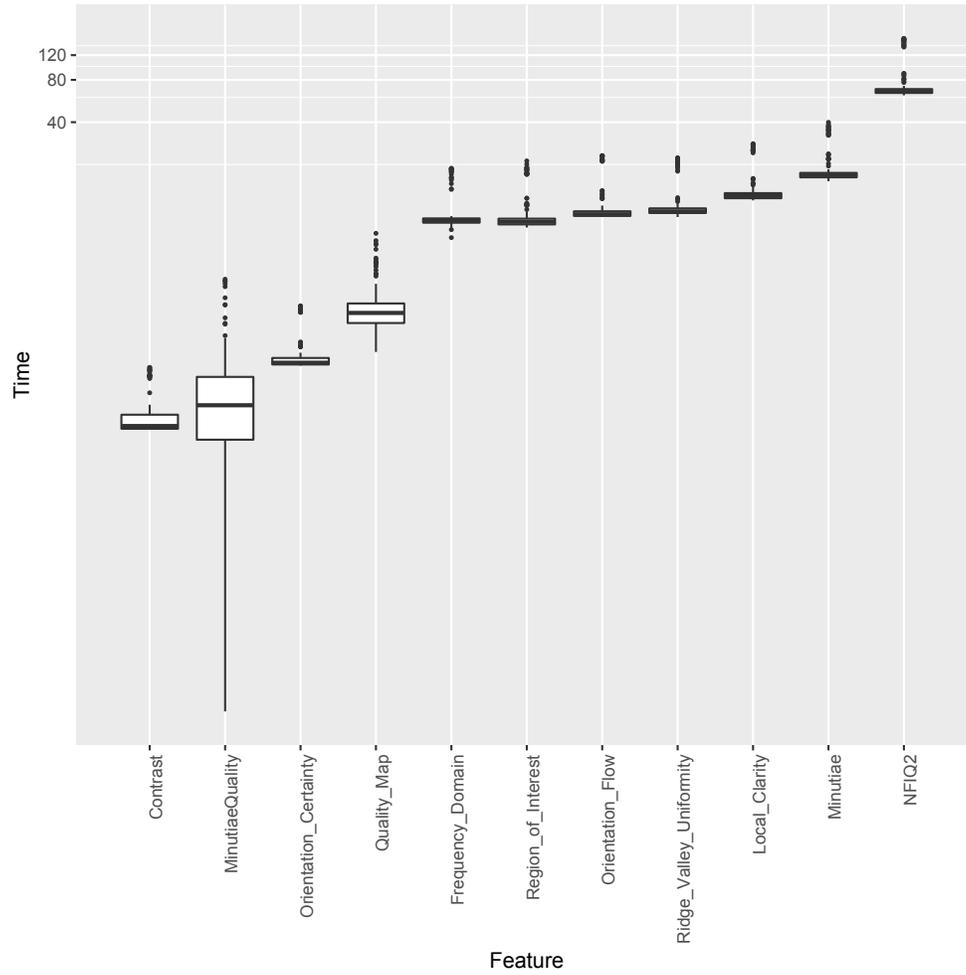


Figure 17: Computation time for NFIQ 2 and its features for FVC2000/DB1 and FVC2000/DB3 on Windows 7 Enterprise, 64-bit OS platform, shown on a logarithmic scale.

References

- [1] Working Group 5. *ISO/IEC 19795-1 Biometric Performance Testing and Reporting: Principles and Framework*. JTC 1/SC 37, international standard edition, August 2005. <https://www.iso.org/standard/41447.html>.
- [2] T.P. Chen, X. Jiang, and W.Y. Yau. Fingerprint image quality analysis. In *Image Processing, 2004. ICIP '04. 2004 International Conference on*, volume 2, pages 1253–1256 Vol.2, 2004.
- [3] Gregory Fiumara, Patricia Flanagan, John Grantham, Bruce Bandini, Kenneth Ko, and John Libert. National institute of standards and technology special database 300:uncompressed plain and rolled images from fingerprint cards. Technical Note 1993, National Institute of Standards and Technology, June 2018.
- [4] Gregory Fiumara, Patricia Flanagan, John Grantham, Kenneth Ko, Karen Marshall, Matthew Schwarz, Elham Tabassi, Brian Woodgate, and Christopher Boehnen. National Institute of Standards and Technology Special Database 302: Nail to Nail Fingerprint Challenge. Technical Note 2007, National Institute of Standards and Technology, August 2018.
- [5] Patrick Grother and Elham Tabassi. Performance of Biometric Quality Measures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29:531–543, 2007.
- [6] ISO/IEC. IS 29794-1:2009. information technology — biometric sample quality — part 1: Framework. ISO Standard, January 2009.
- [7] ISO/IEC. TR 29794-4:2010. information technology — biometric sample quality — part 4: Finger image data. Technical report, JTC 1/SC 37, 2010.
- [8] ISO/IEC. IS 29794-1:2016. information technology — biometric sample quality — part 1: Framework. ISO Standard, January 2016.
- [9] Michael Kass and Andrew Witkin. Analyzing oriented patterns. *Computer vision, graphics, and image processing*, 37(3):362–385, 1987.
- [10] P. D. Kovesi. MATLAB and Octave functions for computer vision and image processing, 2005. <https://www.peterkovesi.com/matlabfns>.
- [11] E. Lim, K.-A. Toh, P.N. Suganthan, X. Jiang, and W.-Y. Yau. Fingerprint image quality analysis. In *Image Processing, 2004. ICIP '04. 2004 International Conference on*, volume 2, pages 1241–1244 Vol.2, 2004.
- [12] Eyung Lim, Xudong Jiang, and Weiyun Yau. Fingerprint quality and validity analysis. In *Proc. Int. Conf. Image Processing 2002*, volume 1, 2002.
- [13] Davide Maltoni, Dario Maio, Anil K. Jain, and Salil Prabhakar. *Handbook of Fingerprint Recognition*. Springer Publishing Company, Incorporated, 2nd edition, 2009.
- [14] A. Martin, G. Doddington, T. Kamm, M. Ordowski, and M. Przybocki. The det curve in assessment of detection task performance. In *Proceedings of Eurospeech 97*, pages 1895–1898, 1997.
- [15] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *Automatica*, 11(285-296):23–27, 1975.
- [16] Satoshi Suzuki et al. Topological structural analysis of digitized binary images by border following. *Computer Vision, Graphics, and Image Processing*, 30(1):32–46, 1985.

Appendices

Appendix A

NFIQ 2 conformance test criteria

A.1 Purpose

The NFIQ 2 algorithm for finger image quality assessment is released as open-source software. This appendix specifies conformance test criteria and a conformance test set by which any interested party can determine whether an implementation complies with the NFIQ 2 reference implementation.

A.2 NFIQ 2 conformance test set

The NFIQ 2 conformance test set consists of a set of publicly available finger images where all NFIQ 2 scores are sufficiently represented. Access to this data set may be requested from https://nigos.nist.gov/datasets/nfiq2_conformance. Images are sourced from NIST Special Databases 300 and 302. Where more than one resolution was present for an image, the 500 PPI version was used. There are also three fingerprints included that fail to obtain a score (marked as NA). These images serve as references for when a fingerprint is unable to be successfully processed by NFIQ 2. Parties who have obtained the conformance test set are free to use it for verifying the conformance of their implementations with NFIQ 2.

Table 5 contains the filenames of all images contained in the test set corresponding with their NFIQ 2 score, generated using NFIQ 2.1. Figure 18 shows some examples of finger images from the NFIQ 2 conformance test set with their NFIQ 2 scores, generated using NFIQ 2.1.

Table 5: NFIQ 2 conformance test set

Score	Finger image identifiers	
97	00002504_S_500_slap_06	
96	00002440_R_500_slap_03,	00002433_J_500_palm_08
95	00002544_S_500_slap_03,	00002433_J_500_palm_09
94	00002489_J_500_palm_02,	00002373_R_500_slap_06
93	00002610_S_500_slap_06,	00002433_N_500_palm_07
93	00001057_plain_500_03	
92	00002645_S_500_slap_01,	00002596_J_500_palm_07
91	00002439_Q_500_palm_03,	00002574_S_500_slap_06
90	00002571_S_500_slap_07,	00002611_Q_500_palm_03
89	00002314_R_500_slap_01,	00002579_J_500_palm_08
88	00002528_N_500_palm_07,	00002338_R_500_slap_06
87	00002424_R_500_slap_01,	00002504_N_500_palm_03
86	00002454_N_500_palm_02,	00002571_S_500_slap_10
85	00002626_S_500_slap_04,	00002495_J_500_palm_03
84	00002523_S_500_slap_06,	00002488_S_500_slap_03
83	00002392_R_500_slap_04,	00002371_J_500_palm_07
83	00001581_plain_500_04	
82	00002644_S_500_slap_01,	00002435_N_500_palm_03
81	00002569_S_500_slap_01,	00002375_J_500_palm_02
80	00002573_Q_500_palm_03,	00002578_S_500_slap_09
79	00002602_S_500_slap_08,	00002445_J_500_palm_03
78	00002493_Q_500_palm_02,	00002376_R_500_slap_03
77	00002439_Q_500_palm_02,	00002568_S_500_slap_07
76	00002538_Q_500_palm_04,	00002392_R_500_slap_10
75	00002466_N_500_palm_04,	00002641_S_500_slap_02

Table 5: Continued

Score	Finger image identifiers	
74	00002591_S_500_slap_01,	00002495_J_500_palm_02
73	00002624_Q_500_palm_07,	00002641_S_500_slap_03
73	00001308_plain_500_08	
72	00002466_N_500_palm_05,	00002633_S_500_slap_09
71	00002315_R_500_slap_06,	00002489_J_500_palm_03
70	00002550_S_500_slap_09,	00002434_N_500_palm_04
69	00002581_S_500_slap_08,	00002514_Q_500_palm_03
68	00002423_J_500_palm_02,	00002494_S_500_slap_03
67	00002437_S_500_slap_08,	00002547_N_500_palm_08
66	00002445_J_500_palm_02,	00002650_S_500_slap_08
65	00002528_N_500_palm_05,	00002420_R_500_slap_04
64	00002533_S_500_slap_08,	00002573_Q_500_palm_04
63	00002457_N_500_palm_07,	00002319_R_500_slap_09
63	00001723_plain_500_05	
62	00002401_R_500_slap_08,	00002588_J_500_palm_07
61	00002401_N_500_palm_02,	00002409_R_500_slap_02
60	00002523_J_500_palm_03,	00002547_Q_500_palm_03
59	00002555_S_500_slap_08,	00002516_J_500_palm_05
58	00002399_J_500_palm_08,	00002373_R_500_slap_05
57	00002376_R_500_slap_05,	00002362_N_500_palm_04
56	00002518_N_500_palm_02,	00002559_S_500_slap_07
55	00002640_S_500_slap_02,	00002356_N_500_palm_05
54	00002559_J_500_palm_02,	00002636_S_500_slap_09
53	00002578_N_500_palm_04,	00002502_S_500_slap_08
53	00001428_plain_500_07	
52	00002447_R_500_slap_06,	00002319_N_500_palm_04
51	00002573_N_500_palm_08,	00002649_S_500_slap_08
50	00002408_R_500_slap_02,	00002534_N_500_palm_05
49	00002443_Q_500_palm_10,	00002550_J_500_palm_09
48	00002547_S_500_slap_04,	00002353_N_500_palm_10
47	00002357_R_500_slap_08,	00002362_N_500_palm_05
46	00002359_J_500_palm_10,	00002559_Q_500_palm_07
45	00002523_S_500_slap_04,	00002383_J_500_palm_05
44	00002401_R_500_slap_09,	00002641_J_500_palm_04
43	00002361_R_500_slap_03,	00002310_R_500_slap_05
43	00001574_plain_500_05	
42	00002458_J_500_palm_02,	00002304_R_500_slap_08
41	00002445_S_500_slap_05,	00002330_N_500_palm_10
40	00002569_S_500_slap_07,	00002533_Q_500_palm_09
39	00002320_R_500_slap_06,	00002593_Q_500_palm_05
38	00002325_N_500_palm_08,	00002569_S_500_slap_03
37	00002439_S_500_slap_10,	00002478_Q_500_palm_03
36	00002478_S_500_slap_08,	00002584_Q_500_palm_03
35	00002362_R_500_slap_09,	00002462_N_500_palm_02
34	00002415_R_500_slap_05,	00002649_Q_500_palm_09
33	00002396_R_500_slap_01,	00002357_N_500_palm_10
33	00001655_plain_500_04	
32	00002325_R_500_slap_04,	00002494_J_500_palm_04
31	00002502_S_500_slap_09,	00002554_J_500_palm_08
30	00002391_R_500_slap_07,	00002577_Q_500_palm_10
29	00002447_R_500_slap_01,	00002478_S_500_slap_03
28	00002358_R_500_slap_05,	00002531_Q_500_palm_09
27	00002579_J_500_palm_05,	00002593_J_500_palm_07

Table 5: Continued

Score	Finger image identifiers	
26	00002458_S_500_slap_02,	00002542_Q_500_palm_10
25	00002581_Q_500_palm_10,	00002568_J_500_palm_10
24	00002368_J_500_palm_10,	00002384_N_500_palm_10
23	00002374_R_500_slap_10,	00002577_N_500_palm_09
23	00001449_plain_500_08	
22	00002475_S_500_slap_04,	00002398_J_500_palm_09
21	00002568_J_500_palm_04,	00002349_N_500_palm_05
20	00002645_S_500_slap_10,	00002560_N_500_palm_10
19	00002458_N_500_palm_09,	00002361_J_500_palm_09
18	00002523_S_500_slap_05,	00002484_Q_500_palm_05
17	00002475_S_500_slap_01,	00002533_N_500_palm_10
16	00002384_N_500_palm_05,	00002616_Q_500_palm_07
15	00002406_R_500_slap_09,	00002494_J_500_palm_10
14	00002339_N_500_palm_07,	00002349_R_500_slap_07
13	00002502_Q_500_palm_05,	00002302_N_500_palm_10
13	00001610_plain_500_10	
12	00002612_S_500_slap_10,	00002478_N_500_palm_05
11	00002391_R_500_slap_06,	00002569_N_500_palm_08
10	00002616_J_500_palm_07,	00002403_J_500_palm_05
9	00002569_N_500_palm_09,	00002475_J_500_palm_04
8	00002521_Q_500_palm_10,	00002396_R_500_slap_10
7	00002640_Q_500_palm_04,	00002505_N_500_palm_10
6	00002559_S_500_slap_05,	00002326_R_500_slap_05
5	00002342_R_500_slap_05,	00002584_J_500_palm_09
4	00002475_J_500_palm_09,	00002523_J_500_palm_05
3	00002321_R_500_slap_07,	00002497_N_500_palm_08
3	00001665_plain_500_08	
2	00002315_R_500_slap_05,	00002475_J_500_palm_10
1	00002320_R_500_slap_03,	00002497_N_500_palm_09
0	00002502_J_500_palm_09,	00002564_S_500_slap_09
0	00001880_plain_500_10	
NA	00001052_plain_500_10	
NA	00002314_R_500_slap_02	
NA	00002303_N_500_palm_02	

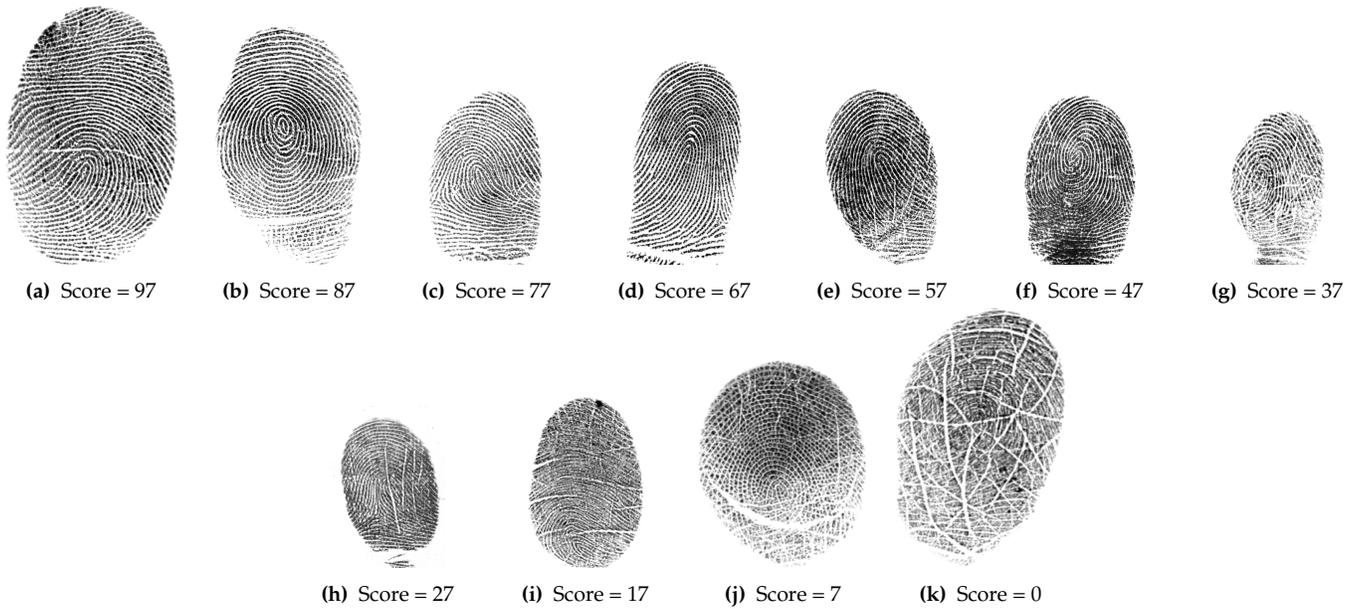


Figure 18: Examples of finger images from the NFIQ 2 conformance test set and their NFIQ 2 scores, generated using NFIQ 2.1: ((a)) 00002504_S_500_slap_06; ((b)) 00002424_R_500_slap_01; ((c)) 00002439_Q_500_palm_02; ((d)) 00002437_S_500_slap_08; ((e)) 00002376_R_500_slap_05; ((f)) 00002357_R_500_slap_08; ((g)) 00002439_S_500_slap_10; ((h)) 00002579_J_500_palm_05; ((i)) 00002475_S_500_slap_01; ((j)) 00002640_Q_500_palm_04; ((k)) 00002564_S_500_slap_09. Fingerprint images sourced from NFIQ 2 Conformance Test Set, subset from NIST Special Databases 300 and 302 [3, 4].

A.3 Use of conformance testing

Parties who wish to verify the conformance of their implementations with the NFIQ 2 reference implementation shall run their implementation on all images in the NFIQ 2 conformance test set. Scores must match the scores generated by the reference implementation. These scores may be found on the GitHub repository under the conformance directory. These scores must remain consistent across all operating systems and platforms.

Appendix B

Results for provider 1F

This appendix contains results for provider 1F.

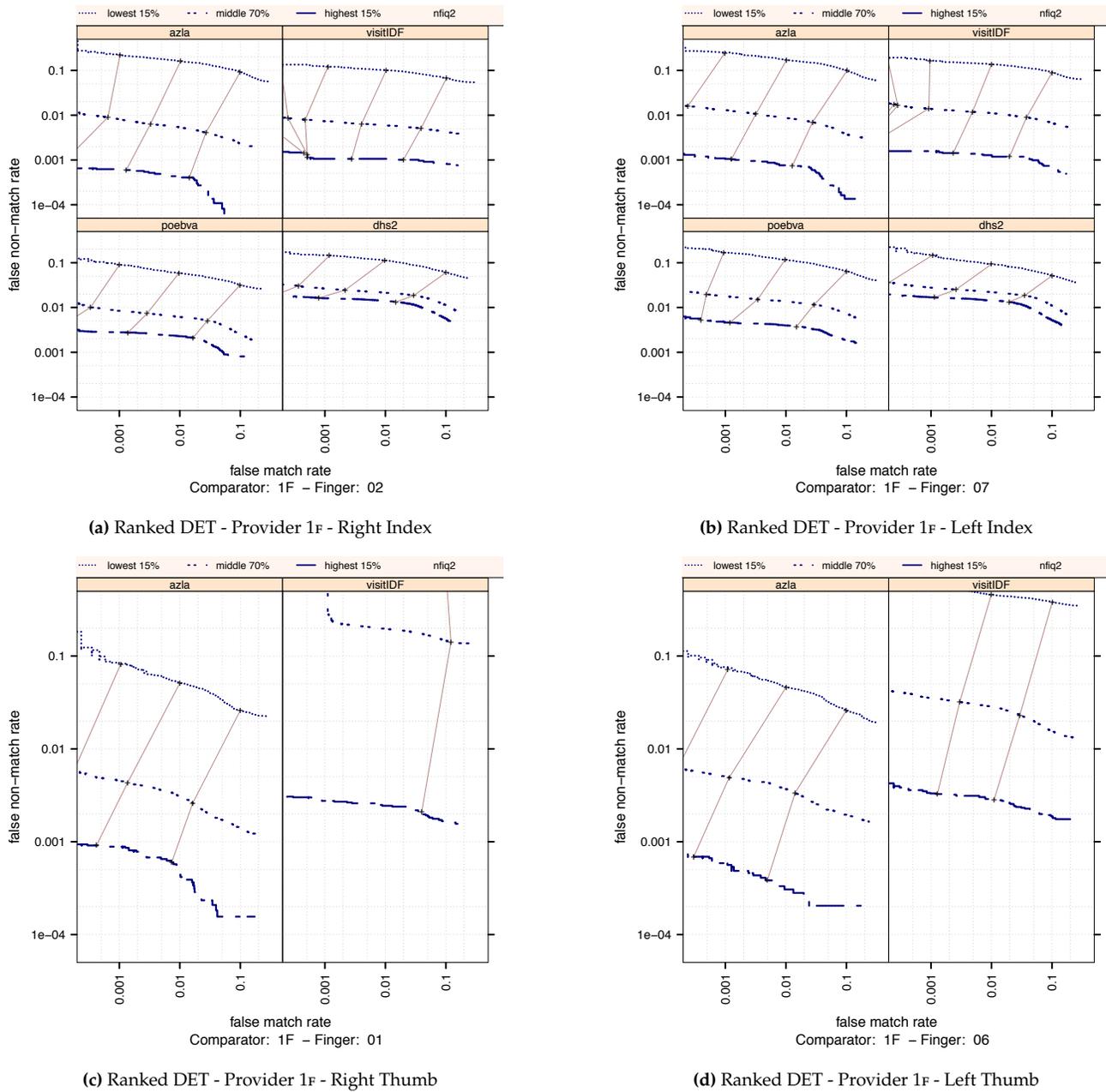


Figure 19:

Ranked DET - Provider 1F. The set of all comparisons are partitioned into three groups based on the pair-wise NFIQ 2 scores of the images being compared. The lowest quality set contains comparisons with pairwise quality in the lowest 15 percentile. The highest quality set contains comparisons with pairwise quality in the highest 15 percentile. The rest of the comparisons, namely the middle 70%, make up the third set. The DETs are connected at the same score threshold values (brown lines). Lower false non-match rate and false match rates are expected for higher quality images. That means well separated curves in each cell, with the DET curve corresponding to the lowest NFIQ 2 values appearing above, and the DET curve of highest NFIQ 2 values below all the other curves.

