NFIQ 2.0

NIST Fingerprint Image Quality

NIST Interagency Report XXXX

DRAFT

FOR COMMENT

DO NOT EXCERPT

April 28, 2016

Acknowledgements

The authors would like to thank the sponsors of this project, Department of Homeland Security (DHS) the Science and Technology Directorate, and the Federal Bureau of Investigation (FBI) 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 2.0 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 2.0 on their data.

Furthermore, the authors extend their gratitude to the fingerprint community and NFIQ 2.0 community in particular for their time and contribution to this project.

Disclaimer

Specific hardware and software products identified in this report were used in order to perform the evaluations described in this document. In no case does identification of any commercial product, trade name, or vendor, imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the products and equipment identified are necessarily the best available for the purpose.

Contents

List of Figures

List of Tables

Executive Summary

NIFQ 2.0 is a revision of the open source software NIST Finger Image Quality NFIQ. In 2004 NIST developed the first publicly available fingerprint quality assessment tool NFIQ. The major innovation of NFIQ 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 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 2.0 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 2.0 provides a higher resolution quality score (in range of 0-100 according to the international biometric sample quality standard ISO/IEC 29794-1:2016 [6] as opposed to 1-5), lower computation complexity, as well as support for quality assessment in mobile platform.

Furthermore, NFIQ 2.0 is the basis for a revision of the Technical Report ISO/IEC 29794-4 Biometric sample quality – Part 4: Finger image [5] into an international standard. Specifically, NFIQ 2.0 quality features are being formally standardized as part of ISO/IEC 29794-4 Biometric sample quality – Part 4: Finger image data and NFIQ 2.0 source code serves as the reference implementation of the standard.

Release Notes

All NFIQ 2.0 related reports, drafts, announcements and news items may be found on the homepage http://fingerprint.nist.gov/nfiq.

- **Concept of Operations**
  NFIQ 2.0 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 2.0 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 2.0 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 libFRFXLL/COPYRIGHT.txt provided with this software are met.
  NFIQ 2.0 employs openCV implementation of random forest. openCV source code is included in NFIQ 2.0 distribution and its copyright notice, and the disclaimer are in OpenCV/doc/license.txt.

- **Source code**
  NFIQ 2.0 source code is publicly available from http://www.nist.gov/itl/iad/ig/development_nfiq_2.cfm. NFIQ 2.0 documentation and user guide are posted at www.nist.gov/itl/iad/ig/development_nfiq_2.cfm. The NFIQ 2.0 compliance test is available as part of NFIQ 2.0 distribution and is documented in Appendix A.

- **Supplemental reports**
  The following supplemental reports will follow:
  Calibration of NFIQ 2.0 values including its mapping to NFIQ 1.0 values.
  Concept and baseline implementation for NFIQ lite.
  NFIQ 2.0 Framework and Research kit.
  Technical guidance on setting quality threshold.
  Technical guidance on use of NFIQ 2.0 for quality assessment of slap impressions.
  Performance of NFIQ 2.0 on rolled impressions.

- **Appendices**
  The main body of this document describes the technical aspects of design, implementation and evaluation of NFIQ 2.0. The appendices consists of compliance test, provider specific results, and impact of WSQ decoder on NFIQ 2.0 values.

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

- **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 2.0 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' [4] 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 2.0 fills this gap by providing a publicly available algorithm that quantifies the utility of a finger image. NFIQ 2.0 is a new and improved version of NFIQ 1.0 developed in consultation and collaboration with users and industry. NFIQ 2.0 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 2.0 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 2.0 software produces a quality score which in compliance with the international biometric sample quality standard ISO/IEC 29794-1:2016 [6] is in [0-100], where 0 means no utility value and 100 is the highest utility value. A value of 255 indicates that computation of quality score has failed.

This document details the design, implementation and evaluation of NFIQ 2.0 'operational kit'. Source code can be obtained from `www.nist.gov/itl/iad/ig/development_nfiq_2.cfm` as well as documentaion and User guide. NFIQ 2.0 compliance test is included in the NFIQ 2.0 distribution and documented in Appendix A of this document.

Additionally, to facilitate and promote research in biometric sample quality, we will release NFIQ 2.0 'research kit' which includes source codes of the framework used during the development of the NFIQ 2.0 algorithm along with source code for computing features that were not selected for final NFIQ 2.0 'operational kit'.

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

Table 1: NFIQ 2.0 providers.

2 Data

Three sources of images were used in the development and evaluation of NFIQ 2.0: 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 2.0.

3 Algorithm submission and use

Participation in NFIQ 2.0 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 2.0 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 2.0 .

Recognition accuracy of these implementations are reported in Appendix K.

[1] http://biometrics.nist.gov/cs_links/quality/NFIQ_2/NFIQ_2_call_for_participation_v0.0.pdf

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
POEBVA	Live scan	Operational	180,000	Index	120,000 mated. 120,000 non-mated	training + testing
VISITIDF	Live scan	Operational	220,000	Index and Thumb	95,000 mated. 120,000 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 BKA	Live scan + Scanned ink	Operational	342,000 images	10 fingers	—	testing
SD 29	Scanned ink	Public	209	10 fingers	1912 mated. 35,791 non-mated	testing
FVC 2000 DB1	Live scan	Public	110	8 fingers	—	compliance testing
FVC 2000 DB3	Live scan	Public	110	8 fingers	—	compliance testing
FVC 2002 DB1	Live scan	Public	110	8 fingers	—	compliance testing

Table 2: NFIQ 2.0 datasets.

4 Quality Features

This section describes NFIQ 2.0 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 2.0 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 2.0 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 2.0 but did not make it to the final selection, are included in the NFIQ 2.0 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.

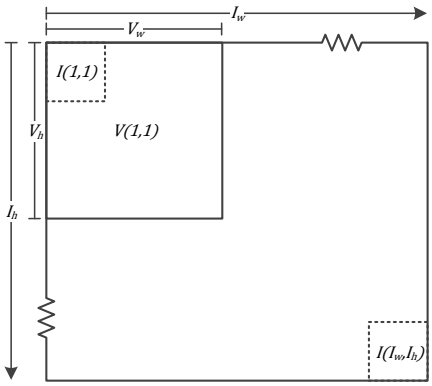


Figure 1: Illustration of block and pixel indexing within an image $\mathbf{I}$ with dimensions $\mathbf{I}_w$, $\mathbf{I}_h$. Shown is the pixel $\mathbf{I}(1,1)$, the block $\mathbf{V}(1,1)$, with dimensions $\mathbf{V}_w$, $\mathbf{V}_h$

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

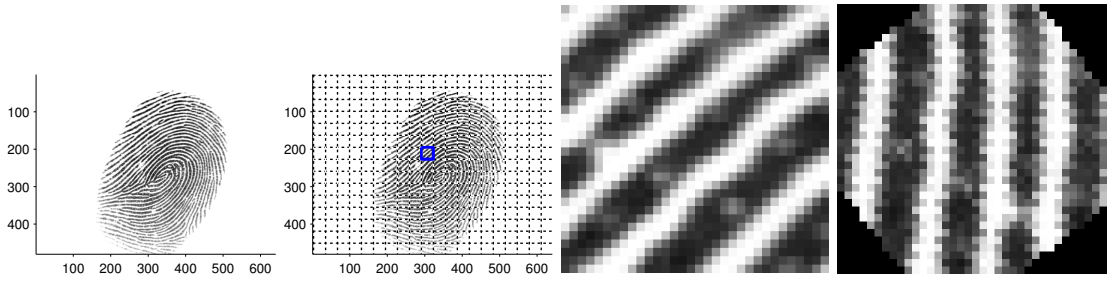


Figure 2: Input image used examples of the processing of quality. Input shown in (a) and division into blocks shown in (b). (c) shows an enlarged view of $\mathbf{V}(10,7)$ and (d) shows $\mathbf{V}(10,7)$ rotated according to ridge orientation its dominant ridge orientation as determined using eq. (10).

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).

$$\mathbf{f_x} = \frac{\mathbf{I}(x+1, y) - \mathbf{I}(x-1, y)}{2} \tag{1}$$

$$\mathbf{f_y} = \frac{\mathbf{I}(x, y+1) - \mathbf{I}(x, y-1)}{2}. \tag{2}$$

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

$$a = \overline{\mathbf{f_x}^2} \tag{3}$$

$$b = \overline{\mathbf{f_y}^2} \tag{4}$$

$$c = \overline{\mathbf{f_x} \cdot \mathbf{f_y}} \tag{5}$$

$$\mathbf{C} = \begin{bmatrix} a & c \\ c & b \end{bmatrix} \tag{6}$$

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

$$\theta_{sin} = \frac{c}{d} \tag{8}$$

$$\theta_{cos} = \frac{a-b}{d} \tag{9}$$

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

Prior to computing features, fingerprint images are cropped to remove white pixels on the margins according to algorithm 1. 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 larger than $T_\mu = 250$ 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 [8] (see algorithm 2).

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 $\widehat{\mathbf{I}}$

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

Algorithm 2: Foreground segmentation based on local standard deviation.

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

Normalize $\mathbf{I}$ to zero mean, unit standard deviation to produce $\widehat{\mathbf{I}}$
for *each block* $\mathbf{V}$ *in* $\widehat{\mathbf{I}}$ **do**
  Compute the standard deviation of $\mathbf{V}$ as $\sigma_{\mathbf{V}}$
  Mark the corresponding block in $\mathbf{I}_{\text{mask}}$ as foreground if $\sigma_{\mathbf{V}} > T$
end

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 the ridge-valley structure [9, 5].

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

Algorithm 3: fda algorithm

Input: Fingerprint image $\mathbf{I}$
Output: fda quality score Q_{FDA}

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

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

$$\mathbf{T} = \frac{1}{\mathbf{V}_h} \sum_{k=1}^{\mathbf{V}_h} \mathbf{V}(x, k), \tag{11}$$

where $\mathbf{V}_h$ is the height of $\mathbf{V}$.

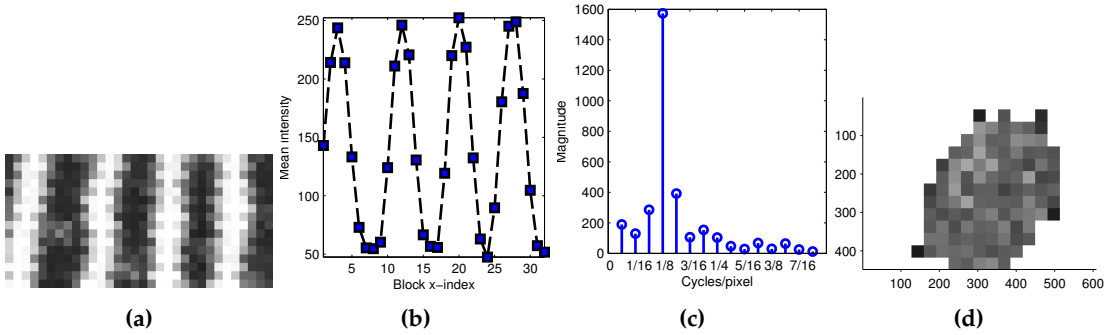


Figure 3: Processing steps of fda quality algorithm. (a) central area of input block; (b) ridge-valley profile; (c) dft of ridge-valley profile; (d) map of $\mathbf{Q}_{\text{FDA}}^{local}$.

Computing the local Frequency Domain Analysis quality score The local fda quality score, $\mathbf{Q}_{\text{FDA}}^{local}$ is computed as:

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

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 $\mathbf{Q}_{\text{FDA}}^{local}$ is set to 1 when maximum frequency amplitude bin is $F_{max} = \mathbf{A}_1$ or $F_{max} = \mathbf{A}_{|\mathbf{A}|}$ 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, 5] 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 quality score Q_{LCS}
for *each block* $\mathbf{V}$ *in* $\mathbf{I}$ **do**
  rotate $\mathbf{V}$ such that dominant ridge flow is perpendicular to x-axis
  crop $\mathbf{V}$ such that no invalid regions are included
  with $\mathbf{V}$ obtain the ridge-valley signature $\mathbf{T}$ (eq. (11))
  Determine DT using linear regression on $\mathbf{T}$
  For each element $\mathbf{T}(x)$ set threshold $\mathbf{P}(x)$ of x being ridge or valley based on DT
  Predict columns in $\mathbf{V}$ as ridge (1) or valley (0) with $\mathbf{S}(x) = \begin{cases} 1, \textit{if} \quad \mathbf{T}(x) < \mathbf{P}(x) \\ 0, \textit{otherwise} \end{cases}$
  Determine ridge-valley transition vector $\mathbf{C}$ from $\mathbf{S}$
  Compute the vector $\mathbf{W}$ containing ridge and valley widths from $\mathbf{C}$
  Determine normalized ridge width and valley width $\bar{W}_r$ and $\bar{W}_v$ (eqs. (15) and (16))
  Determine proportion of misclassified pixel β and α in the ridge and valley regions (eqs. (13) and (14))
  Compute the local quality score $\mathbf{Q}_{\text{LCS}}^{local}$ (eq. (19))
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)).

$$\alpha = \frac{v_B}{v_T} \tag{13}$$

$$\beta = \frac{r_B}{r_T} \tag{14}$$

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

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

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

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

$$\hat{W}_v = \frac{W_v}{20} \tag{17}$$

$$\hat{W}_r = \frac{W_r}{20} \tag{18}$$

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

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

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

4.4 Orientation Certainty Level

ocl [10, 5] 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 [5] in that we subtract the ratio between the eigen values from 1 such that $\mathbf{Q}_{\text{OCL}}^{local}(i,j) = 0$ reflects the lowest local orientation certainty and $\mathbf{Q}_{\text{OCL}}^{local}(i,j) = 1$ the highest local orientation certainty.

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

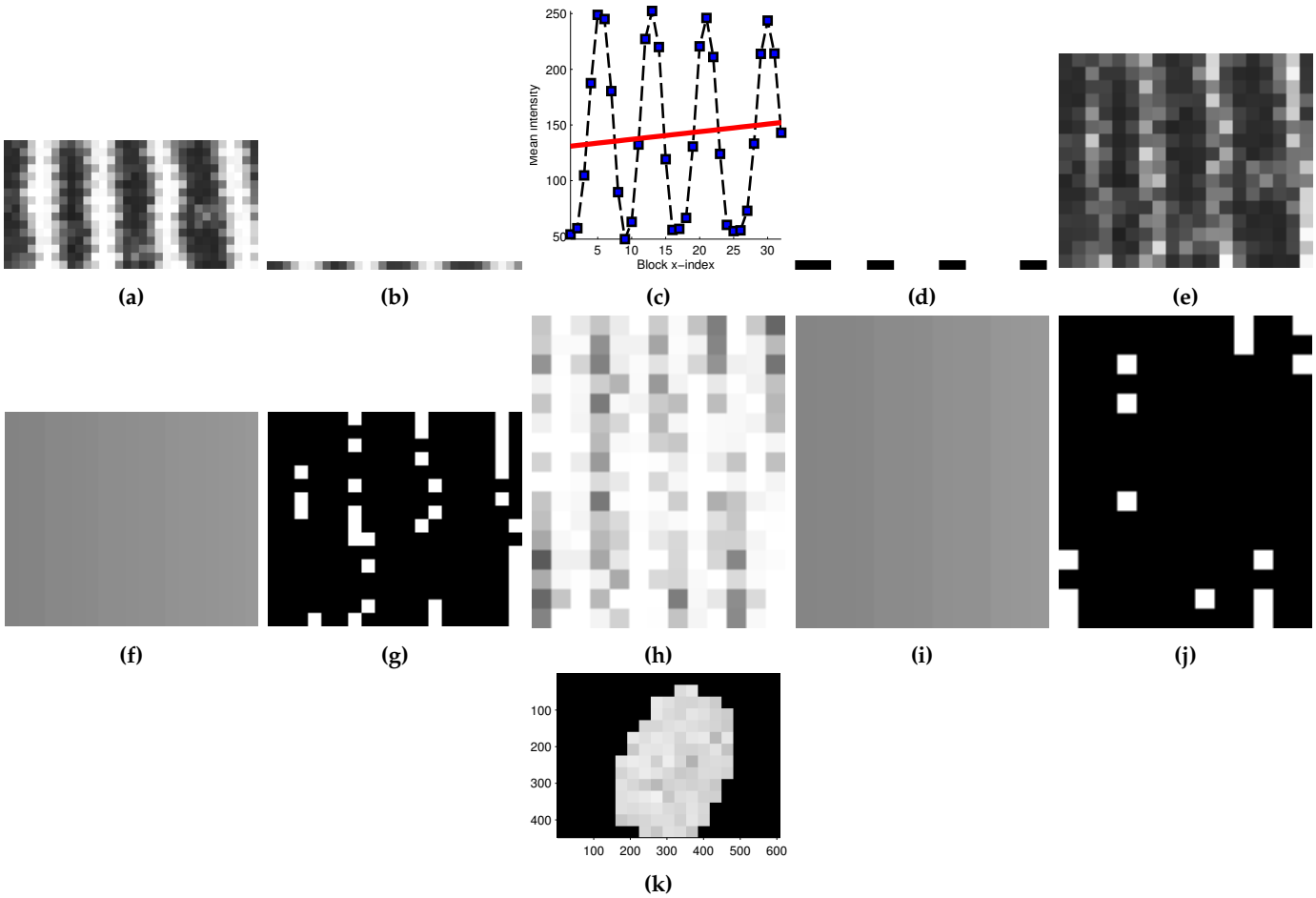


Figure 4: Processing steps of Local Clarity Score algorithm. (a) crop of current block; (b) average profile of block; (c) average block profile with linear regression line; (d) binarisation mask with ridge and valley regions based on regression line; (e) pixels determined to be ridge based on mask; (f) 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 $\mathbf{I}$
Output: ocl quality score Q_{OCL}
for *each block* $\mathbf{V}$ *in* $\mathbf{I}$ **do**
  Compute the intensity gradient with centered differences method
  Compute the covariance matrix $\mathbf{C}$
  Compute the eigenvalues of $\mathbf{C}$ to obtain $\mathbf{Q}_{\text{OCL}}^{local}$ (eqs. (20) and (22))
end

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

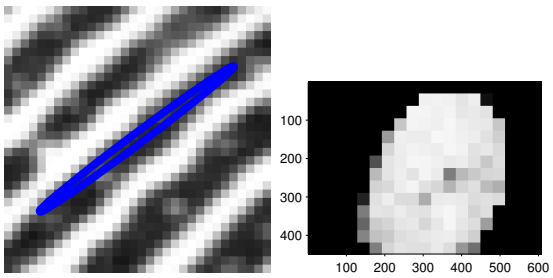
$$\lambda_{min} = \frac{a + b - \sqrt{(a-b)^2 + 4c^2}}{2} \tag{20}$$

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

this yields a local orientation certainty level $\mathbf{Q}_{\text{OCL}}^{local}$:

$$\mathbf{Q}_{\text{OCL}}^{local} = \begin{cases} 1 - \dfrac{\lambda_{min}}{\lambda_{max}}, & if \quad \lambda_{max} > 0 \\ 0, & otherwise \end{cases} \tag{22}$$

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



(a) **(b)**

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, 5] 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 [5] 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 $\mathbf{V}(i, j)$ is:

Algorithm 6: ofl algorithm

Input: Fingerprint image $\mathbf{I}$
Output: ofl quality score Q_{OFL}
Determine the dominant ridgeflow orientation $\mathbf{O}$ of blocks $\mathbf{V}$ in $\mathbf{I}$
for *each block* $\mathbf{V}$ *in* $\mathbf{I}$ **do**
  Compute the absolute orientation difference $\mathbf{D}(i,j)$ using orientations $\mathbf{O}$ (eq. (23))
  Compute the local orientation quality score $\mathbf{Q}_{\text{OFL}}^{local}(i,j)$ (eq. (24))
end

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

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

$$\mathbf{Q}_{\text{OFL}}^{local}(i,j) = \begin{cases} \dfrac{\mathbf{D}(i,j) - \theta_{min}}{90\,\text{deg} - \theta_{min}} & ,\mathbf{D}(i,j) > \theta_{min} \\ 0 & ,otherwise \end{cases} \tag{24}$$

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

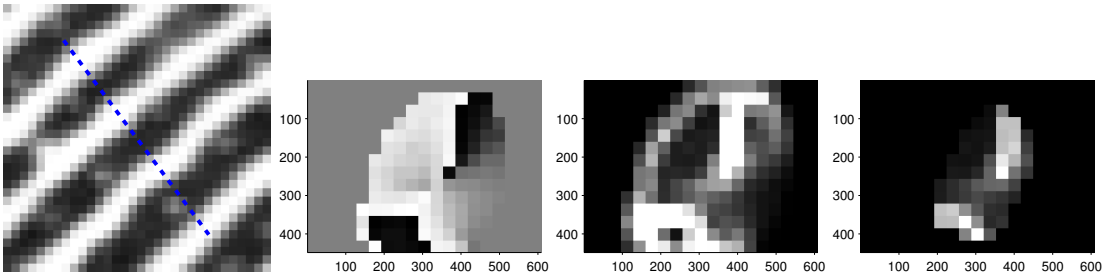


Figure 6: Processing steps of Orientation Flow quality algorithm. (a) line marking the normal to the ridgeline orientation; (b) local orientations; (c) orientation differences; (d) local quality scores.

4.6 Ridge Valley Uniformity

Ridge Valley Uniformity is a measure of the consistency of the ridge and valley widths [10, 5]. 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 $\mathbf{I}$
Output: rvu quality score Q_{RVU}

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

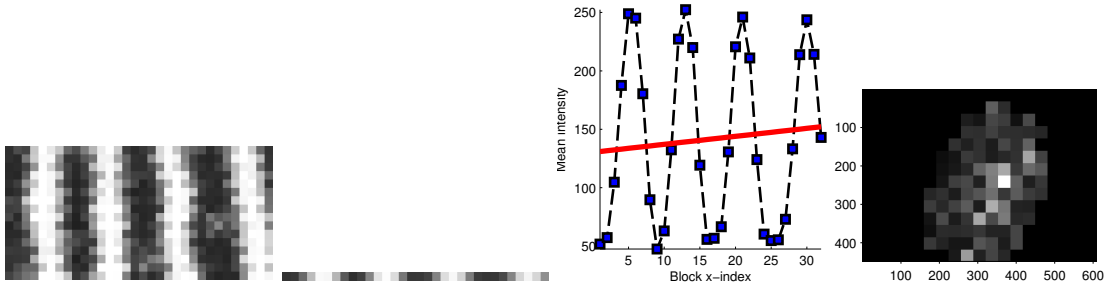


Figure 7: Processing steps of Ridge Valley Uniformity quality algorithm. (a) crop of current block; (b) average profile of block; (c) average profile with regression line; (d) local quality score as the standard deviation of local ridge to valley ratios.

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 $\mathbf{I}$
Output: MU quality score Q_{MU}
Compute Q_{MU} as the arithmetic mean of pixel values in $\mathbf{I}$;

Algorithm 9: MMB algorithm

Input: Fingerprint image $\mathbf{I}$
Output: MMB quality score Q_{MMB}
for *each block* $\mathbf{V}$ *in* $\mathbf{I}$ **do**
  compute the arithmetic mean of pixels in $\mathbf{V}$ as $\mathbf{Q}_{\text{MMB}}^{local}$;
end
Compute Q_{MMB} as the arithmetic mean of set of $\mathbf{Q}_{\text{MMB}}^{local}$;

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

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

  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 $\mathbf{I}$
Output: Binarized image R specifying the ROI
Erode the fingerprint image $\mathbf{I}$ using a square of size 5×5 as structuring element
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
Binarize the image resulting from the previous step using the threshold method of Otsu [15] (see description below)
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
One again, binarize the image resulting from the previous step using the threshold method of Otsu [15]
Find contours in the image, e.g., using the algorithm of Suzuki [16]
Fill all white holes, i.e., regions completely surrounded by black pixels, by setting the pixels in that area to black
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
The resulting binarized image R contains the ROI specified by all black pixels.

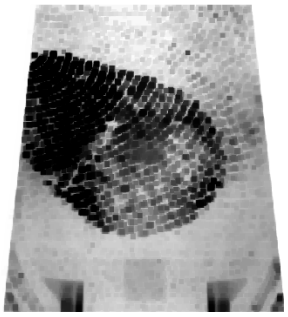
The processing of the ROI determination algorithm is illustrated with two example input images in fig. 8.

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

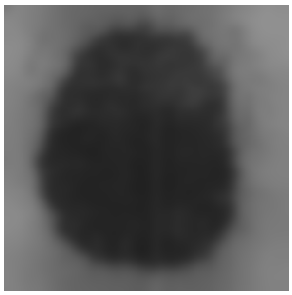
(a) Input image A

(b) Input image B

(c) Result of step 1 for input image A



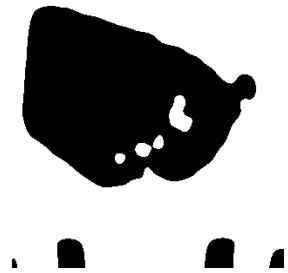
(d) Result of step 1 for input image B



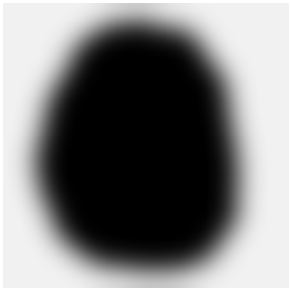
(e) Result of step 2 for input image A

(f) Result of step 2 for input image B

(g) Result of step 3 for input image A

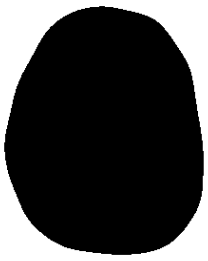


(h) Result of step 3 for input image B

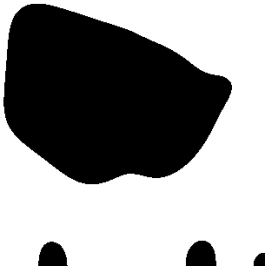


(i) Result of step 4 for input image A

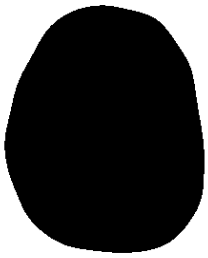
(j) Result of step 4 for input image B



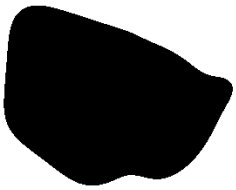
(k) Result of step 5 for input image A



(l) Result of step 5 for input image B



(m) Result of step 8 for input image A



(n) Result of step 8 for input image B

Figure 8: Processing steps of the ROI determination algorithm.

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

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

4.11.1 ROI Area Mean

The feature Q^{μ}_{AREA} 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^{sum}_{COH} computes the coherence map of the orientation field estimation as specified in [7], 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

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

and

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

4.11.3 ROI Relative Orientation Map Coherence Sum

The feature Q^{rel}_{COH} computes the coherence map of the orientation field estimation as specified in [7], and returns the average coherence values over all image blocks of size 16×16 in the ROI . The returned quality value Q^{rel}_{COH} is computed as $Q^{rel}_{\text{COH}} = Q^{sum}_{\text{COH}}/n$, 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_{COH}^{sum}
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)
Compute the (so-called) *squared gradient* field $\mathbf{g}_s = (g_x^2 - g_y^2, 2g_x g_y)^T$
Determine the ROI of $\mathbf{I}$ using algorithm 10
for *each* 16×16 *block* $\mathbf{V}$ *in* $\mathbf{I}$ **do**
    if *Block* $\mathbf{V}$ *contains at least one pixel of the ROI* **then**
        Compute the coherence of the orientation map of block $\mathbf{V}$ as $Coh(\mathbf{V}) = |\sum g_s(i,j)| \,/\, \sum |g_s(i,j)|$, where the sums are taken over all pixels (i,j) in $\mathbf{V}$ and $|\cdot|$ denotes the Euclidean norm
    else
        Set $Coh(\mathbf{V}) = 0$
    end
end
Compute Q_{COH}^{sum} as the sum of the coherence $Coh(\mathbf{V})$ 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 $\mathbf{Q}_{qname}^{local}$ is computed with eq. (28).

$$Q_{qname}^{\mu} = \frac{1}{N * M} \sum_{i=1}^{N} \sum_{j=1}^{M} \mathbf{Q}_{qname}^{local}, \tag{28}$$

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

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

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

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).

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

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

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

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

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

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

$$\left[B_{\text{Q}}^{i}, B_{\text{Q}}^{i+1}\right), \qquad 1 \leq i \leq |B_{\text{Q}}|\,. \tag{35}$$

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

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

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

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

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

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

4.13 The NFIQ 2.0 feature vector

The NFIQ 2.0 feature vector is produced as a concatenation of individual quality features:

$$\begin{aligned}\mathbf{Q}_{\text{NFIQ 2.0}} = \Big(&Q_{\text{FDA}}^{\mu}, Q_{\text{LCS}}^{\mu}, Q_{\text{OCL}}^{\mu}, Q_{\text{OFL}}^{\mu}, Q_{\text{RVU}}^{\mu},\\ &Q_{\text{FDA}}^{\sigma}, Q_{\text{LCS}}^{\sigma}, Q_{\text{OCL}}^{\sigma}, Q_{\text{OFL}}^{\sigma}, Q_{\text{RVU}}^{\sigma},\\ &\mathbf{Q}_{\text{FDA}}, \mathbf{Q}_{\text{LCS}}, \mathbf{Q}_{\text{OCL}}, \mathbf{Q}_{\text{OFL}}, \mathbf{Q}_{\text{RVU}},\\ &Q_{\text{MU}}, Q_{\text{MMB}}, Q_{\text{COH}}^{rel}, Q_{\text{COH}}^{sum}, Q_{\text{AREA}}^{\mu},\\ &Q_{\text{MIN}}^{cnt}, Q_{\text{MIN}}^{com}, Q_{\text{MIN}}^{mu}, Q_{\text{MIN}}^{ocl}\Big)\,.\end{aligned} \tag{41}$$

4.14 Predictive power of NFIQ 2.0 features

A useful feature will give statistically different values for images with low or high utility. Figures 9 shows NFIQ 2.0 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).

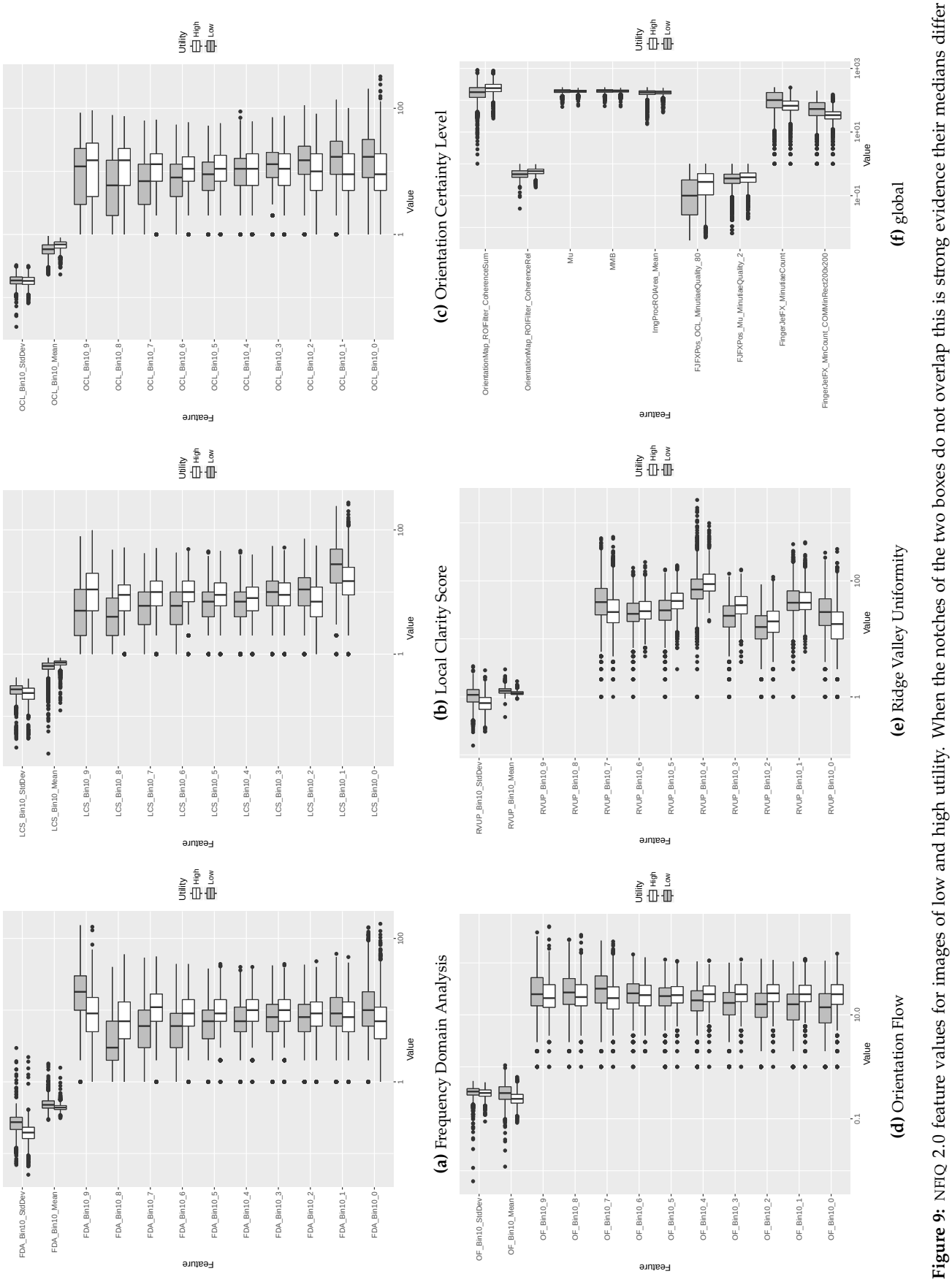


Figure 9: NFIQ 2.0 feature values for images of low and high utility. When the notches of the two boxes do not overlap this is strong evidence their medians differ (Chambers et al., 1983, p. 62).

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 2.0 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 6629 images (3295 in Class 0 and 3334 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 2.0 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	3295	0	0.000
1	0	3334	0.000

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

	Name	MeanDreaseGini
Q^{σ}_{FDA}	Frequency Domain Analysis_Standard Deviation	140.760
Q^{com}_{MIN}	FingerJet FX OSE COM Minutiae Count	92.089
Q^{ocl}_{MIN}	FingerJet FX OSE OCL MinutiaeQuality	83.027
Q^{μ}_{RVU}	Ridge Valley Uniformity_Mean	69.517
Q^{μ}_{FDA}	Frequency Domain Analysis_Mean	62.229
Q^{cnt}_{MIN}	FingerJet FX OSE Total Minutiae Count	57.565
Q^{σ}_{RVU}	Ridge Valley Uniformity_Standard Deviation	50.946
Q^{7}_{LCS}	Local Clarity Score_Bin_7	50.688
Q^{8}_{LCS}	Local Clarity Score_Bin_8	50.100
Q^{9}_{FDA}	Frequency Domain Analysis_Bin_9	47.844
Q^{sum}_{COH}	ROI Orientation Map Coherence Sum	38.104
Q^{2}_{OFL}	Orientation Flow_Bin_2	37.172
Q^{μ}_{LCS}	Local Clarity Score_Mean	36.483
Q^{5}_{RVU}	Ridge Valley Uniformity_Bin_5	35.617
Q^{3}_{RVU}	Ridge Valley Uniformity_Bin_3	35.139
Q^{μ}_{AREA}	ROI Area Mean	34.932
Q^{1}_{OFL}	Orientation Flow_Bin_1	33.751
Q^{0}_{OFL}	Orientation Flow_Bin_0	33.513
Q_{MU}	MU	32.914
Q_{MMB}	MMB	32.625
Q^{mu}_{MIN}	FingerJet FX OSE Mu MinutiaeQuality	32.316
Q^{8}_{FDA}	Frequency Domain Analysis_Bin_8	31.428
Q^{7}_{FDA}	Frequency Domain Analysis_Bin_7	31.236
Q^{μ}_{OFL}	Orientation Flow_Mean	31.172
Q^{4}_{RVU}	Ridge Valley Uniformity_Bin_4	30.801
Q^{μ}_{OCL}	Orientation Certainty Level_Mean	30.035
Q^{3}_{OFL}	Orientation Flow_Bin_3	29.721
Q^{σ}_{LCS}	Local Clarity Score_Standard Deviation	28.777
Q^{rel}_{COH}	ROI Relative Orientation Map Coherence Sum	28.700
Q^{σ}_{OCL}	Orientation Certainty Level_Standard Deviation	28.429
Q^{σ}_{OFL}	Orientation Flow_Standard Deviation	27.556
Q^{8}_{OCL}	Orientation Certainty Level_Bin_8	26.425
Q^{6}_{FDA}	Frequency Domain Analysis_Bin_6	25.161
Q^{6}_{LCS}	Local Clarity Score_Bin_6	23.837
Q^{5}_{OFL}	Orientation Flow_Bin_5	23.431
Q^{9}_{LCS}	Local Clarity Score_Bin_9	23.283
Q^{4}_{OFL}	Orientation Flow_Bin_4	22.883
Q^{2}_{RVU}	Ridge Valley Uniformity_Bin_2	22.843
Q^{1}_{RVU}	Ridge Valley Uniformity_Bin_1	22.380
Q^{9}_{OFL}	Orientation Flow_Bin_9	21.413
Q^{6}_{RVU}	Ridge Valley Uniformity_Bin_6	21.306
Q^{7}_{OFL}	Orientation Flow_Bin_7	21.296
Q^{6}_{OFL}	Orientation Flow_Bin_6	20.939
Q^{1}_{LCS}	Local Clarity Score_Bin_1	20.910
Q^{8}_{OFL}	Orientation Flow_Bin_8	20.867
Q^{7}_{RVU}	Ridge Valley Uniformity_Bin_7	20.798
Q^{0}_{RVU}	Ridge Valley Uniformity_Bin_0	20.470
Q^{0}_{OCL}	Orientation Certainty Level_Bin_0	20.234
Q^{7}_{OCL}	Orientation Certainty Level_Bin_7	19.887
Q^{1}_{OCL}	Orientation Certainty Level_Bin_1	19.434
Q^{6}_{OCL}	Orientation Certainty Level_Bin_6	19.188
Q^{5}_{LCS}	Local Clarity Score_Bin_5	19.073
Q^{4}_{OCL}	Orientation Certainty Level_Bin_4	18.865
Q^{3}_{LCS}	Local Clarity Score_Bin_3	18.861
Q^{2}_{OCL}	Orientation Certainty Level_Bin_2	18.663
Q^{0}_{FDA}	Frequency Domain Analysis_Bin_0	18.657
Q^{5}_{FDA}	Frequency Domain Analysis_Bin_5	18.438
Q^{5}_{OCL}	Orientation Certainty Level_Bin_5	18.262
Q^{9}_{OCL}	Orientation Certainty Level_Bin_9	18.164
Q^{2}_{LCS}	Local Clarity Score_Bin_2	18.136
Q^{3}_{FDA}	Frequency Domain Analysis_Bin_3	17.847
Q^{3}_{OCL}	Orientation Certainty Level_Bin_3	17.834
Q^{1}_{FDA}	Frequency Domain Analysis_Bin_1	17.710
Q^{4}_{LCS}	Local Clarity Score_Bin_4	17.598
Q^{4}_{FDA}	Frequency Domain Analysis_Bin_4	17.283
Q^{2}_{FDA}	Frequency Domain Analysis_Bin_2	17.145
Q^{8}_{RVU}	Ridge Valley Uniformity_Bin_8	0.000
Q^{9}_{RVU}	Ridge Valley Uniformity_Bin_9	0.000
Q^{0}_{LCS}	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.

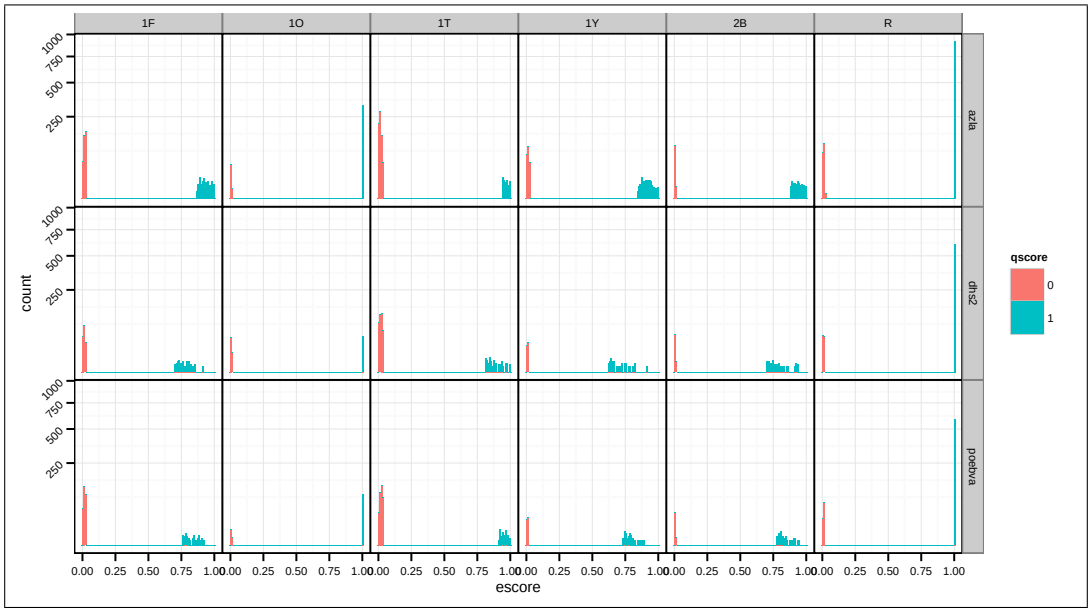


Figure 10: Training set composition: Histograms of sample counts according to provider and dataset, colored by ground truth class

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 2.0 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 2.0 scores).

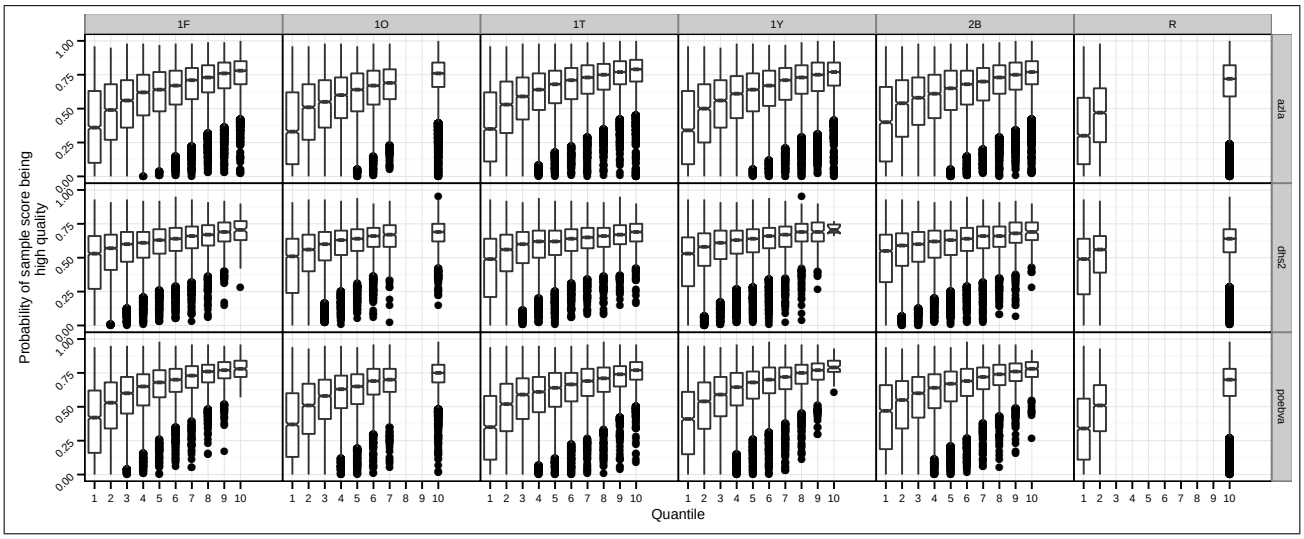


Figure 11: Boxplot of the output of random forest per quantized comparison score bin for the validation set.

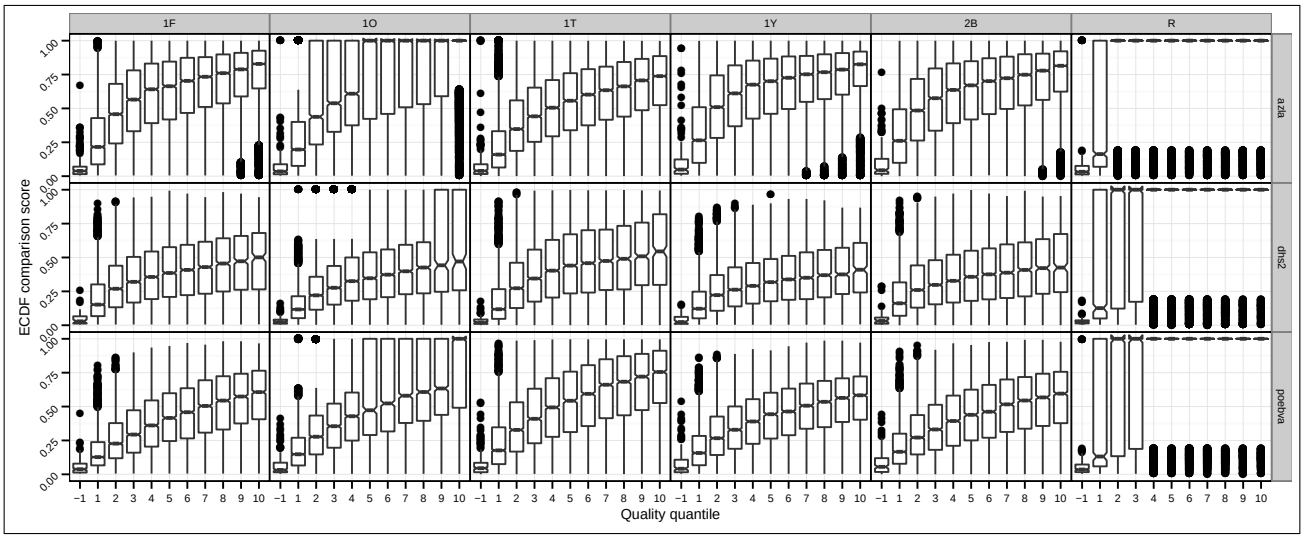


Figure 12: Boxplot of the CDF comparison score per quantile quality score bin for the validation set.

6 Predictive power of NFIQ 2.0

6.1 Rejection curves

One metric for comparative evaluation of quality assessment algorithms is the error versus reject curves [3]. 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

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

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

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

where

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

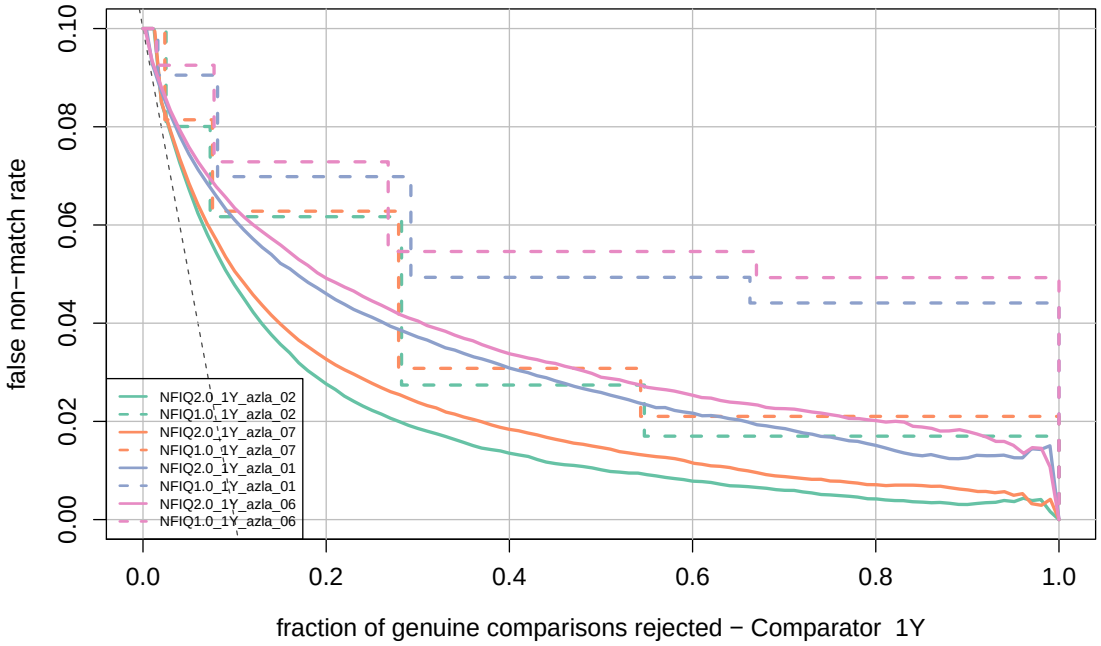
where R^C is the complement of R.

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.

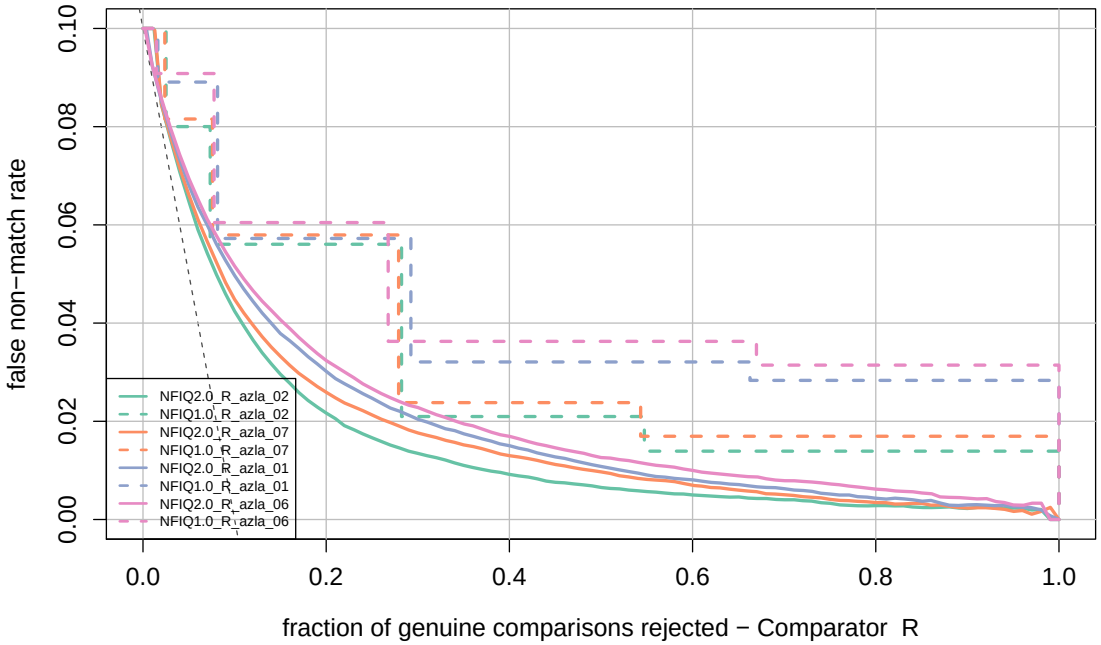
Figure 13 shows the error vs. reject curves for NFIQ 2.0 (solid lines) and NFIQ 1.0 (dotted lines) for images from AZLA dataset and comparison algorithms R and 1Y. Results for other datasets and comparison algorithms are in provider specific appendices.

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 algorith is displayed by the gray dotted line where the rejection of the lowest 10% quality would result in an false non-match ratefalse non-match rate of zero. Both NFIQ 1.0 and NFIQ 2.0 trend in the correct direction (false non-match rate improves as more low-quality samples are rejected). NFIQ 2.0 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 **??** shows the error vs. reject curves for rejection rate of up to 25%.

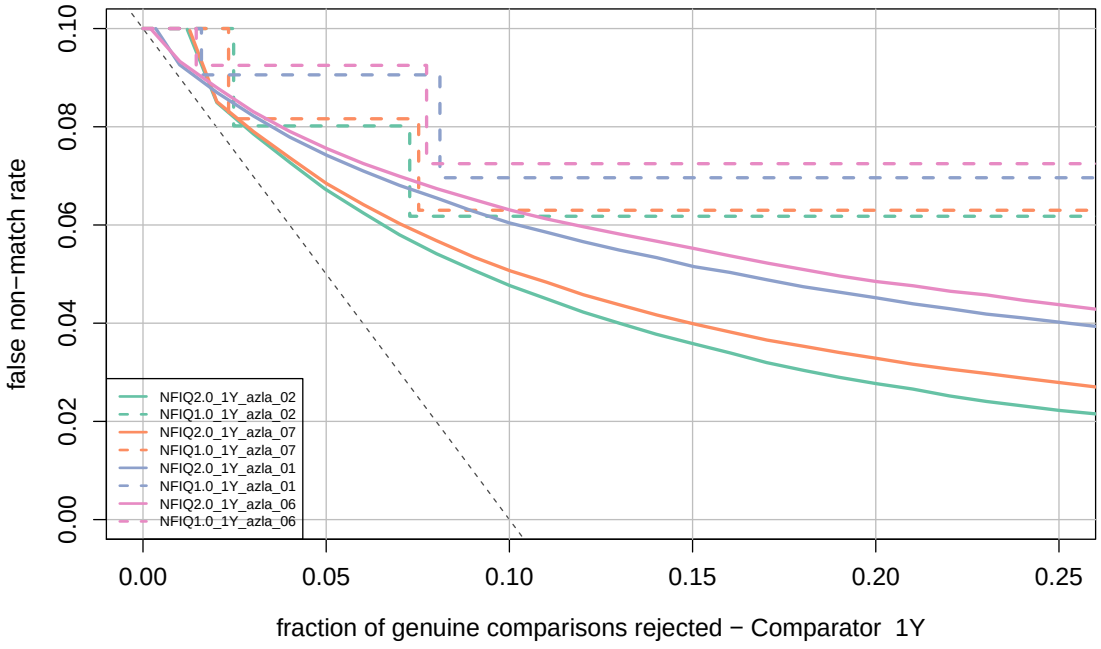


(a)

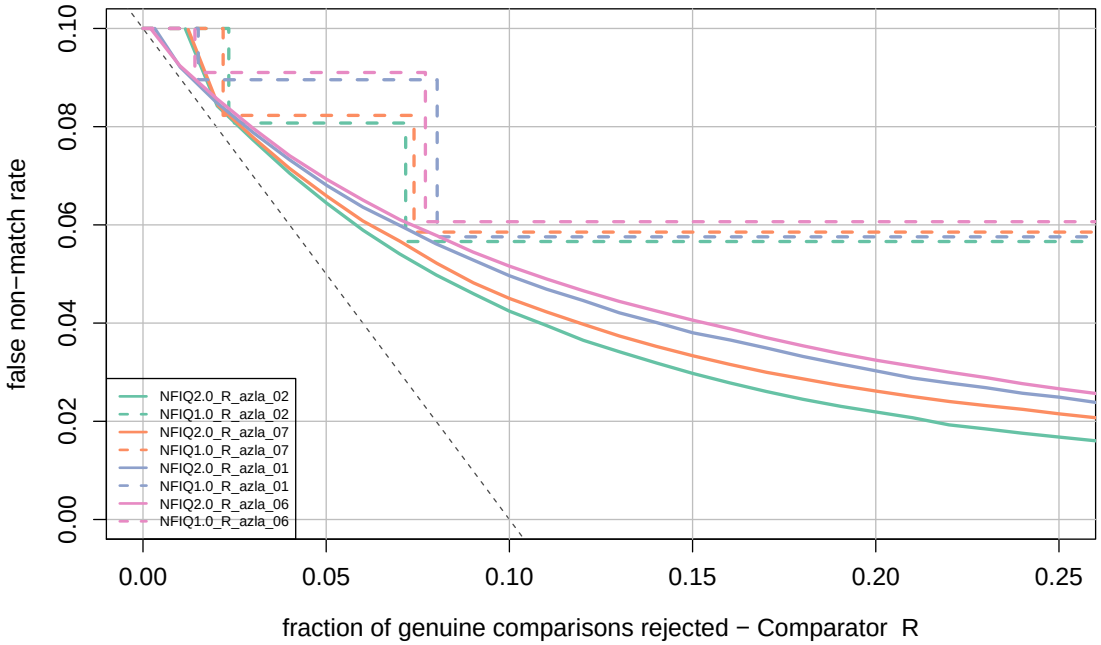


(b)

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 2.0 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 2.0 is a better predictive of performance than NFIQ 1.0 because its error vs. reject curves are closer to the gray dotted line.



(a)



(b)

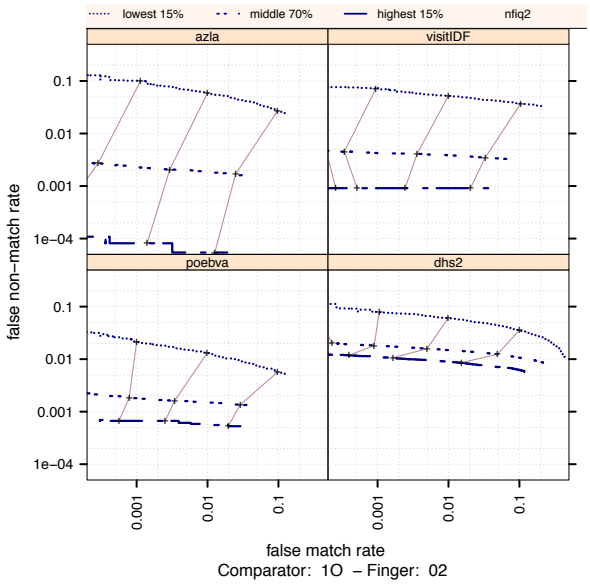
Figure 14:
Error vs. reject curve. Dataset AZLA, for rejection rate of smaller than 25%. Finger positions right index (02), left index (07), right thumb (01) and left thumb (06). The threshold is set to give an initial false non-match rate of 0.1. NFIQ 2.0 results are shown in solid lines and NFIQ 1.0 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 2.0 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

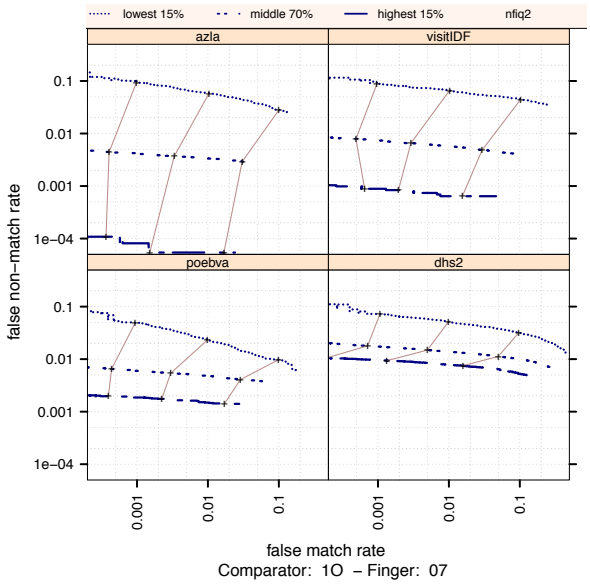
Detection error trade-off (DET) characteristic curves are the primary performance metric for offline testing of biometric recognition algorithms [14], [1]. Each point on a DET curve exhibits the false match and false non-match rates associated with a certain threshold value. The DET curve spans the whole 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.

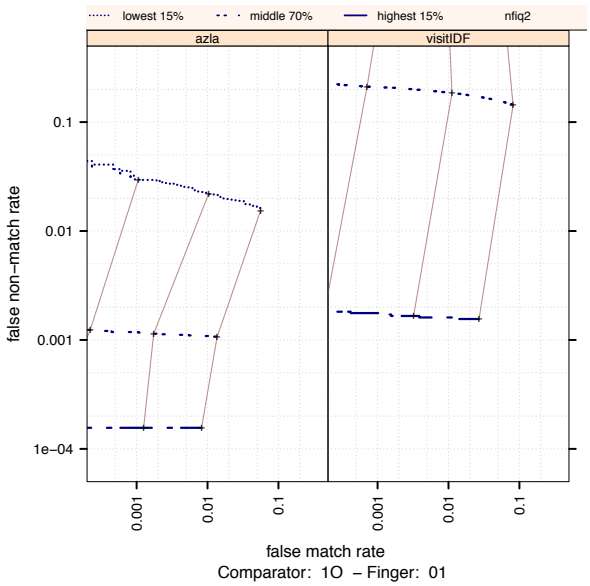
Figure 25 shows ranked DET results for provider 1O. Ranked DET for other providers are in provider specific appendices. The ranking and the separation of the DET curves in figure 25, as explained above, show that NFIQ 2.0 is predictive of the false non-match rates. Effect of quality on false match rate 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.



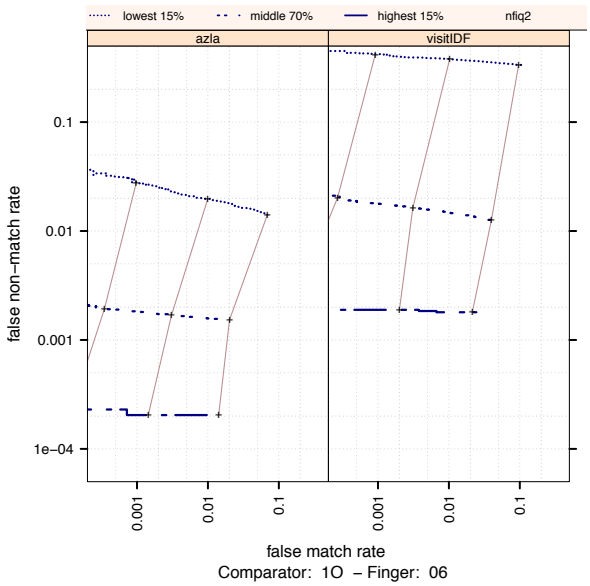
(a) provvider 1O. Right Index



(b) Provider 1O. Left Index



(c) Provider 1O. Right Thumb



(d) Provider 1O. Left Thumb

Figure 15:
Ranked DET. Provider 1O.The set of all comparisons are partitioned into three groups based on the pair-wise NFIQ2.0 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 2.0 values appearing above, and the DET curve of highest NFIQ 2.0 values below all the other curves.

6.3 Computation efficiency

NFIQ 2.0 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 x 300 pixels. NFIQ 2.0 computation time for FVC2000/DB3 is around 150 milliseconds, and their image sizes are 448 x 478 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 NFIQ 2.0 compliance test.