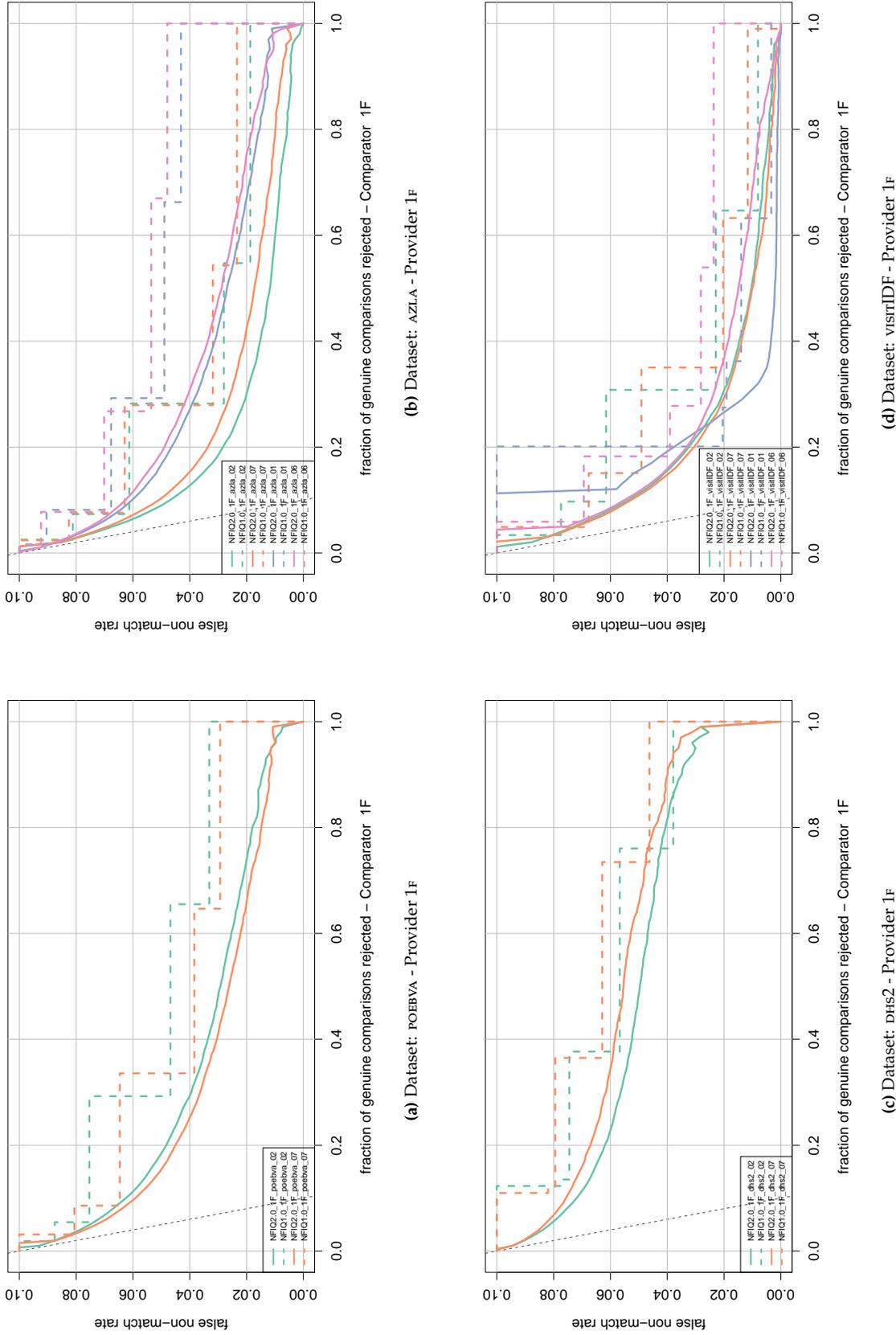


Figure 20:

Error vs. reject curve - Provider 1F. The threshold is set to give an initial false non-match rate of 0.1. The gray dotted line shows the ideal case where the rejection of the ten percent lowest quality results in zero false non-match rate. NFIQ 2 is a better predictive of performance than NFIQ 1.0 because its error vs. reject curves are closer to the gray dotted line.

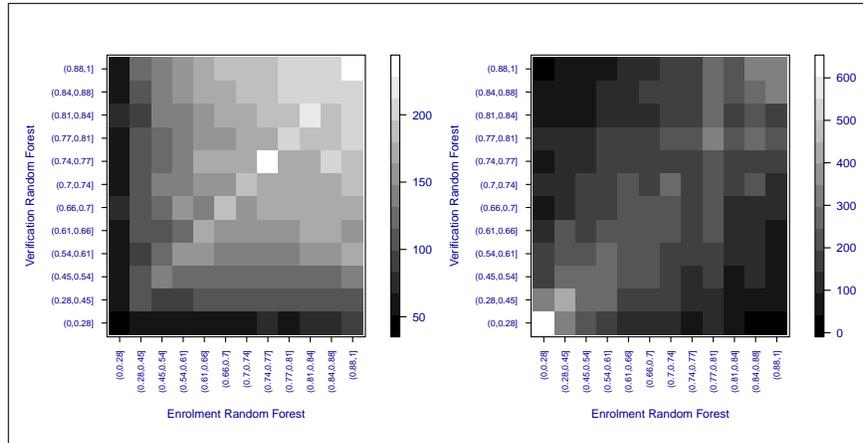


Figure 22: Heatmap for provider 1f on dataset azla: Left plot shows the mean comparison score, right plot shows the sample count.

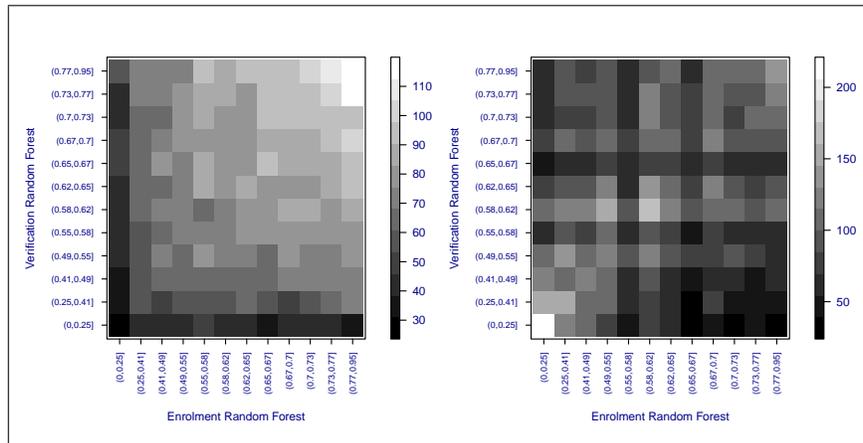


Figure 23: Heatmap for provider 1f on dataset dhs2: Left plot shows the mean comparison score, right plot shows the sample count.

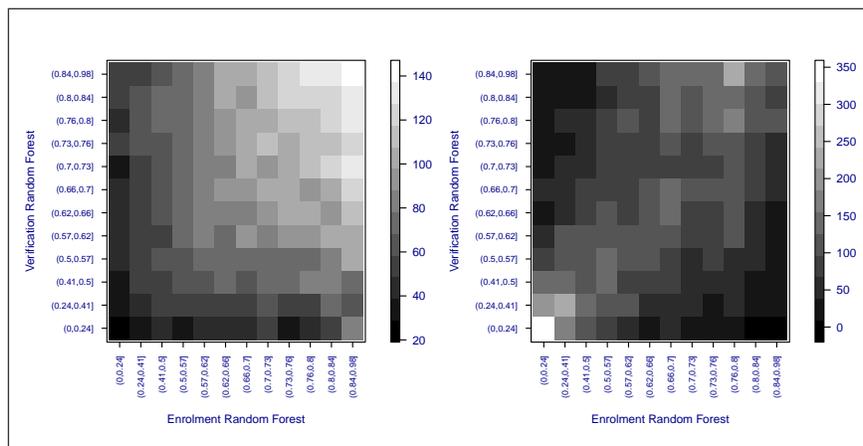


Figure 24: Heatmap for provider 1f on dataset poebva: Left plot shows the mean comparison score, right plot shows the sample count.

Appendix C

Results for provider 1o

This appendix contains results for provider 1o.

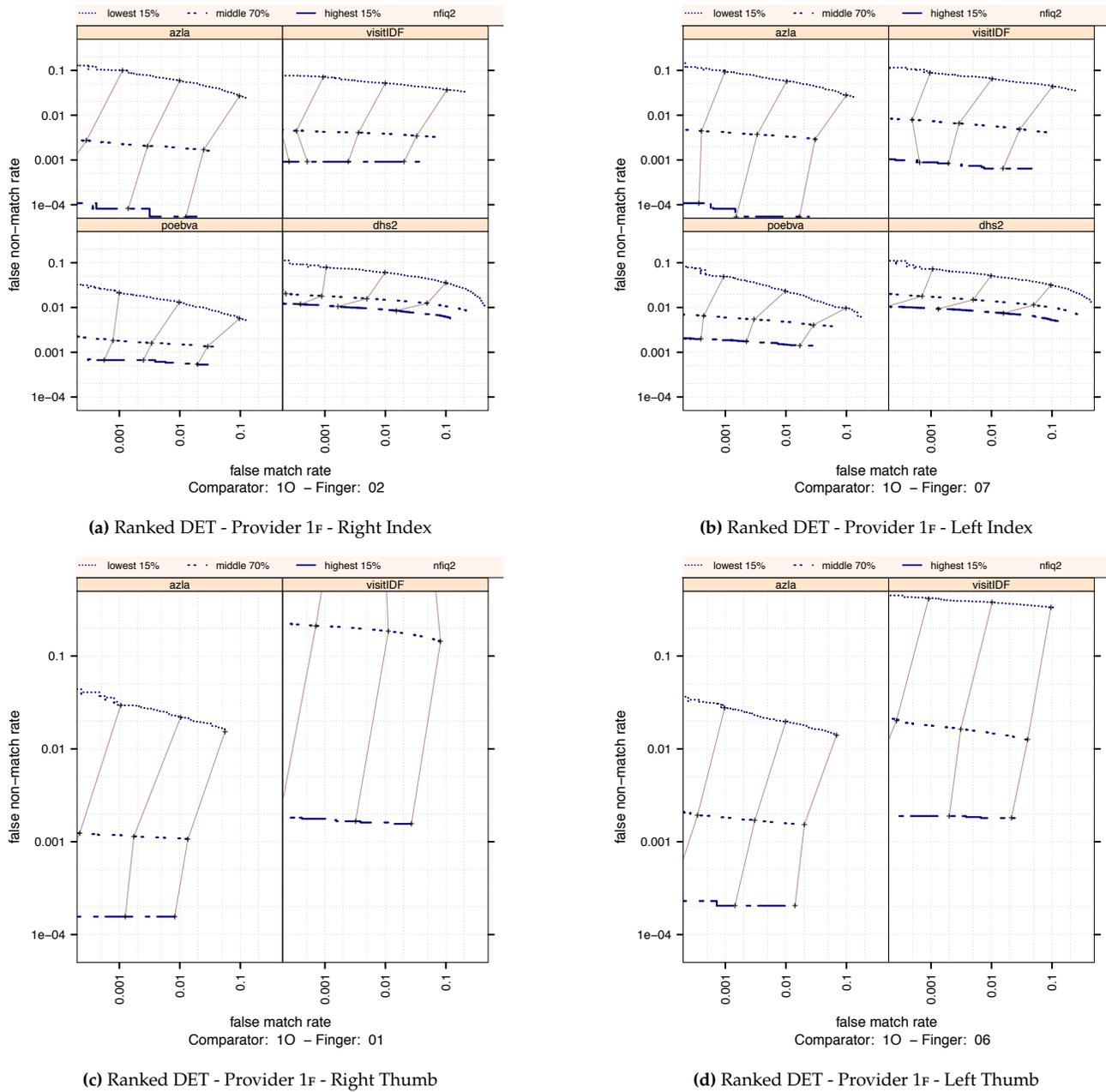


Figure 25:

Ranked DET - Provider 1o. The set of all comparisons are partitioned into three groups based on the pair-wise NFIQ 2 scores of the images being compared. The lowest quality set contains comparisons with pairwise quality in the lowest 15 percentile. The highest quality set contains comparisons with pairwise quality in the highest 15 percentile. The rest of the comparisons, namely the middle 70%, make up the third set. The DETs are connected at the same score threshold values (brown lines). Lower false non-match rate and false match rates are expected for higher quality images. That means well separated curves in each cell, with the DET curve corresponding to the lowest NFIQ 2 values appearing above, and the DET curve of highest NFIQ 2 values below all the other curves.

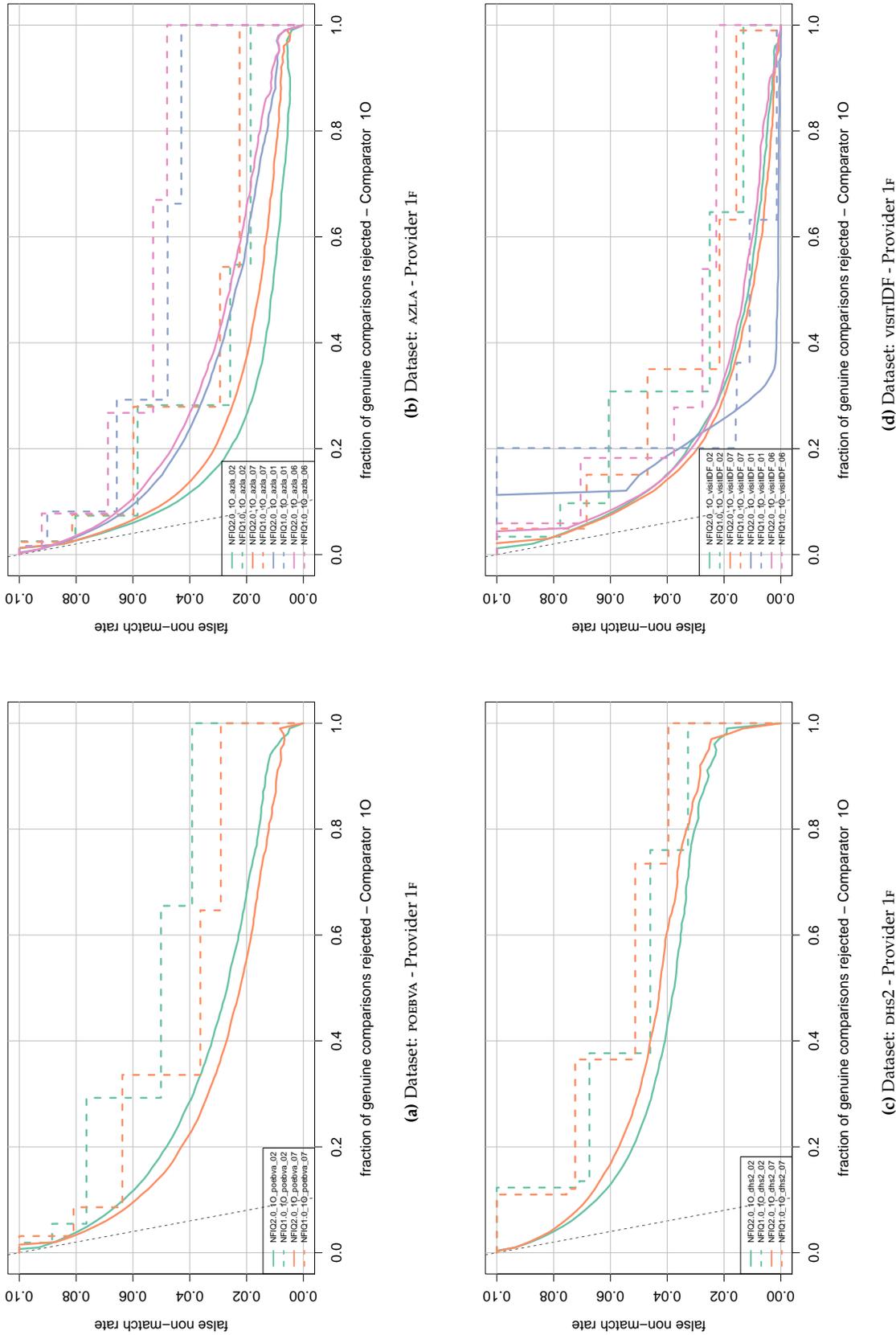


Figure 26:

Error vs. reject curve - Provider 1o. The threshold is set to give an initial false non-match rate of 0.1. The gray dotted line shows the ideal case where the rejection of the ten percent lowest quality results in zero false non-match rate. NFIQ 2 is a better predictive of performance than NFIQ 1.0 because its error vs. reject curves are closer to the gray dotted line.

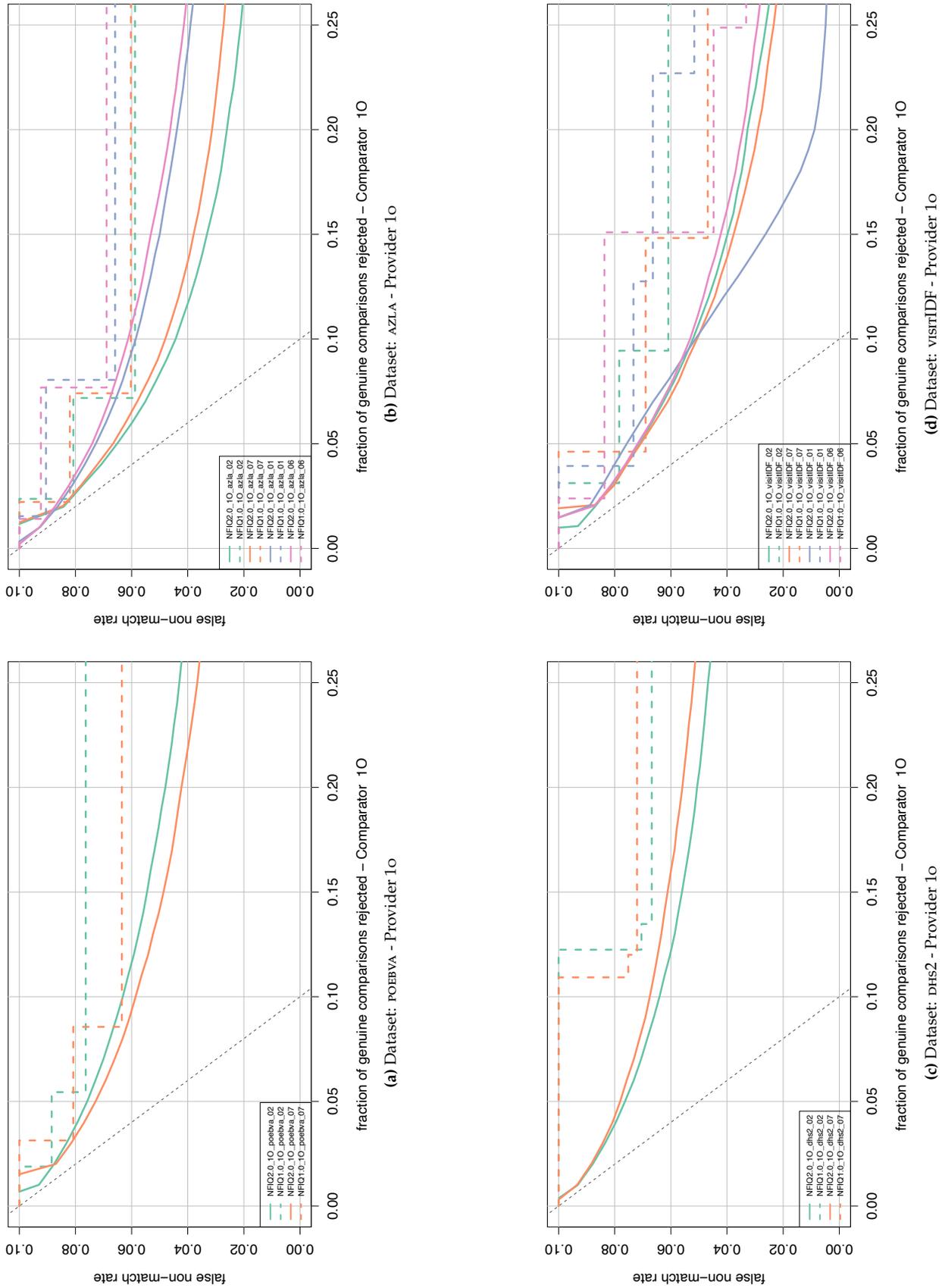


Figure 27:

Error vs. reject curve $\leq 25\%$. Provider 10. The threshold is set to give an initial false non-match rate of 0.1. The gray dotted line shows the ideal case where the rejection of the ten percent lowest quality results in zero false non-match rate. Same result as in fig. 26 but for rejection rate of up to 25%.

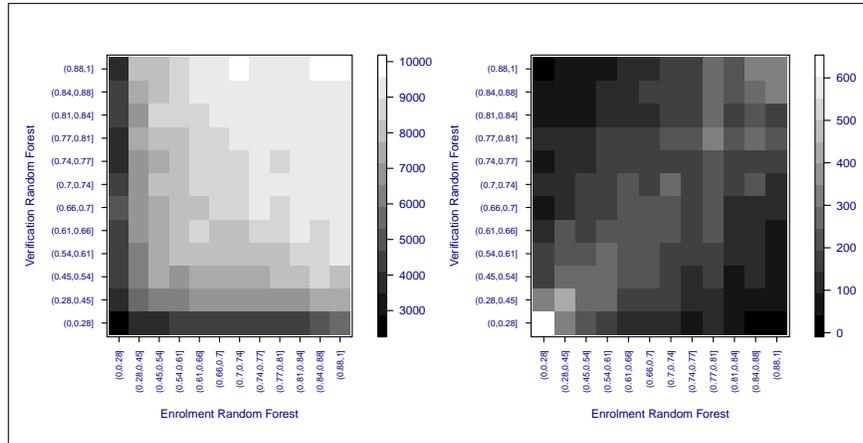


Figure 28: Heatmap for provider 1o on dataset azla: Left plot shows the mean comparison score, right plot shows the sample count.

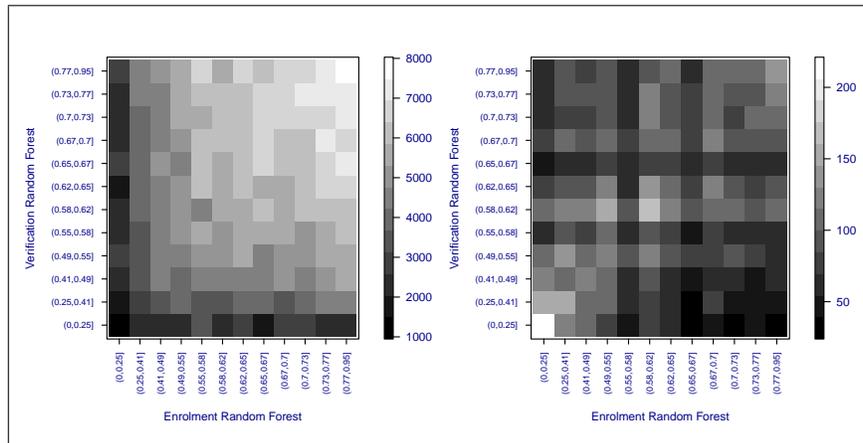


Figure 29: Heatmap for provider 1o on dataset dhs2: Left plot shows the mean comparison score, right plot shows the sample count.

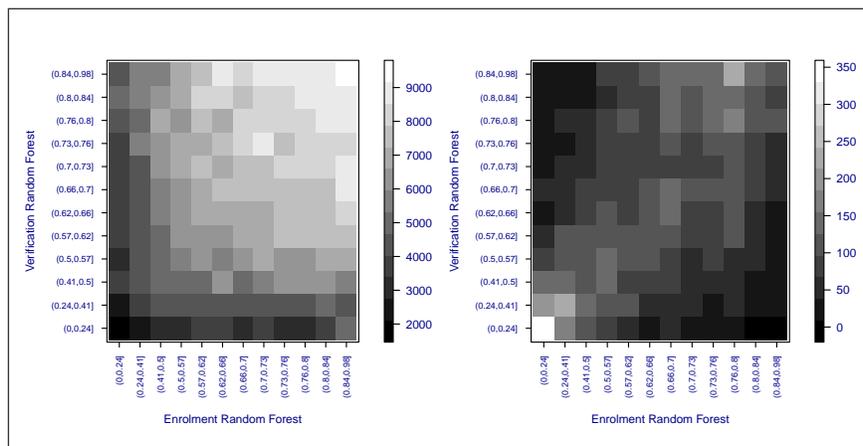


Figure 30: Heatmap for provider 1o on dataset poebva: Left plot shows the mean comparison score, right plot shows the sample count.

Appendix D

Results for provider 1τ

This appendix contains results for provider 1τ.

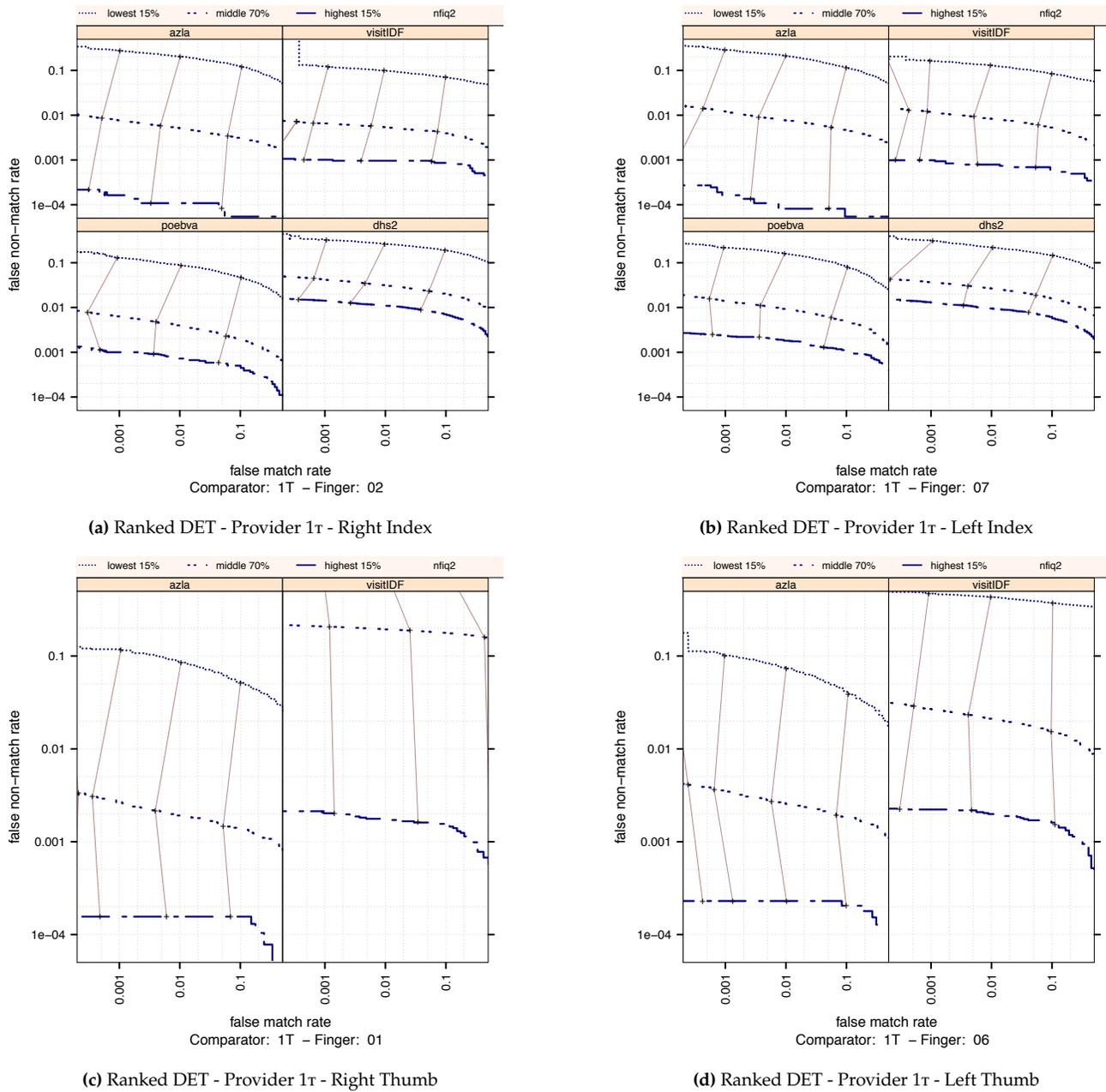


Figure 31:

Ranked DET - Provider 1τ. The set of all comparisons are partitioned into three groups based on the pair-wise NFIQ 2 scores of the images being compared. The lowest quality set contains comparisons with pairwise quality in the lowest 15 percentile. The highest quality set contains comparisons with pairwise quality in the highest 15 percentile. The rest of the comparisons, namely the middle 70%, make up the third set. The DETs are connected at the same score threshold values (brown lines). Lower false non-match rate and false match rates are expected for higher quality images. That means well separated curves in each cell, with the DET curve corresponding to the lowest NFIQ 2 values appearing above, and the DET curve of highest NFIQ 2 values below all the other curves.

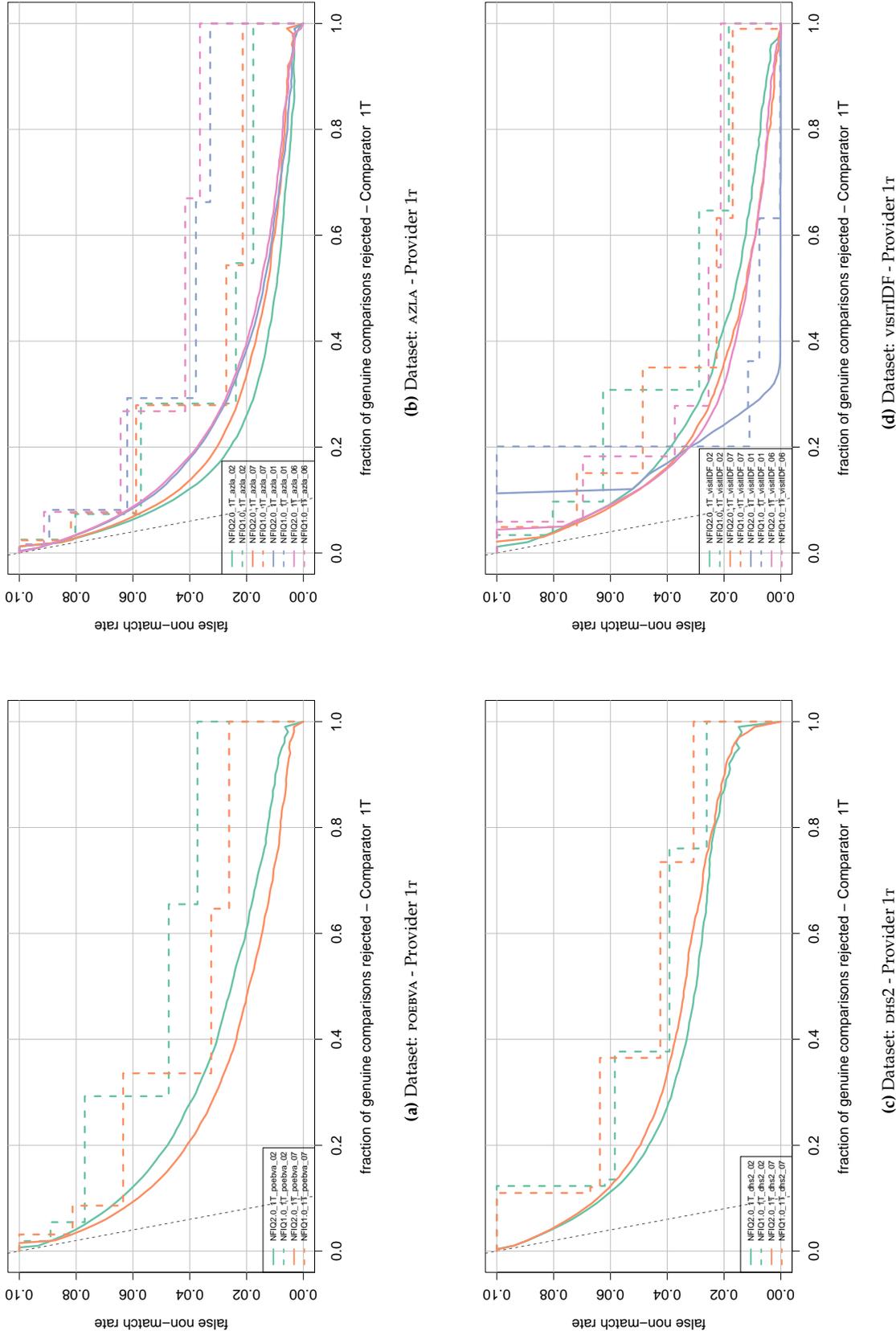


Figure 32:

Error vs. reject curve - Provider 1r. The threshold is set to give an initial false non-match rate of 0.1. The gray dotted line shows the ideal case where the rejection of the ten percent lowest quality results in zero false non-match rate. NFIQ 2 is a better predictive of performance than NFIQ 1.0 because its error vs. reject curves are closer to the gray dotted line.

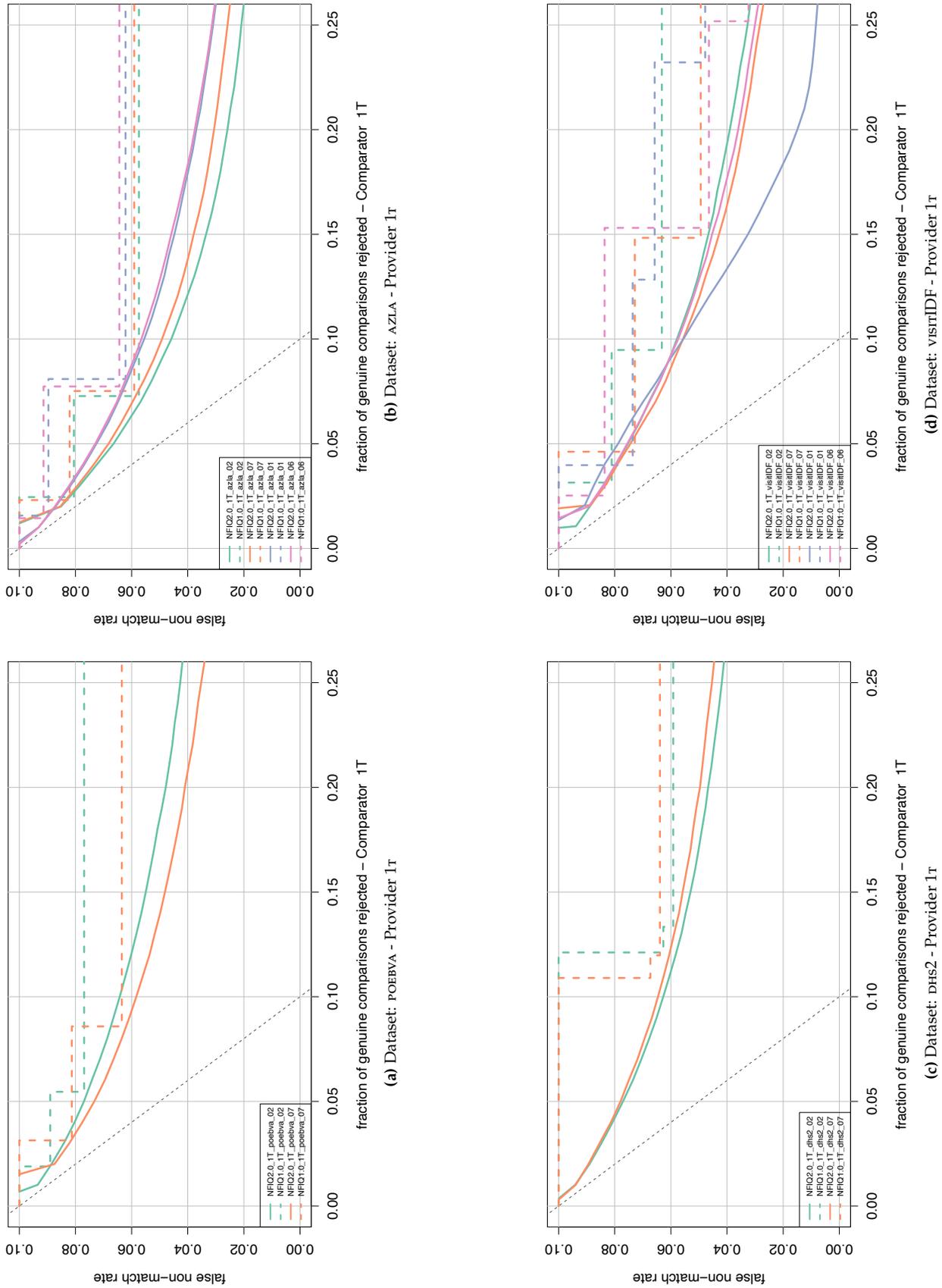


Figure 33:

Error vs. reject curve $\leq 25\%$. **Provider 1T**. The threshold is set to give an initial false non-match rate of 0.1. The gray dotted line shows the ideal case where the rejection of the ten percent lowest quality results in zero false non-match rate. Same result as in fig. 32 but for rejection rate of up to 25%.

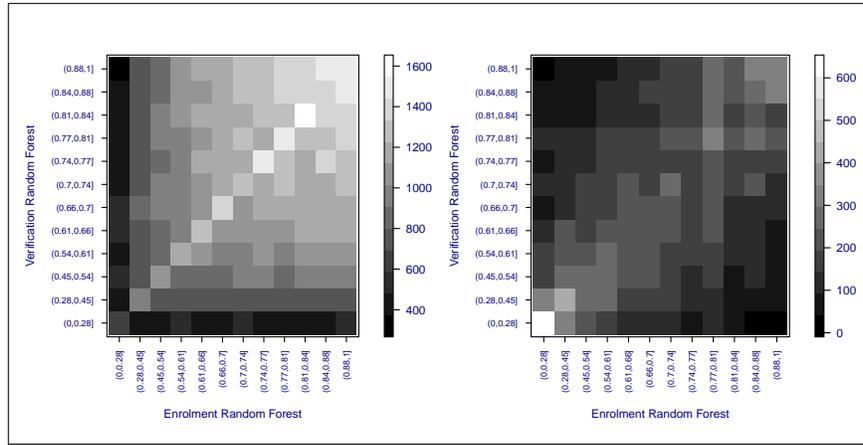


Figure 34: Heatmap for provider 1r on dataset azla: Left plot shows the mean comparison score, right plot shows the sample count.

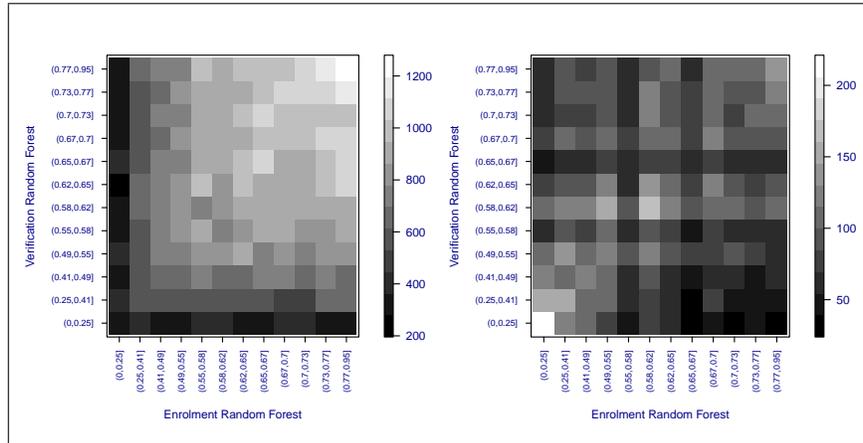


Figure 35: Heatmap for provider 1r on dataset dhs2: Left plot shows the mean comparison score, right plot shows the sample count.

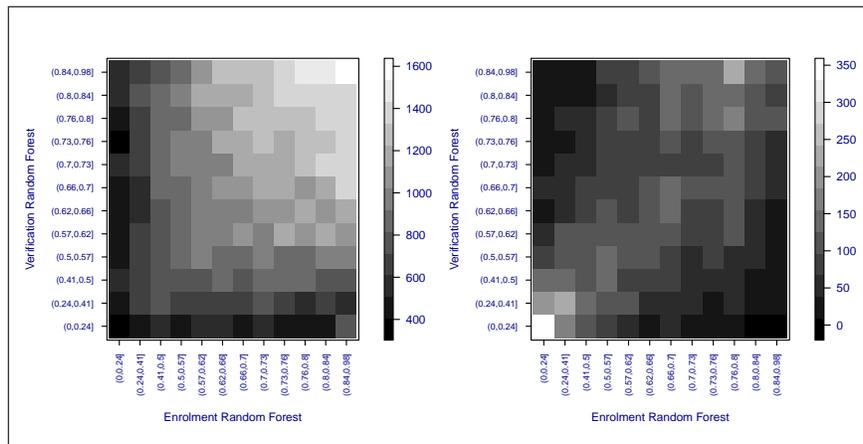
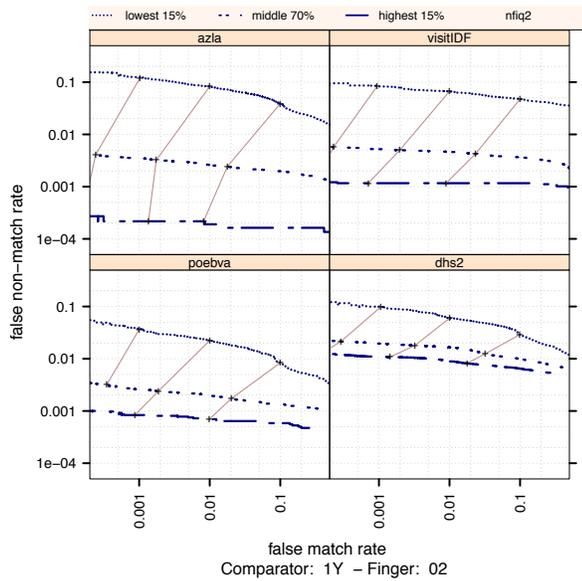


Figure 36: Heatmap for provider 1r on dataset poebva: Left plot shows the mean comparison score, right plot shows the sample count.

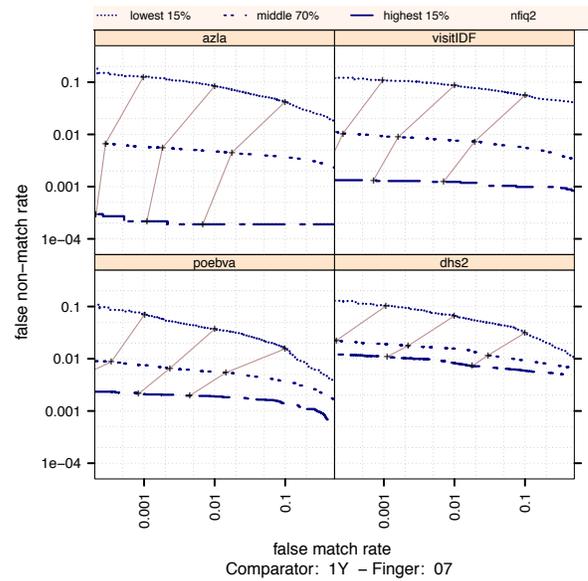
Appendix E

Results for provider 1y

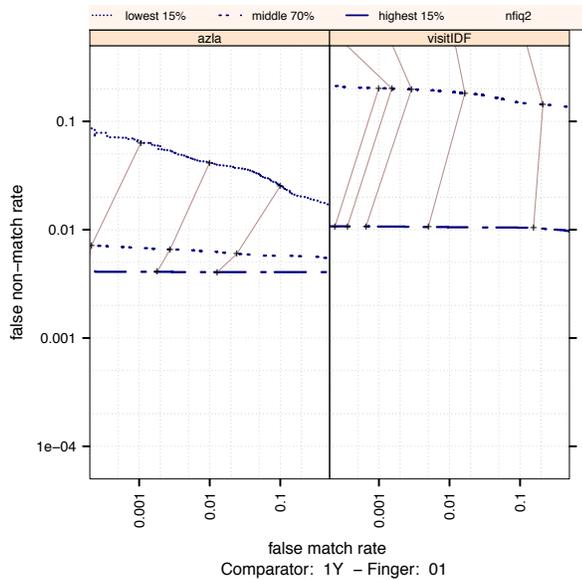
This appendix contains results for provider 1y.



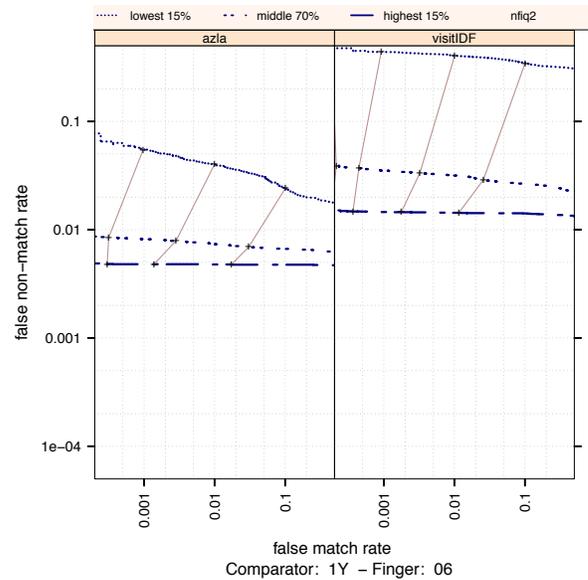
(a) Ranked DET - Provider 1γ - Right Index



(b) Ranked DET - Provider 1γ - Left Index



(c) Ranked DET - Provider 1γ - Right Thumb



(d) Ranked DET - Provider 1γ - Left Thumb

Figure 37:

Ranked DET - Provider 1γ. The set of all comparisons are partitioned into three groups based on the pair-wise NFIQ 2 scores of the images being compared. The lowest quality set contains comparisons with pairwise quality in the lowest 15 percentile. The highest quality set contains comparisons with pairwise quality in the highest 15 percentile. The rest of the comparisons, namely the middle 70%, make up the third set. The DETs are connected at the same score threshold values (brown lines). Lower false non-match rate and false match rates are expected for higher quality images. That means well separated curves in each cell, with the DET curve corresponding to the lowest NFIQ 2 values appearing above, and the DET curve of highest NFIQ 2 values below all the other curves.