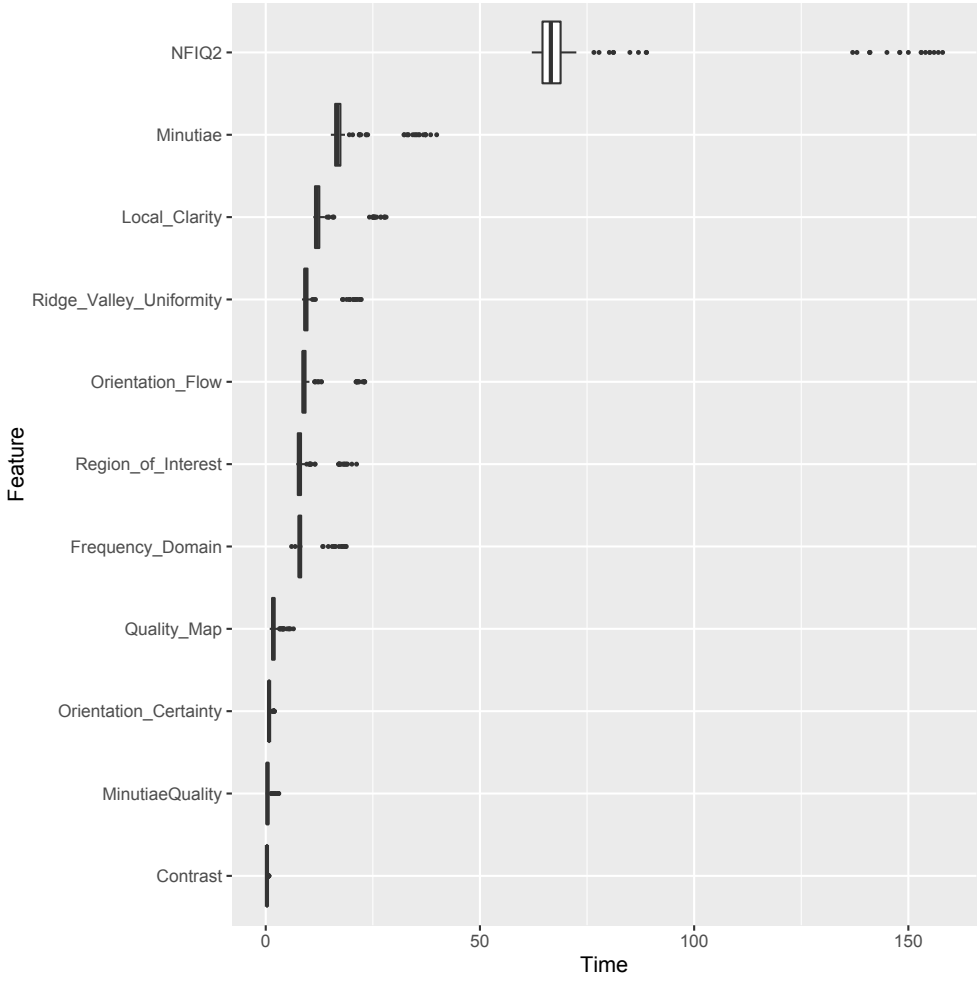


Figure 16: Computation time for NFIQ 2.0 and its features for FVC2000/DB1 and FVC2000/DB3 on Windows 7 Enterprise, 64 bit OS platform.

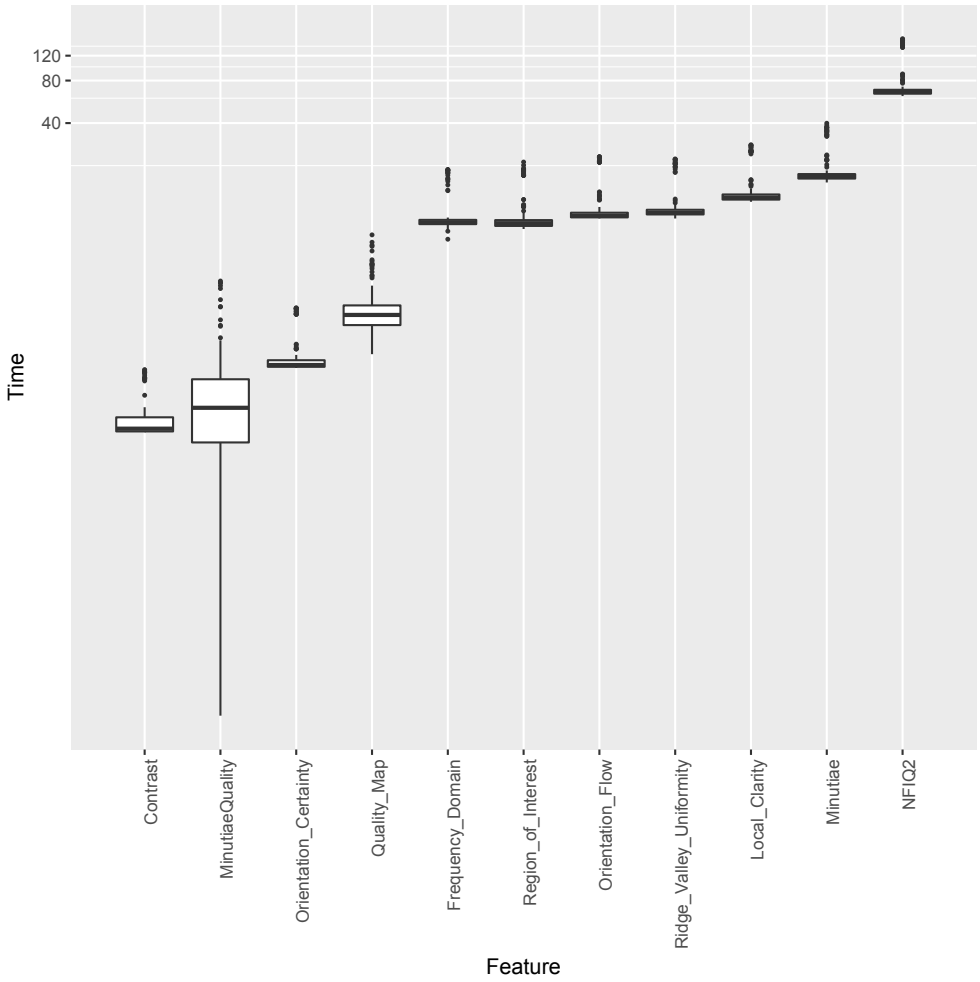


Figure 17: Computation time for NFIQ 2.0 and its features for FVC2000/DB1 and FVC2000/DB3 on Windows 7 Enterprise, 64 bit OS platform.

References

[1] Working Group 5. *ISO/IEC 19795-1 Biometric Performance Testing and Reporting: Principles and Framework*. JTC1 :: SC37, international standard edition, August 2005. http://isotc.iso.org/isotcportal.

[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 1253–1256Vol.2, 2004.

[3] Patrick Grother and Elham Tabassi. Performance of Biometric Quality Measures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29:531–543, 2007.

[4] ISO/IEC. IS 29794-1:2009. information technology âĂŤ biometric sample quality âĂŤ part 1: Framework. ISO Standard, January 2009.

[5] ISO/IEC. TR 29794-4:2010. information technology âĂŤ biometric sample quality âĂŤ part 4: Finger image data. Technical report, JTC 1/SC 37, 2010.

[6] ISO/IEC. IS 29794-1:2016. information technology âĂŤ biometric sample quality âĂŤ part 1: Framework. ISO Standard, January 2016.

[7] Michael Kass and Andrew Witkin. Analyzing oriented patterns. *Computer vision, graphics, and image processing*, 37(3):362–385, 1987.

[8] P. D. Kovesi. MATLAB and Octave functions for computer vision and image processing, 2005. Available from: <http://www.peterkovesi.com/matlabfns/>.

[9] 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 1241–1244Vol.2, 2004.

[10] Eyung Lim, Xudong Jiang, and Weiyun Yau. Fingerprint quality and validity analysis. In *Proc. Int. Conf. Image Processing 2002*, volume 1, 2002.

[11] D. Maio, D. Maltoni, R. Cappelli, J.L. Wayman, and A.K. Jain. FVC2000: Fingerprint verification competition, 2000.

[12] D. Maio, D. Maltoni, R. Cappelli, J.L. Wayman, and A.K. Jain. FVC2002: Second fingerprint verification competition, 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 2.0 compliance test criteria

A.1 Purpose

The NFIQ 2.0 algorithm for finger image quality assessment is released as open-source software. This appendix specifies compliance test criteria and a compliance test set by which any interested party can determine whether an implementation complies with the NFIQ 2.0 reference implementation.

A.2 NFIQ 2.0 compliance test set

The NFIQ 2.0 compliance test set consists of a set of publicly available finger images where all NFIQ 2.0 scores are sufficiently represented. Most of the possible NFIQ 2.0 scores are assumed by two finger images in the compliance test set. Because the NFIQ 2.0 algorithm has been trained on live-scan finger images captured with optical fingerprint sensors with a resolution of 500 dpi, the compliance test set includes only 500-dpi flat finger images captured with optical sensors. The image format is grayscale BMP. Images are selected from databases DB1[2] and DB3[3] of FVC2000 [11] and database DB1[4] of FVC2002 [12].

Table 5 contains the identifiers of all images of the NFIQ 2.0 compliance test set and their NFIQ 2.0 scores computed with the reference implementation. Figure 18 shows some examples of finger images from the NFIQ 2.0 compliance test set with their NFIQ 2.0 scores.

The NFIQ 2.0 compliance test set together with the expected NFIQ 2.0 scores is provided in the NFIQ 2.0 Perforce depot at //depot/projects/NFIQ2/Main/data. Parties who have obtained the compliance test set are free to use it for verifying the compliance of their implementations with NFIQ 2.0.

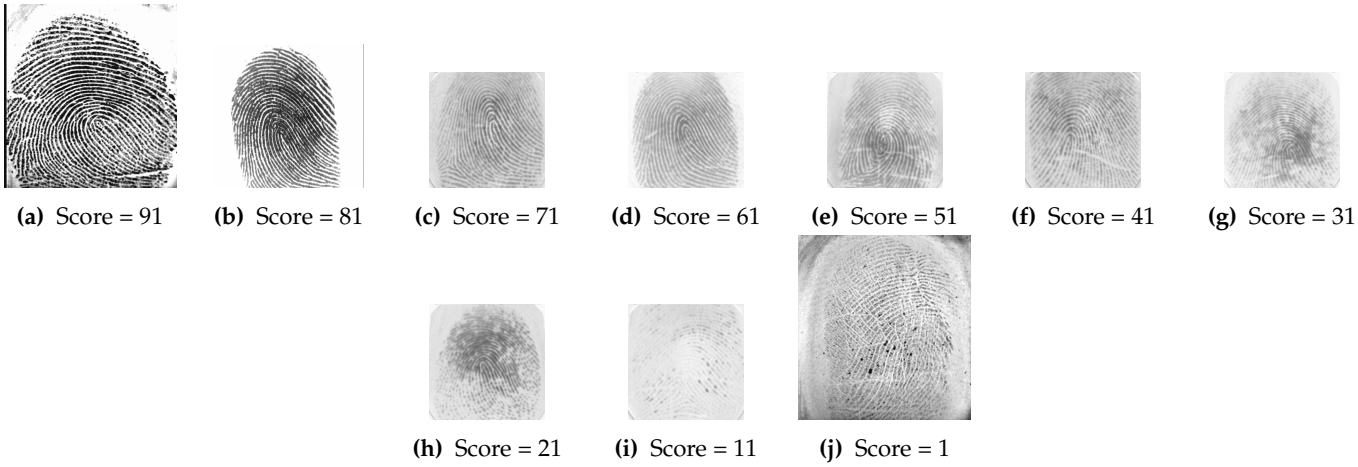
Table 5: NFIQ 2.0 compliance test set

NFIQ 2.0 score	Finger image identifiers	
91	FVC2000/Db3/25_3	
89	FVC2002/Db1/49_6	
88	FVC2002/Db1/51_4,	FVC2002/Db1/89_1
87	FVC2000/Db3/108_3,	FVC2000/Db3/108_4
86	FVC2000/Db1/69_6,	FVC2000/Db3/26_2
85	FVC2000/Db3/24_4,	FVC2002/Db1/29_2
84	FVC2002/Db1/51_5,	FVC2002/Db1/55_6
83	FVC2002/Db1/110_7,	FVC2002/Db1/14_6
82	FVC2000/Db1/69_8,	FVC2000/Db3/108_6
81	FVC2002/Db1/105_2,	FVC2002/Db1/106_5
80	FVC2000/Db1/36_6,	FVC2000/Db1/9_5
79	FVC2000/Db1/104_8,	FVC2000/Db1/69_5
78	FVC2000/Db1/28_5,	FVC2000/Db1/58_6
77	FVC2000/Db1/28_6,	FVC2000/Db1/33_1
76	FVC2000/Db1/11_5,	FVC2000/Db1/19_5
75	FVC2000/Db1/104_6,	FVC2000/Db1/105_3
74	FVC2000/Db1/24_5,	FVC2000/Db1/2_5
73	FVC2000/Db1/25_8,	FVC2000/Db1/28_2

[2] captured with the low-cost optical sensor "Secure Desktop Scanner" of KeyTronic [11]
[3] captured with the optical sensor "DF-90" of Identicator Technology [11]
[4] captured with the optical sensor "TouchView II" of Identix [12]

72	FVC2000/Db1/101_7,	FVC2000/Db1/101_8
71	FVC2000/Db1/20_6,	FVC2000/Db1/23_2
70	FVC2000/Db1/101_1,	FVC2000/Db1/16_5
69	FVC2000/Db1/101_6,	FVC2000/Db1/105_8
68	FVC2000/Db1/102_5,	FVC2000/Db1/104_3
67	FVC2000/Db1/101_5,	FVC2000/Db1/103_4
66	FVC2000/Db1/101_2,	FVC2000/Db1/102_8
65	FVC2000/Db1/102_2,	FVC2000/Db1/104_2
64	FVC2000/Db1/106_1,	FVC2000/Db1/106_8
63	FVC2000/Db1/101_3,	FVC2000/Db1/102_4
62	FVC2000/Db1/102_7,	FVC2000/Db1/104_5
61	FVC2000/Db1/105_6,	FVC2000/Db1/106_7
60	FVC2000/Db1/105_2,	FVC2000/Db1/105_5
59	FVC2000/Db1/102_6,	FVC2000/Db1/106_2
58	FVC2000/Db1/107_3,	FVC2000/Db1/15_1
57	FVC2000/Db1/102_3,	FVC2000/Db1/103_1
56	FVC2000/Db1/103_3,	FVC2000/Db1/10_8
55	FVC2000/Db1/102_1,	FVC2000/Db1/107_7
54	FVC2000/Db1/106_5,	FVC2000/Db1/107_5
53	FVC2000/Db1/109_3,	FVC2000/Db1/109_5
52	FVC2000/Db1/100_7,	FVC2000/Db1/103_6
51	FVC2000/Db1/101_4,	FVC2000/Db1/108_4
50	FVC2000/Db1/105_7,	FVC2000/Db1/107_2
49	FVC2000/Db1/12_2,	FVC2000/Db1/30_6
48	FVC2000/Db1/108_7,	FVC2000/Db1/109_7
47	FVC2000/Db1/108_3,	FVC2000/Db1/14_3
46	FVC2000/Db1/16_7,	FVC2000/Db1/22_1
45	FVC2000/Db1/15_5,	FVC2000/Db1/21_4
44	FVC2000/Db1/108_1,	FVC2000/Db1/11_3
43	FVC2000/Db1/110_5,	FVC2000/Db1/12_4
42	FVC2000/Db1/13_1,	FVC2000/Db1/20_3
41	FVC2000/Db1/61_2,	FVC2000/Db1/67_3
40	FVC2000/Db1/42_6,	FVC2000/Db1/4_1
39	FVC2000/Db1/75_7,	FVC2000/Db1/79_8
38	FVC2000/Db1/108_8,	FVC2000/Db1/110_7
37	FVC2000/Db1/64_7,	FVC2000/Db1/75_4
36	FVC2000/Db1/110_8,	FVC2000/Db1/27_1
35	FVC2000/Db1/100_5,	FVC2000/Db1/76_4
34	FVC2000/Db1/109_2,	FVC2000/Db1/50_8
33	FVC2000/Db1/100_8,	FVC2000/Db1/108_2
32	FVC2000/Db1/78_3,	FVC2000/Db1/91_1
31	FVC2000/Db1/75_2,	FVC2000/Db1/89_3
30	FVC2000/Db1/78_2,	FVC2000/Db1/78_4
29	FVC2000/Db1/48_3,	FVC2000/Db1/70_6
28	FVC2000/Db1/85_7,	FVC2000/Db3/100_2
27	FVC2000/Db1/84_4,	FVC2000/Db3/109_1
26	FVC2000/Db1/86_8,	FVC2000/Db1/94_7
25	FVC2000/Db1/76_2,	FVC2000/Db1/87_7
24	FVC2000/Db1/93_5,	FVC2000/Db1/94_1
23	FVC2000/Db1/87_5,	FVC2000/Db1/94_6
22	FVC2000/Db1/92_6,	FVC2000/Db1/93_3
21	FVC2000/Db1/100_6,	FVC2000/Db1/87_6
20	FVC2000/Db1/110_6,	FVC2000/Db1/92_8
19	FVC2000/Db3/104_1,	FVC2000/Db3/105_2
18	FVC2000/Db1/110_4,	FVC2000/Db1/91_5

17	FVC2000/Db1/76_7,	FVC2000/Db3/15_4
16	FVC2000/Db1/97_5,	FVC2000/Db1/99_5
15	FVC2000/Db1/94_5,	FVC2000/Db1/95_8
14	FVC2000/Db1/84_3,	FVC2000/Db1/89_2
13	FVC2000/Db1/93_6,	FVC2000/Db1/95_1
12	FVC2000/Db1/100_4,	FVC2000/Db1/91_2
11	FVC2000/Db1/92_7,	FVC2000/Db1/98_2
10	FVC2000/Db1/97_4,	FVC2000/Db1/97_8
9	FVC2000/Db1/110_2,	FVC2000/Db1/90_8
8	FVC2000/Db1/90_6,	FVC2000/Db1/91_7
7	FVC2000/Db1/89_4,	FVC2000/Db1/93_2
6	FVC2000/Db1/96_8,	FVC2000/Db3/102_4
5	FVC2000/Db1/90_5,	FVC2000/Db1/90_7
4	FVC2000/Db1/100_2,	FVC2000/Db1/100_3
3	FVC2000/Db1/100_1,	FVC2000/Db1/97_7
2	FVC2000/Db1/110_1,	FVC2000/Db3/18_3
1	FVC2000/Db3/102_3,	FVC2000/Db3/19_4
0	FVC2000/Db3/29_8,	FVC2000/Db3/30_5



(a) Score = 91 **(b)** Score = 81 **(c)** Score = 71 **(d)** Score = 61 **(e)** Score = 51 **(f)** Score = 41 **(g)** Score = 31

(h) Score = 21 **(i)** Score = 11 **(j)** Score = 1

Figure 18: Examples of finger images from the NFIQ 2.0 compliance test set and their NFIQ 2.0 scores: ((a)) FVC2000/Db3/25_3; ((b)) FVC2002/Db1/106_5; ((c)) FVC2000/Db1/20_6; ((d)) FVC2000/Db1/105_6; ((e)) FVC2000/Db1/108_4; ((f)) FVC2000/Db1/67_3; ((g)) FVC2000/Db1/89_3; ((h)) FVC2000/Db1/87_6; ((i)) FVC2000/Db1/98_2; ((j)) FVC2000/Db3/19_4.

A.3 Conduction of compliance testing

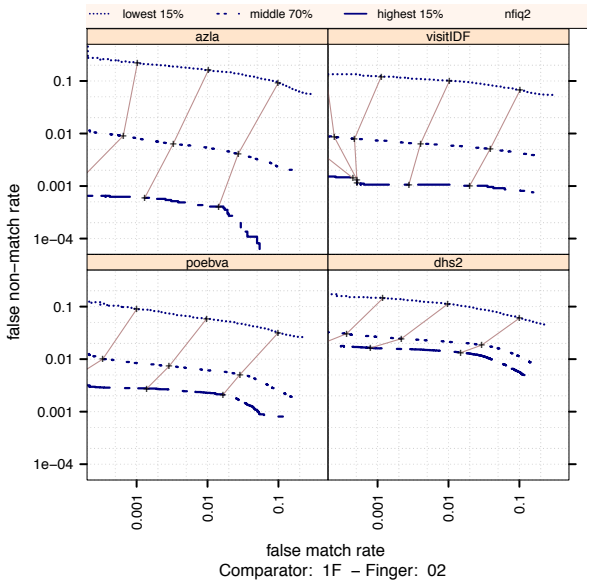
Parties who wish to verify the compliance of their implementations with the NFIQ 2.0 reference implementation shall run their implementation on all images in the NFIQ 2.0 compliance test set. In order to claim compliance of an implementation to NFIQ 2.0, at most five of the quality scores computed for the NFIQ 2.0 compliance test set are allowed to differ from the given expected NFIQ 2.0 scores, and no quality score must differ from the expected NFIQ 2.0 score by more than one score point.

In order to aid analysis of errors, for all finger images of the NFIQ 2.0 compliance test set also all feature values computed by the NFIQ 2.0 reference implementation on a Windows platform are stored in the NFIQ 2.0 Perforce depot at //depot/projects/NFIQ2/Main/data. Running the NFIQ 2.0 reference implementation on a Linux platform or MacOS platform yields slightly different results; yet at most five of the quality scores differ from those computed on the Windows platform, and no quality score differs by more than one score point. The deviations are mainly due to differences among the FingerJet FX feature values computed on different platforms.

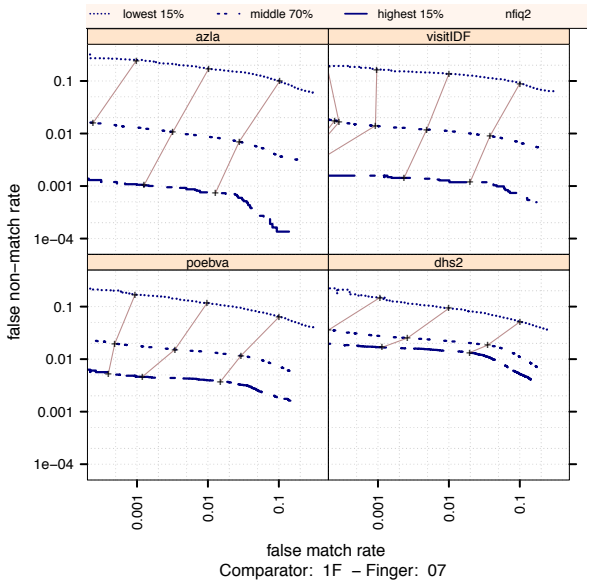
Appendix B

Results for provider 1F

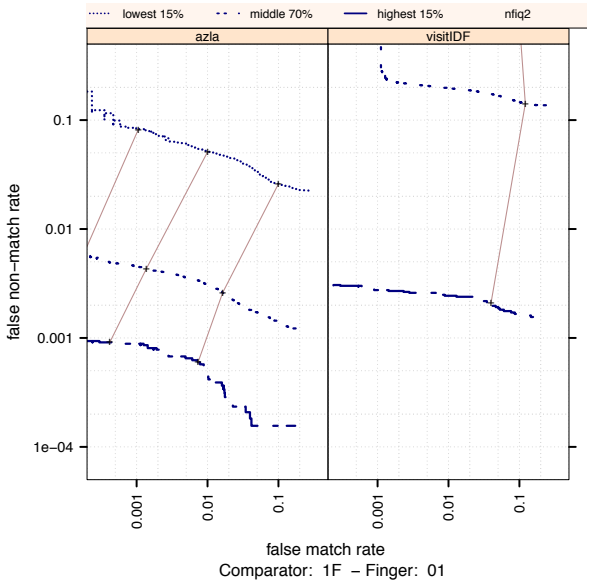
This appendix contains results for provider 1F.



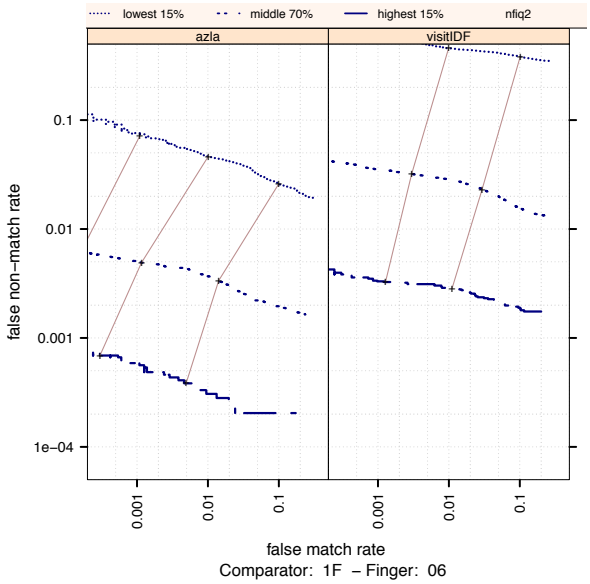
(a) Provider 1F. Right Index



(b) Provider 1F. Left Index

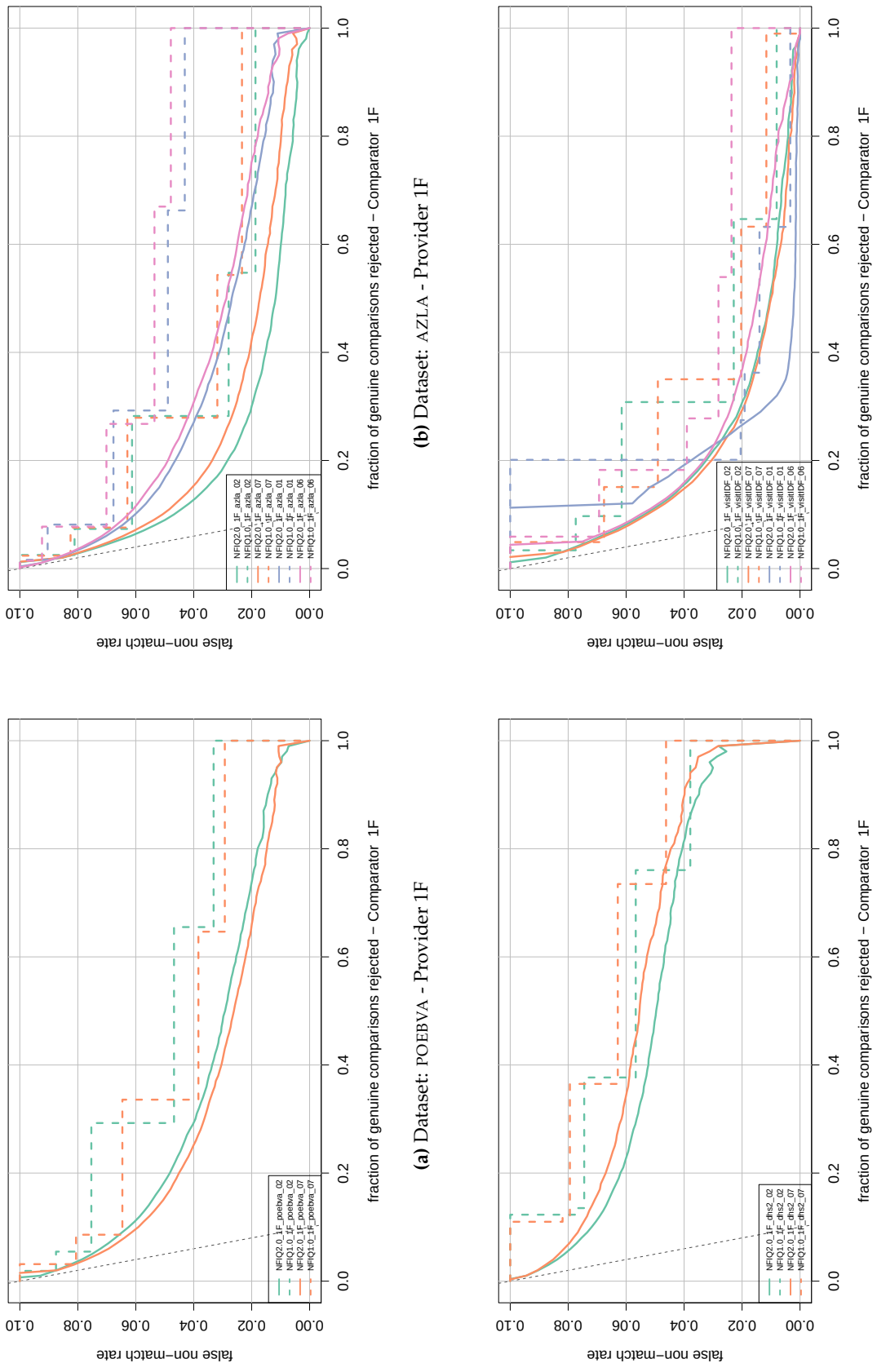


(c) Provider 1F. Right Thumb



(d) Provider 1F. Left Thumb

Figure 19:
Ranked DET. Provider 1F. The set of all comparisons are partitioned into three groups based on the pair-wise NFIQ2.0 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 2.0 values appearing above, and the DET curve of highest NFIQ 2.0 values below all the other curves.



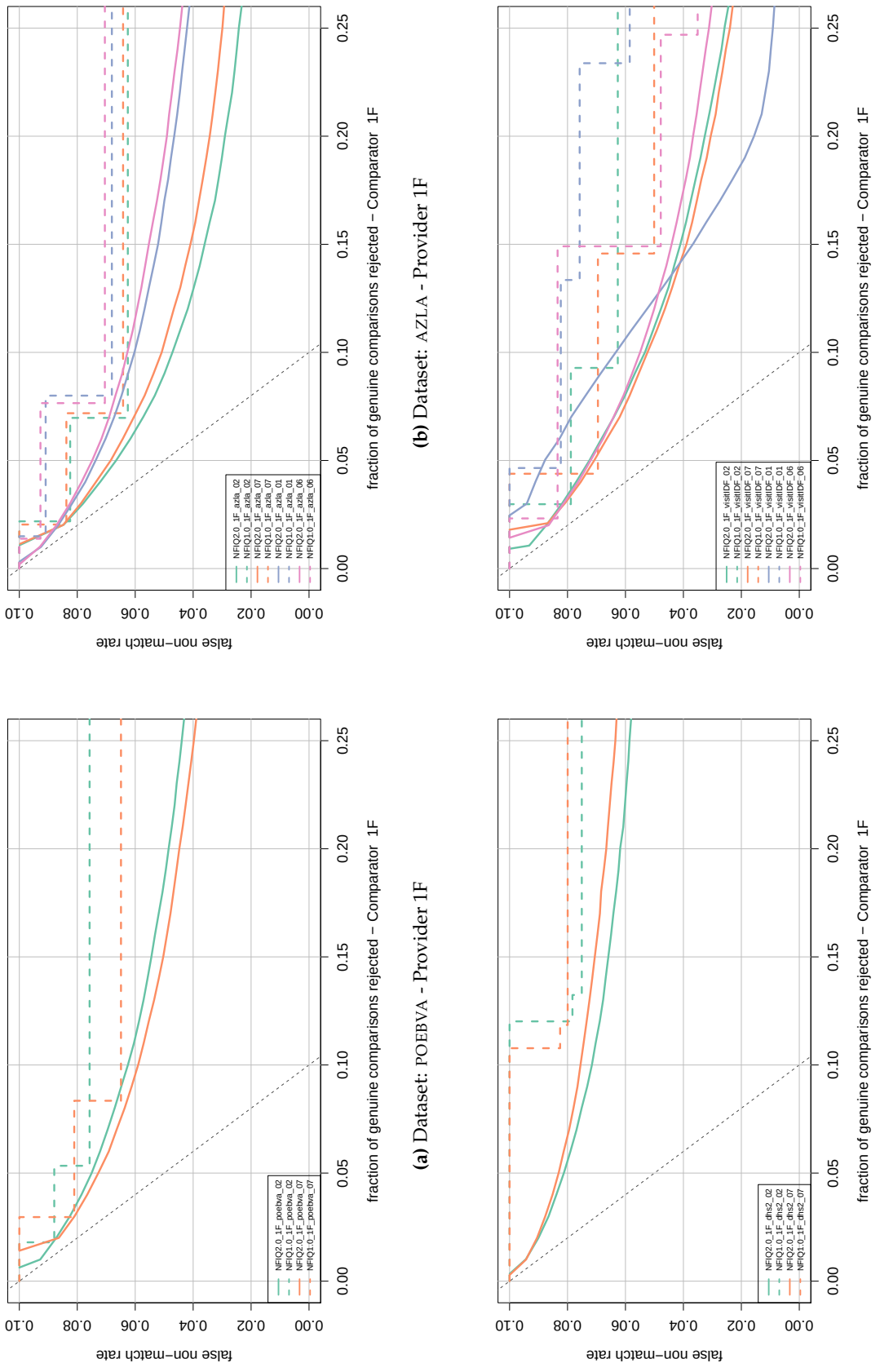
(a) Dataset: POEBVA - Provider 1F

(b) Dataset: AZLA - Provider 1F

(c) Provider 1F

(d) Dataset: VISITIDF - Provider 1F

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 2.0 is a better predictive of performance than NFIQ 1.0 because its error vs. reject curves are closer to the gray dotted line.