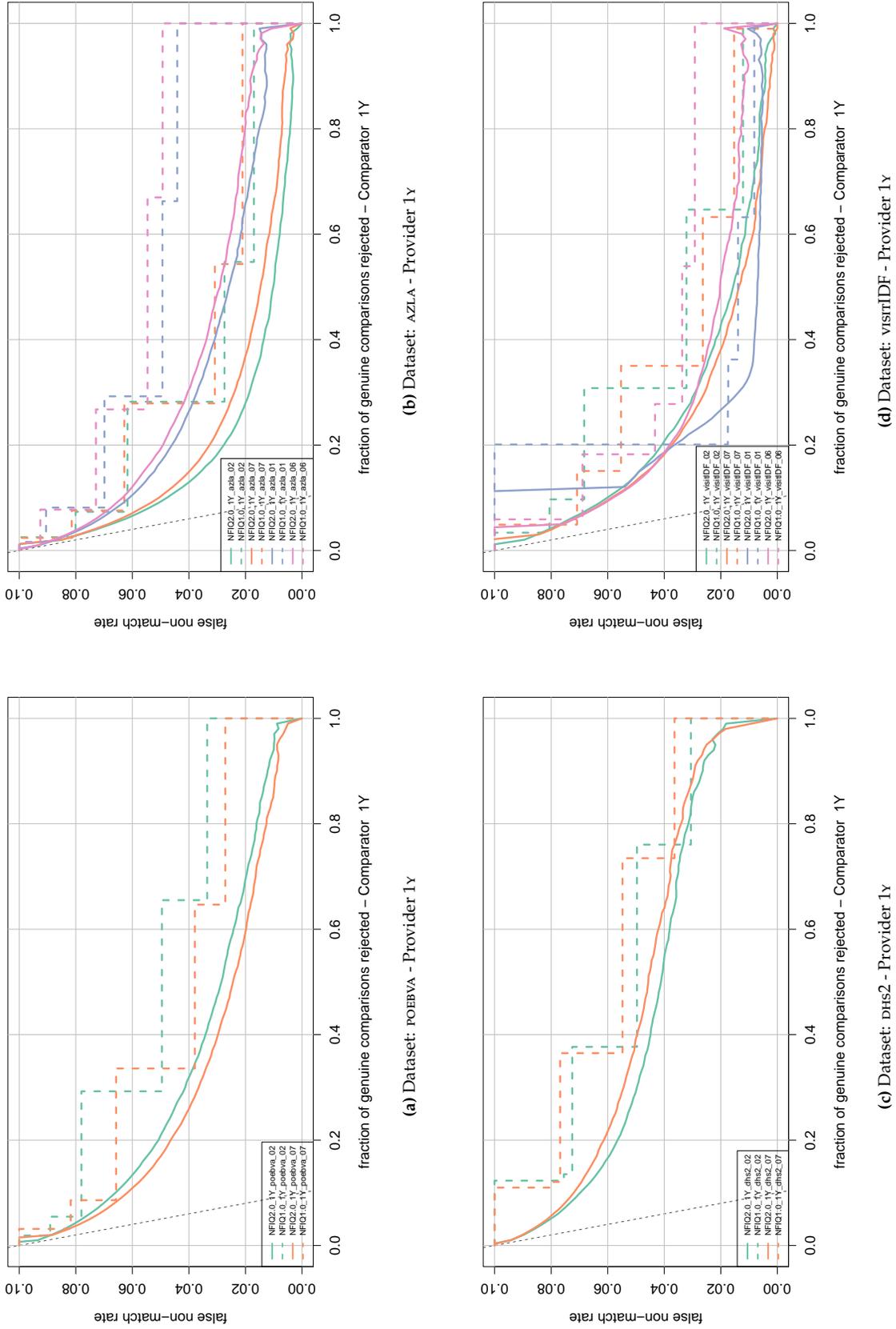


Figure 38:

Error vs. reject curve - Provider 1y. The threshold is set to give an initial false non-match rate of 0.1. The gray dotted line shows the ideal case where the rejection of the ten percent lowest quality results in zero false non-match rate. NFIQ 2 is a better predictive of performance than NFIQ 1.0 because its error vs. reject curves are closer to the gray dotted line.

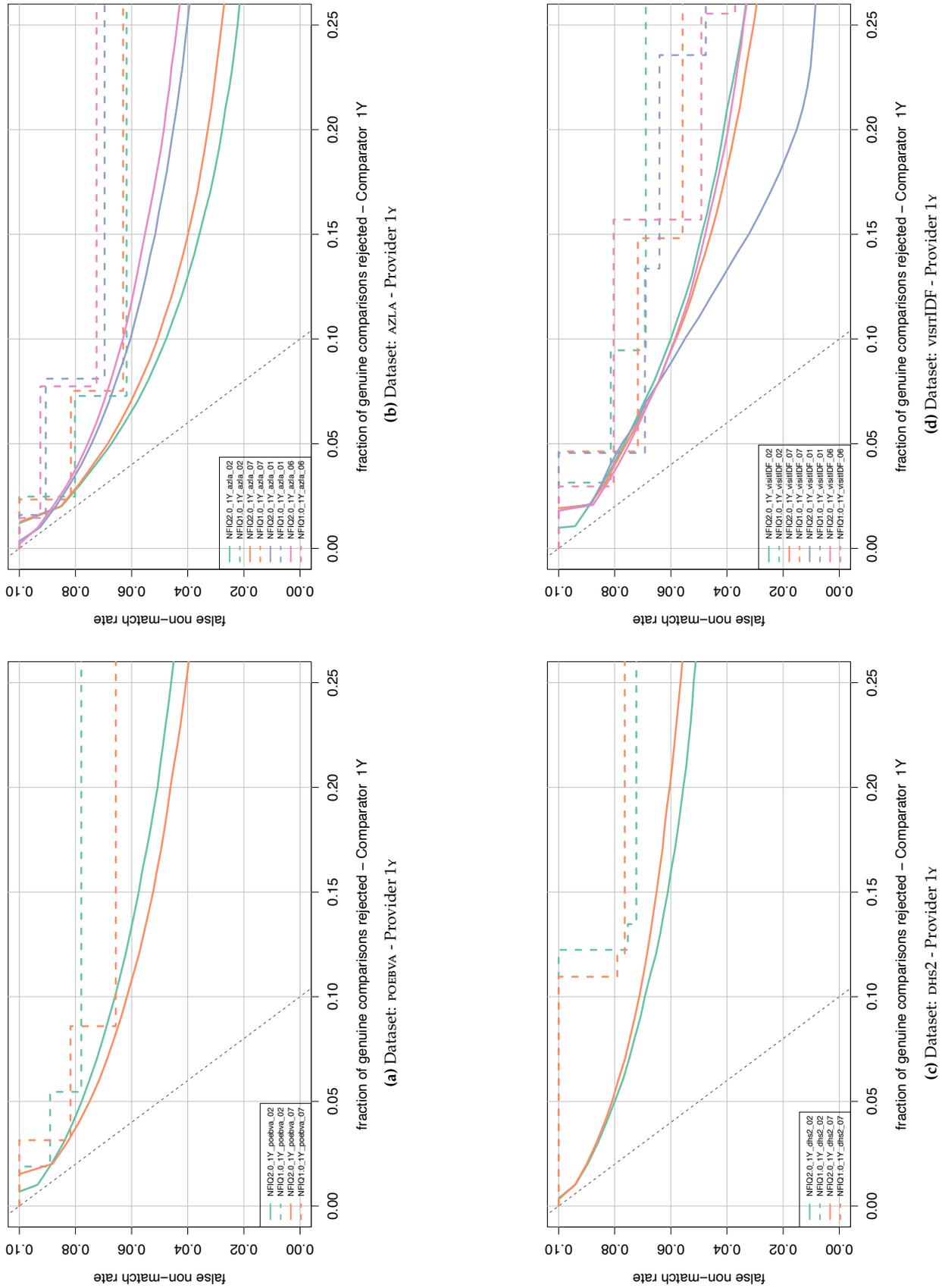


Figure 39:

Error vs. reject curve $\leq 25\%$. Provider 1y. The threshold is set to give an initial false non-match rate of 0.1. The gray dotted line shows the ideal case where the rejection of the ten percent lowest quality results in zero false non-match rate. Same result as in fig. 38 but for rejection rate of up to 25%.

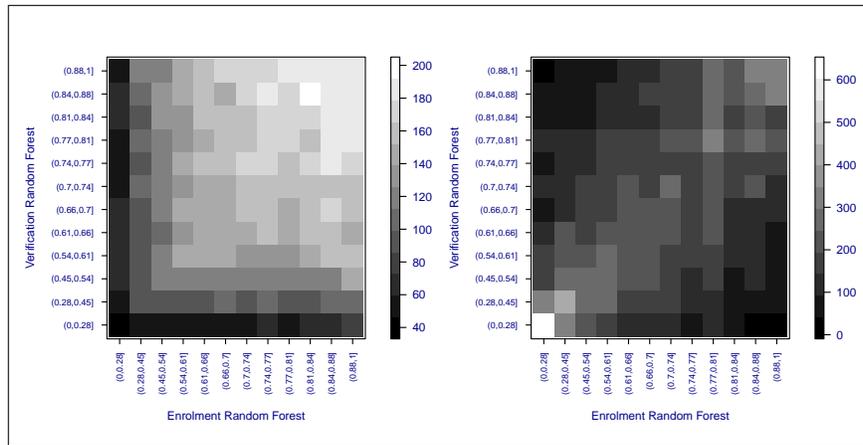


Figure 40: Heatmap for provider 1y on dataset azla: Left plot shows the mean comparison score, right plot shows the sample count.

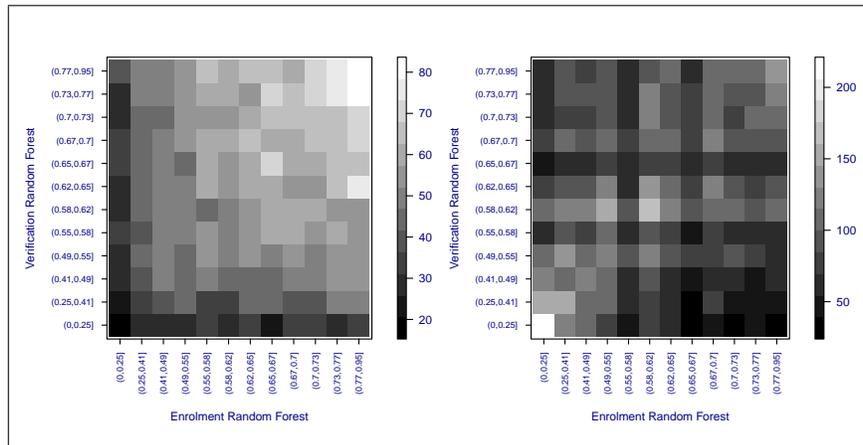


Figure 41: Heatmap for provider 1y on dataset dhs2: Left plot shows the mean comparison score, right plot shows the sample count.

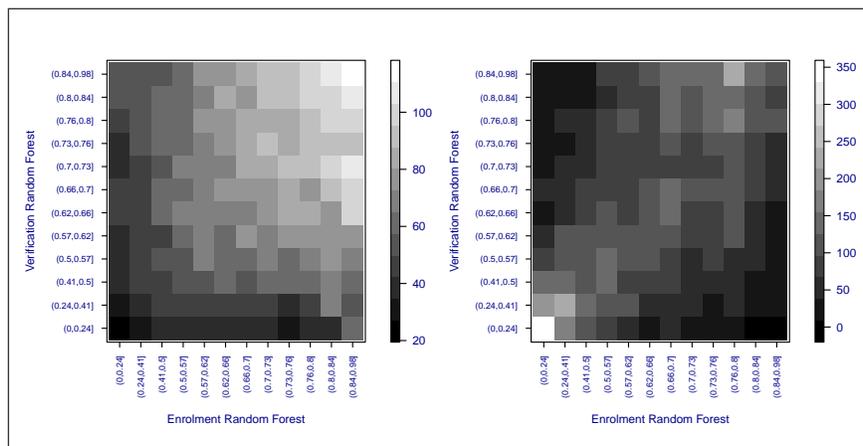
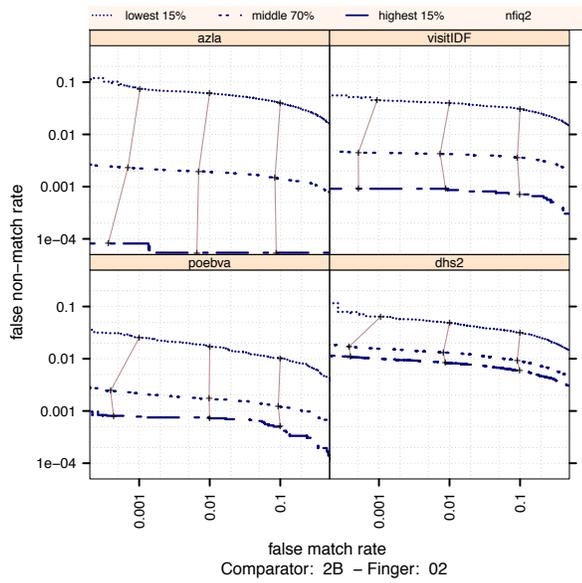


Figure 42: Heatmap for provider 1y on dataset poebva: Left plot shows the mean comparison score, right plot shows the sample count.

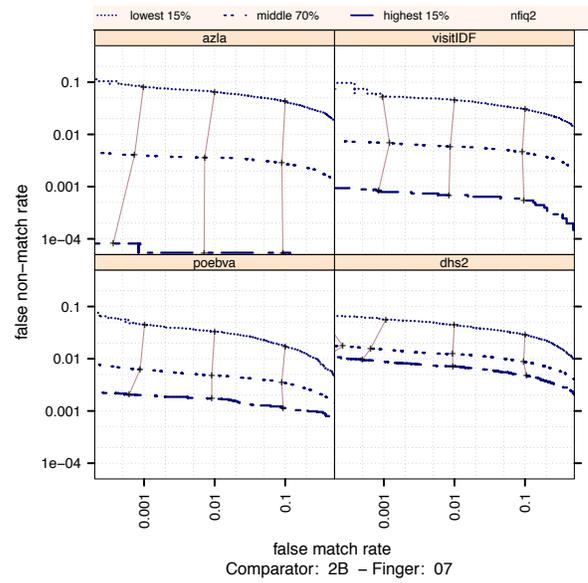
Appendix F

Results for provider 2B

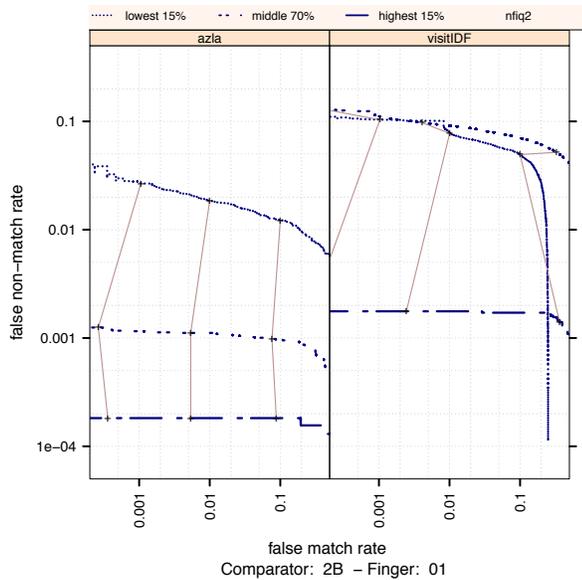
This appendix contains results for provider 2B.



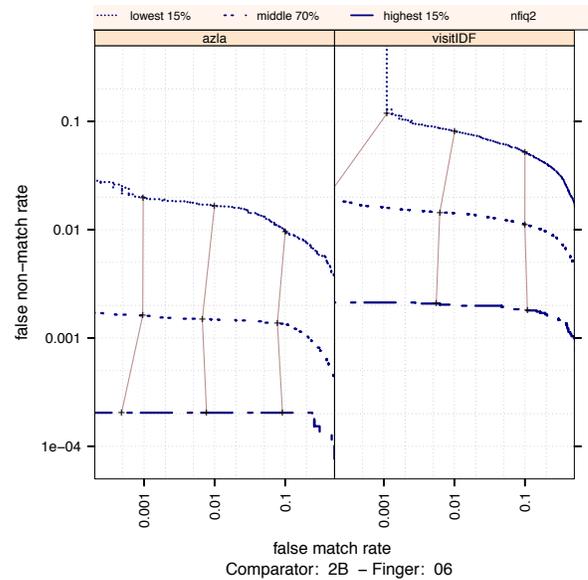
(a) Ranked DET - Provider 2B - Right Index



(b) Ranked DET - Provider 2B - Left Index



(c) Ranked DET - Provider 2B - Right Thumb



(d) Ranked DET - Provider 2B - Left Thumb

Figure 43:

Ranked DET - Provider 2B. The set of all comparisons are partitioned into three groups based on the pair-wise NFIQ 2 scores of the images being compared. The lowest quality set contains comparisons with pairwise quality in the lowest 15 percentile. The highest quality set contains comparisons with pairwise quality in the highest 15 percentile. The rest of the comparisons, namely the middle 70%, make up the third set. The DETs are connected at the same score threshold values (brown lines). Lower false non-match rate and false match rates are expected for higher quality images. That means well separated curves in each cell, with the DET curve corresponding to the lowest NFIQ 2 values appearing above, and the DET curve of highest NFIQ 2 values below all the other curves.

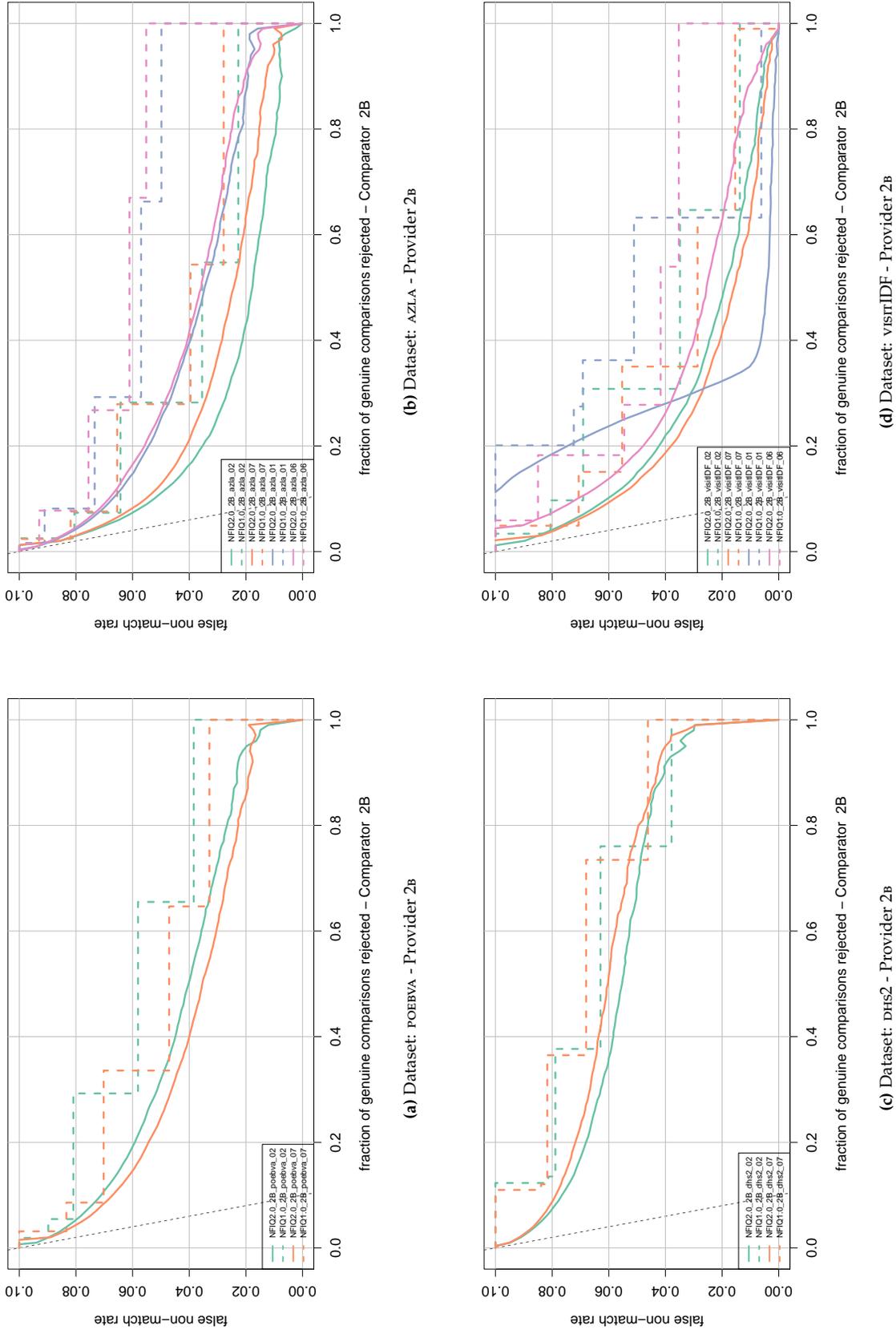


Figure 44:

Error vs. reject curve - Provider 2b. The threshold is set to give an initial false non-match rate of 0.1. The gray dotted line shows the ideal case where the rejection of the ten percent lowest quality results in zero false non-match rate. NFIQ 2 is a better predictive of performance than NFIQ 1.0 because its error vs. reject curves are closer to the gray dotted line.

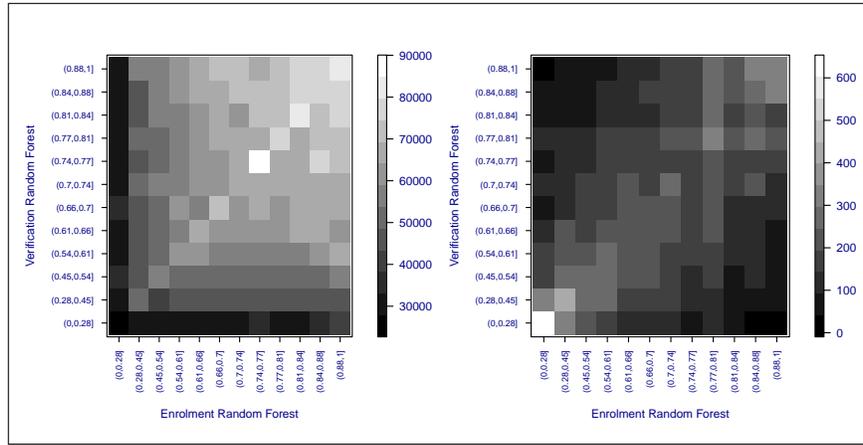


Figure 46: Heatmap for provider 2b on dataset azla: Left plot shows the mean comparison score, right plot shows the sample count.

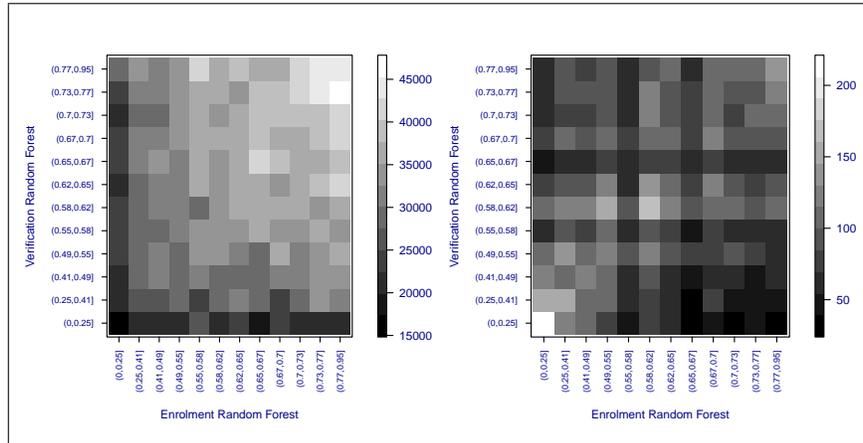


Figure 47: Heatmap for provider 2b on dataset dhs2: Left plot shows the mean comparison score, right plot shows the sample count.