(a) Dataset: POEBVA - Provider 1F

(b) Dataset: AZLA - Provider 1F

(c) Provider 1F

(d) Dataset: VISITIDF - Provider 1F

Figure 21:
Error vs. reject curve. Provider 1F.The threshold is set to give an initial false non-match rate of 0.1. Same result as in fig 20 but for rejection rate of up to 25%.

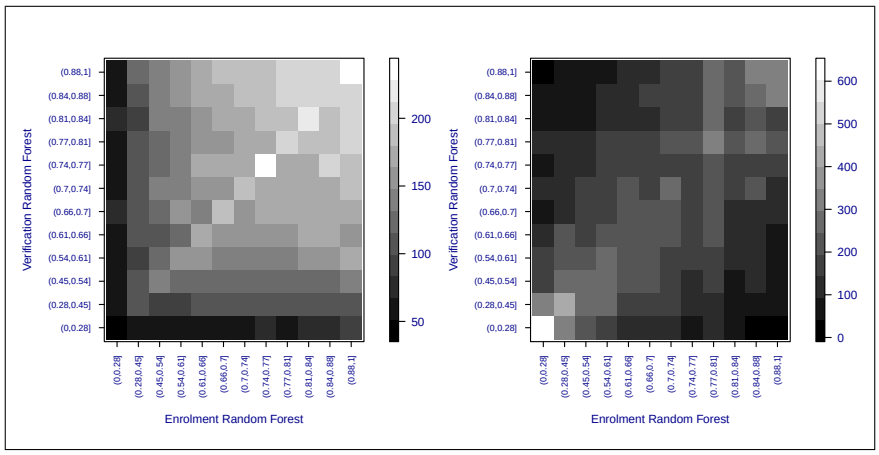


Figure 22: Heatmap for provider 1F on dataset azla: Left plot shows the mean comparison score, right plot shows the sample count.

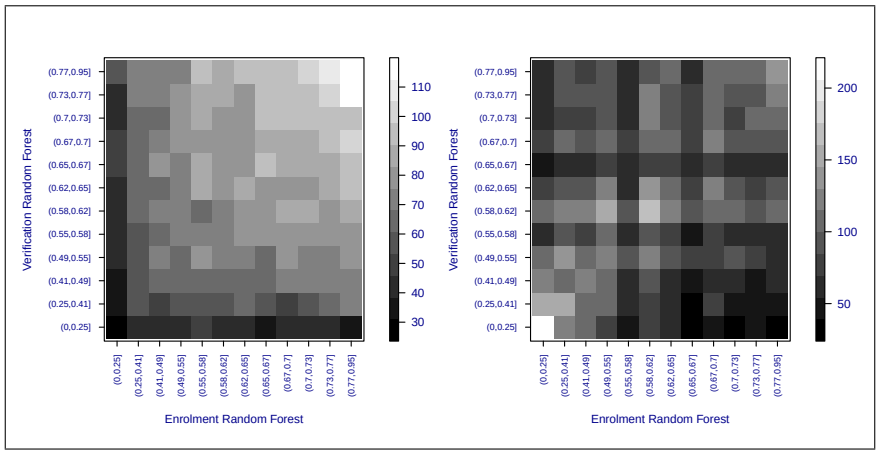


Figure 23: Heatmap for provider 1F on dataset dhs2: Left plot shows the mean comparison score, right plot shows the sample count.

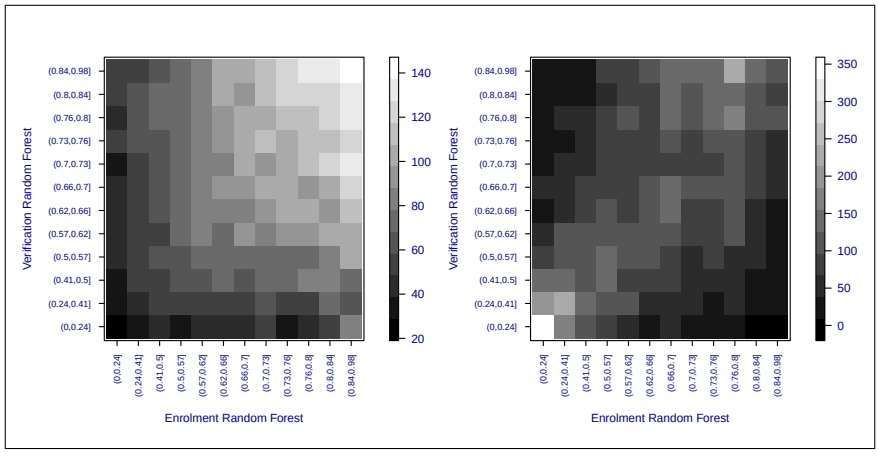
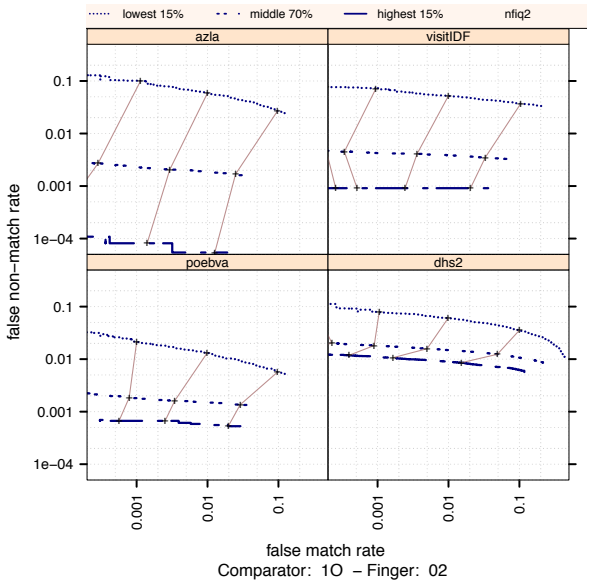


Figure 24: Heatmap for provider 1F on dataset poebva: Left plot shows the mean comparison score, right plot shows the sample count.

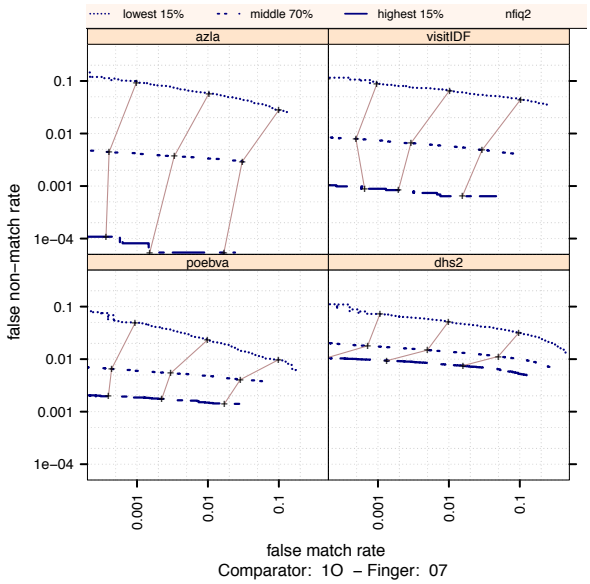
Appendix C

Results for provider 1O

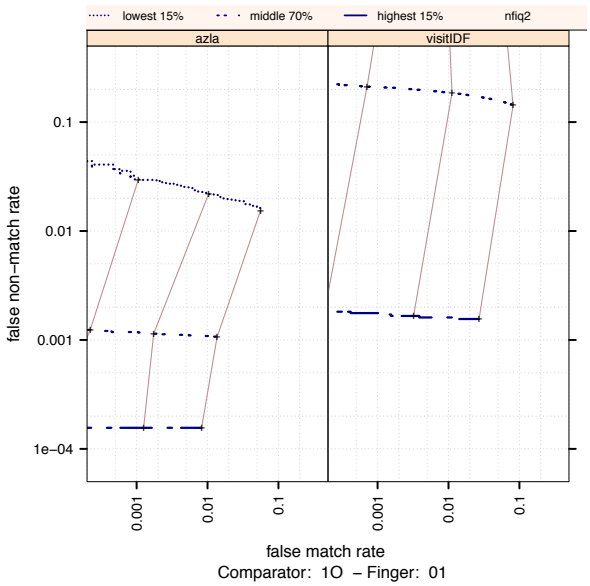
This appendix contains results for provider 1O.



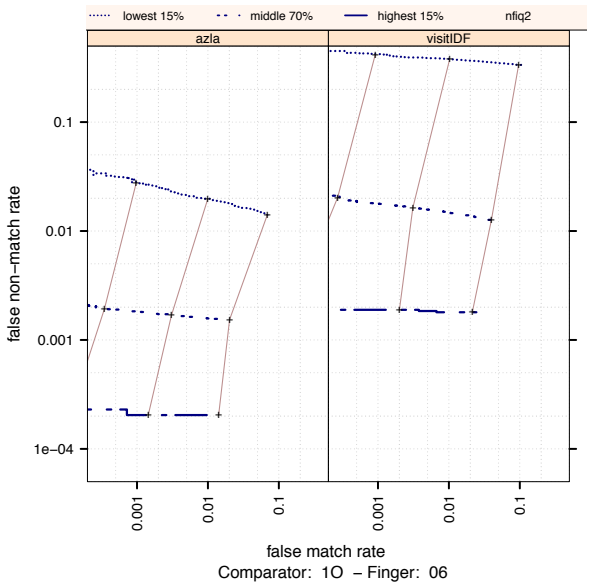
(a) provvider 1O. Right Index



(b) Provider 1O. Left Index

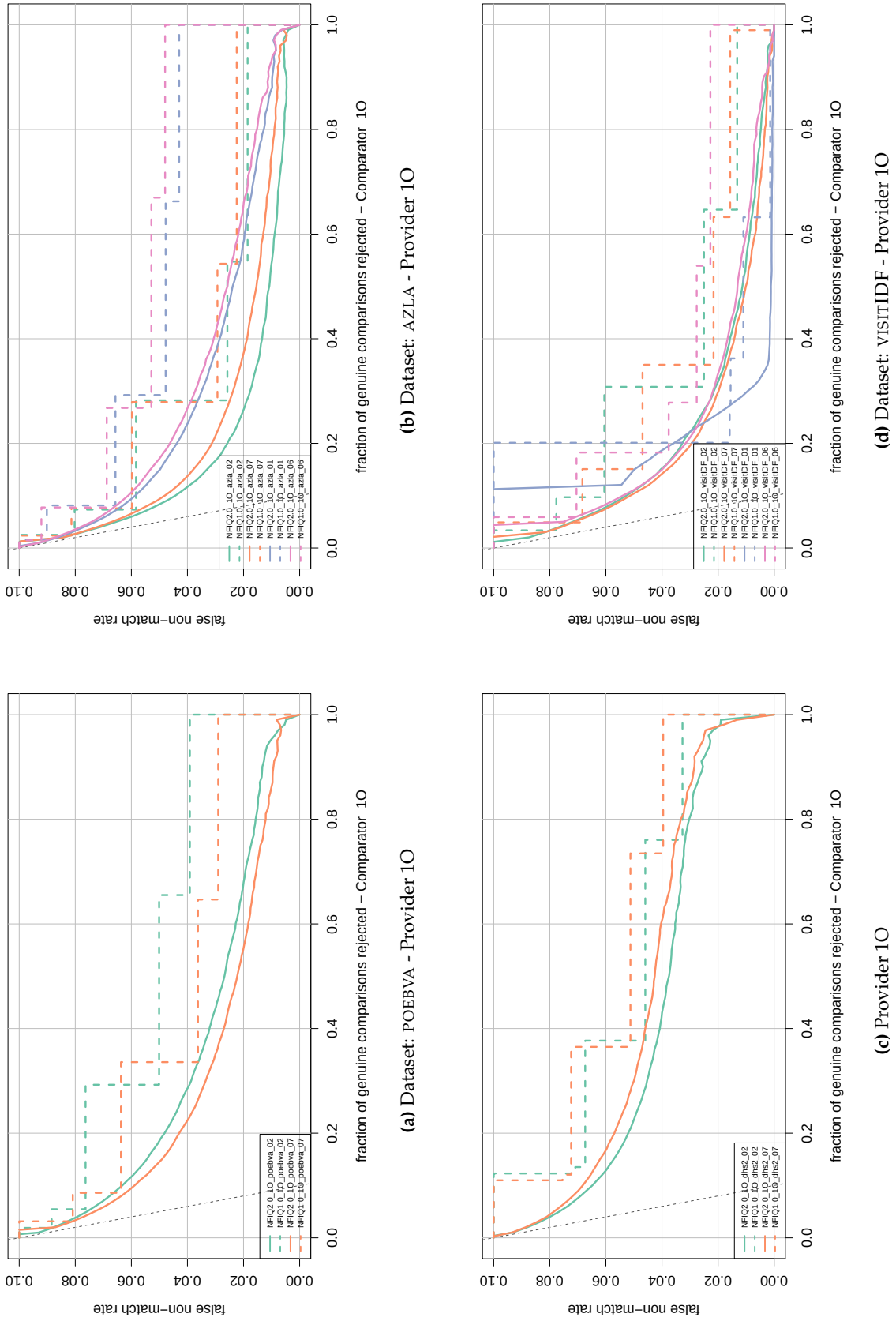


(c) Provider 1O. Right Thumb



(d) Provider 1O. Left Thumb

Figure 25:
Ranked DET. Provider 1O.The set of all comparisons are partitioned into three groups based on the pair-wise NFIQ2.0 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 2.0 values appearing above, and the DET curve of highest NFIQ 2.0 values below all the other curves.



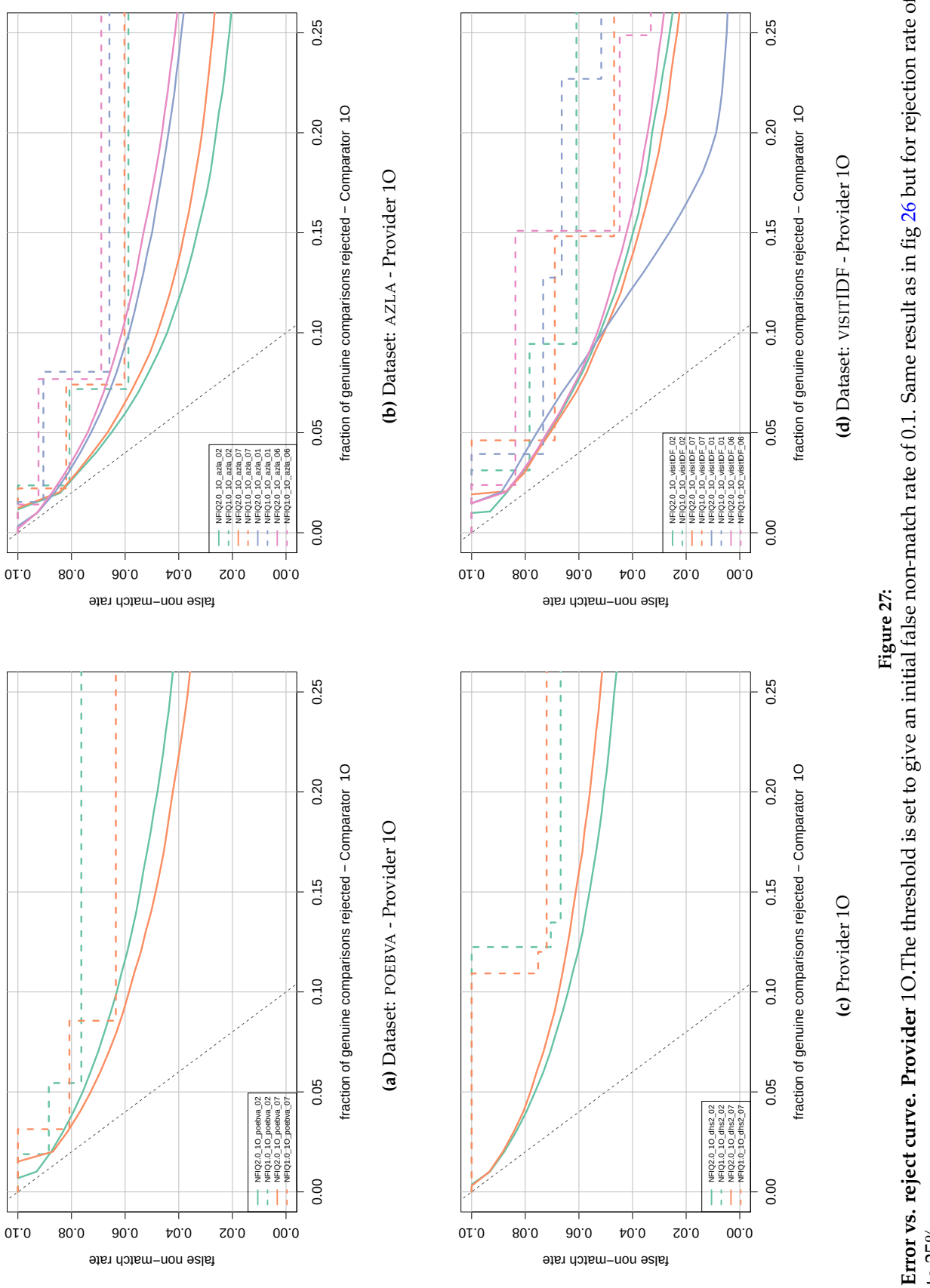
(a) Dataset: POEBVA - Provider 1O

(b) Dataset: AZLA - Provider 1O

(c) Provider 1O

(d) Dataset: VISITIDF - Provider 1O

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 2.0 is a better predictive of performance than NFIQ 1.0 because its error vs. reject curves are closer to the gray dotted line.



(a) Dataset: POEBVA - Provider 1O

(b) Dataset: AZLA - Provider 1O

(c) Provider 1O

(d) Dataset: VISITIDF - Provider 1O

Figure 27:
Error vs. reject curve. Provider 1O.The threshold is set to give an initial false non-match rate of 0.1. Same result as in fig 26 but for rejection rate of up to 25%.

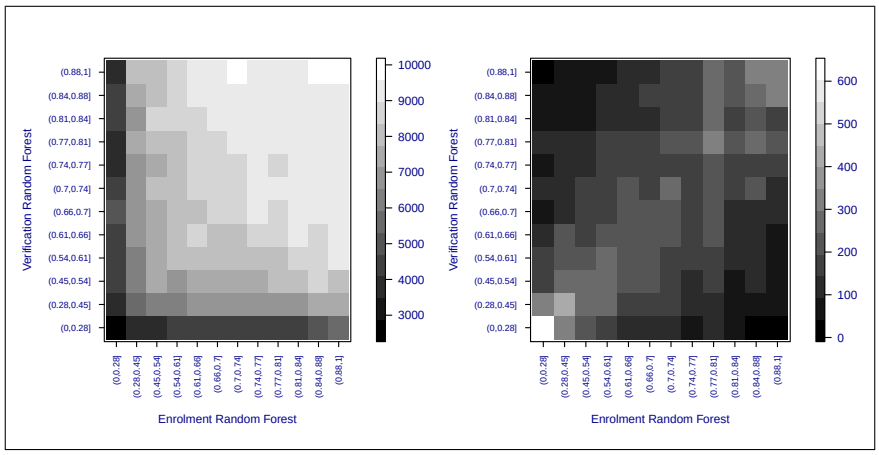


Figure 28: Heatmap for provider 1O on dataset azla: Left plot shows the mean comparison score, right plot shows the sample count.

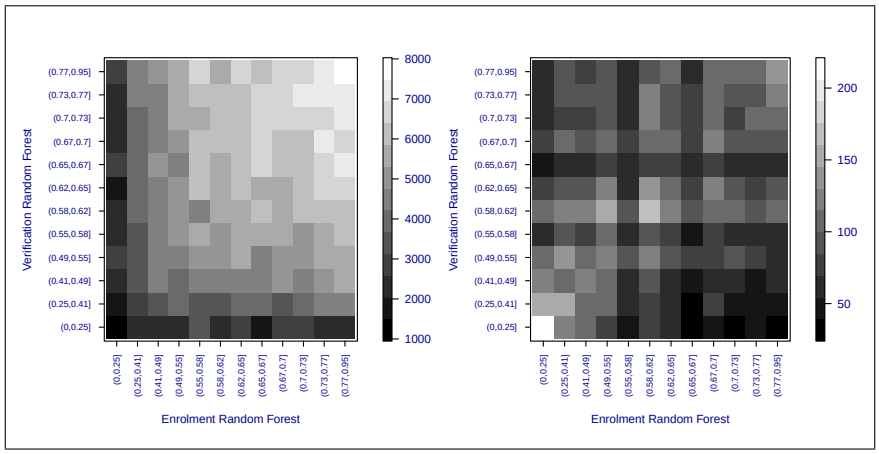


Figure 29: Heatmap for provider 1O on dataset dhs2: Left plot shows the mean comparison score, right plot shows the sample count.

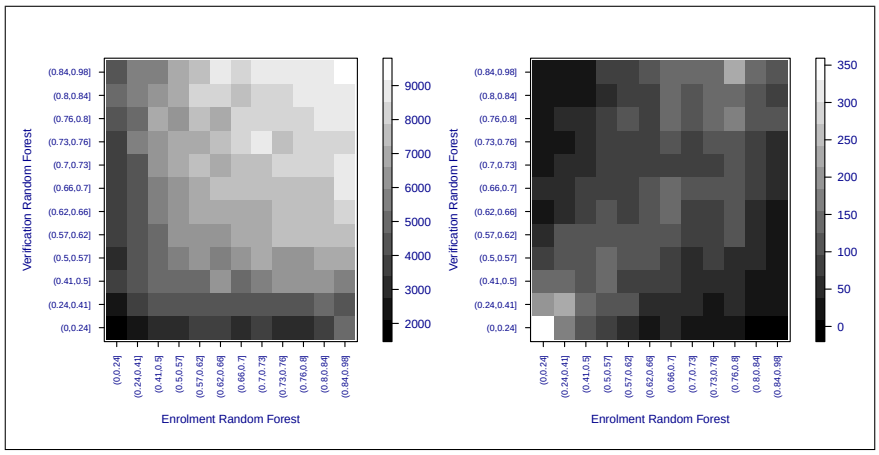
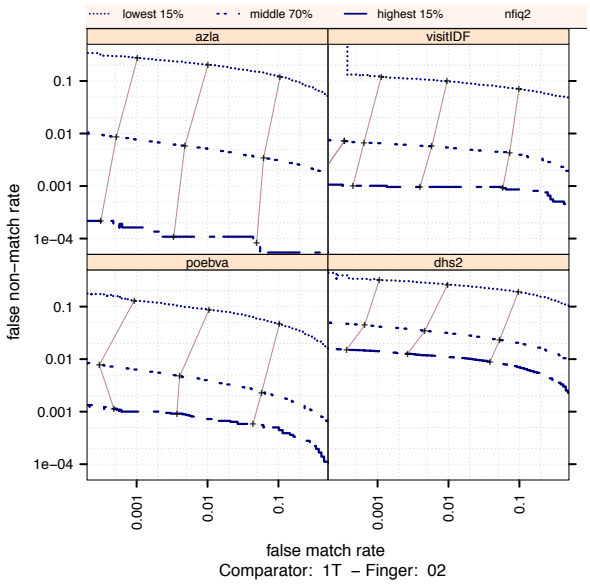


Figure 30: Heatmap for provider 1O on dataset poebva: Left plot shows the mean comparison score, right plot shows the sample count.

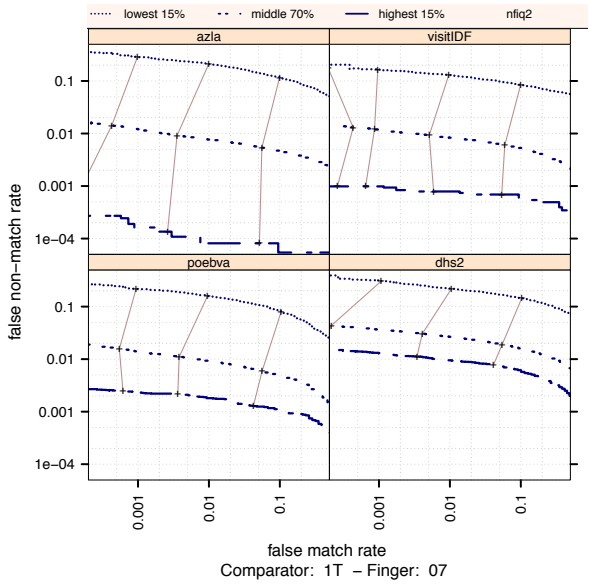
Appendix D

Results for provider 1T

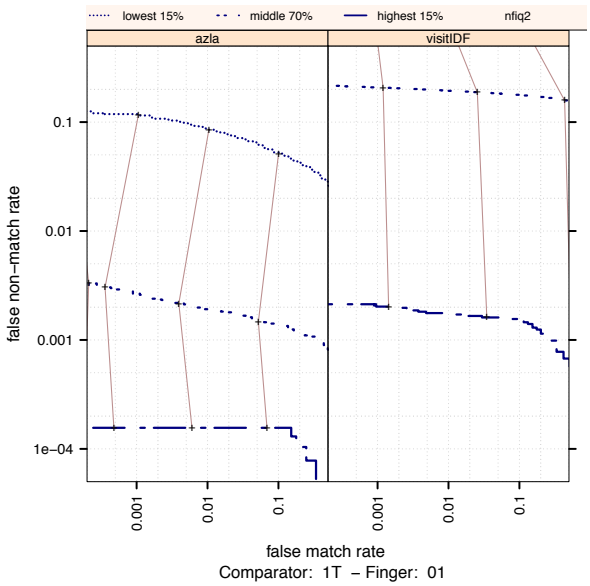
This appendix contains results for provider 1T.



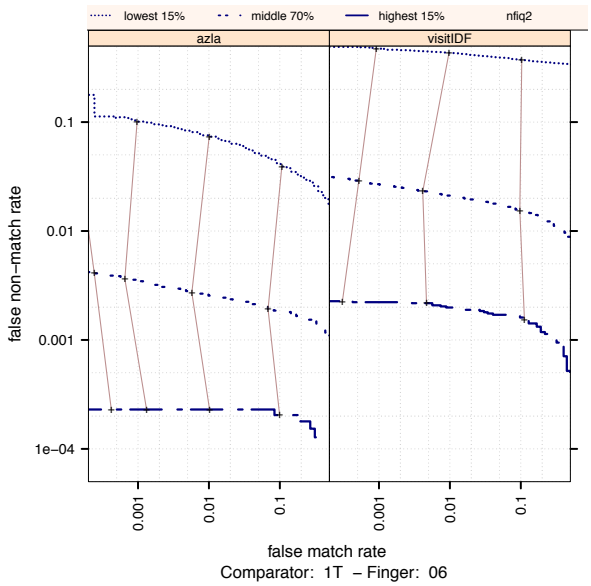
(a) provvider 1T. Right Index



(b) Provider 1T. Left Index

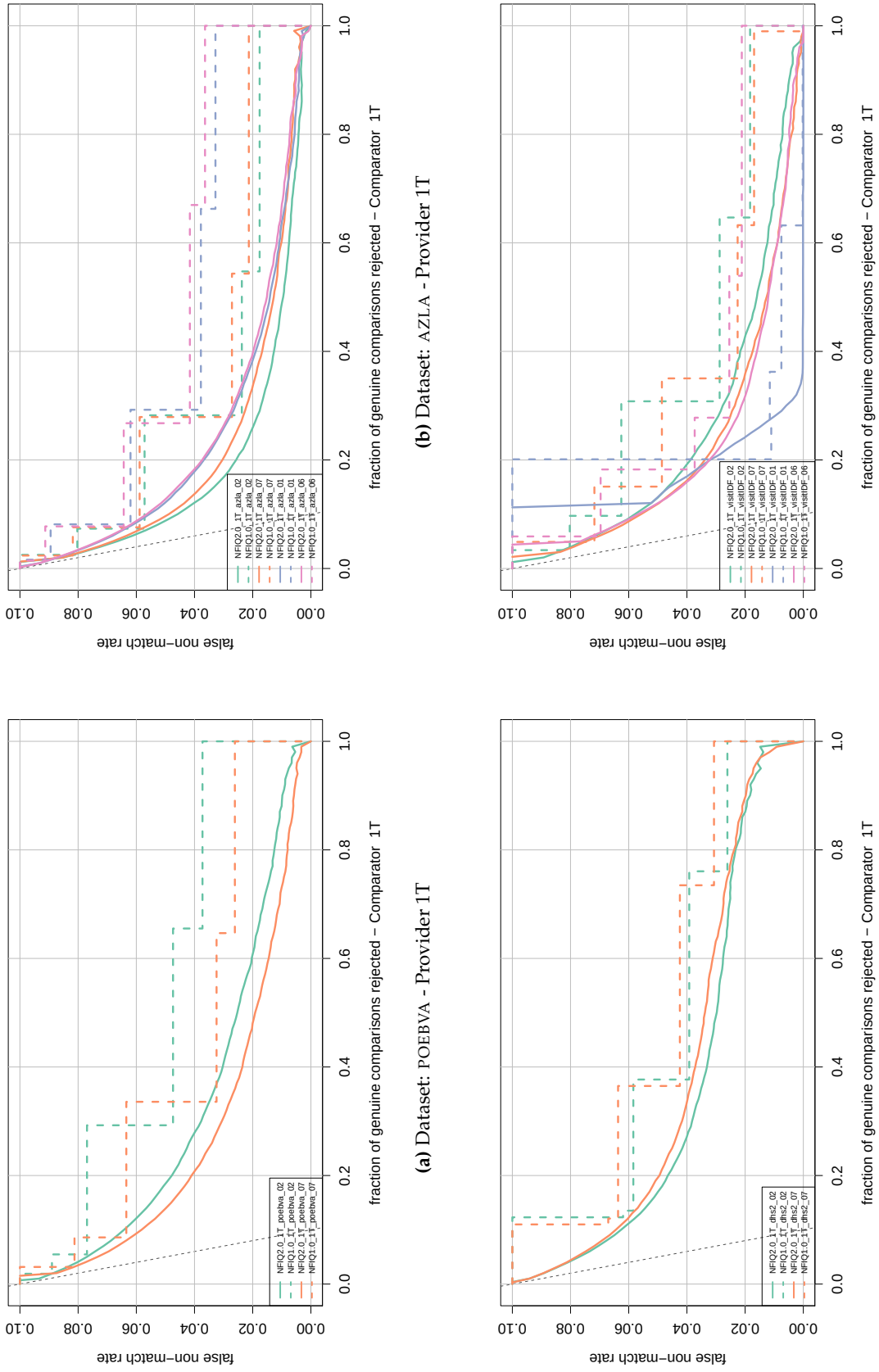


(c) Provider 1T. Right Thumb



(d) Provider 1T. Left Thumb

Figure 31:
Ranked DET. Provider 1T.The set of all comparisons are partitioned into three groups based on the pair-wise NFIQ2.0 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 2.0 values appearing above, and the DET curve of highest NFIQ 2.0 values below all the other curves.



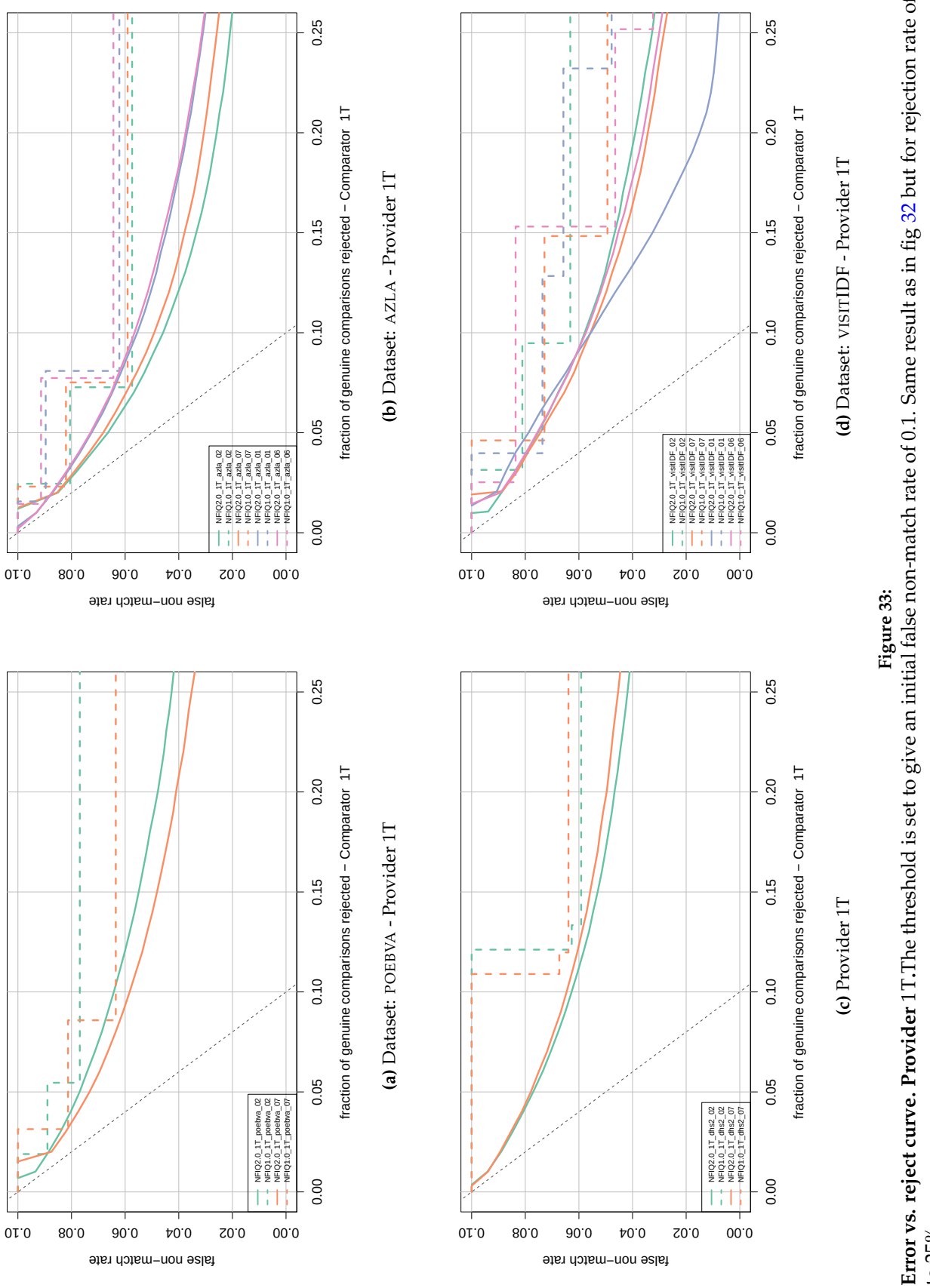
(a) Dataset: POEBVA - Provider 1T

(b) Dataset: AZLA - Provider 1T

(c) Provider 1T

(d) Dataset: VISITIDF - Provider 1T

Figure 32:
Error vs. reject curve. 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. NFIQ 2.0 is a better predictive of performance than NFIQ 1.0 because its error vs. reject curves are closer to the gray dotted line.



(a) Dataset: POEBVA - Provider 1T

(b) Dataset: AZLA - Provider 1T

(c) Provider 1T

(d) Dataset: VISITIDF - Provider 1T

Figure 33:
Error vs. reject curve. Provider 1T.The threshold is set to give an initial false non-match rate of 0.1. Same result as in fig 32 but for rejection rate of up to 25%.

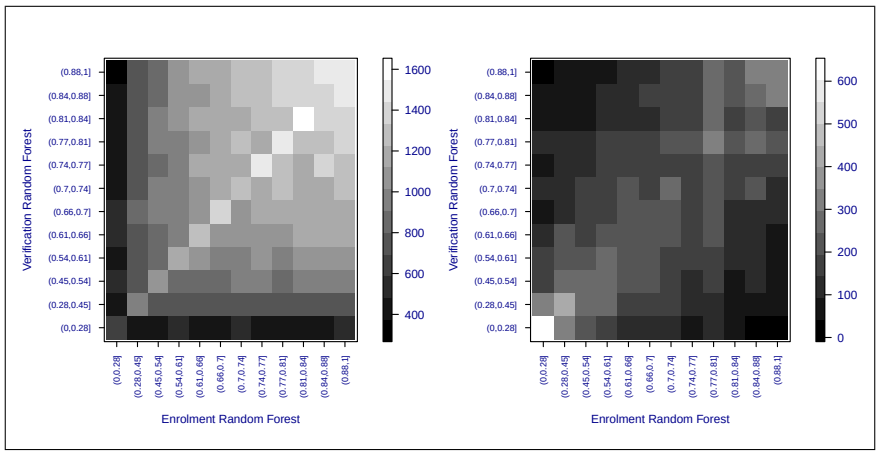


Figure 34: Heatmap for provider 1T on dataset azla: Left plot shows the mean comparison score, right plot shows the sample count.

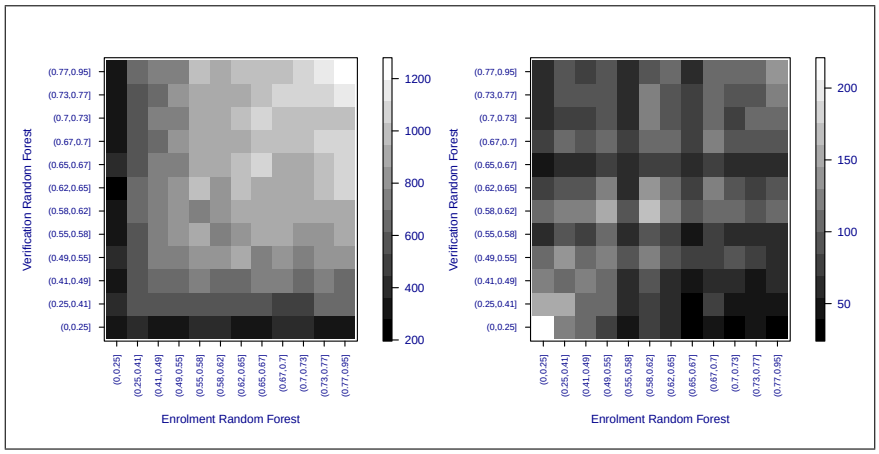


Figure 35: Heatmap for provider 1T on dataset dhs2: Left plot shows the mean comparison score, right plot shows the sample count.

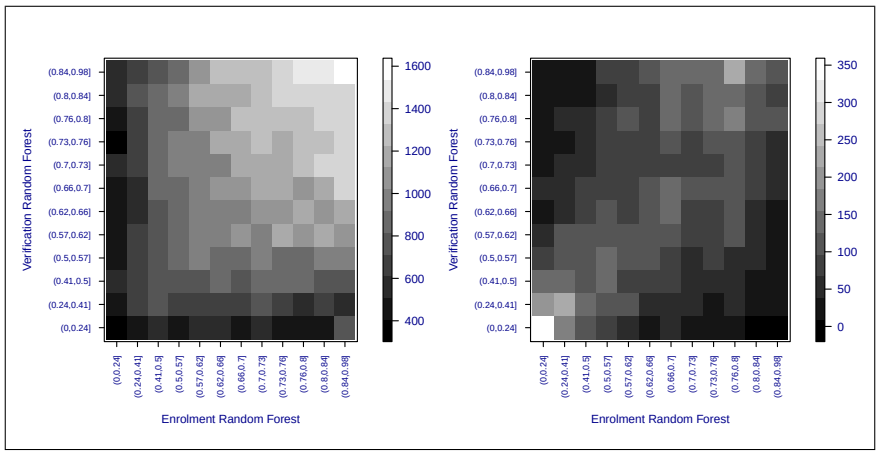
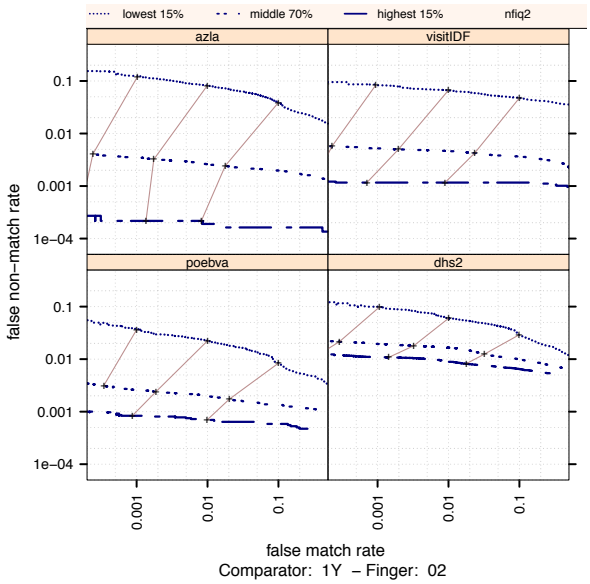


Figure 36: Heatmap for provider 1T on dataset poebva: Left plot shows the mean comparison score, right plot shows the sample count.

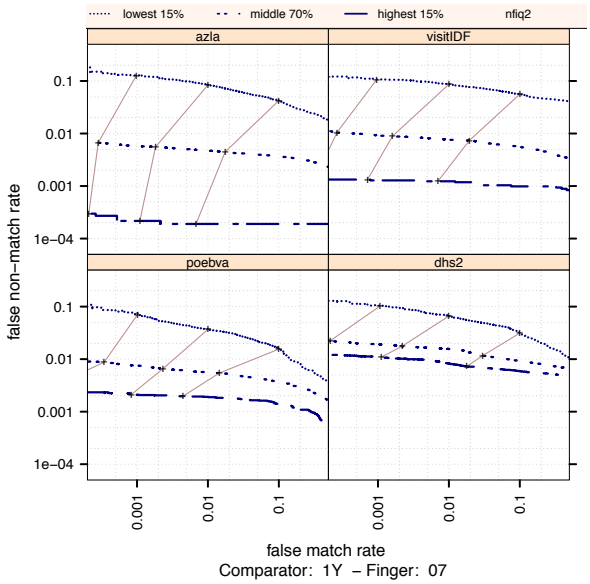
Appendix E

Results for provider 1Y

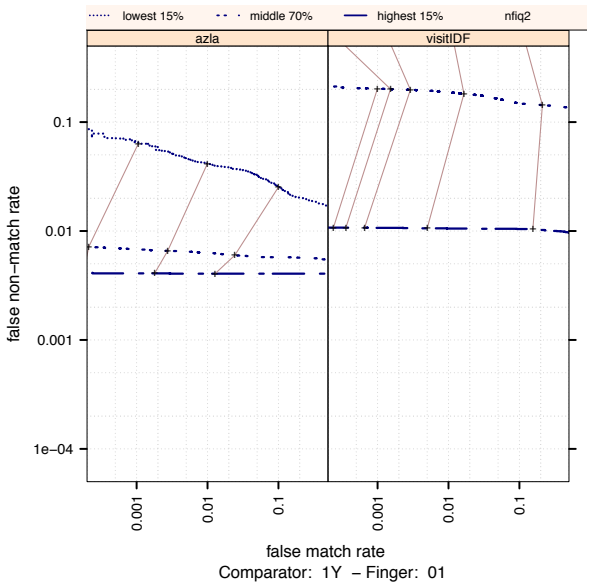
This appendix contains results for provider 1Y.



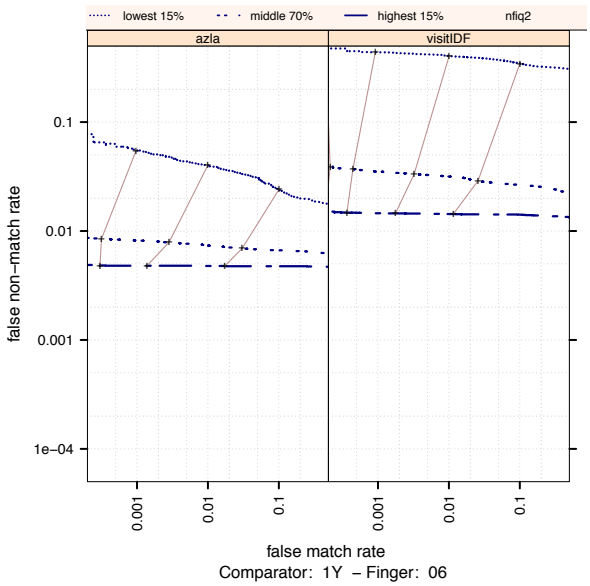
(a) provvider 1Y. Right Index



(b) Provider 1Y. Left Index

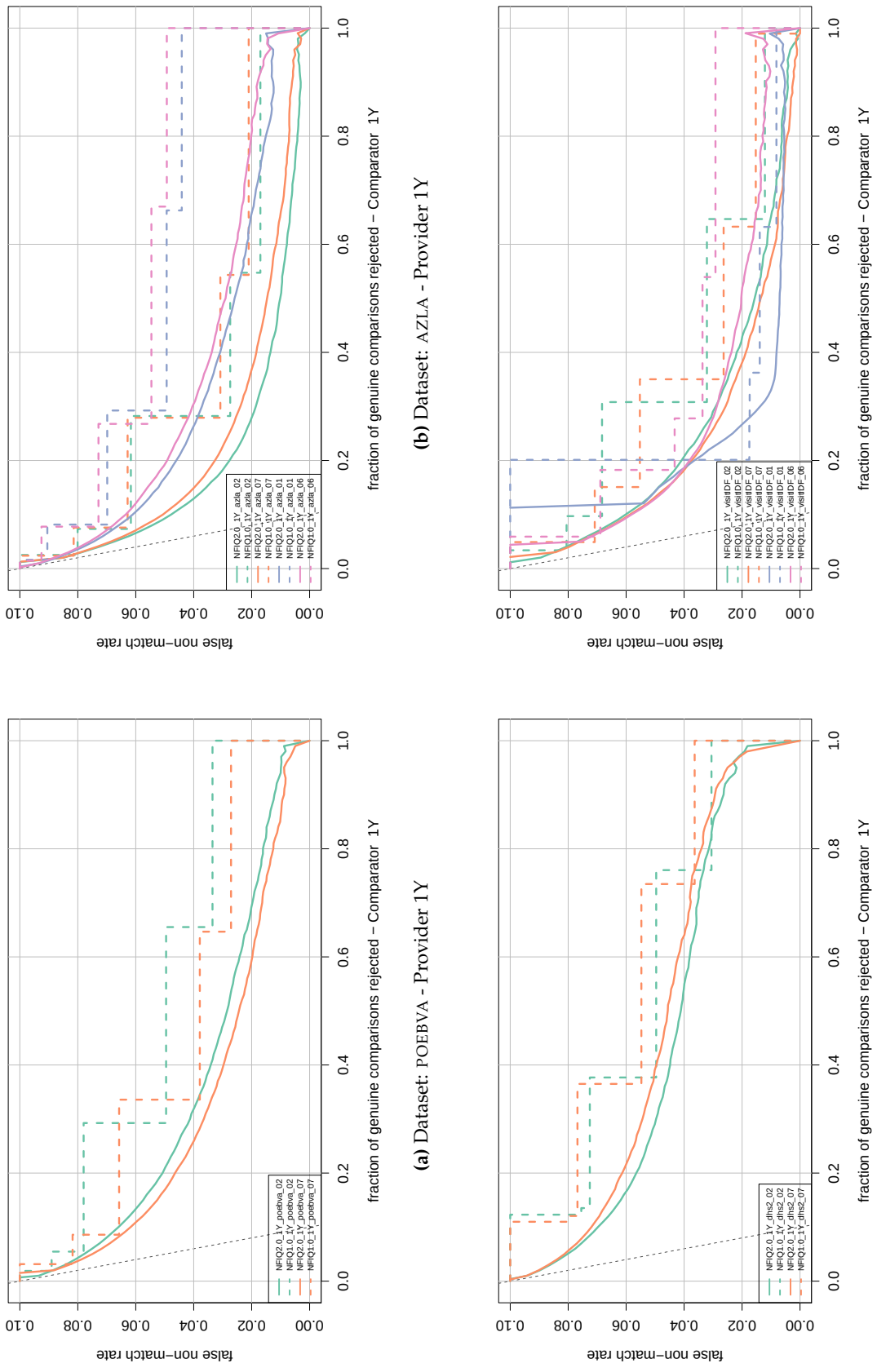


(c) Provider 1Y. Right Thumb



(d) Provider 1Y. Left Thumb

Figure 37:
Ranked DET. Provider 1Y. The set of all comparisons are partitioned into three groups based on the pair-wise NFIQ2.0 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 2.0 values appearing above, and the DET curve of highest NFIQ 2.0 values below all the other curves.



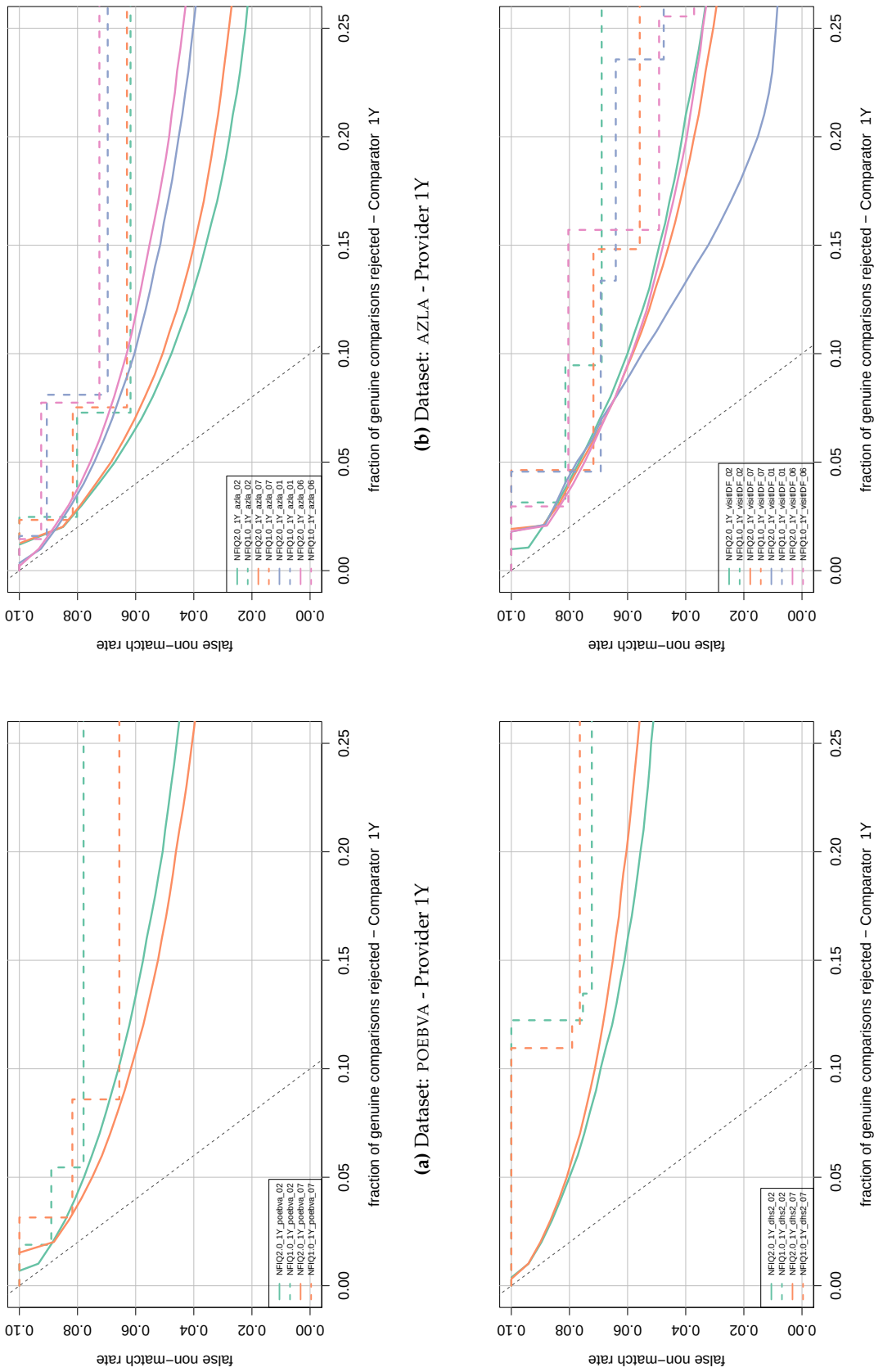
(a) Dataset: POEBVA - Provider 1Y

(b) Dataset: AZLA - Provider 1Y

(c) Provider 1Y

(d) Dataset: VISITIDF - Provider 1Y

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 2.0 is a better predictive of performance than NFIQ 1.0 because its error vs. reject curves are closer to the gray dotted line.



(a) Dataset: POEBVA - Provider 1Y

(b) Dataset: AZLA - Provider 1Y

(c) Provider 1Y

(d) Dataset: VISITIDF - Provider 1Y

Figure 39:
Error vs. reject curve. Provider 1Y.The threshold is set to give an initial false non-match rate of 0.1. Same result as in fig 38 but for rejection rate of up to 25%.

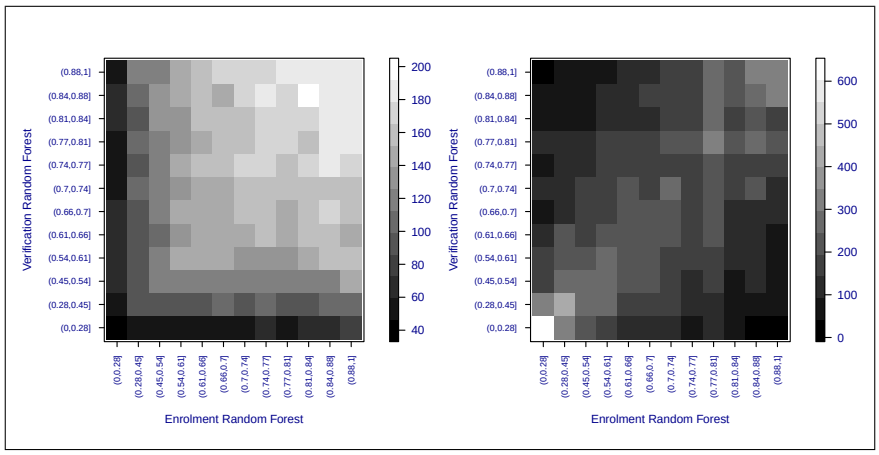


Figure 40: Heatmap for provider 1Y on dataset azla: Left plot shows the mean comparison score, right plot shows the sample count.

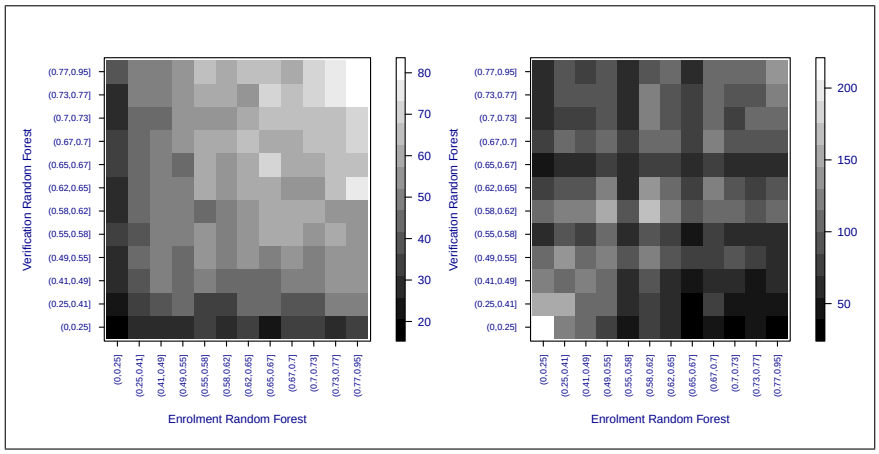


Figure 41: Heatmap for provider 1Y on dataset dhs2: Left plot shows the mean comparison score, right plot shows the sample count.

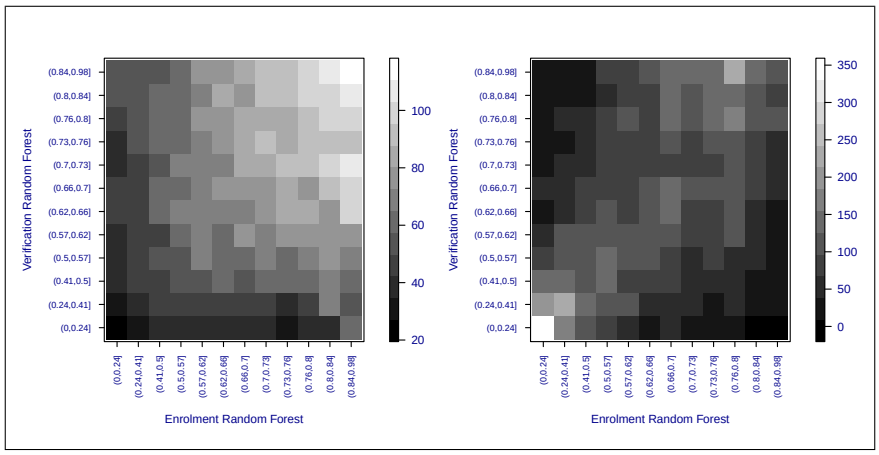
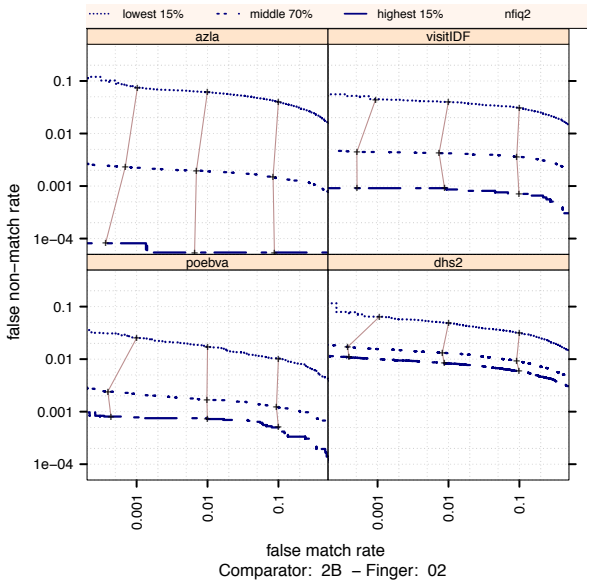


Figure 42: Heatmap for provider 1Y on dataset poebva: Left plot shows the mean comparison score, right plot shows the sample count.

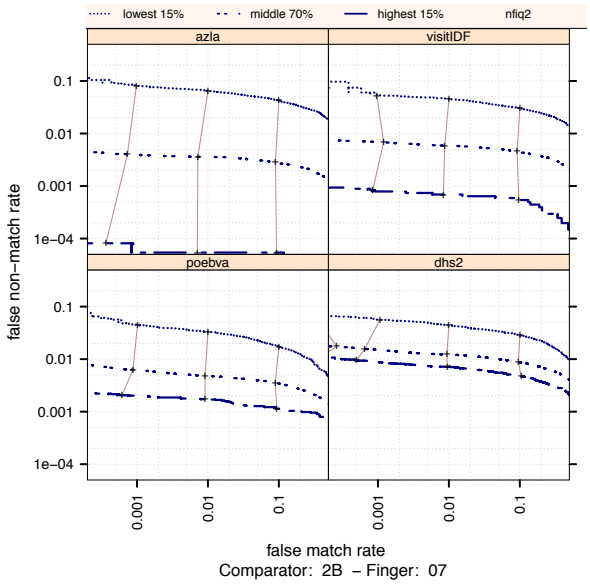
Appendix F

Results for provider 2B

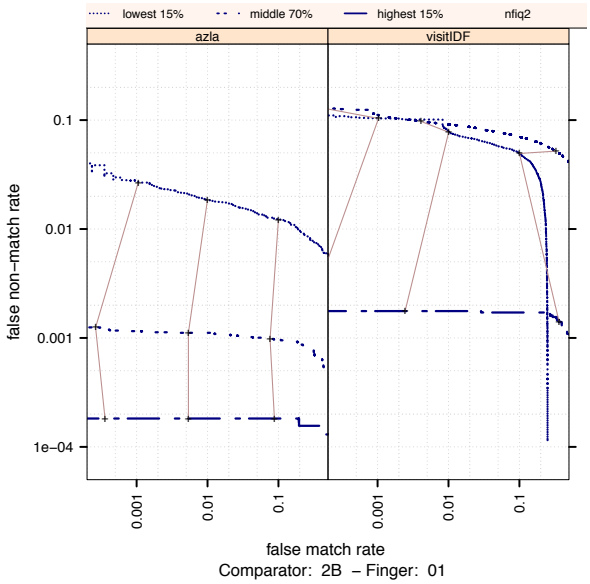
This appendix contains results for provider 2B.