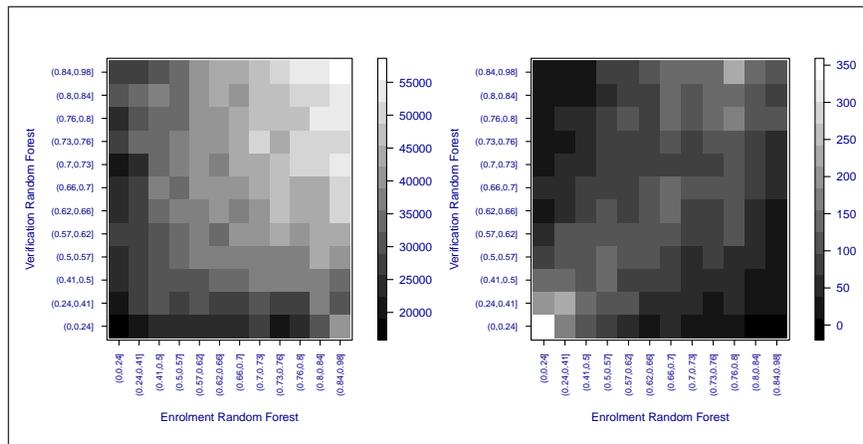
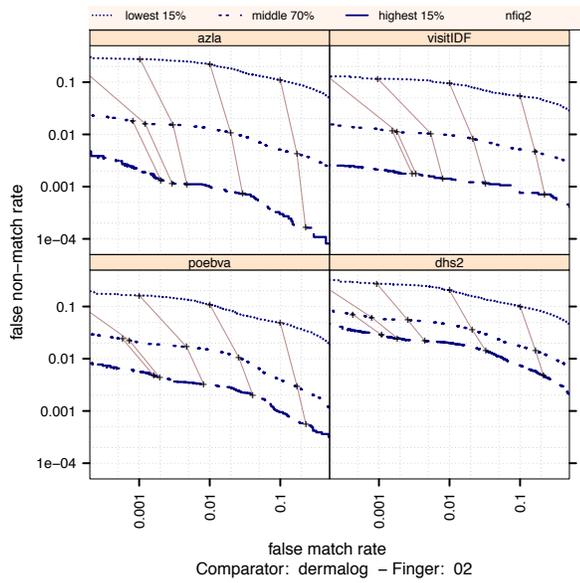


Figure 48: Heatmap for provider 2b on dataset poebva: Left plot shows the mean comparison score, right plot shows the sample count.

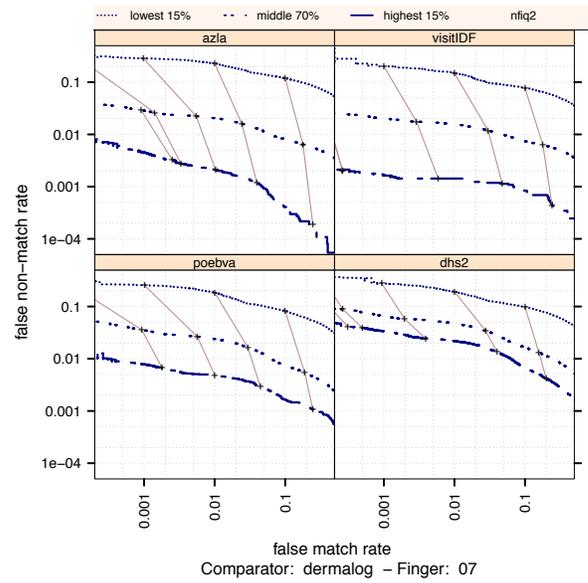
Appendix G

Results for provider DERMALOG

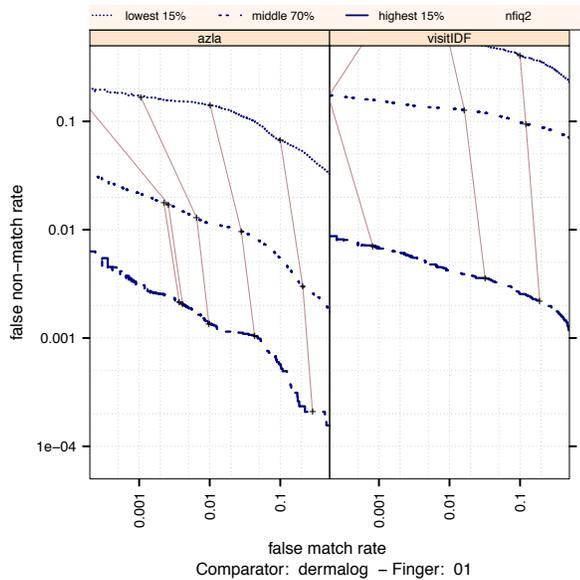
This appendix contains results for provider DERMALOG.



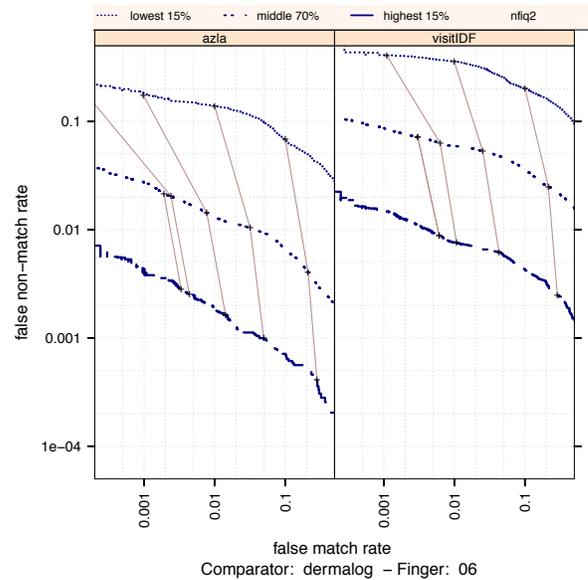
(a) Ranked DET - Provider DERMALOG - Right Index



(b) Ranked DET - Provider DERMALOG - Left Index



(c) Ranked DET - Provider DERMALOG - Right Thumb



(d) Ranked DET - Provider DERMALOG - Left Thumb

Figure 49:

Ranked DET - Provider ID3. The set of all comparisons are partitioned into three groups based on the pair-wise NFIQ 2 scores of the images being compared. The lowest quality set contains comparisons with pairwise quality in the lowest 15 percentile. The highest quality set contains comparisons with pairwise quality in the highest 15 percentile. The rest of the comparisons, namely the middle 70%, make up the third set. The DETs are connected at the same score threshold values (brown lines). Lower false non-match rate and false match rates are expected for higher quality images. That means well separated curves in each cell, with the DET curve corresponding to the lowest NFIQ 2 values appearing above, and the DET curve of highest NFIQ 2 values below all the other curves.

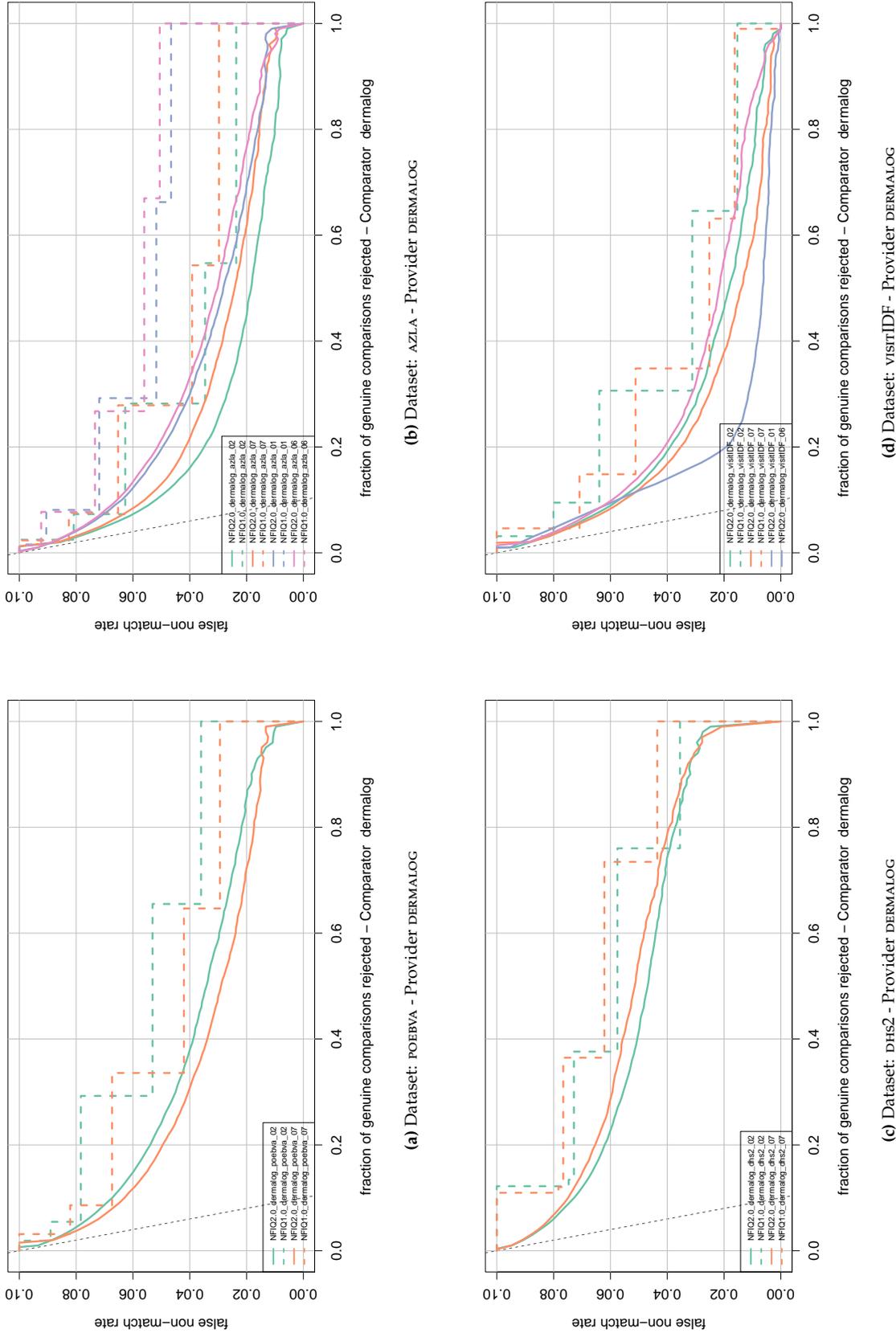


Figure 50:

Error vs. reject curve - Provider DERMALOG. The threshold is set to give an initial false non-match rate of 0.1. The gray dotted line shows the ideal case where the rejection of the ten percent lowest quality results in zero false non-match rate. NFIQ 2 is a better predictive of performance than NFIQ 1.0 because its error vs. reject curves are closer to the gray dotted line.

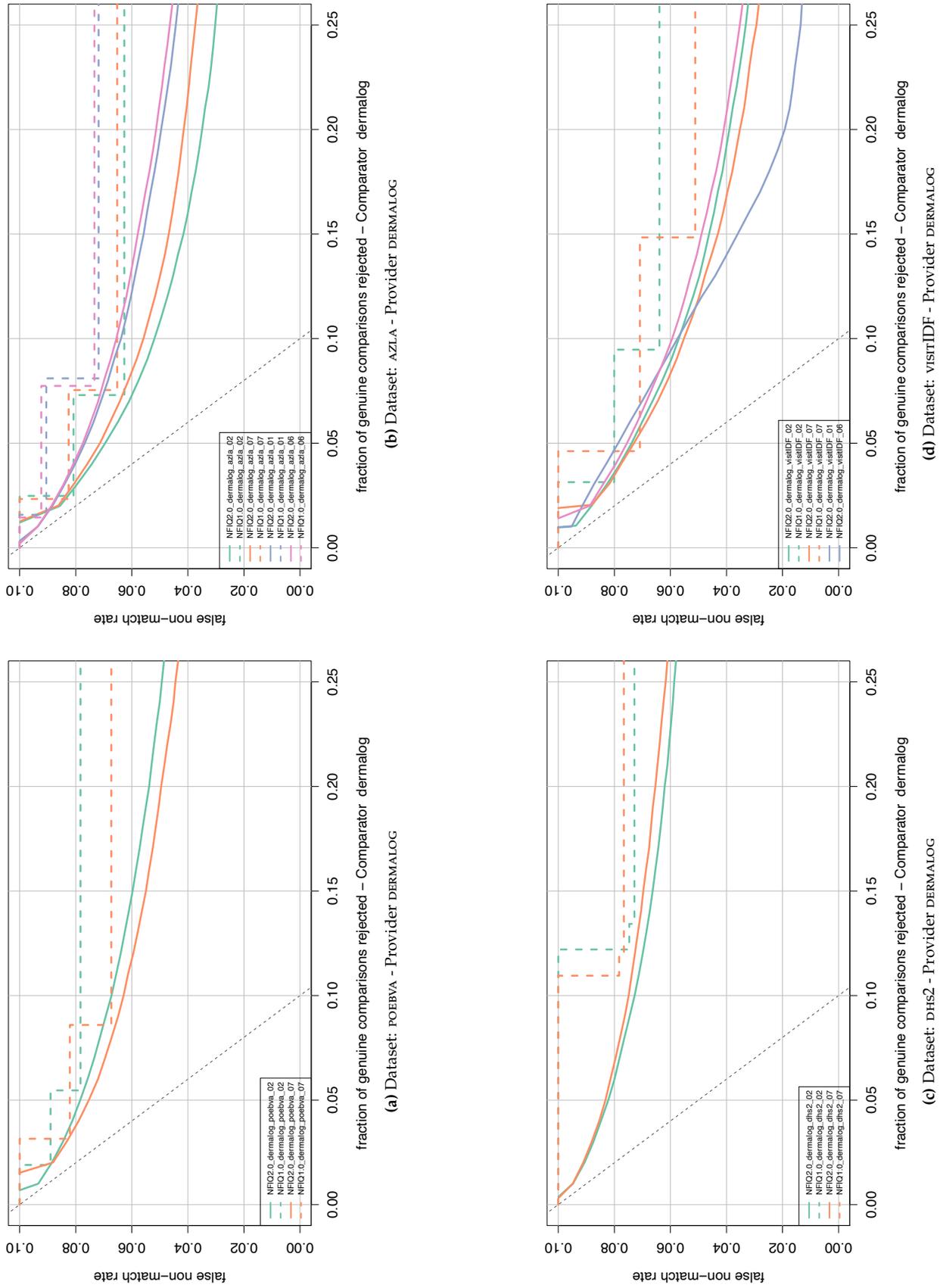


Figure 51:

Error vs. reject curve $\leq 25\%$. Provider DERMALOG. The threshold is set to give an initial false non-match rate of 0.1. The gray dotted line shows the ideal case where the rejection of the ten percent lowest quality results in zero false non-match rate. Same result as in fig. 50 but for rejection rate of up to 25%.

Appendix H

Results for provider ID3

This appendix contains results for provider ID3.

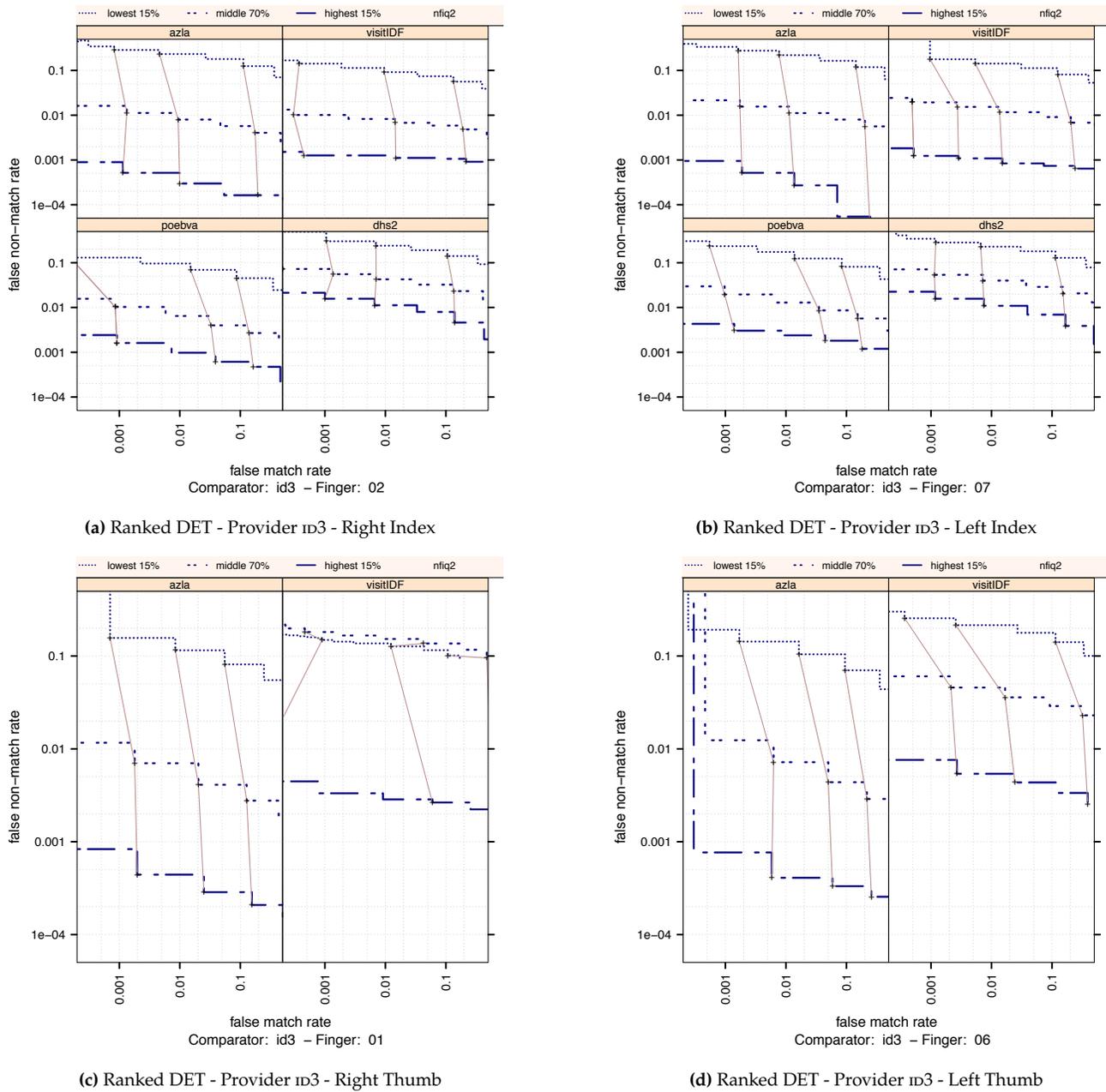


Figure 52:

Ranked DET - Provider id3. The set of all comparisons are partitioned into three groups based on the pair-wise NFIQ 2 scores of the images being compared. The lowest quality set contains comparisons with pairwise quality in the lowest 15 percentile. The highest quality set contains comparisons with pairwise quality in the highest 15 percentile. The rest of the comparisons, namely the middle 70%, make up the third set. The DETs are connected at the same score threshold values (brown lines). Lower false non-match rate and false match rates are expected for higher quality images. That means well separated curves in each cell, with the DET curve corresponding to the lowest NFIQ 2 values appearing above, and the DET curve of highest NFIQ 2 values below all the other curves.

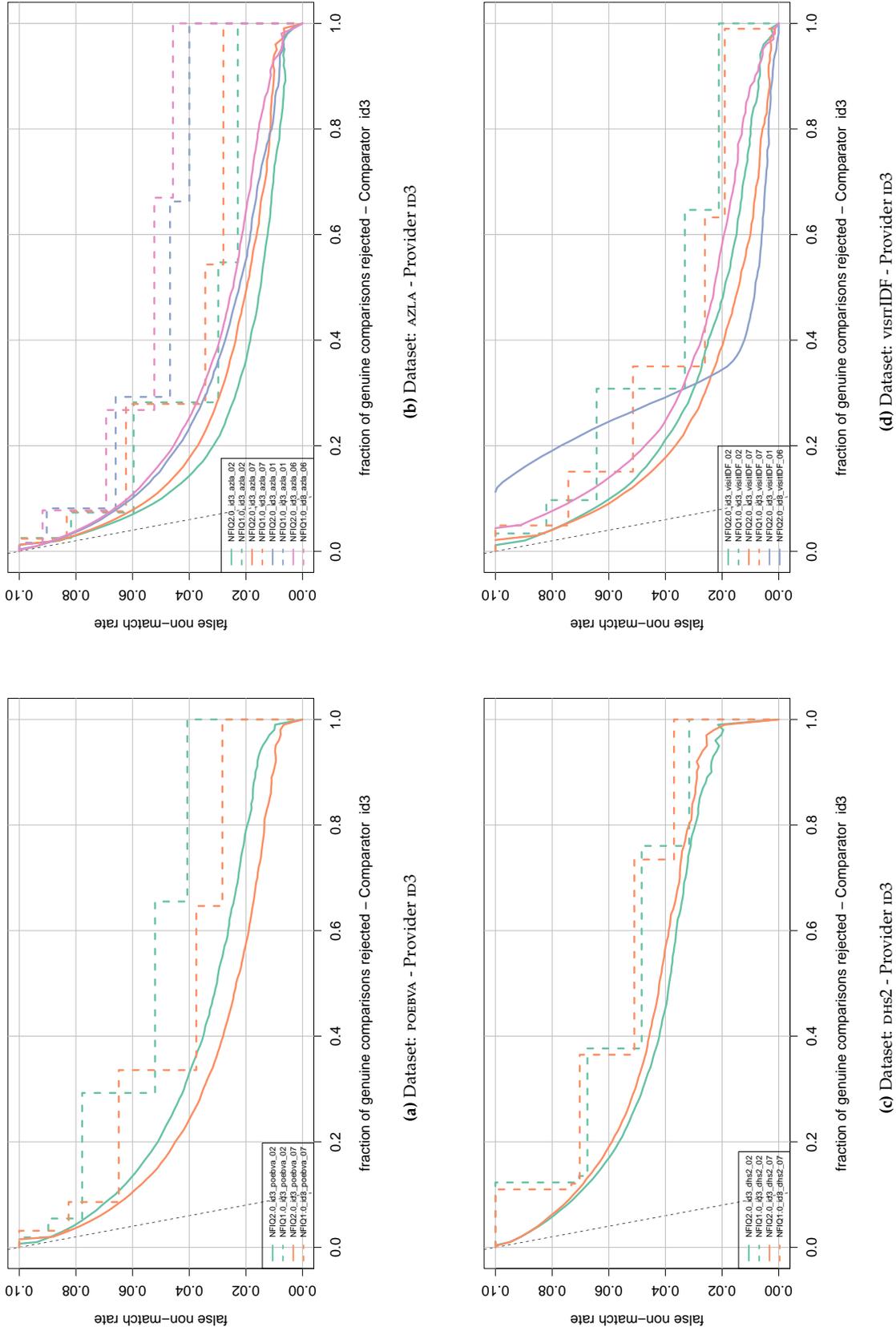


Figure 53:

Error vs. reject curve - Provider id3. The threshold is set to give an initial false non-match rate of 0.1. The gray dotted line shows the ideal case where the rejection of the ten percent lowest quality results in zero false non-match rate. NFIQ 2 is a better predictive of performance than NFIQ 1.0 because its error vs. reject curves are closer to the gray dotted line.