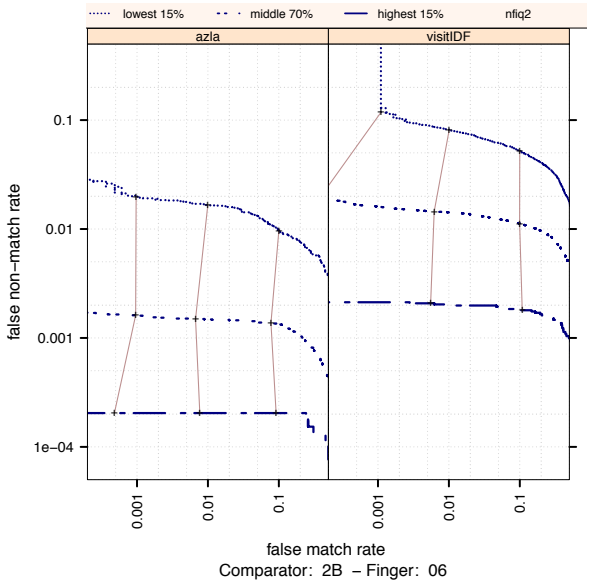
(a) provvider 2B. Right Index



(b) Provider 2B. Left Index

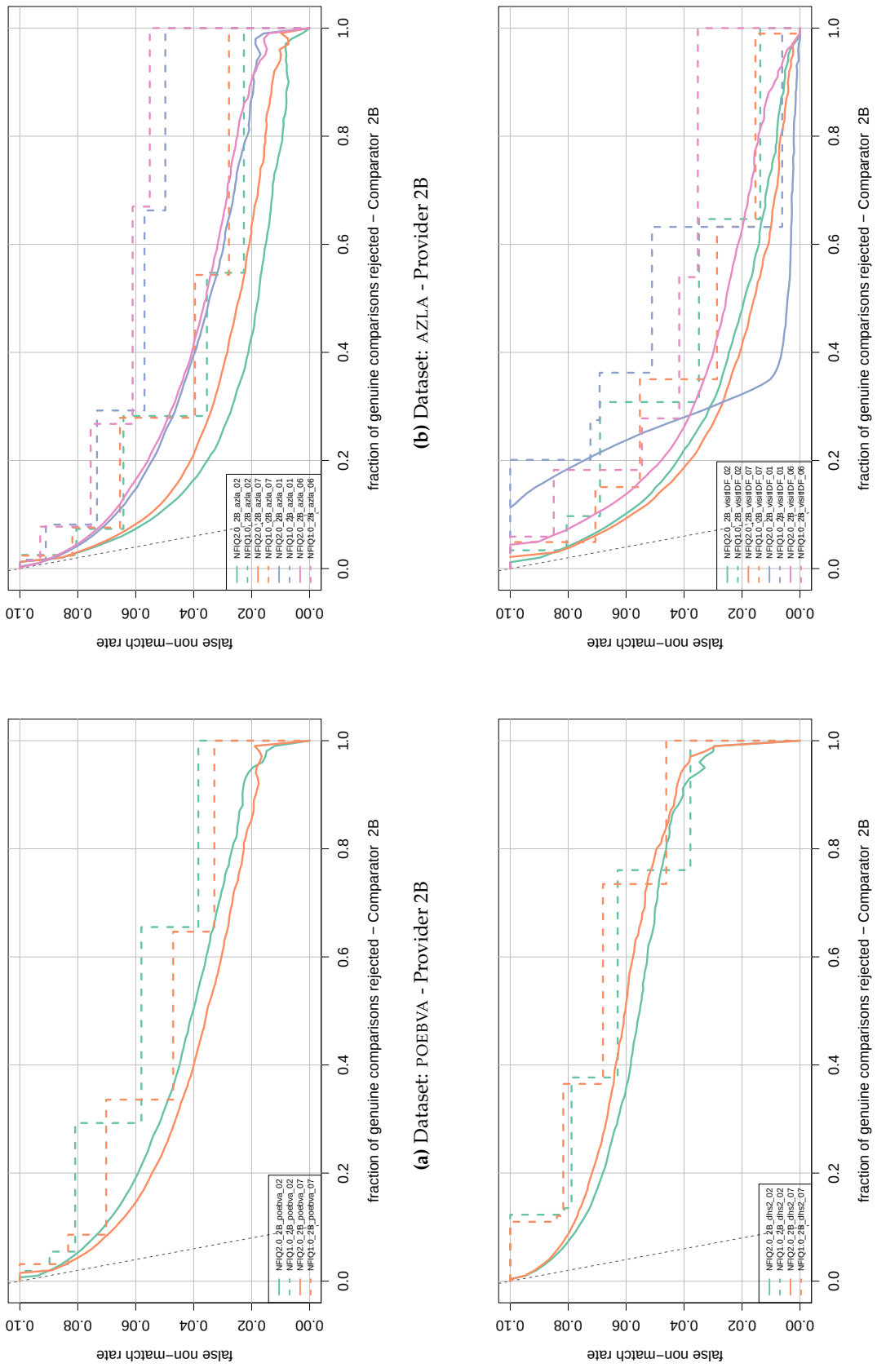


(c) Provider 2B. Right Thumb



(d) 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 NFIQ2.0 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 2.0 values appearing above, and the DET curve of highest NFIQ 2.0 values below all the other curves.



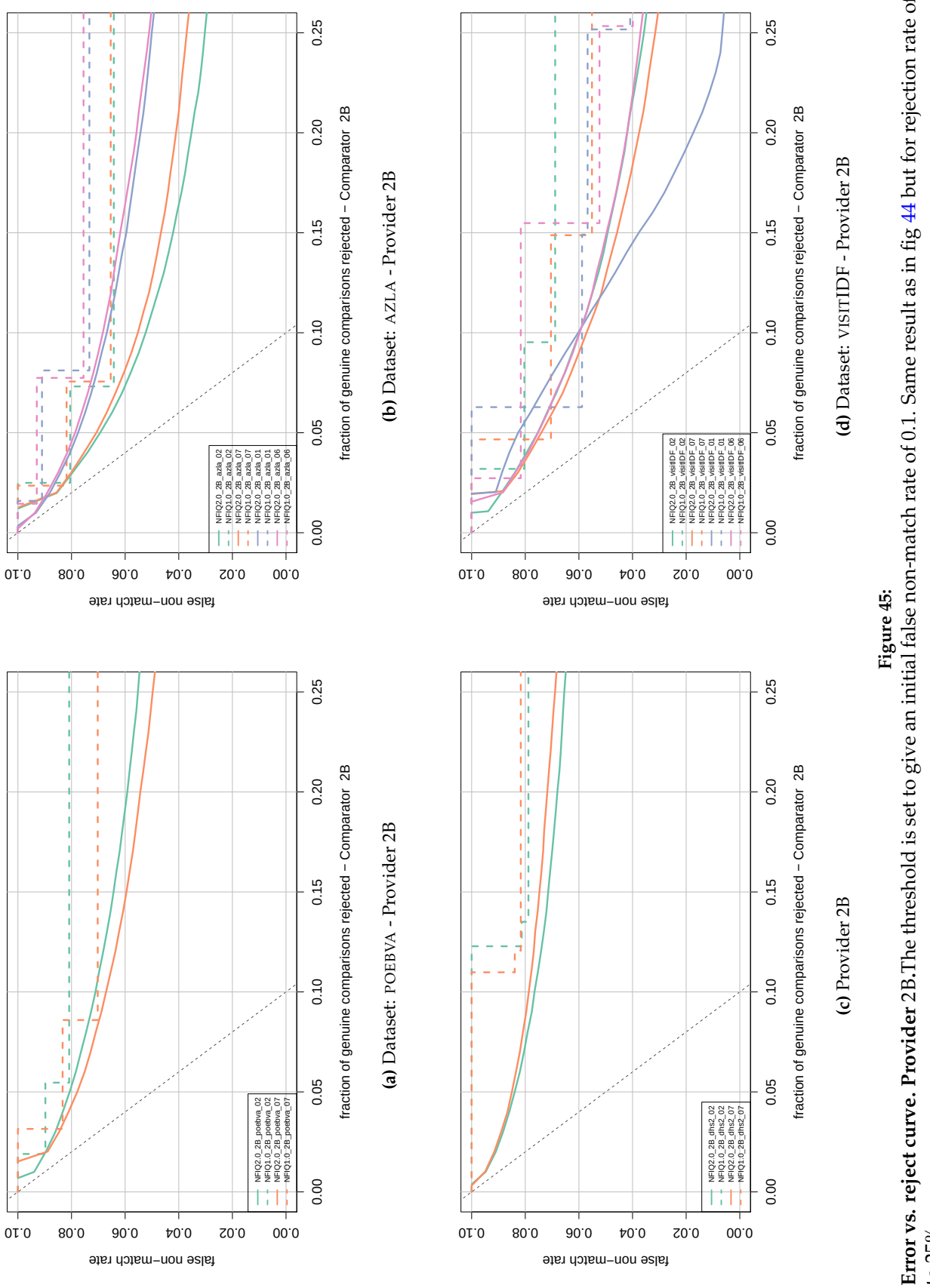
(a) Dataset: POEBVA - Provider 2B

(b) Dataset: AZLA - Provider 2B

(c) Provider 2B

(d) Dataset: VISITIDF - Provider 2B

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 2.0 is a better predictive of performance than NFIQ 1.0 because its error vs. reject curves are closer to the gray dotted line.



(a) Dataset: POEBVA - Provider 2B

(b) Dataset: AZLA - Provider 2B

(c) Provider 2B

(d) Dataset: VISITIDF - Provider 2B

Figure 45:
Error vs. reject curve. Provider 2B.The threshold is set to give an initial false non-match rate of 0.1. Same result as in fig 44 but for rejection rate of up to 25%.

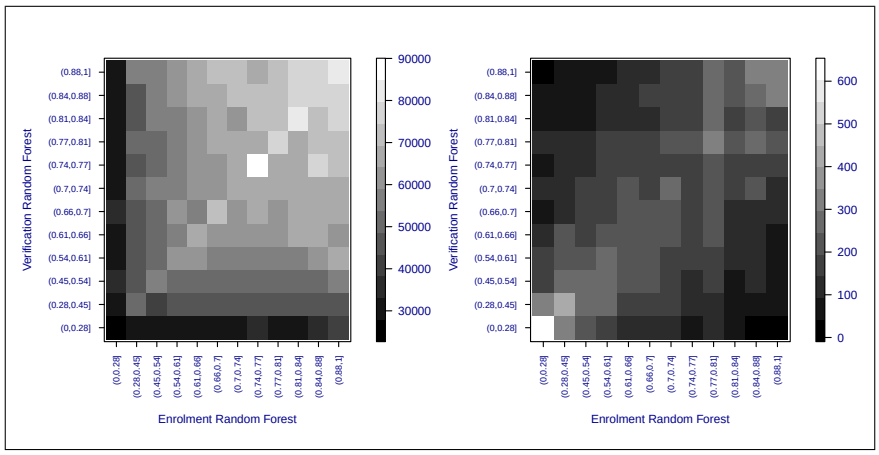


Figure 46: Heatmap for provider 2B on dataset azla: Left plot shows the mean comparison score, right plot shows the sample count.

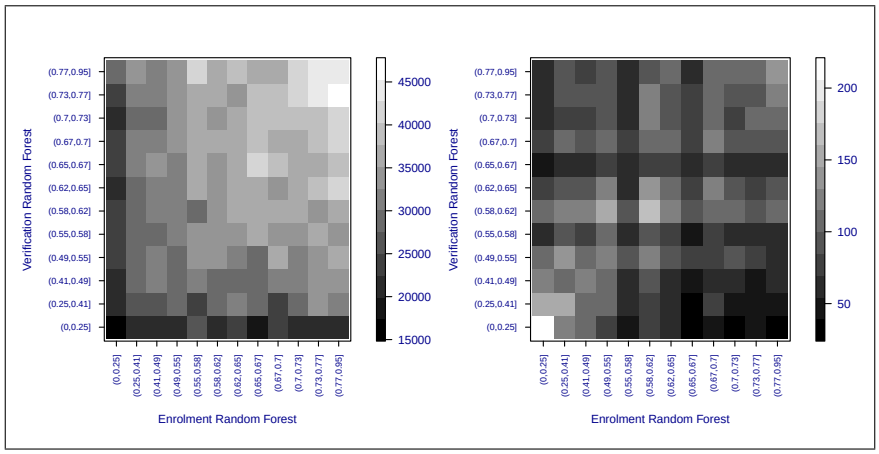


Figure 47: Heatmap for provider 2B on dataset dhs2: Left plot shows the mean comparison score, right plot shows the sample count.

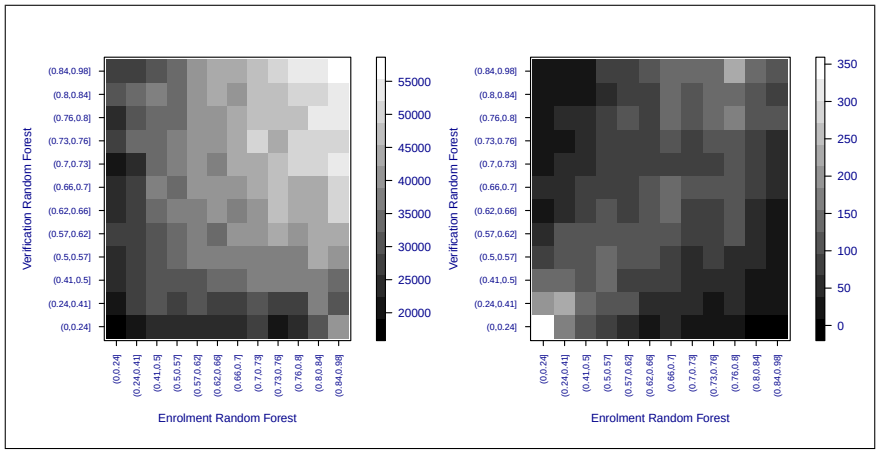
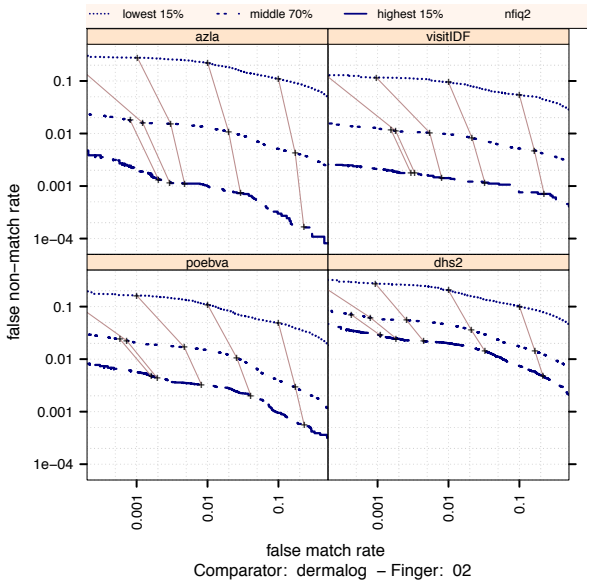


Figure 48: Heatmap for provider 2B on dataset poebva: Left plot shows the mean comparison score, right plot shows the sample count.

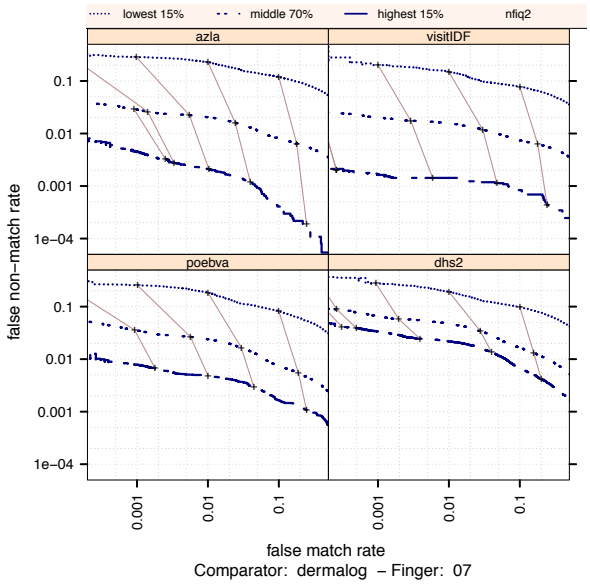
Appendix G

Results for provider DERMALOG

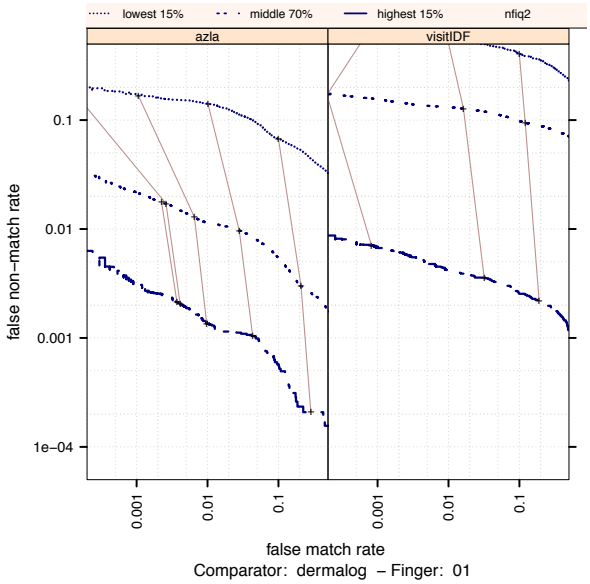
This appendix contains results for provider DERMALOG.



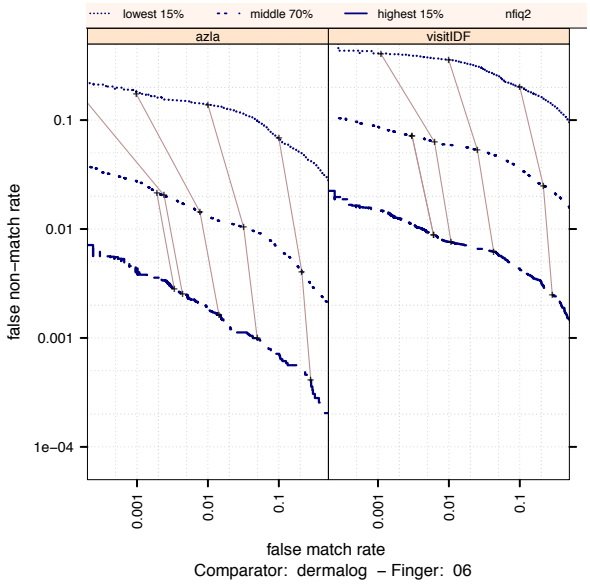
(a) provvider dermalog. Right Index



(b) Provider dermalog. Left Index

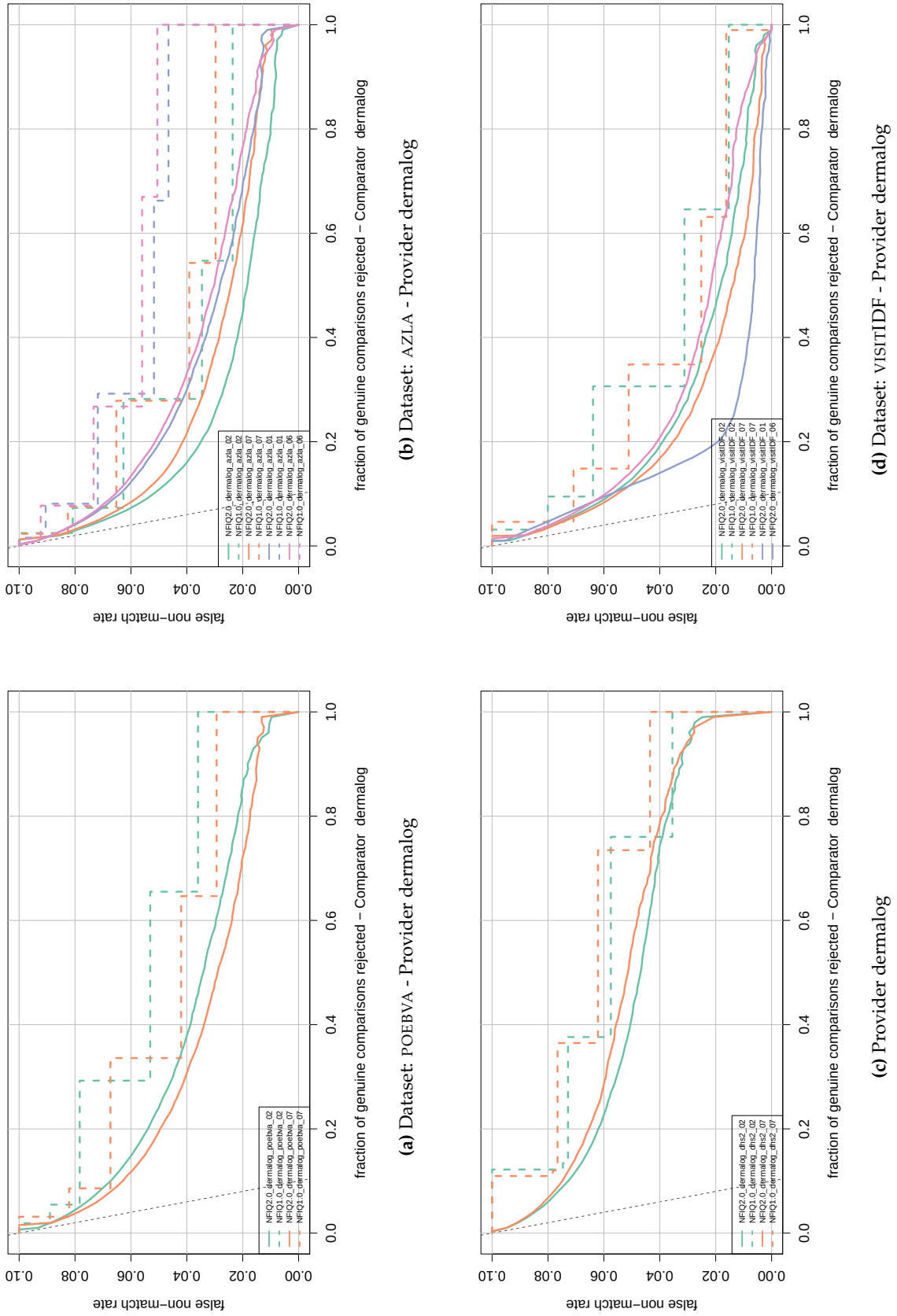


(c) Provider dermalog. Right Thumb



(d) 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 NFIQ2.0 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 2.0 values appearing above, and the DET curve of highest NFIQ 2.0 values below all the other curves.



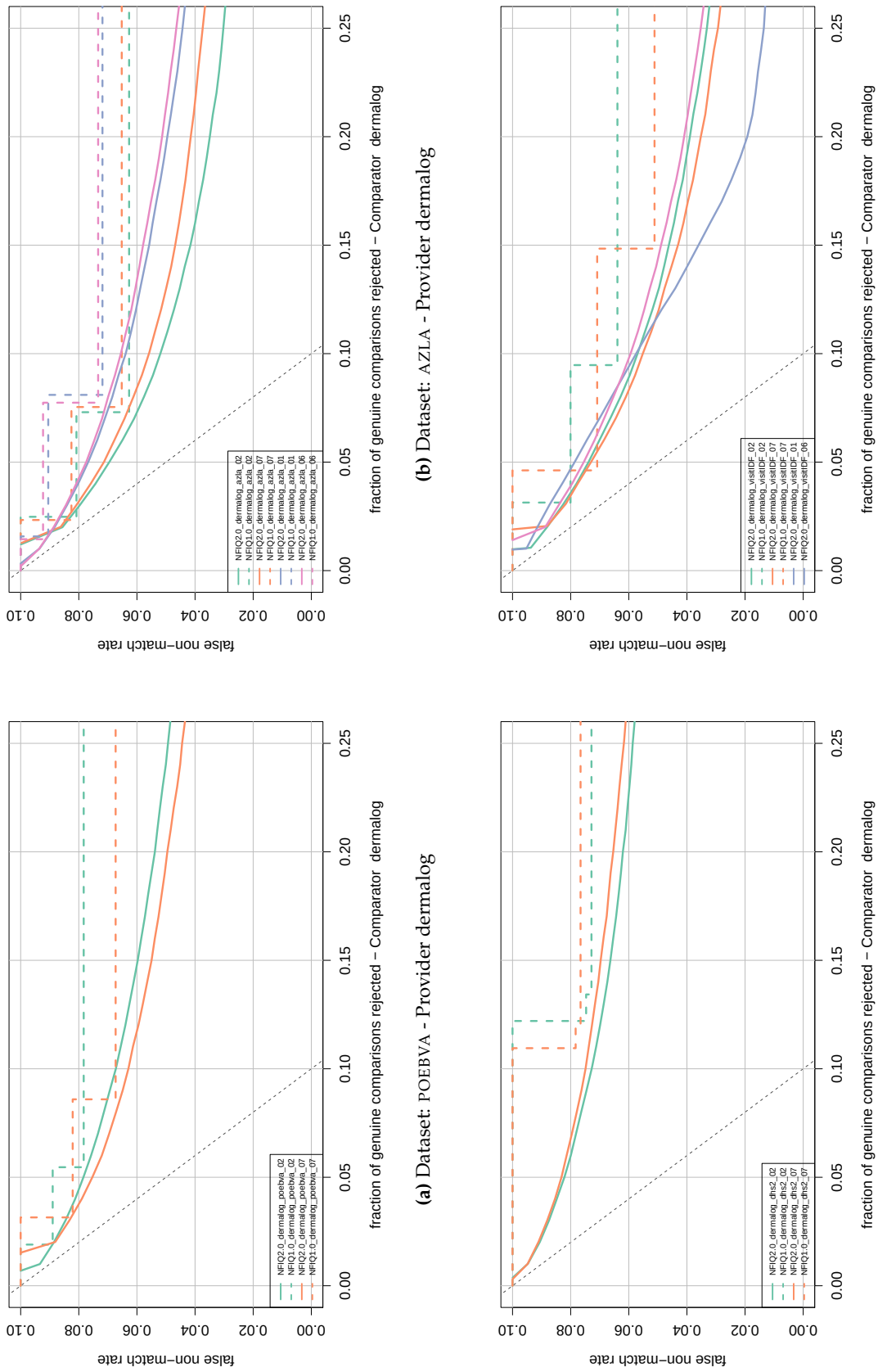
(a) Dataset: POEBVA - Provider dermalog

(b) Dataset: AZLA - Provider dermalog

(c) Provider dermalog

(d) Dataset: VISITIDF - Provider dermalog

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 2.0 is a better predictive of performance than NFIQ 1.0 because its error vs. reject curves are closer to the gray dotted line.



(a) Dataset: POEBVA - Provider dermalog

(b) Dataset: AZLA - Provider dermalog

(c) Provider dermalog

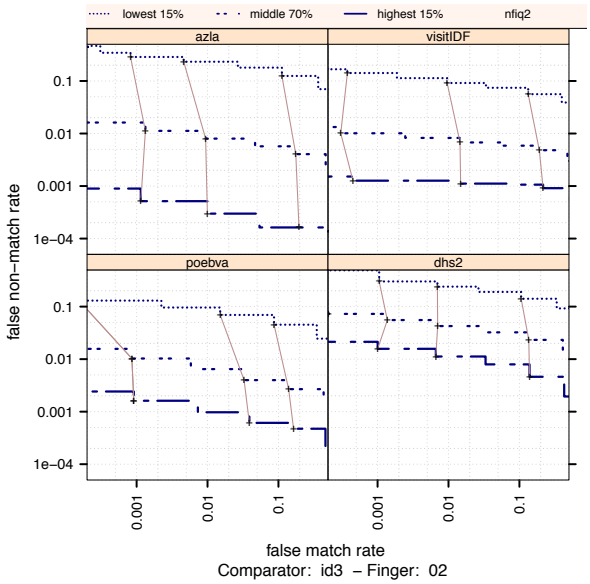
(d) Dataset: VISITIDF - Provider dermalog

Figure 51:
Error vs. reject curve. Provider DERMALOG.The threshold is set to give an initial false non-match rate of 0.1. Same result as in fig 50 but for rejection rate of up to 25%.

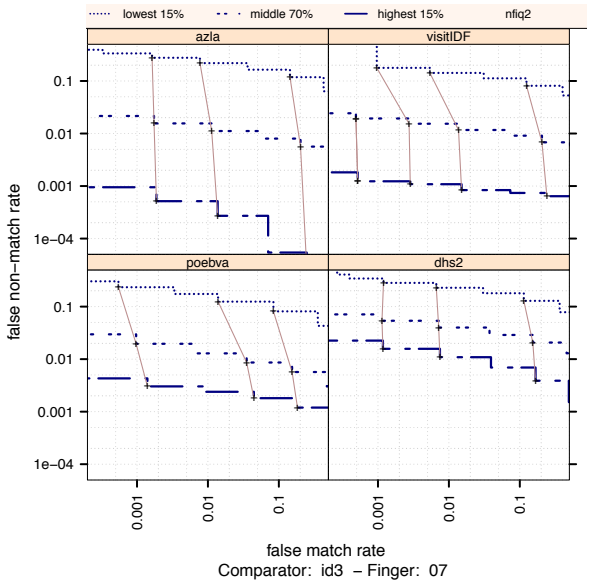
Appendix H

Results for provider ID3

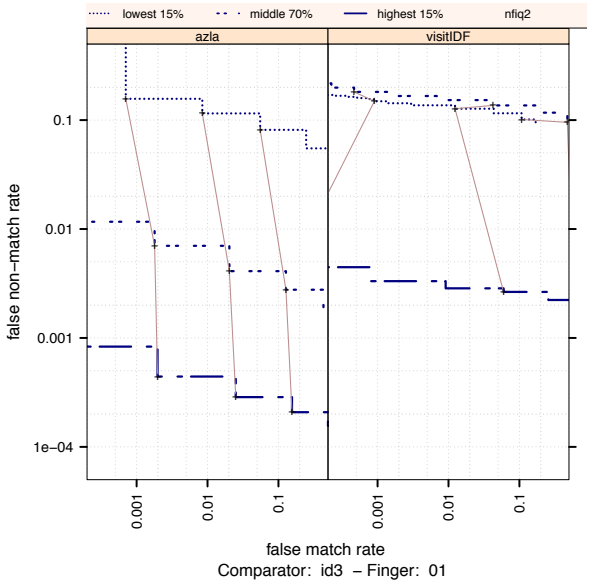
This appendix contains results for provider ID3.