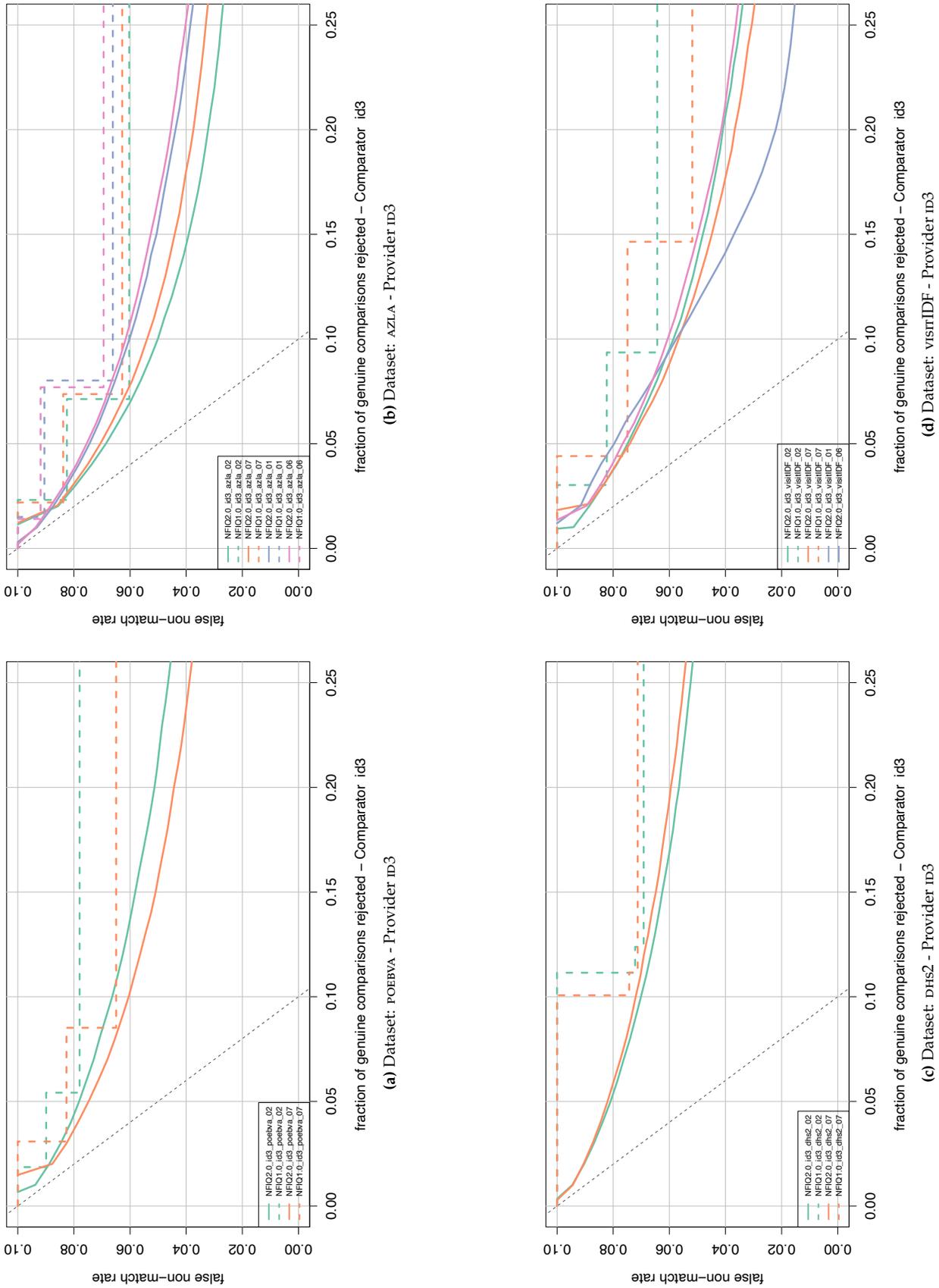


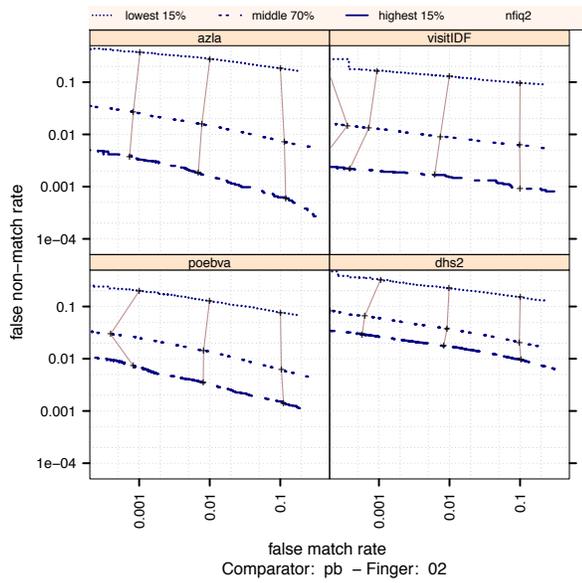
Figure 54:

Error vs. reject curve $\leq 25\%$. Provider id3. The threshold is set to give an initial false non-match rate of 0.1. The gray dotted line shows the ideal case where the rejection of the ten percent lowest quality results in zero false non-match rate. Same result as in fig. 53 but for rejection rate of up to 25%.

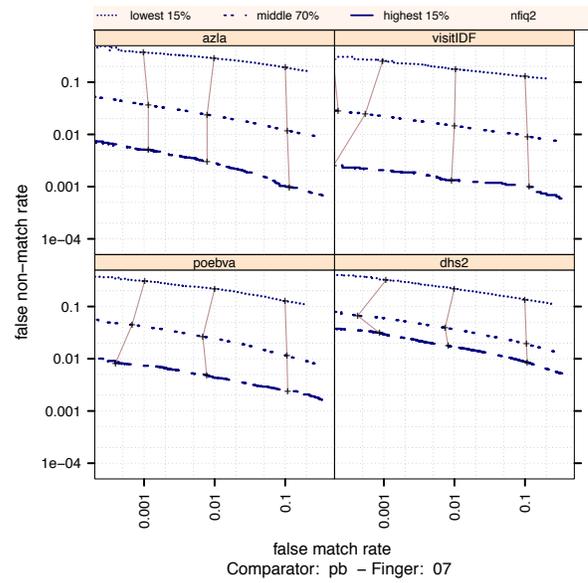
Appendix I

Results for provider PB

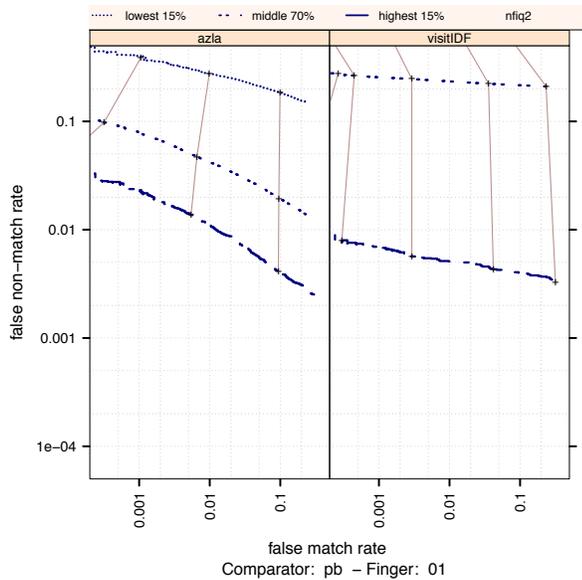
This appendix contains results for provider PB.



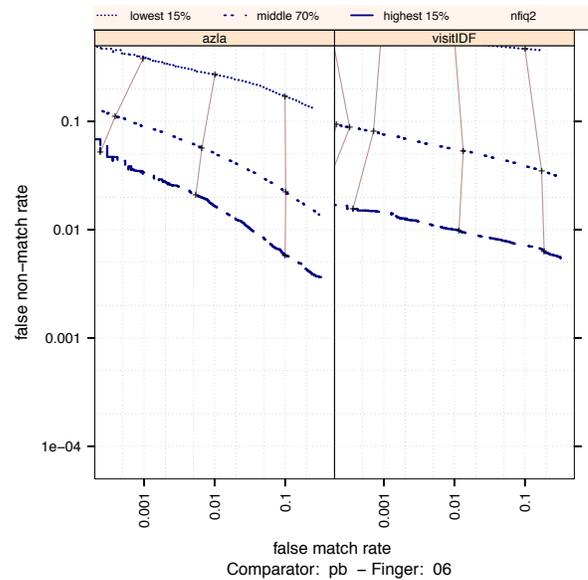
(a) Ranked DET - Provider pb. Right Index



(b) Ranked DET - Provider pb. Left Index



(c) Ranked DET - Provider pb. Right Thumb



(d) Ranked DET - Provider pb. Left Thumb

Figure 55:

Ranked DET - Provider pb. The set of all comparisons are partitioned into three groups based on the pair-wise NFIQ 2 scores of the images being compared. The lowest quality set contains comparisons with pairwise quality in the lowest 15 percentile. The highest quality set contains comparisons with pairwise quality in the highest 15 percentile. The rest of the comparisons, namely the middle 70%, make up the third set. The DETs are connected at the same score threshold values (brown lines). Lower false non-match rate and false match rates are expected for higher quality images. That means well separated curves in each cell, with the DET curve corresponding to the lowest NFIQ 2 values appearing above, and the DET curve of highest NFIQ 2 values below all the other curves.

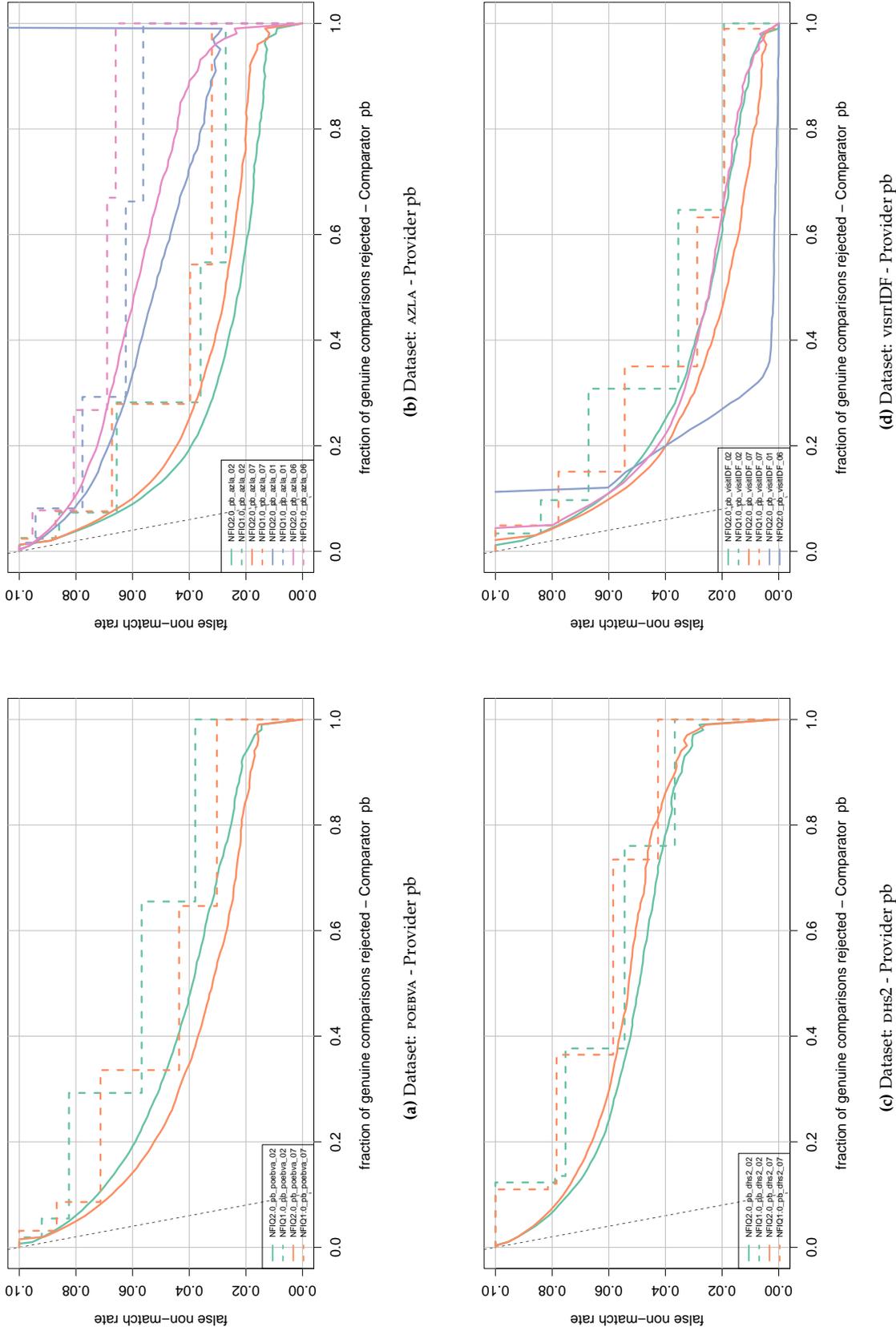


Figure 56:

Error vs. reject curve - Provider pb. The threshold is set to give an initial false non-match rate of 0.1. The gray dotted line shows the ideal case where the rejection of the ten percent lowest quality results in zero false non-match rate. NFIQ 2 is a better predictive of performance than NFIQ 1.0 because its error vs. reject curves are closer to the gray dotted line.

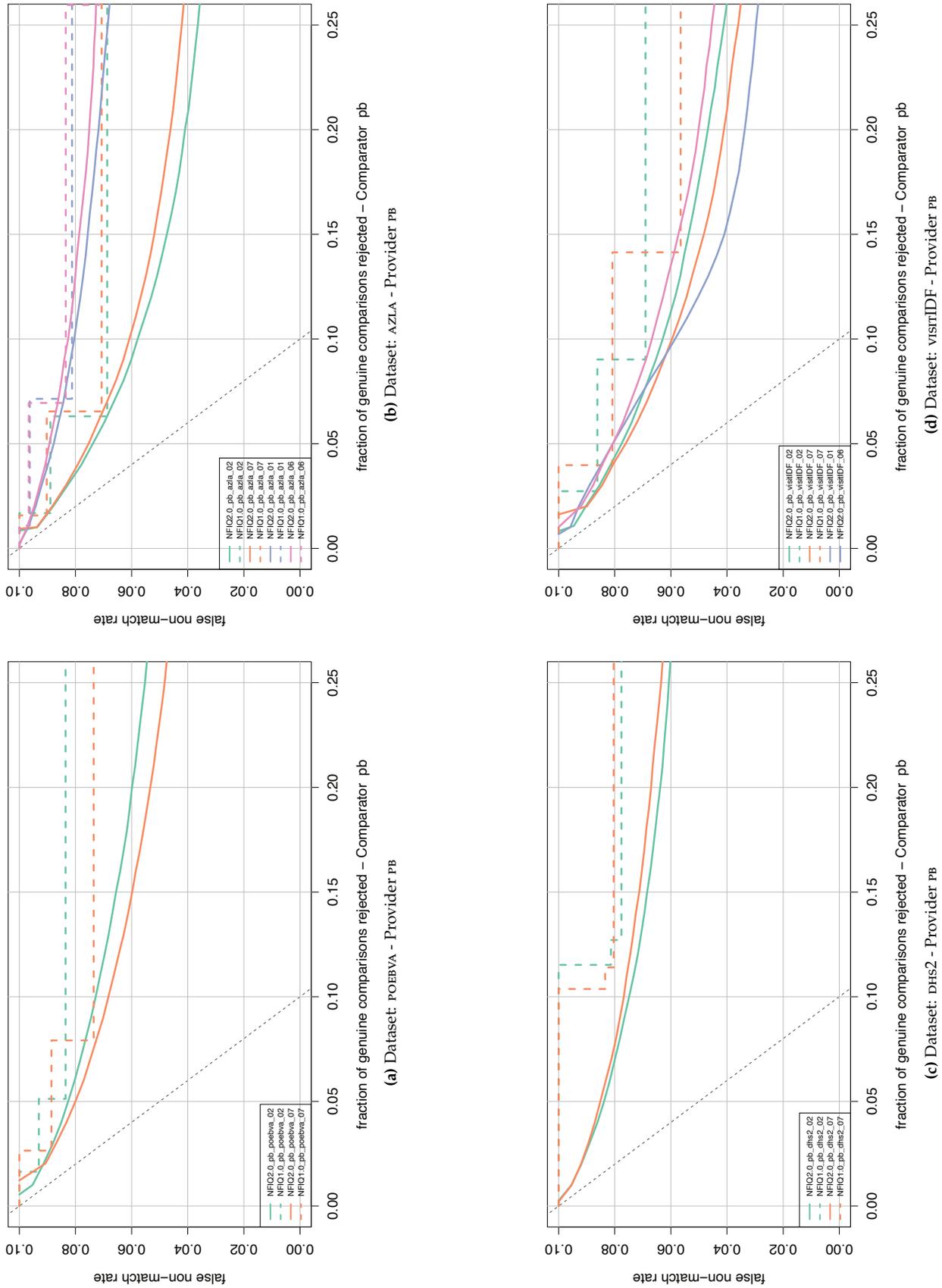


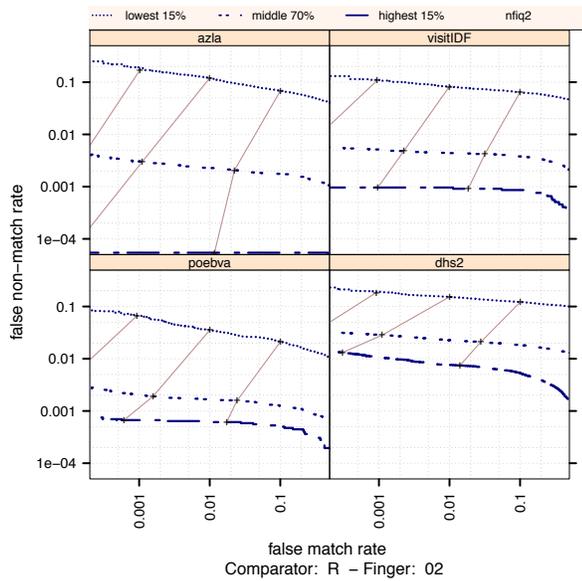
Figure 57:

Error vs. reject curve $\leq 25\%$. Provider pb. The threshold is set to give an initial false non-match rate of 0.1. The gray dotted line shows the ideal case where the rejection of the ten percent lowest quality results in zero false non-match rate. Same result as in fig. 56 but for rejection rate of up to 25%.

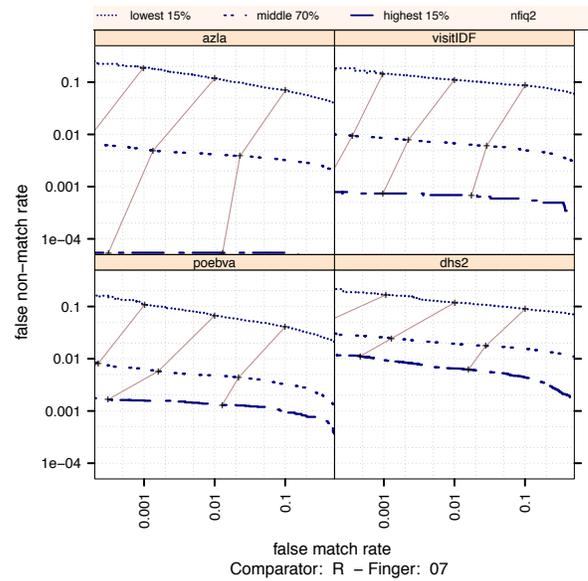
Appendix J

Results for provider R

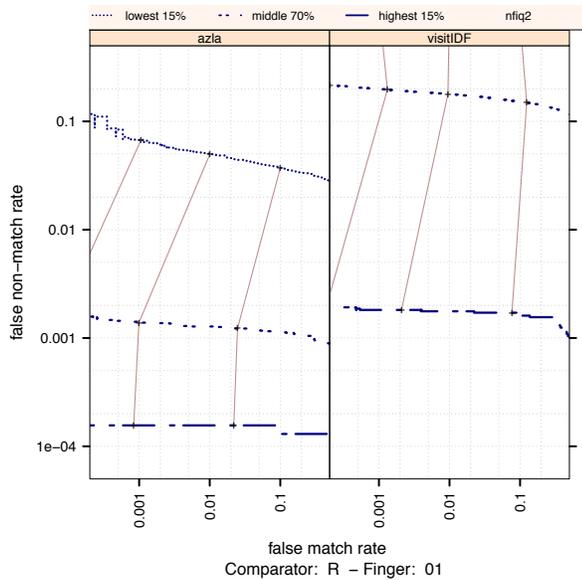
This appendix contains results for provider R.



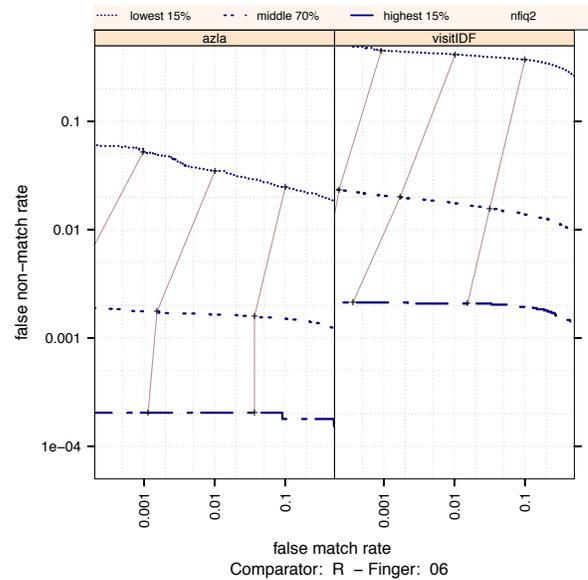
(a) Ranked DET - Provider R - Right Index



(b) Ranked DET - Provider R - Left Index



(c) Ranked DET - Provider R - Right Thumb



(d) Ranked DET - Provider R - Left Thumb

Figure 58:

Ranked DET - Provider R. The set of all comparisons are partitioned into three groups based on the pair-wise NFIQ 2 scores of the images being compared. The lowest quality set contains comparisons with pairwise quality in the lowest 15 percentile. The highest quality set contains comparisons with pairwise quality in the highest 15 percentile. The rest of the comparisons, namely the middle 70%, make up the third set. The DETs are connected at the same score threshold values (brown lines). Lower false non-match rate and false match rates are expected for higher quality images. That means well separated curves in each cell, with the DET curve corresponding to the lowest NFIQ 2 values appearing above, and the DET curve of highest NFIQ 2 values below all the other curves.

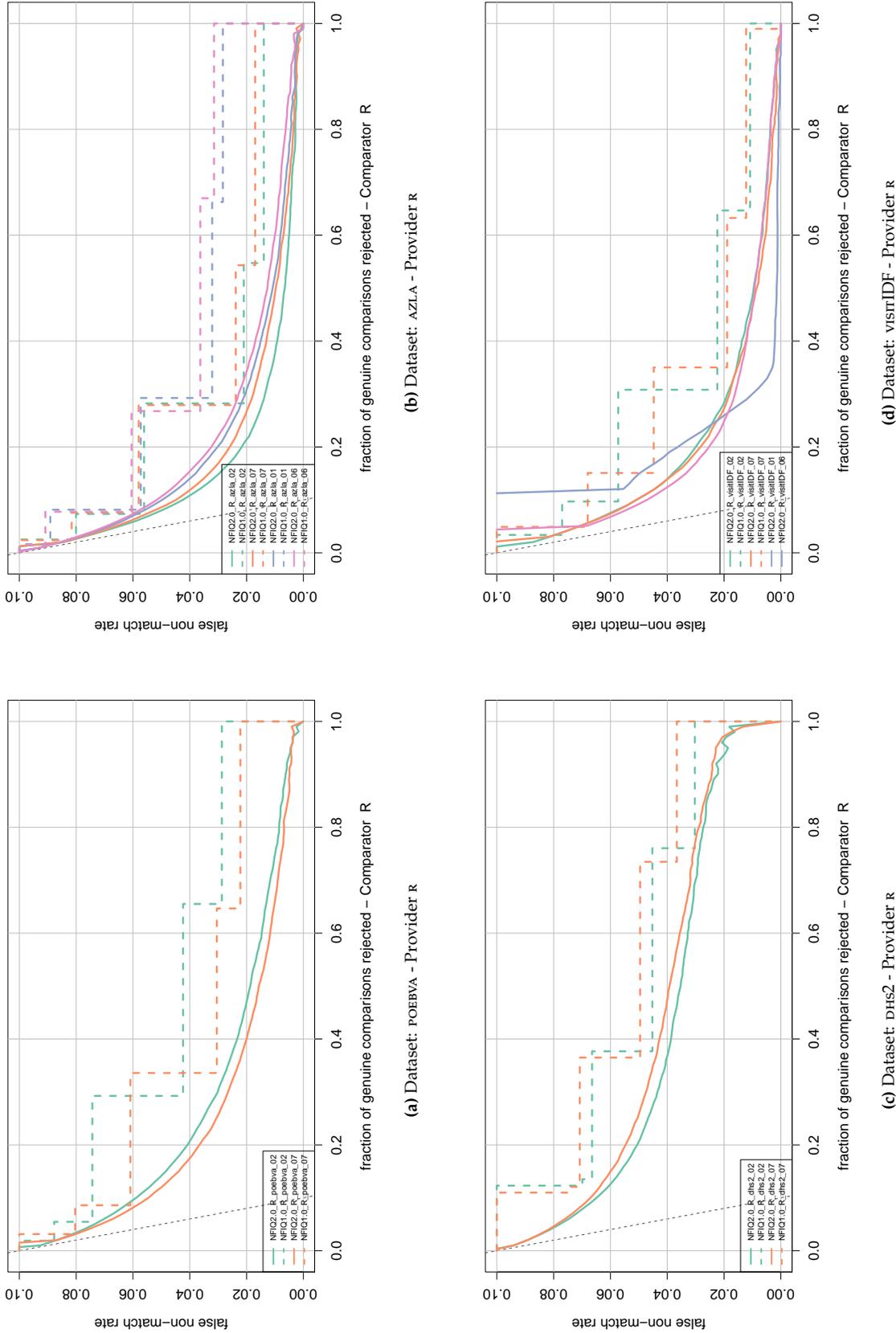


Figure 59:

Error vs. reject curve - Provider R. The threshold is set to give an initial false non-match rate of 0.1. The gray dotted line shows the ideal case where the rejection of the ten percent lowest quality results in zero false non-match rate. NFIQ 2 is a better predictive of performance than NFIQ 1.0 because its error vs. reject curves are closer to the gray dotted line.

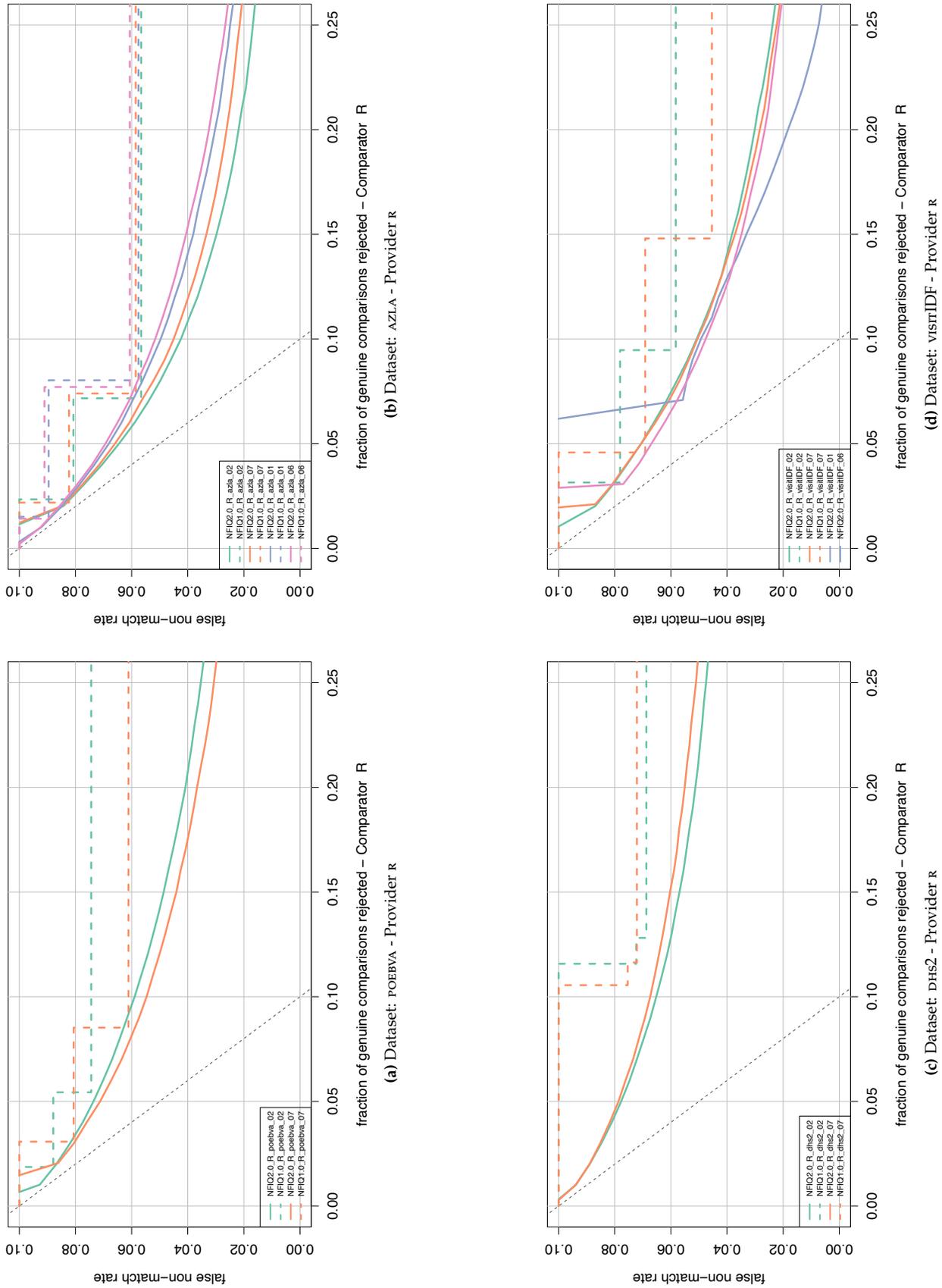


Figure 60:

Error vs. reject curve $\leq 25\%$. Provider r. The threshold is set to give an initial false non-match rate of 0.1. The gray dotted line shows the ideal case where the rejection of the ten percent lowest quality results in zero false non-match rate. Same result as in fig. 59 but for rejection rate of up to 25%.

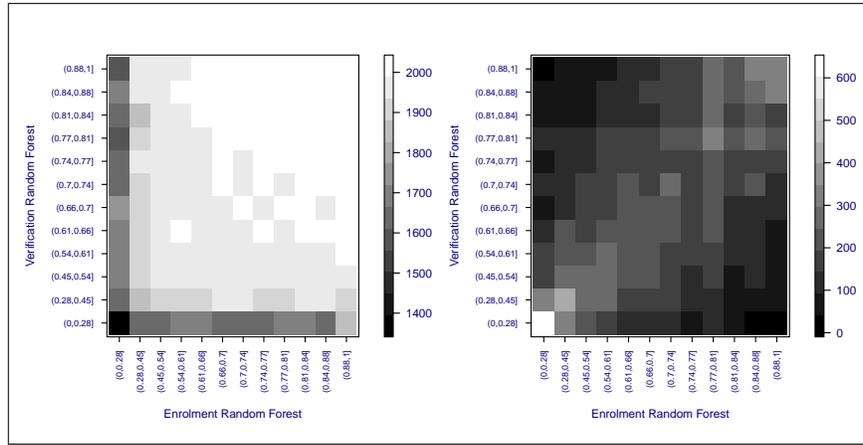


Figure 61: Heatmap for provider κ on dataset azla: Left plot shows the mean comparison score, right plot shows the sample count.

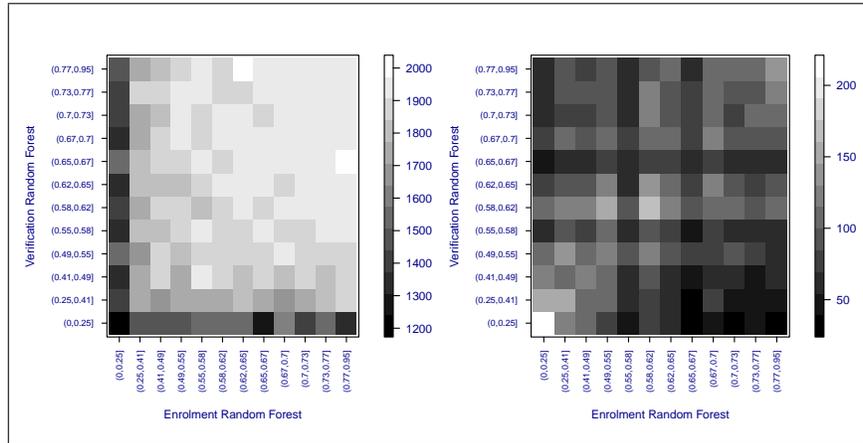


Figure 62: Heatmap for provider κ on dataset dhs2: Left plot shows the mean comparison score, right plot shows the sample count.

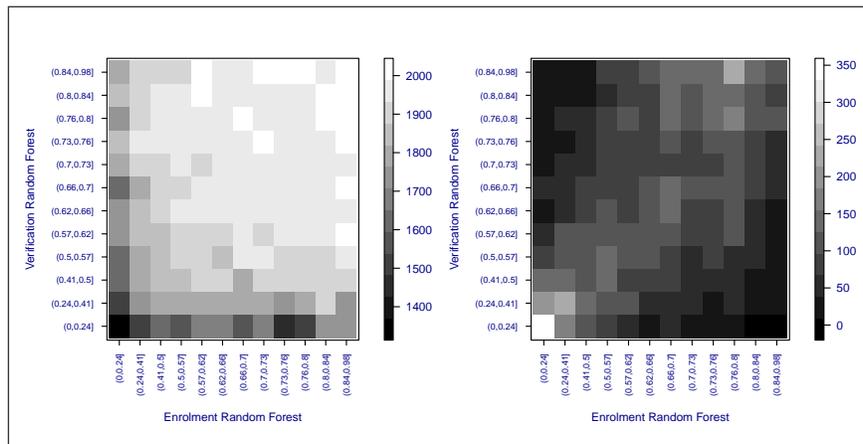


Figure 63: Heatmap for provider κ on dataset poebva: Left plot shows the mean comparison score, right plot shows the sample count.

Appendix K

Results for NFIQ 2 participants

Each plot in figs. 64 to 67 shows detection error tradeoffs of the nine NFIQ 2 participants for each of the four datasets used for training.

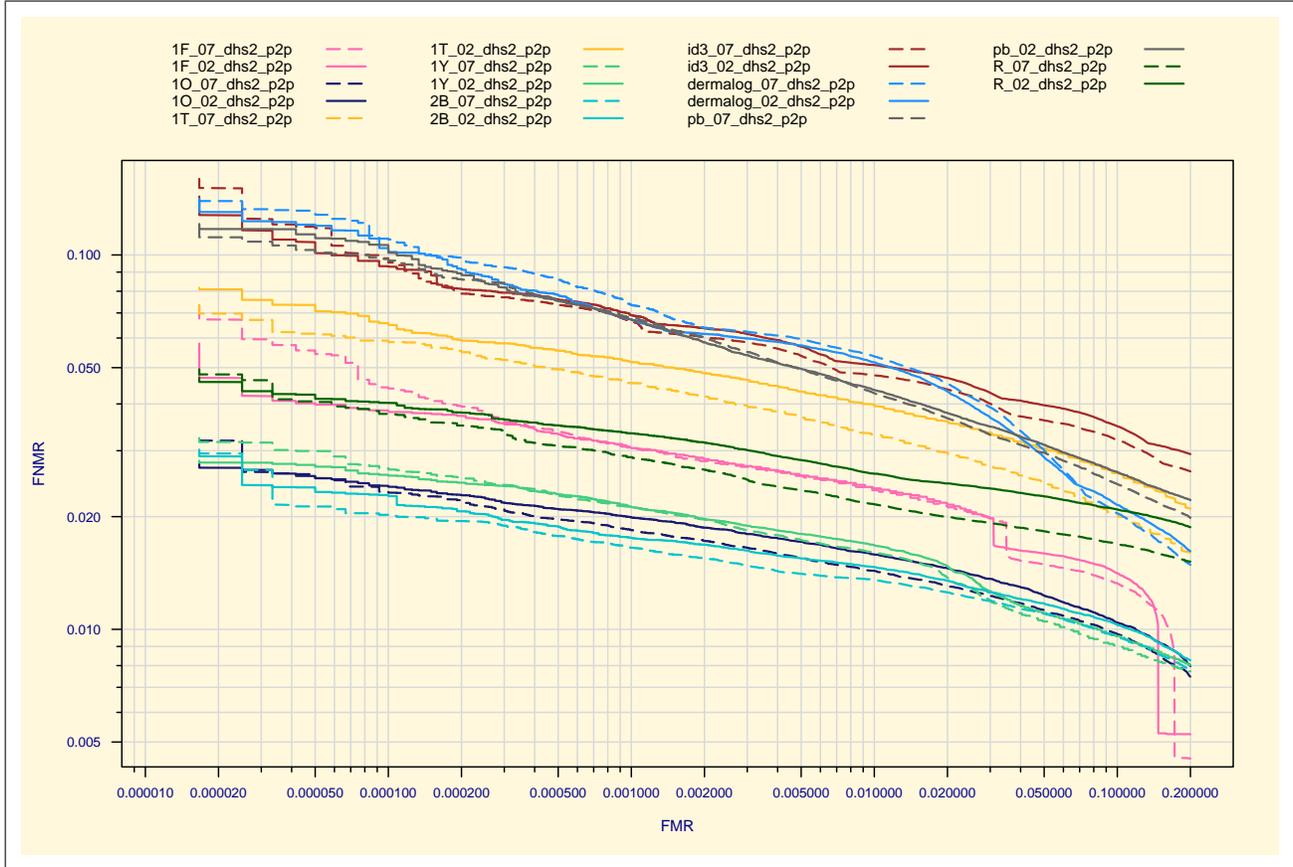


Figure 64: Detection Error Tradeoff (DET) curves of NFIQ 2 submissions on dhs2 images.

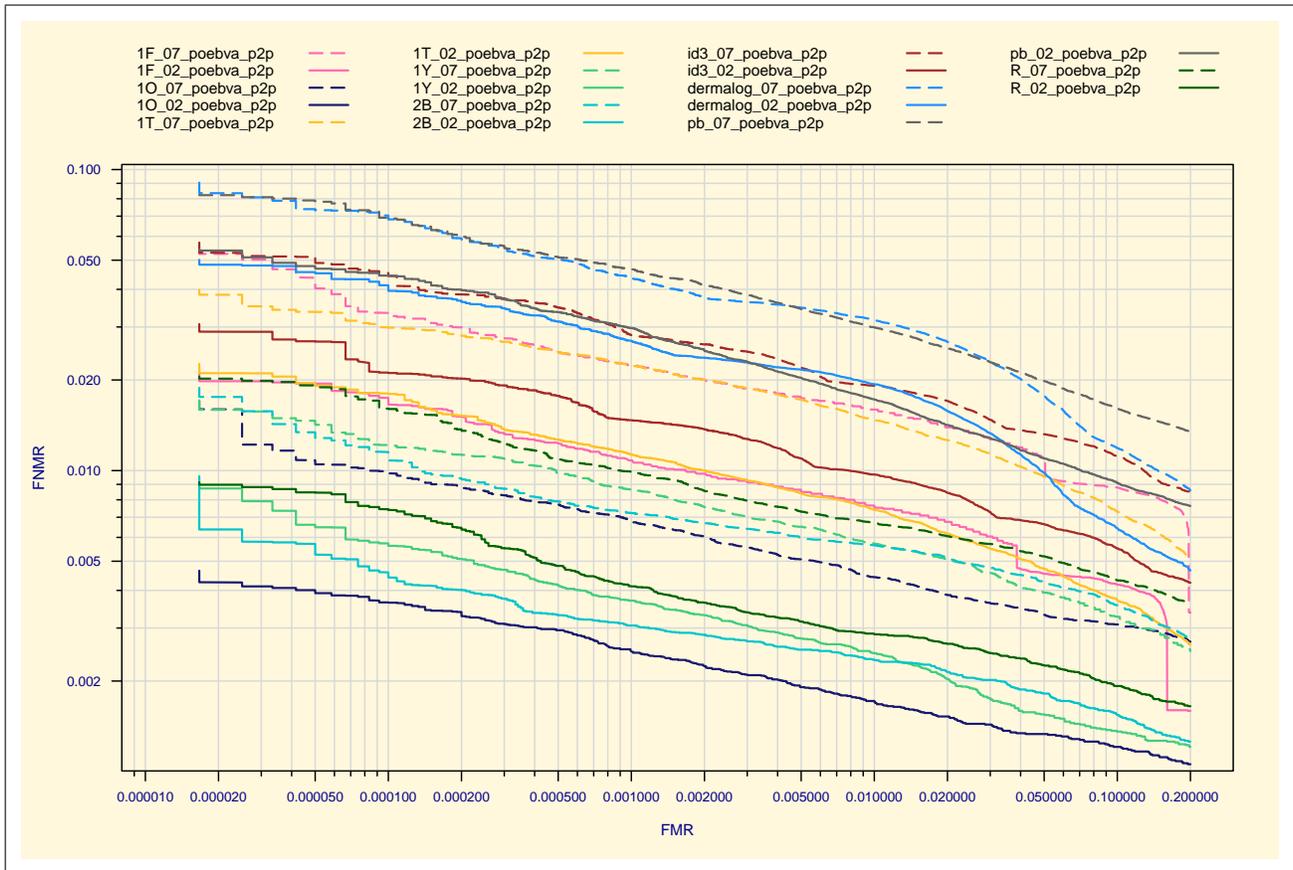


Figure 65: Detection Error Tradeoff (DET) curves of NFIQ 2 submissions on POEBVA images.

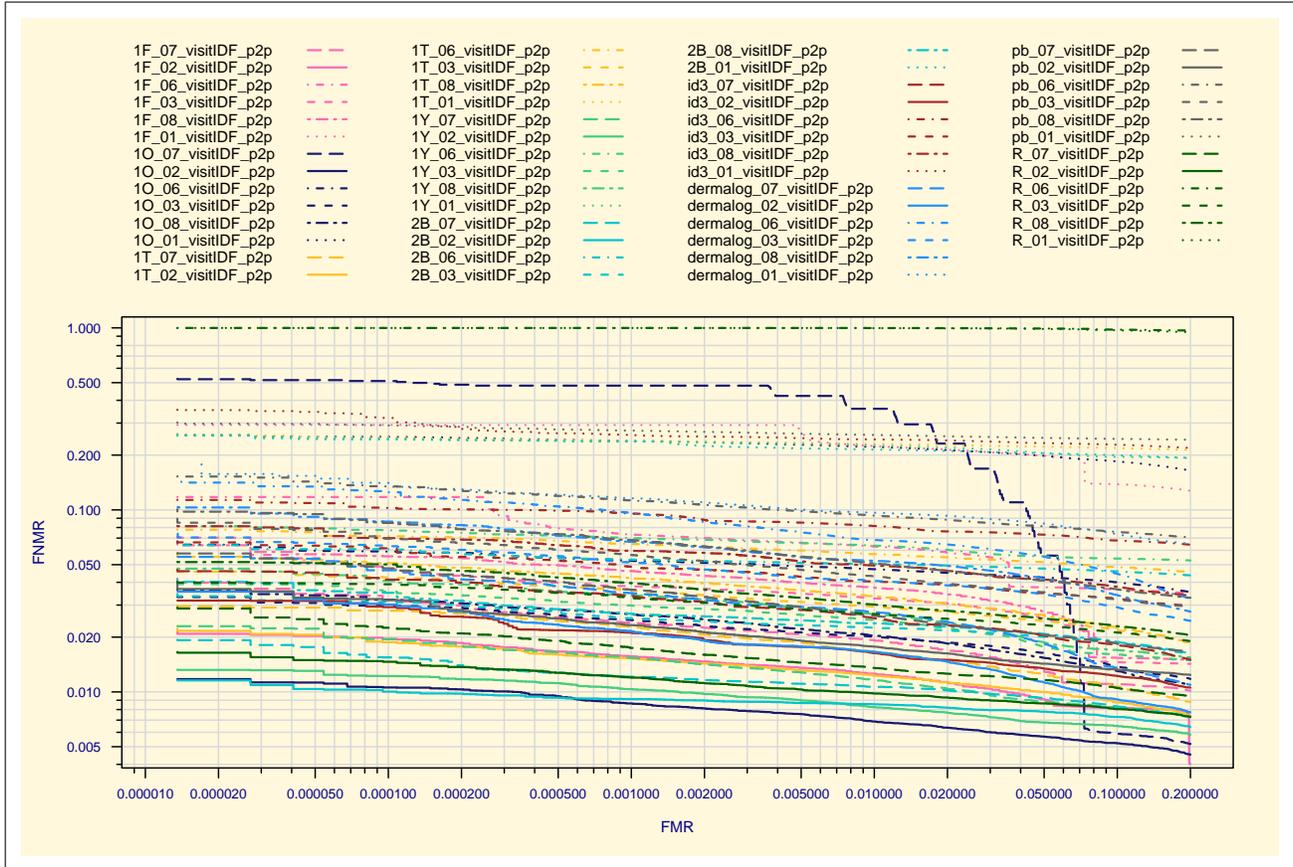


Figure 66: Detection Error Tradeoff (DET) curves of NFIQ 2 submissions on visit images.