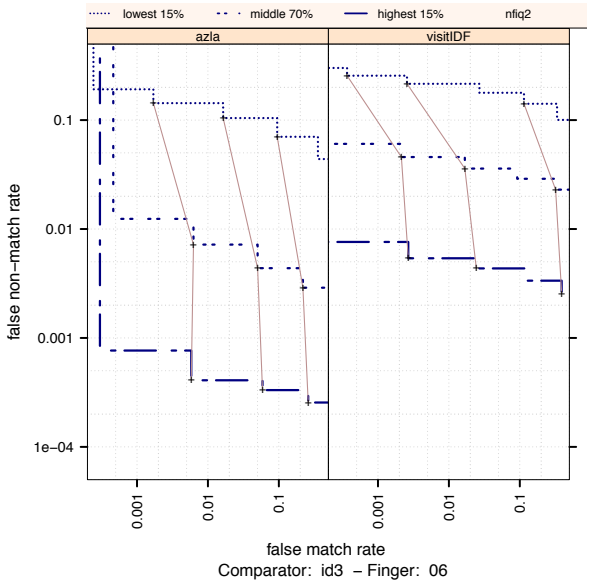
(a) provvider id3. Right Index



(b) Provider id3. Left Index

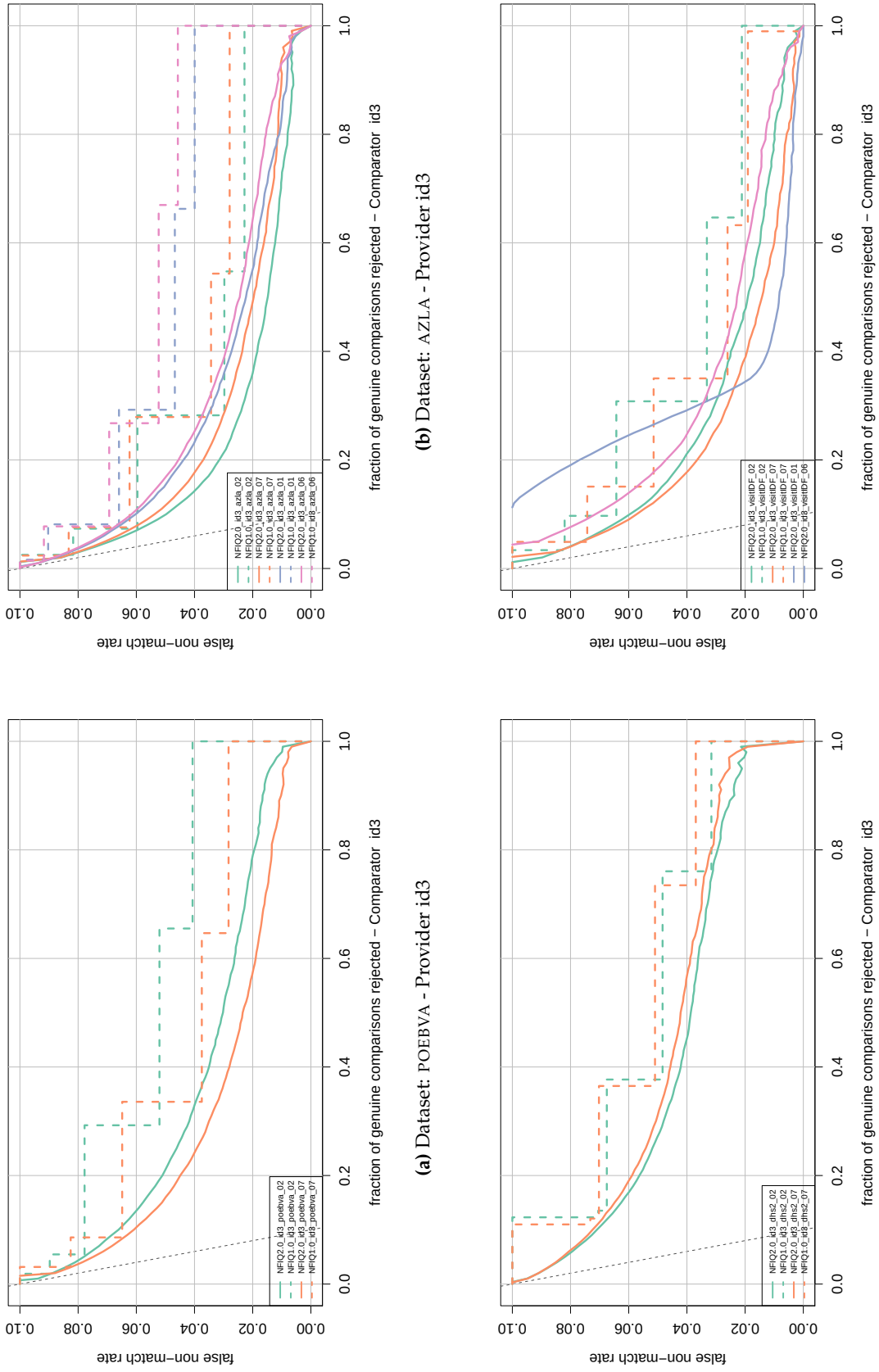


(c) Provider id3. Right Thumb



(d) Provider id3. Left Thumb

Figure 52:
Ranked DET. Provider ID3.The set of all comparisons are partitioned into three groups based on the pair-wise NFIQ2.0 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 2.0 values appearing above, and the DET curve of highest NFIQ 2.0 values below all the other curves.



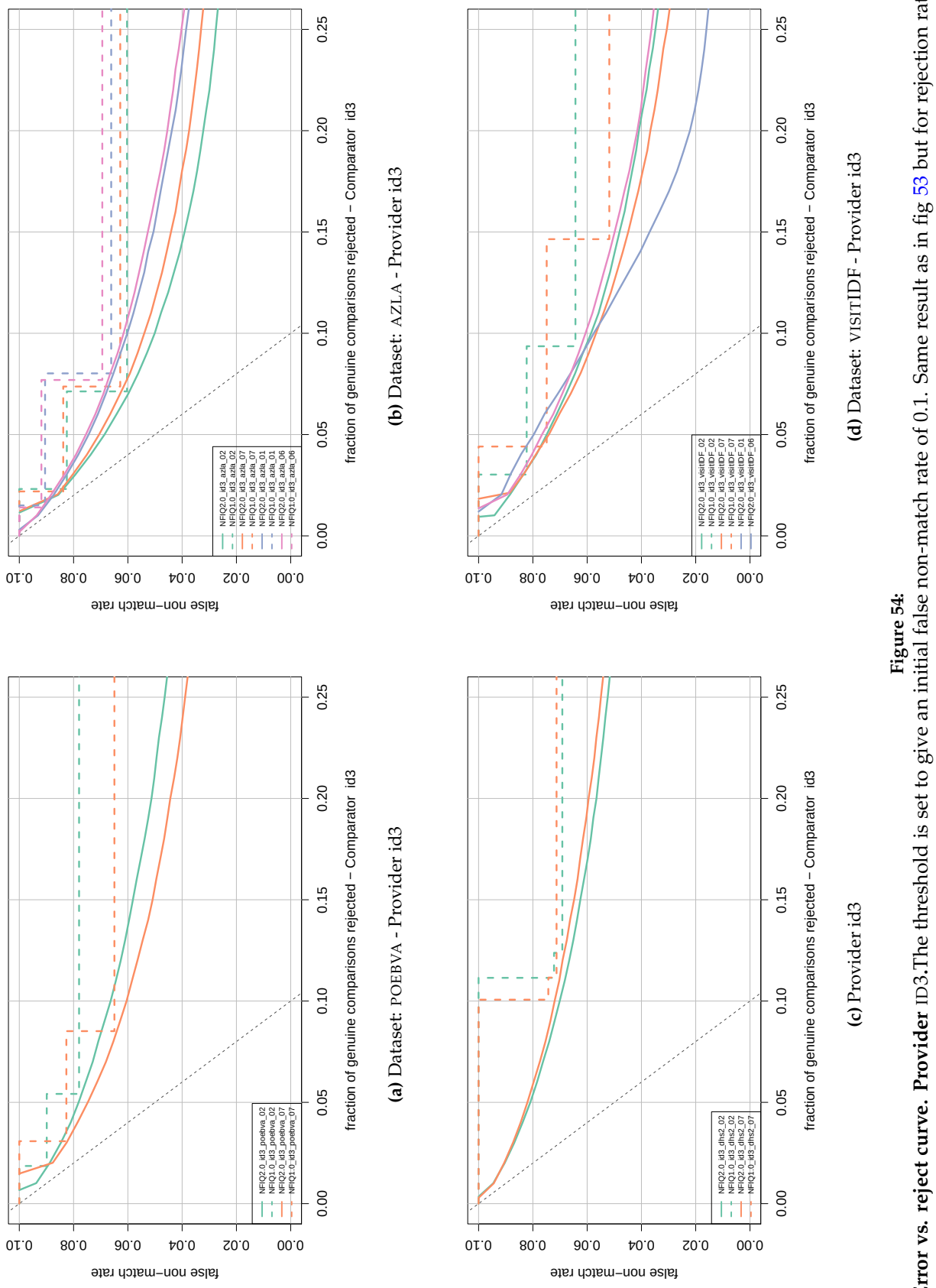
(a) Dataset: POEBVA - Provider id3

(b) Dataset: AZLA - Provider id3

(c) Provider id3

(d) Dataset: VISITIDF - Provider id3

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 2.0 is a better predictive of performance than NFIQ 1.0 because its error vs. reject curves are closer to the gray dotted line.



(a) Dataset: POEBVA - Provider id3

(b) Dataset: AZLA - Provider id3

(c) Provider id3

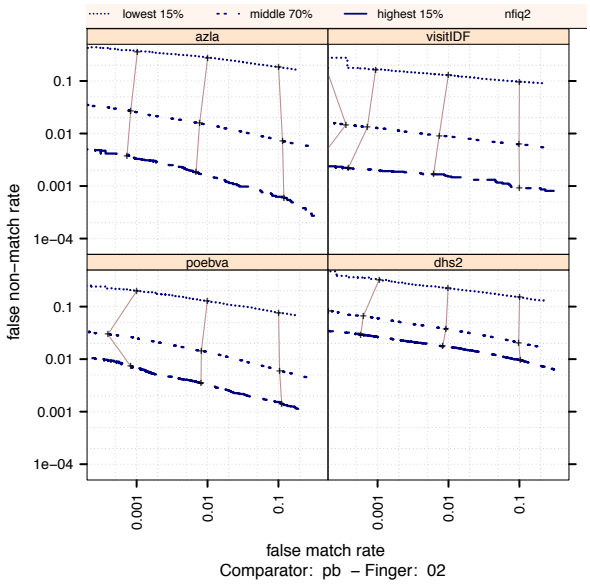
(d) Dataset: VISITIDF - Provider id3

Figure 54:
Error vs. reject curve. Provider ID3.The threshold is set to give an initial false non-match rate of 0.1. Same result as in fig 53 but for rejection rate of up to 25%.

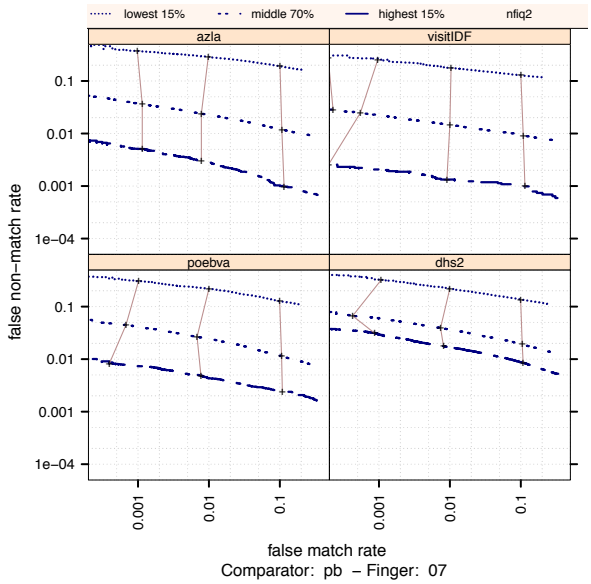
Appendix I

Results for provider PB

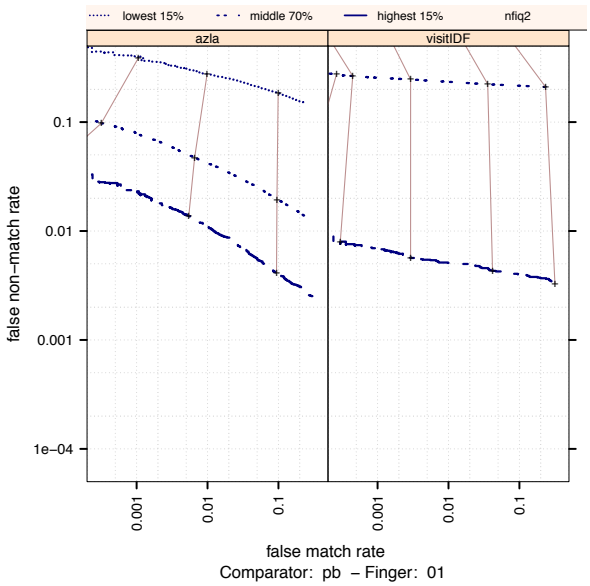
This appendix contains results for provider PB.



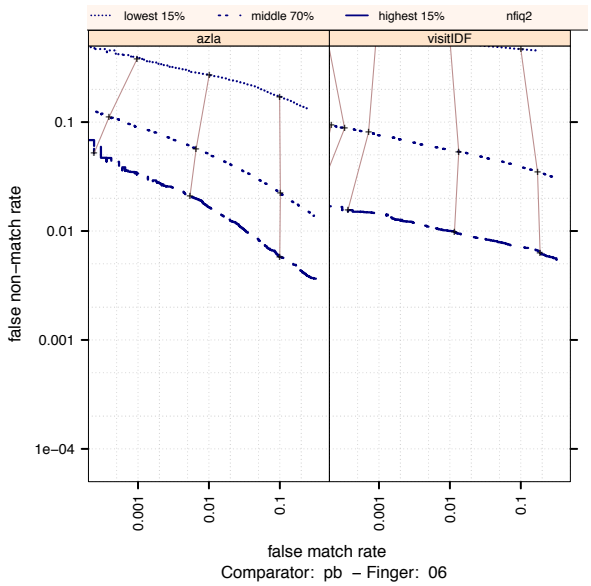
(a) provvider pb. Right Index



(b) Provider pb. Left Index

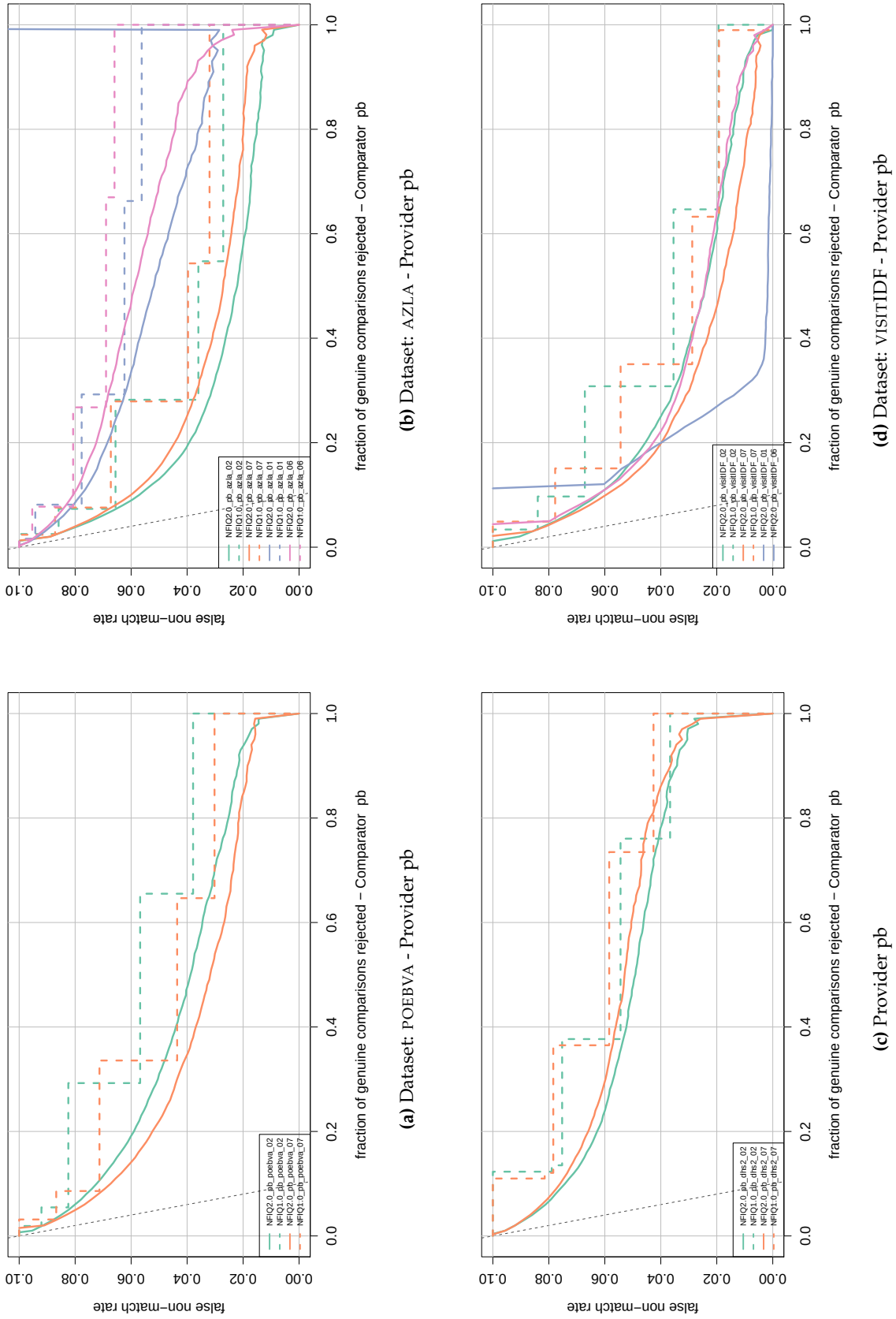


(c) Provider pb. Right Thumb



(d) 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 NFIQ2.0 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 2.0 values appearing above, and the DET curve of highest NFIQ 2.0 values below all the other curves.



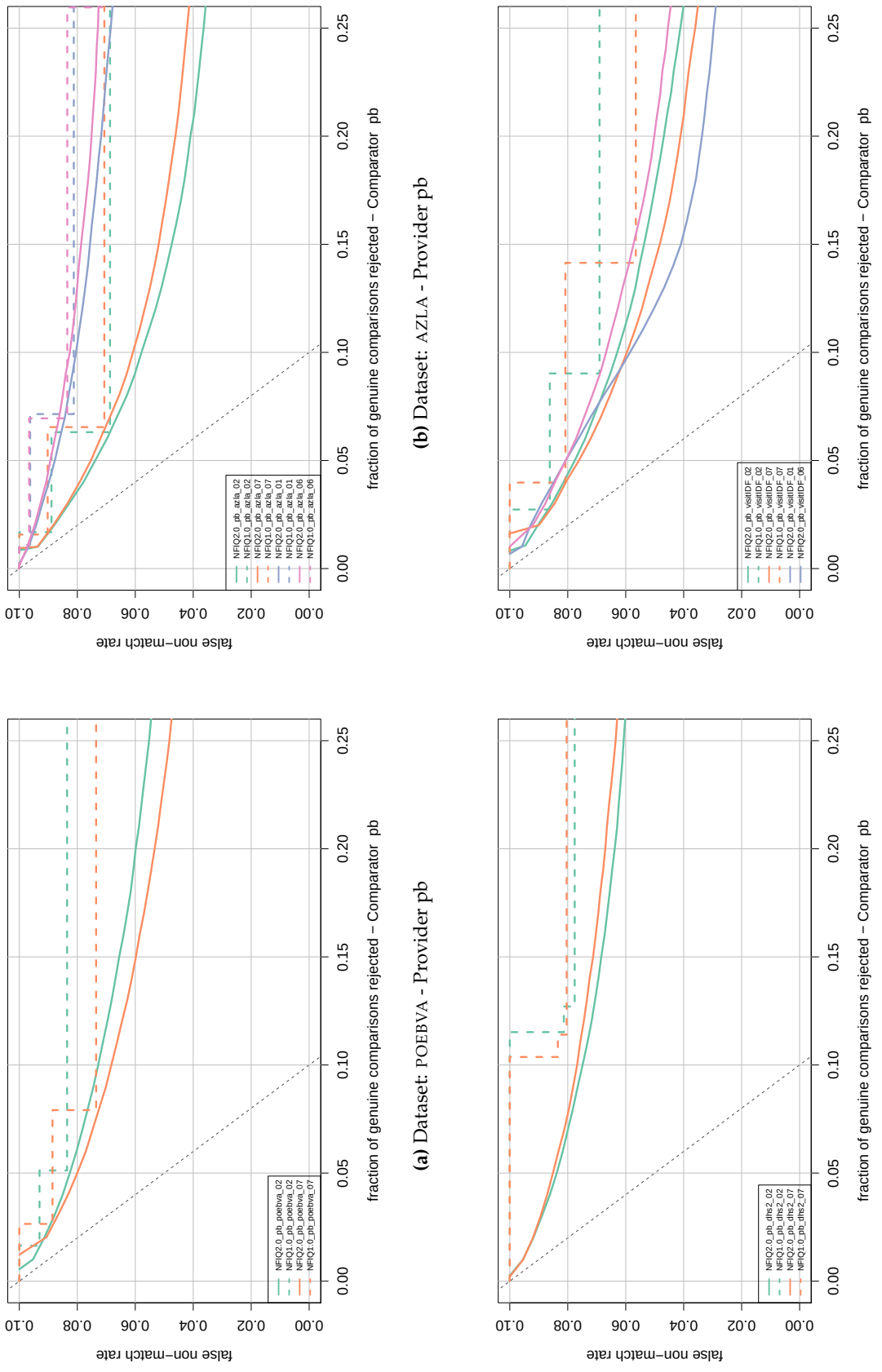
(a) Dataset: POEBVA - Provider pb

(b) Dataset: AZLA - Provider pb

(c) Provider pb

(d) Dataset: VISITIDF - Provider pb

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 2.0 is a better predictive of performance than NFIQ 1.0 because its error vs. reject curves are closer to the gray dotted line.



(a) Dataset: POEBVA - Provider pb

(b) Dataset: AZLA - Provider pb

(c) Provider pb

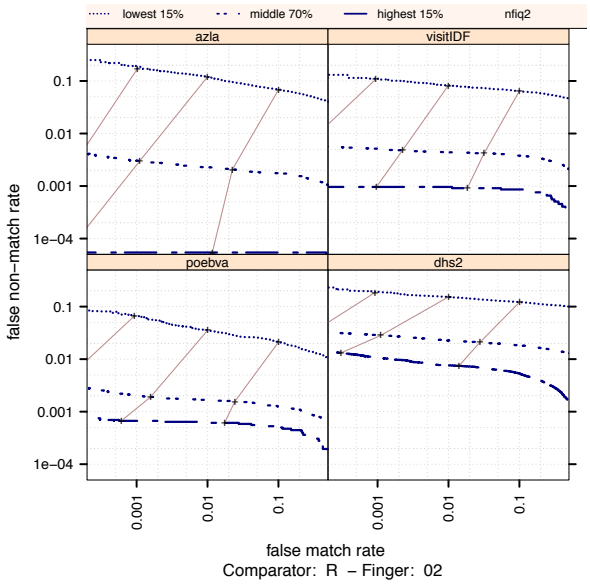
(d) Dataset: VISITIDF - Provider pb

Figure 57:
Error vs. reject curve. Provider PB.The threshold is set to give an initial false non-match rate of 0.1. Same result as in fig 56 but for rejection rate of up to 25%.

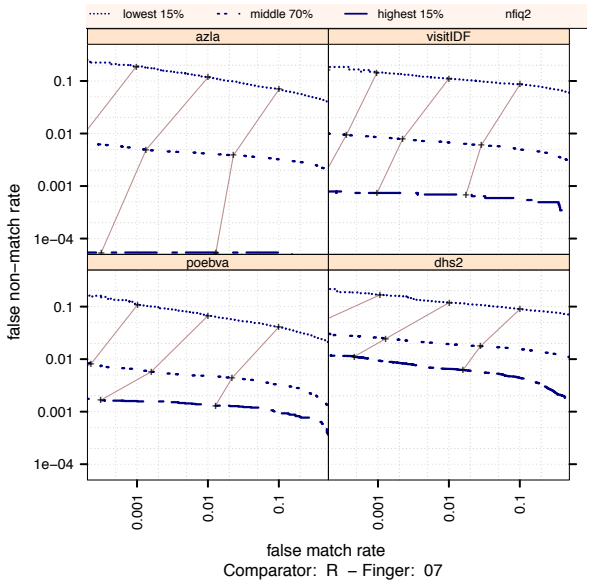
Appendix J

Results for provider R

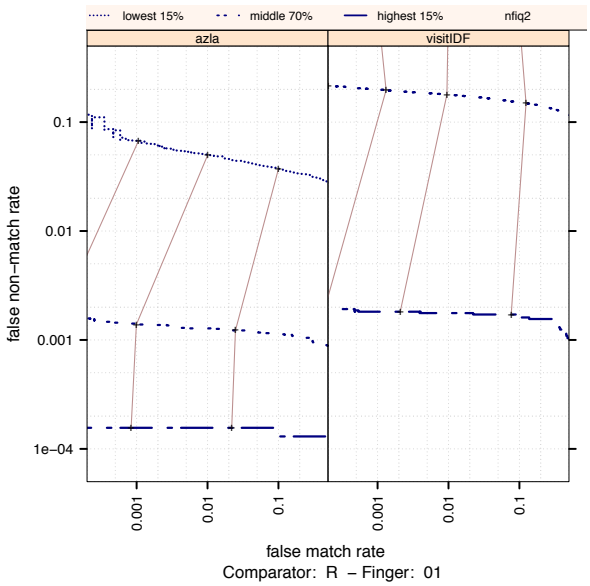
This appendix contains results for provider R.



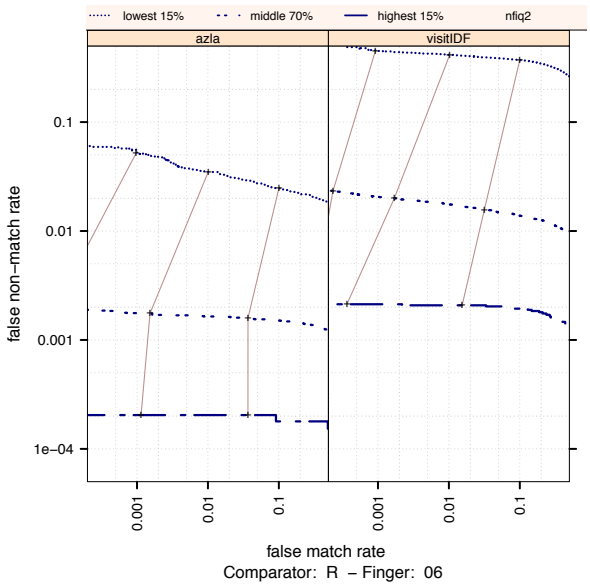
(a) provvider R. Right Index



(b) Provider R. Left Index

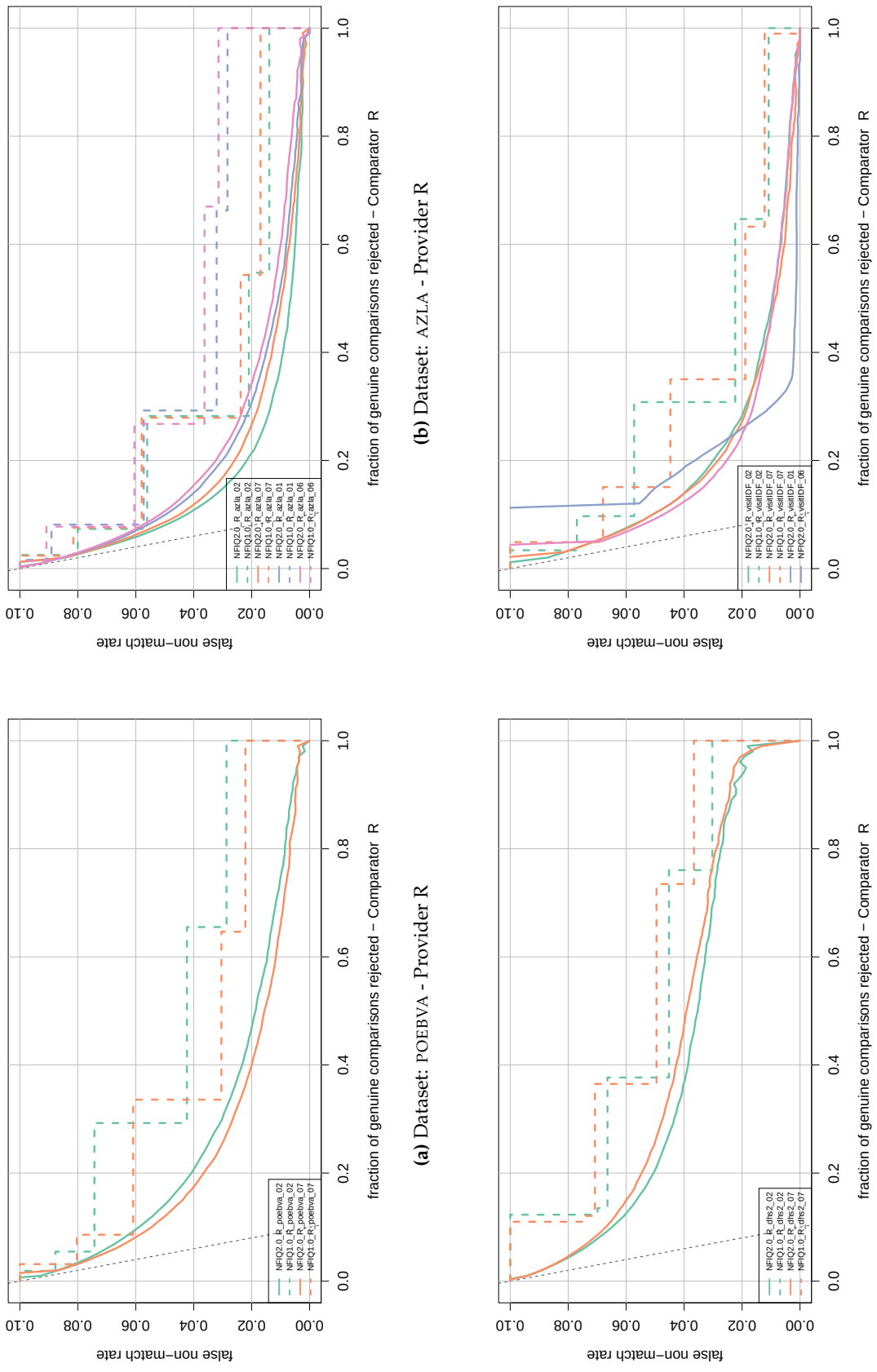


(c) Provider R. Right Thumb



(d) 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 NFIQ2.0 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 2.0 values appearing above, and the DET curve of highest NFIQ 2.0 values below all the other curves.



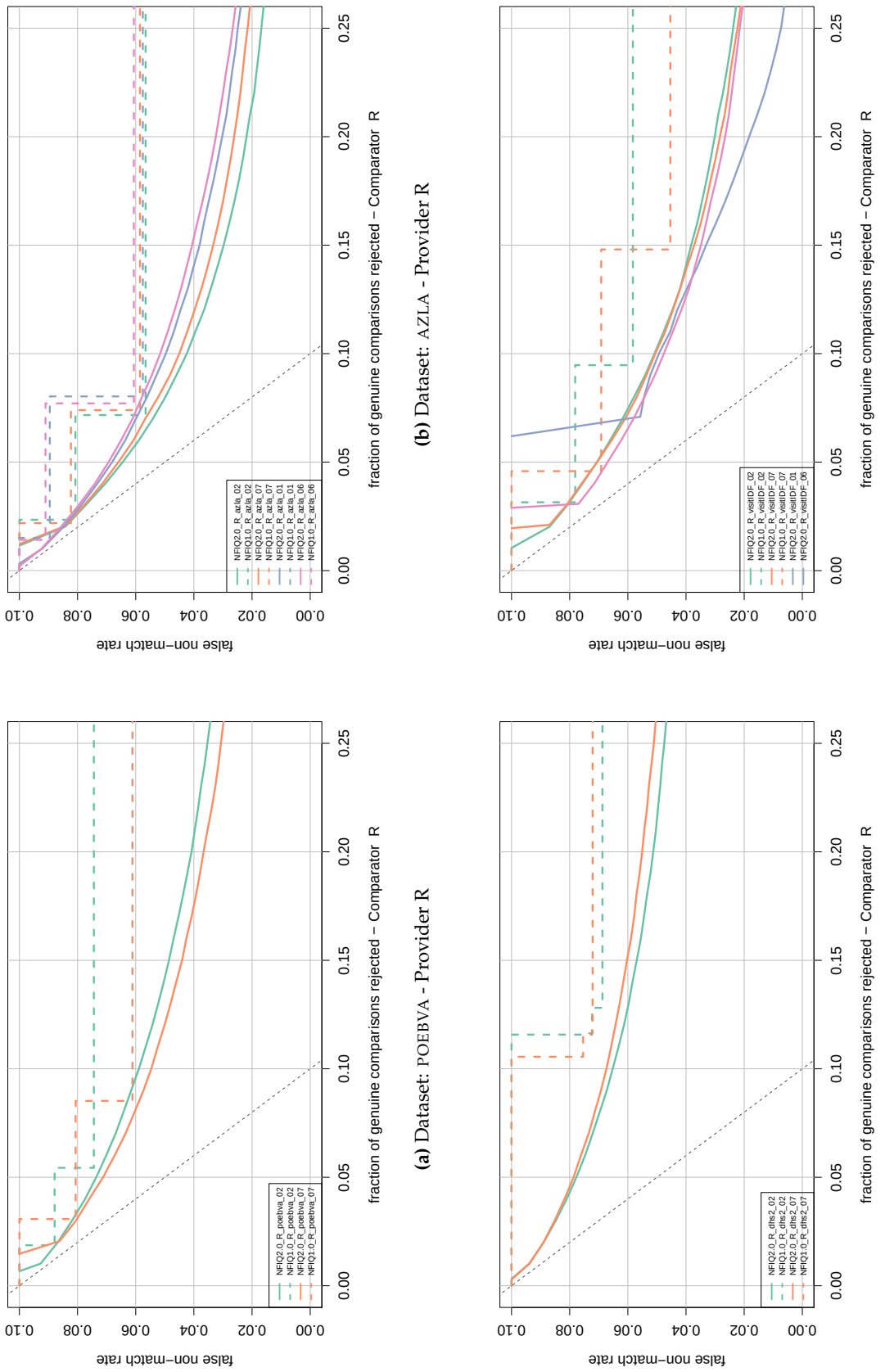
(a) Dataset: POEBVA - Provider R

(b) Dataset: AZLA - Provider R

(c) Provider R

(d) Dataset: VISITIDF - Provider R

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 2.0 is a better predictive of performance than NFIQ 1.0 because its error vs. reject curves are closer to the gray dotted line.