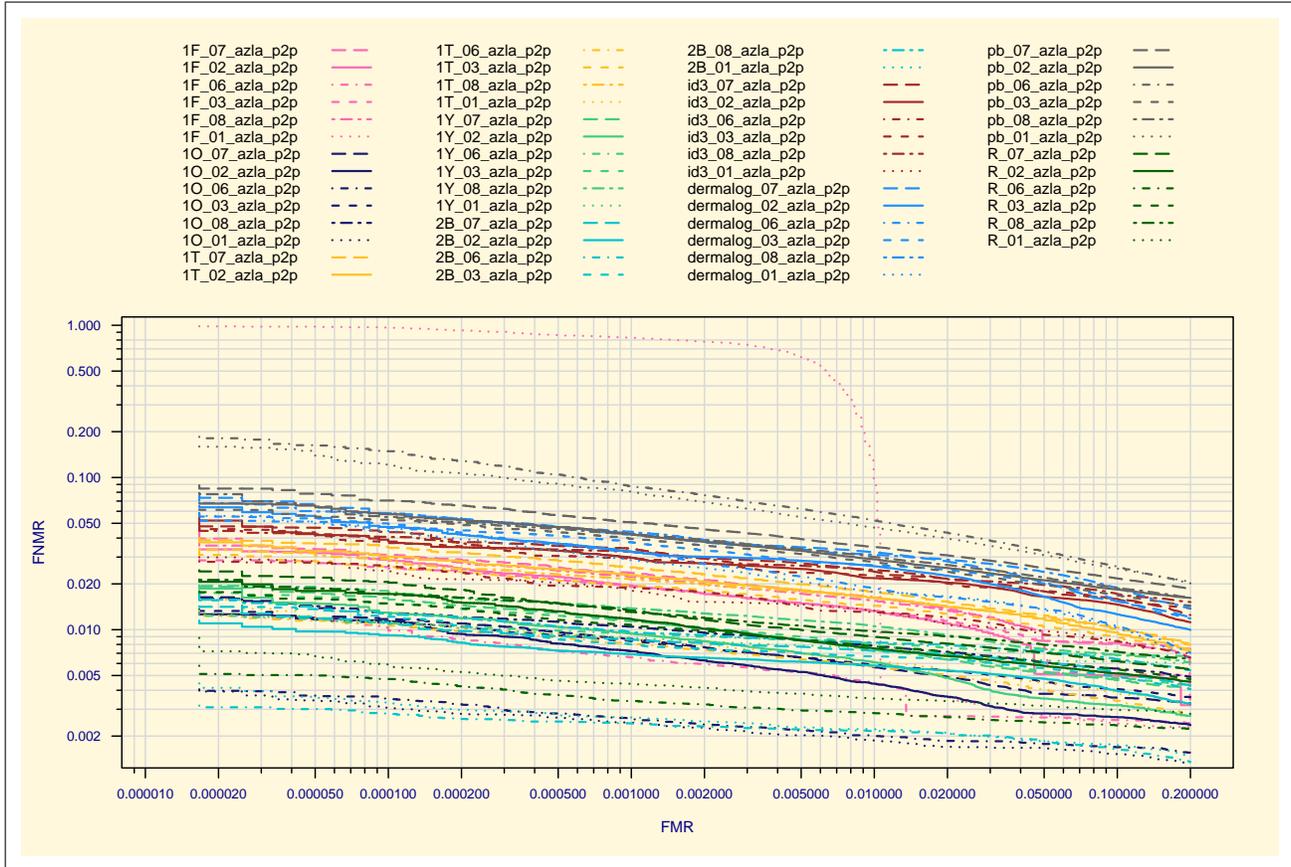


Figure 67: Detection Error Tradeoff (DET) curves of NFIQ 2 submissions on AZLA images.

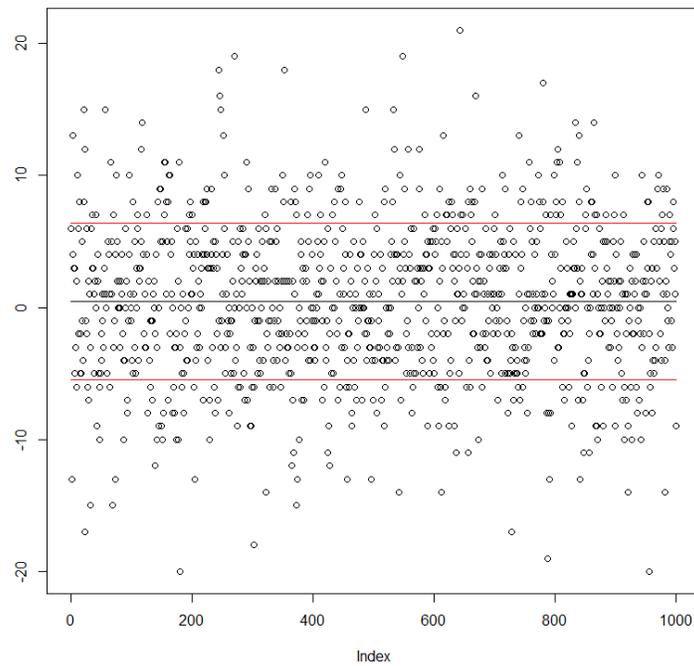


Figure 68: Deviations of NFIQ2 values between uncompressed and wsq-compressed (factor 4) images of 1000 fingerprints from the MCYT database.

Appendix L

Impact of WSQ Compression on the NFIQ 2 Values

During evaluation of the NFIQ2 algorithm, we observed deviations between WSQ compressed and uncompressed images of fingerprints. Figure 68 shows, of 1000 images from the MCYT-100 database, the difference $x-y$ between the NFIQ2 values x and y of the uncompressed (BMP) and compressed (WSQ) images, respectively, when using a compression ratio of 4. The maximum deviation is approximately 20.

The deviations do not become much stronger for higher compression ratios. Figure 69 shows the deviations of the same fingerprints when using a compression ratio of 8.

In order to determine which quality features are chiefly responsible for the deviations, we computed, for each of the NFIQ2 quality features, the mean of the relative absolute deviation, defined as $|x_1 - x_2|/|x_1 + x_2|$, between the feature values x_1 and x_2 of the compressed and uncompressed image, respectively.

It turned out that the lower OCL bins deviate most, but strong deviations are also observed for LCS, in particular for the bins 2, 8, and 9. FDA deviates slightly less, but almost uniformly across all bins.

Another observation is that, sometimes, even the sum of the bins of a feature differs considerably (up to 18% for OCL, up to 8% for FDA, LCS, OF and RVU) between compressed and uncompressed images. Since the sum of the bin is the total number of image blocks considered, we can conclude that WSQ compression may change the foreground area analyzed by the features. However, the sum of the bins and the deviation thereof between BMP and WSQ also varies between different features. For instance, a strong deviation of the number of blocks used by OCL, for instance, does not necessarily imply that the numbers of blocks used by the other features deviate as well. However, the average deviation of the sum of bins per feature is small (below 0.4%).

An inspection of images, where stronger deviations of the sums over the bins was observed, yields the (subjective)

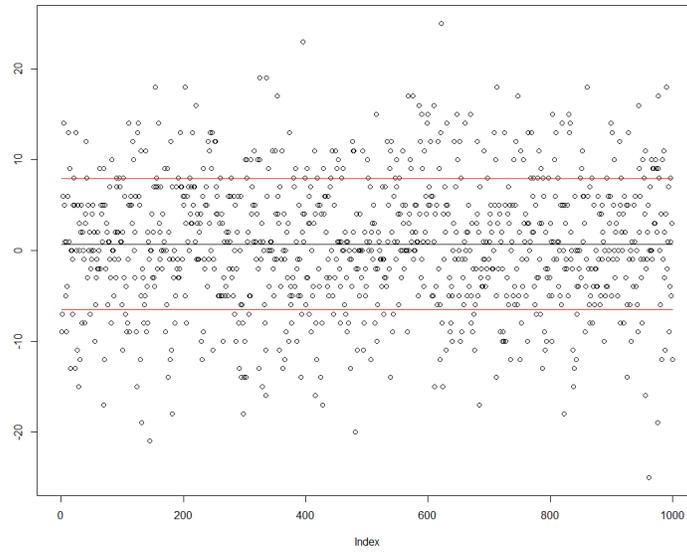


Figure 69: Deviations of NFIQ2 values between uncompressed and wsq-compressed (factor 8) images of 1000 fingerprints from the MCYT database.

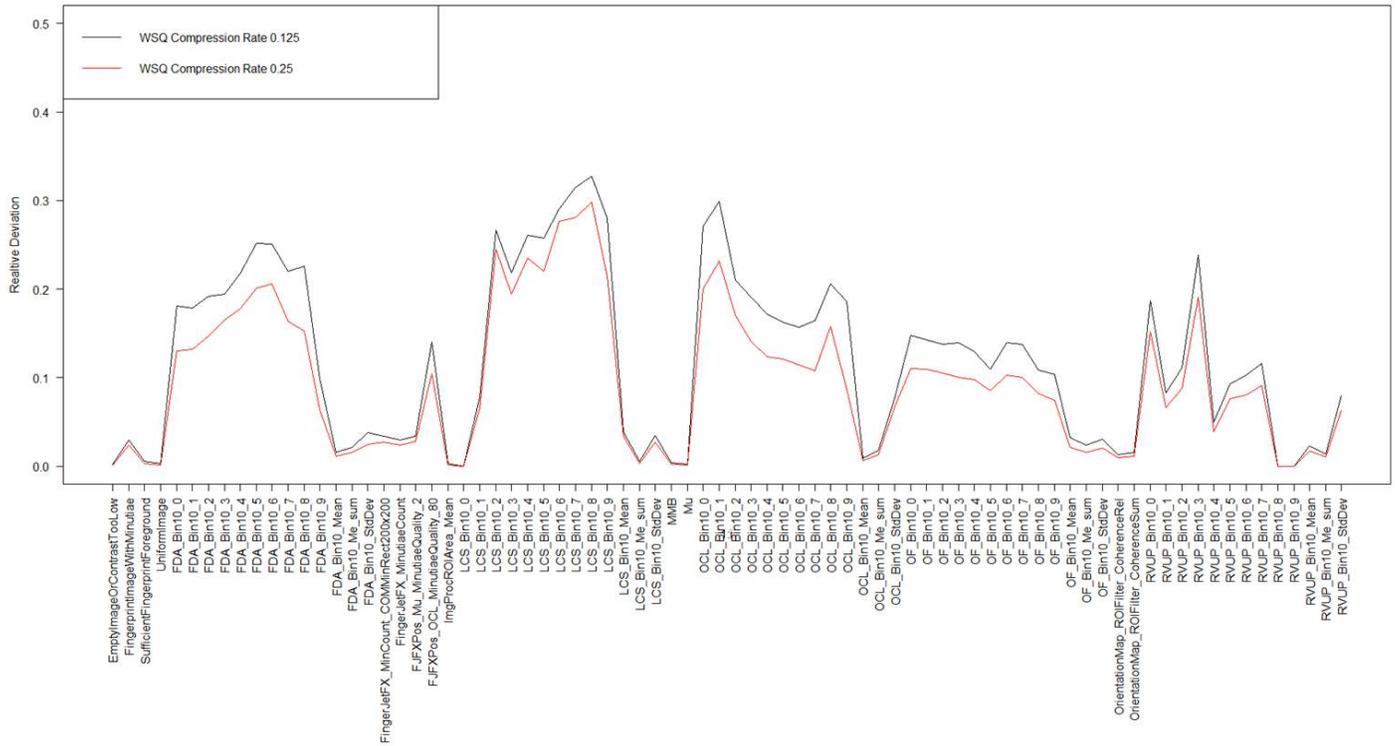


Figure 70: Average relative absolute deviations ($|x_1 - x_2| / |x_1 + x_2|$) in the values of the NFIQ2 quality features values between uncompressed and wsq-compressed images of 1000 fingerprints from the MCYT database.

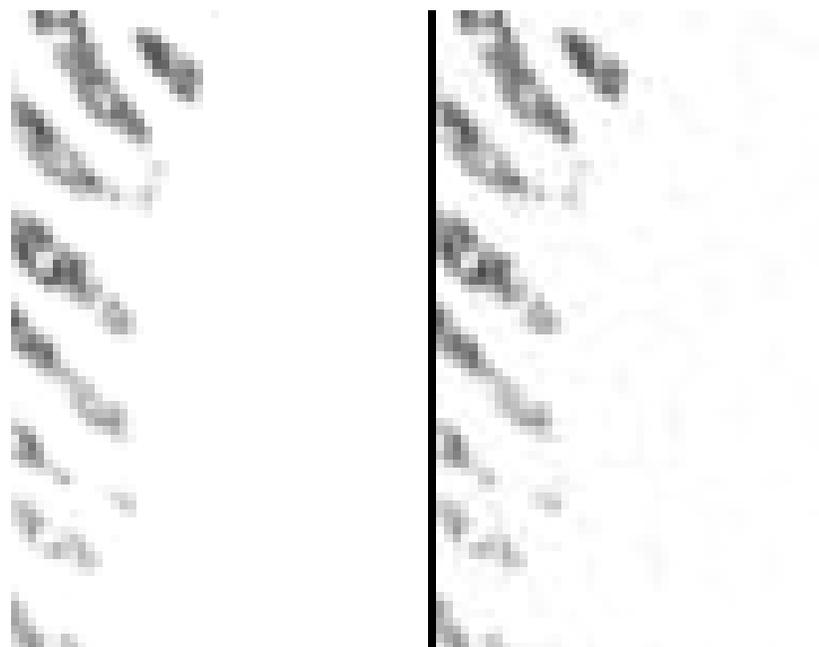


Figure 71: Artifacts at the boundary of a fingerprint resulting from WSQ compression (factor 8).

impression that, in particular in high contrast images, the WSQ compression results in artifacts at the edge of the fingerprint, which may result in consideration of additional image blocks in that area. An example is shown in the Figure 71.

The artifacts become more visible with the naked eye when reducing gamma value to 0.1 as shown in Figure 72.

Also, high deviations are observed for images having ghost image artifacts in the background, which become blurred by WSQ compression. An example is given in the Figure 73.

Again, for better visualization, we present the same image with gamma value reduced to 0.1 as shown in Figure 74.

It seems that the WSQ compression does not only blur the artifacts from the original image but also produces local line-patterns which may result in significant higher values in quality features that measure the clarity or uniformity of ridge-valley structures. A potential remedy could be a pre-processing step applying a brightness threshold on small image blocks, so that areas with artifacts are whitened or neglected from further analysis.

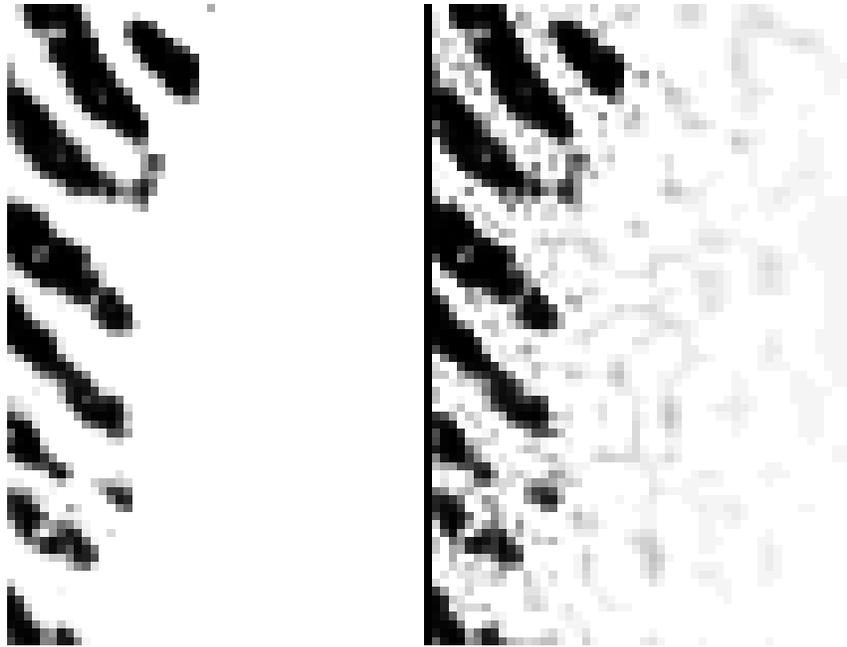


Figure 72: Artifacts at the boundary of a fingerprint resulting from WSQ compression (factor 8) after setting gamma value to 0.1.

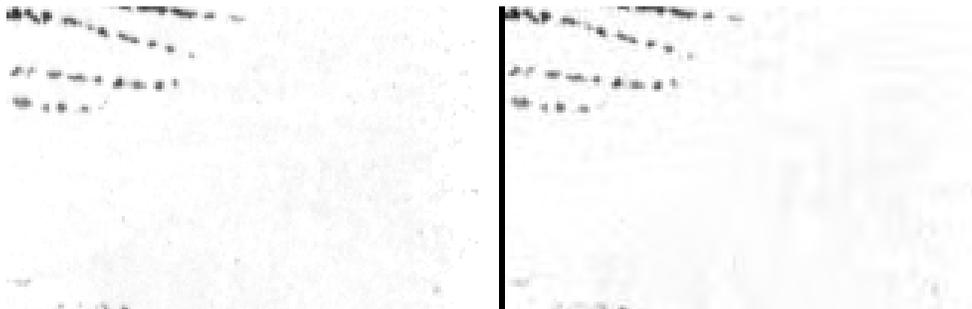


Figure 73: Effect of WSQ compression (factor 8) on artifacts in fingerprint image.

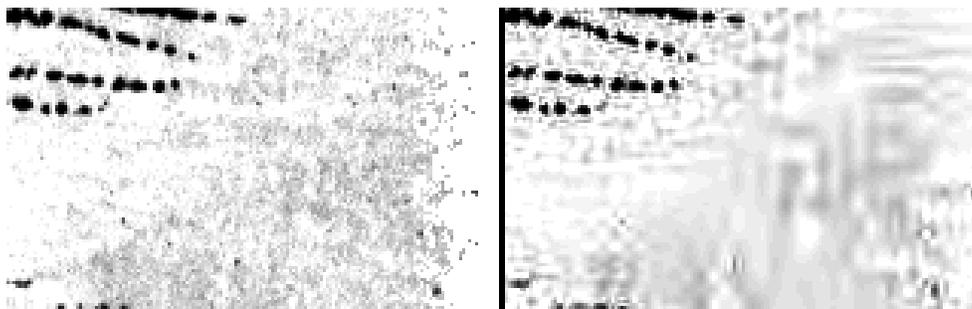


Figure 74: Effect of WSQ compression (factor 8) on artifacts in fingerprint image after setting gamma value to 0.1.

Appendix M

Software Releases

M.1 Operational Kit

The NFIQ 2 Operational Kit is the open-source software release of NFIQ 2 for operational use. It contains the actual released software source of NFIQ 2— including all finally selected features and training data for NFIQ 2.

Furthermore, it consists of a command line tool that provides access to the NFIQ 2 algorithm easily. In the single program mode, the command line tool of the Operational Kit allows to compute the NFIQ 2 quality score, actionable quality values and, optionally, the individual quality feature values for an input fingerprint image. Alternatively, the tool can be used in a batch mode, allowing to compute the NFIQ 2 quality scores, actionable quality values and, optionally, the individual quality feature values for a list of fingerprint images by a single invocation.

The exact usage of the command line tool is described in the NFIQ 2 User Manual.

M.2 Research Kit

The NFIQ 2 Research Kit is based on a so called NFIQ 2 framework that is designed to serve as a layer between modules and applications that use the framework and the available modules. It offers plug and play of different combinations of quality features and machine learning techniques in order to ease the research and development process of NFIQ releases. Hence, it allows flexible integration and sharing and re-using of results, e.g. for further NFIQ releases and variants.

Figure 75 shows a schematic representation of the framework architecture. It consists of four module interfaces:

- The Input/Output Module interface abstracts the storage of all relevant project data. Such data consists of fingerprint images, comparison scores, quality features and utility values. By implementing dedicated Input/Output Modules, developers can use any background systems for storage and retrieval of relevant data. One example could be an Input/Output Module accessing a database in the background, which organizes and stores the data. Another possibility could be to use the file system as a basis for data storage and retrieval.
- The Quality Feature Extraction Module interface allows exchanging algorithms for generating quality features.
- The Utility Estimation Module interface offers functionality for calculating different utility values. Additionally, the interface offers the possibility to fuse utility values computed from different comparison algorithms.
- The Machine Learning Module interface abstracts the machine learning process.

Additionally, the framework offers functions to convert images into different formats such as raw data, BMP, JPEG, JPEG2000 or WSQ.

Several Input/Output Modules, Utility Estimation Modules and Machine Learning Modules may be registered in the framework, but only one instance of every type of these modules may be selected as being active. Selection of the active module is done by calling a dedicated framework function. The only exceptions are Quality Feature Extraction Modules, which are activated automatically by the framework (after they are registered) due to the fact that several features may be extracted from different modules at the same time. In this case, the selection of the Quality Feature Extraction Module is done by unique feature IDs.

The framework and its modules are implemented as C++ classes. For all four module types, software interfaces (abstract classes) are specified, which define the abstract functions for the according module. All implementations of modules have to be derived of the according interface definition and must implement the mandatory functions. The framework offers some high-level functions, which directly access one or more modules to execute a certain task. Framework functions as well as the interface functions, error handling, data types and structures are described in detail in the NFIQ 2 User Manual.

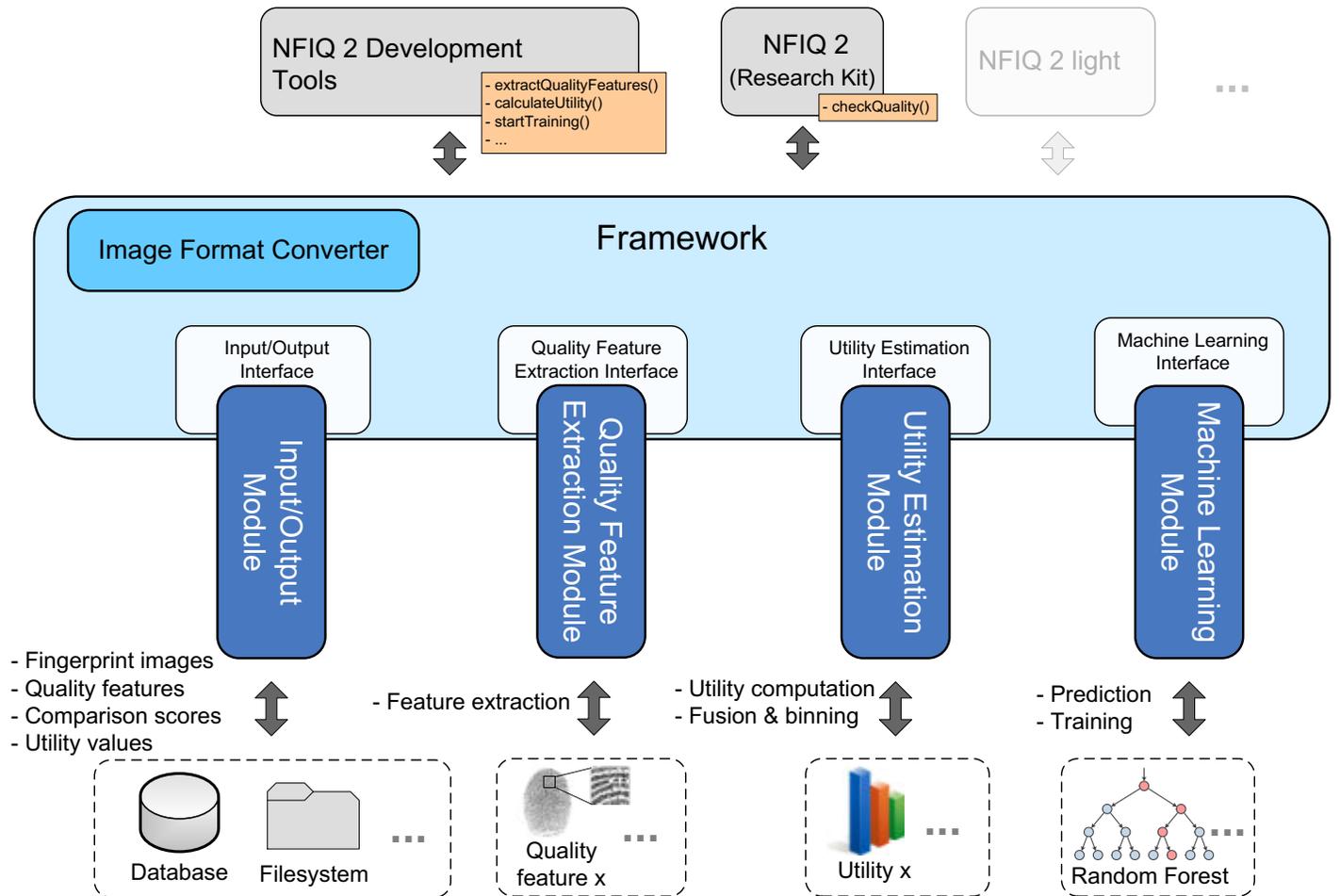


Figure 75: Architecture of the NFIQ 2 Framework used by the NFIQ 2 Research Kit.