(a) Dataset: POEBVA - Provider R

(b) Dataset: AZLA - Provider R

(c) Provider R

(d) Dataset: VISITIDF - Provider R

Figure 60:
Error vs. reject curve. Provider R.The threshold is set to give an initial false non-match rate of 0.1. TSame result as in fig 59 but for rejection rate of up to 25%.

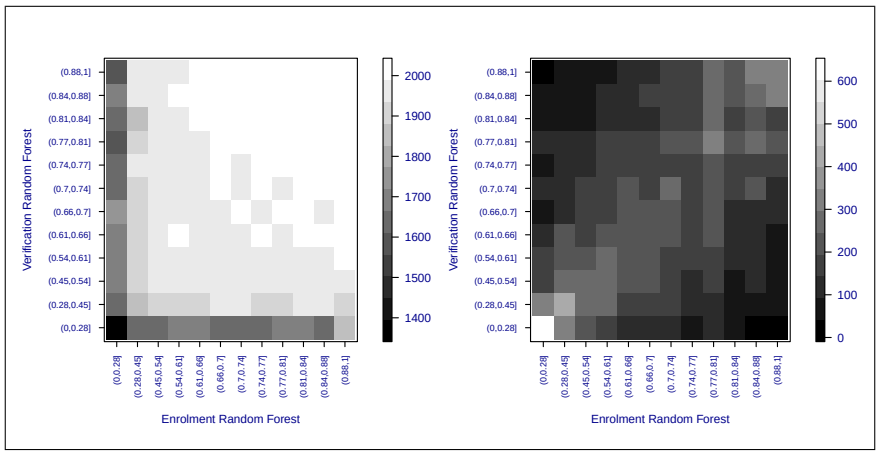


Figure 61: Heatmap for provider R on dataset azla: Left plot shows the mean comparison score, right plot shows the sample count.

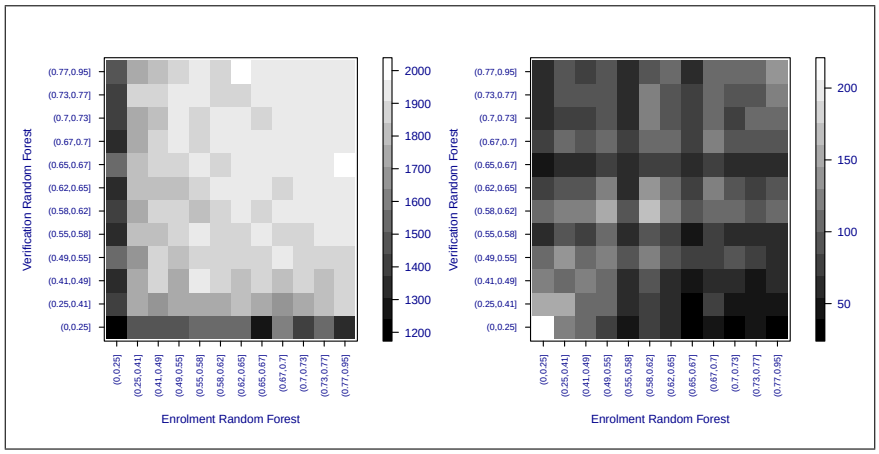


Figure 62: Heatmap for provider R on dataset dhs2: Left plot shows the mean comparison score, right plot shows the sample count.

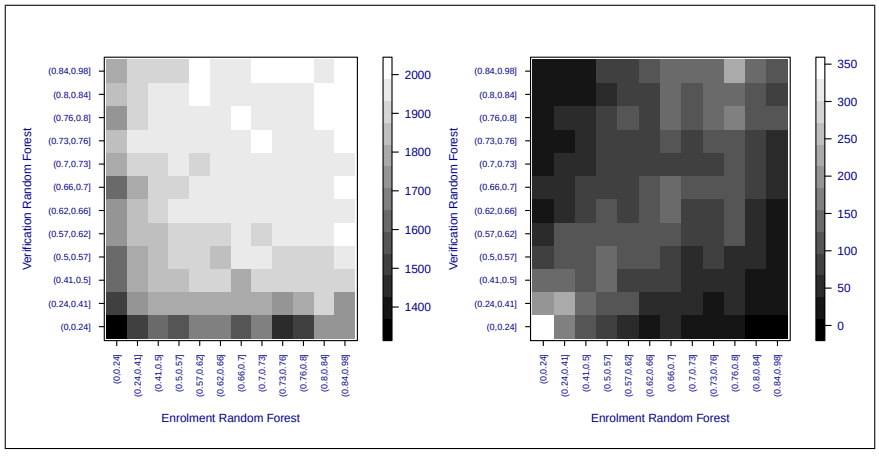


Figure 63: Heatmap for provider R on dataset poebva: Left plot shows the mean comparison score, right plot shows the sample count.

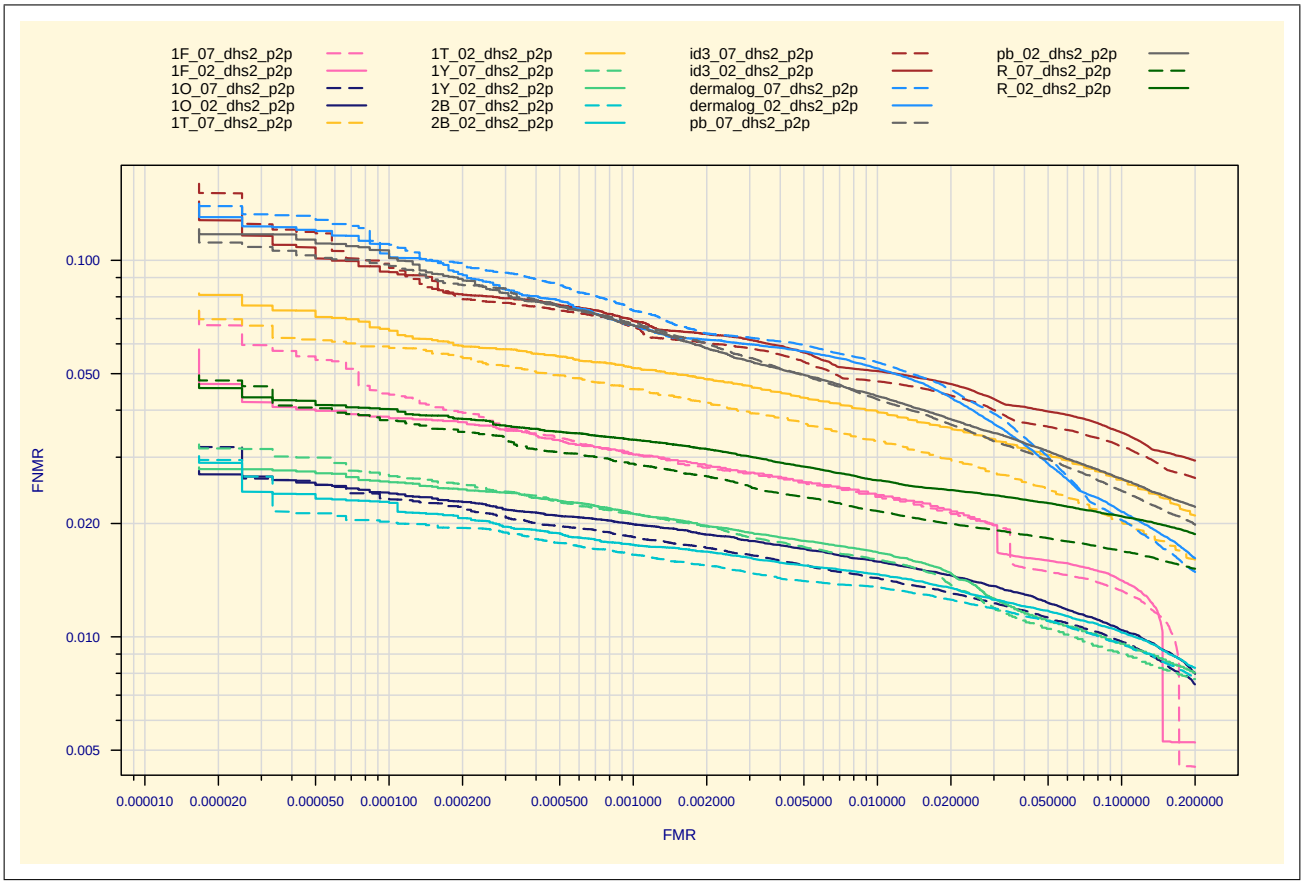


Figure 64: Detection Error Tradeoff (DET) curves of NFIQ 2.0 submissions on DHS2 images.

Appendix K

Results for NFIQ 2.0 participants

Each figure shows detection error tradeoffs of the nine NFIQ 2.0 participants for each of the four datasets used for training.

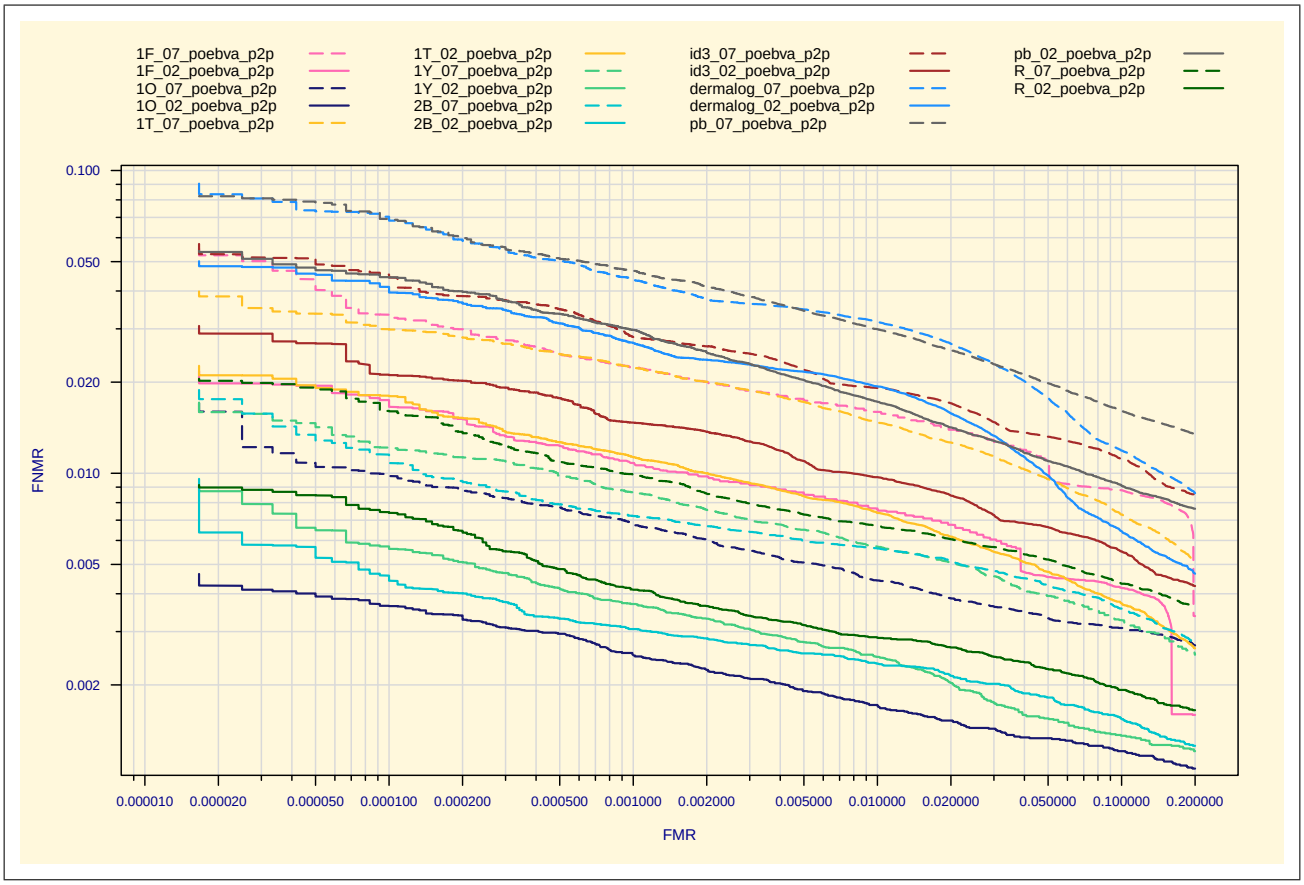


Figure 65: Detection Error Tradeoff (DET) curves of NFIQ 2.0 submissions on POEBVA images.

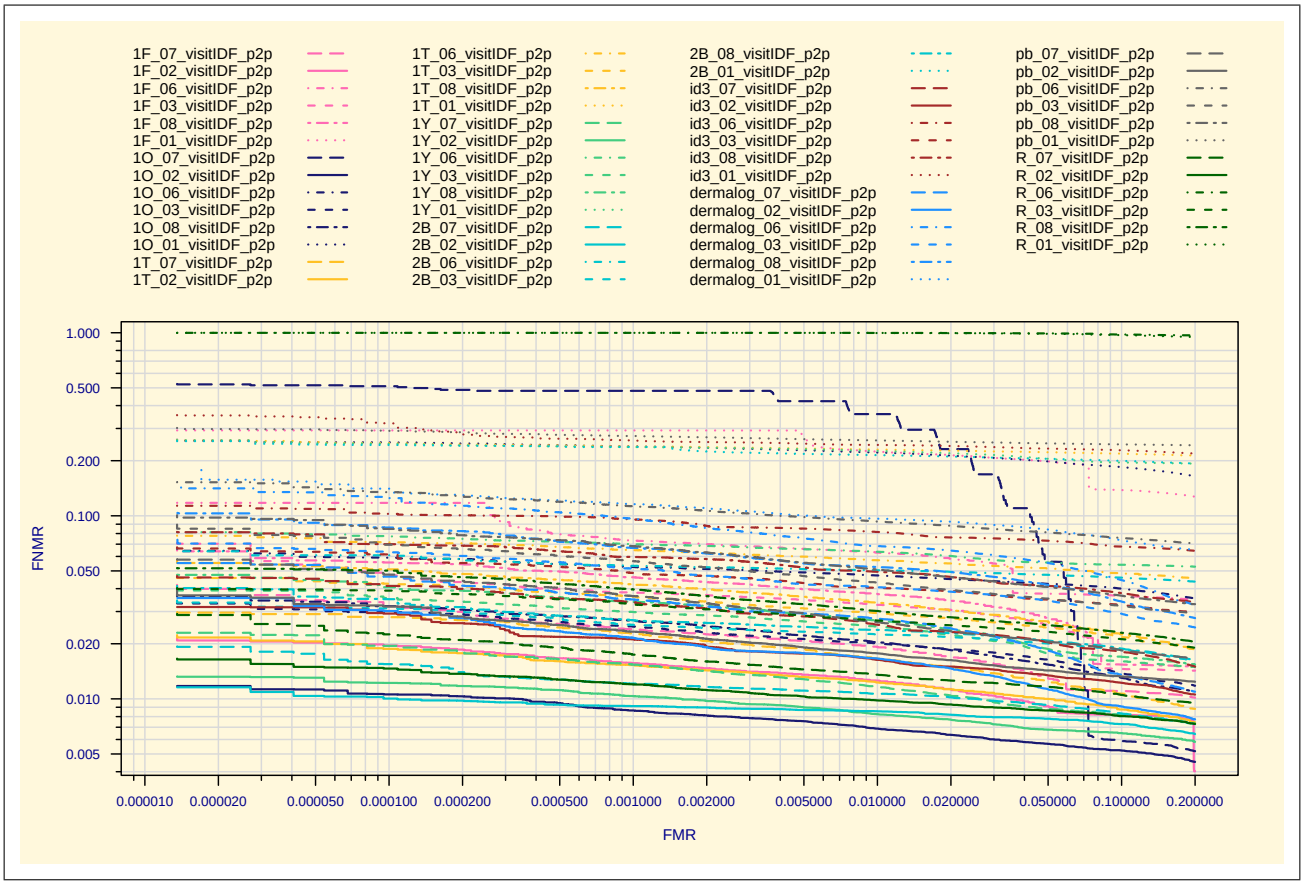


Figure 66: Detection Error Tradeoff (DET) curves of NFIQ 2.0 submissions on VISIT images.

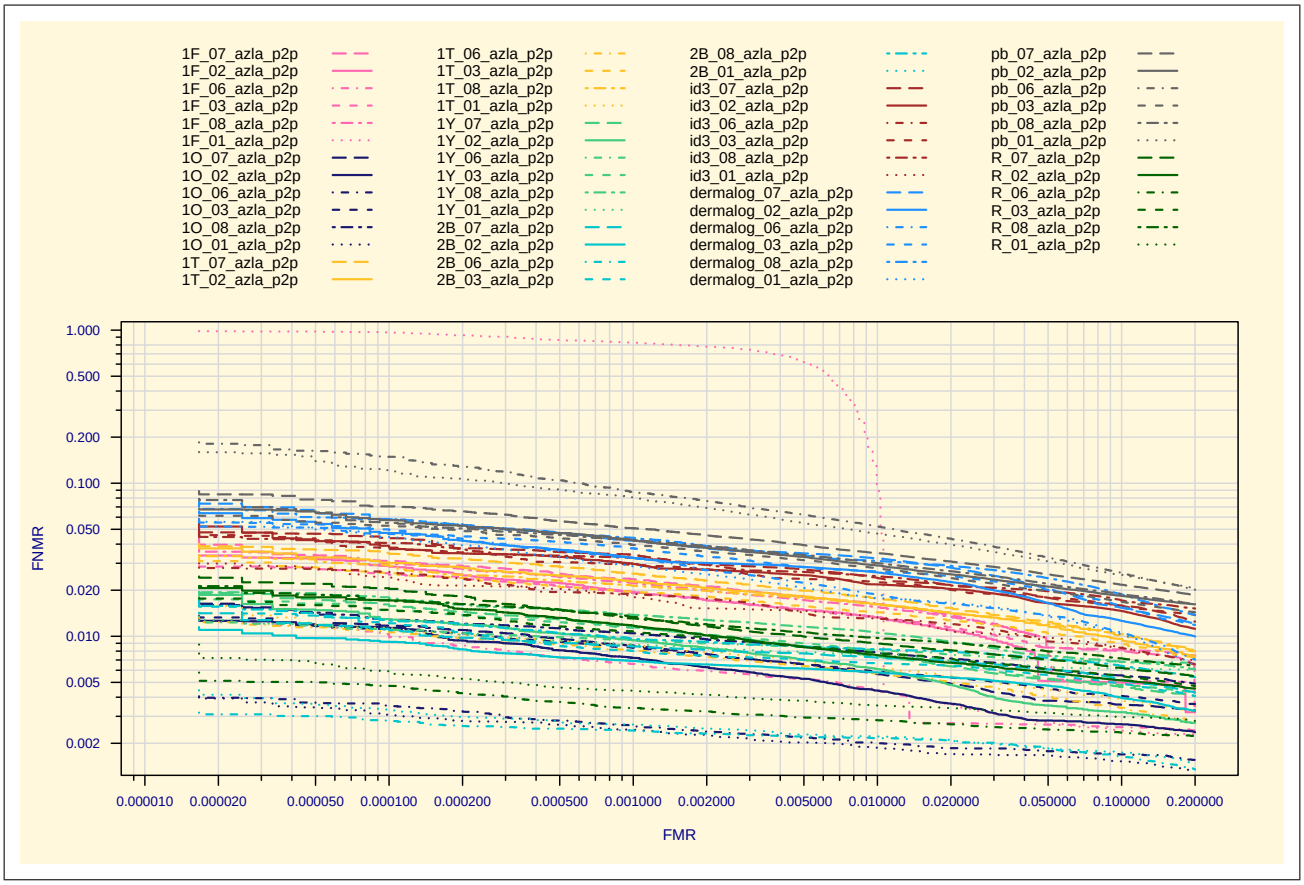


Figure 67: Detection Error Tradeoff (DET) curves of NFIQ 2.0 submissions on AZLA images.

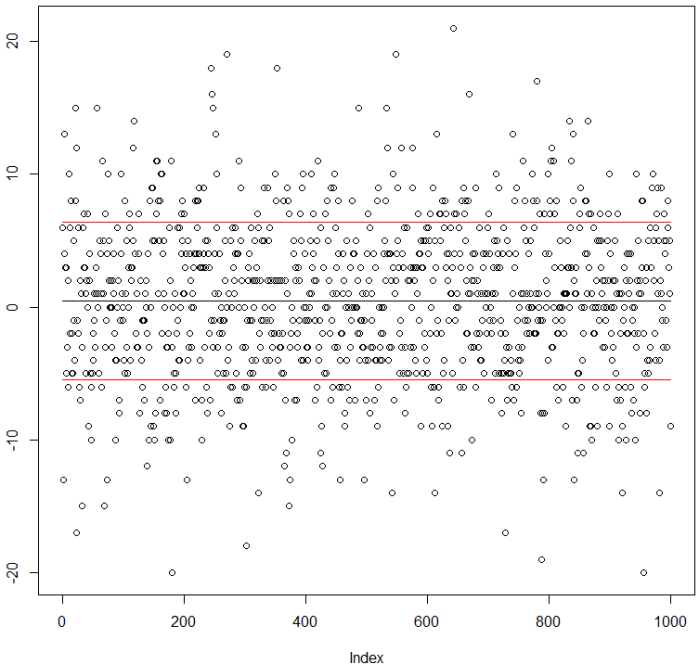


Figure 68: Deviations of NFIQ2 values between uncompressed and wsq-compressed (factor 4) images of 1000 fingerprints from the MCYT database.

Appendix L

Impact of WSQ Compression on the NFIQ 2.0 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) im-

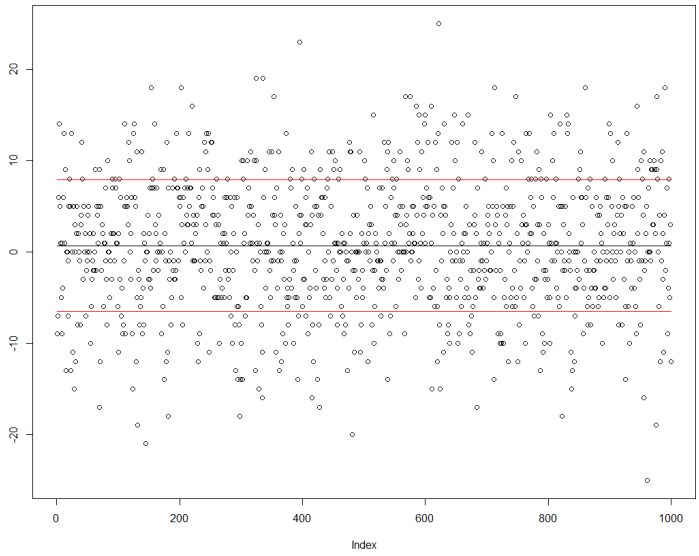


Figure 69: Deviations of NFIQ2 values between uncompressed and wsq-compressed (factor 8) images of 1000 fingerprints from the MCYT database.

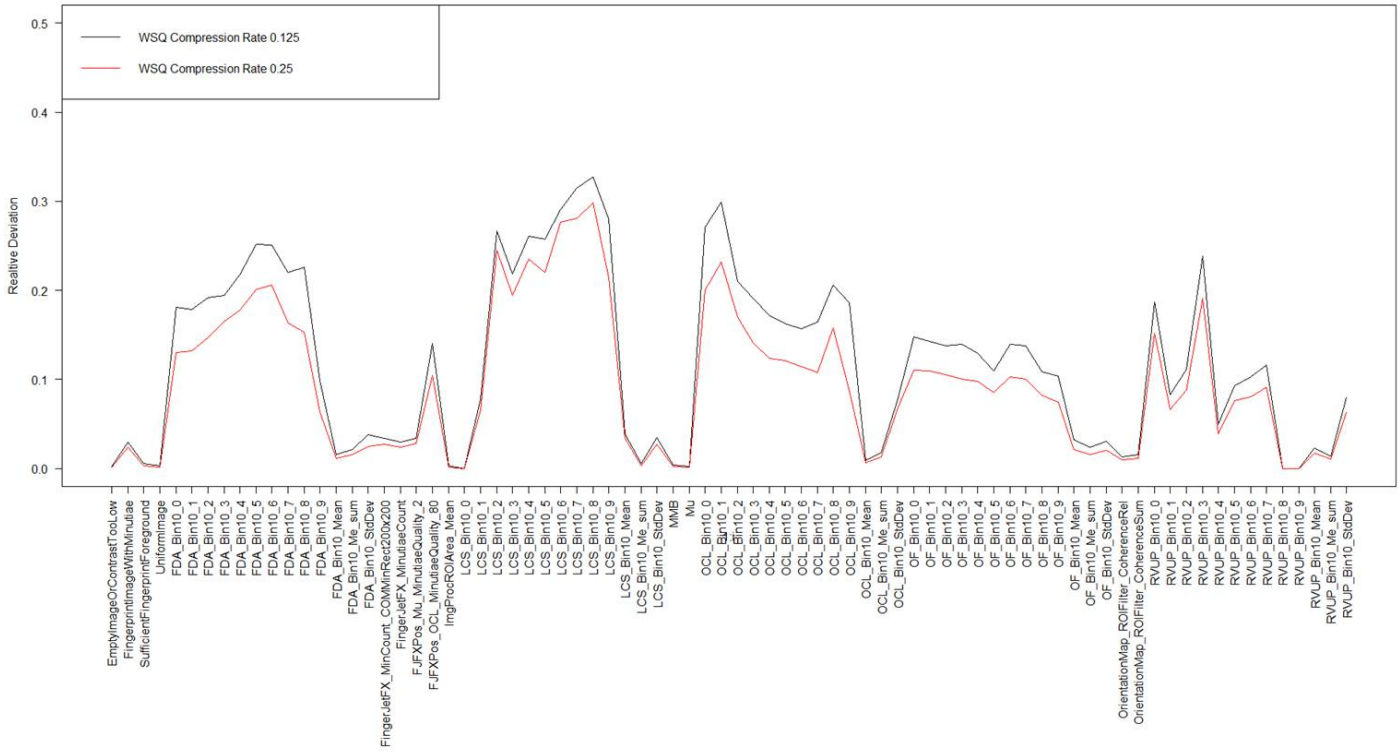


Figure 70: Average relative absolute deviations ($|x_1 - x_2|/|x_1 + x_2|$) in the values of the NFIQ2 quality features values between uncompressed and wsq-compressed images of 1000 fingerprints from the MCYT database.

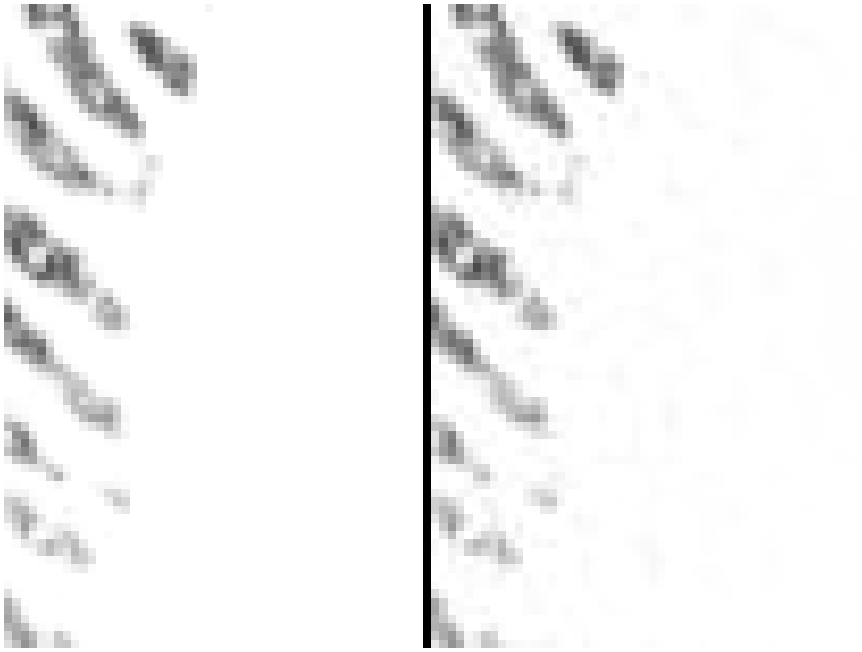


Figure 71: Artifacts at the boundary of a fingerprint resulting from WSQ compression (factor 8).

pression 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.

If 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.

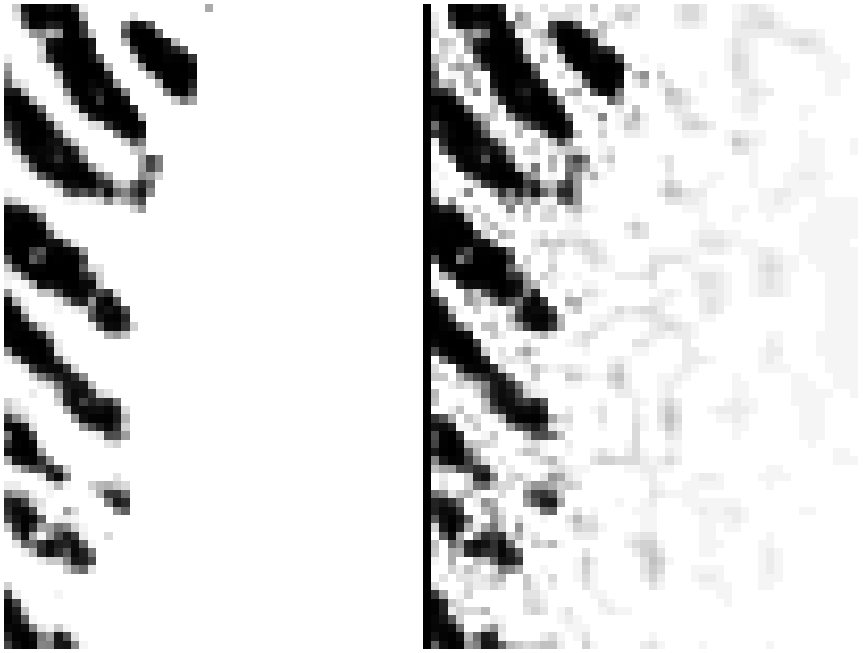


Figure 72: Artifacts at the boundary of a fingerprint resulting from WSQ compression (factor 8) after setting gamma value to 0.1.

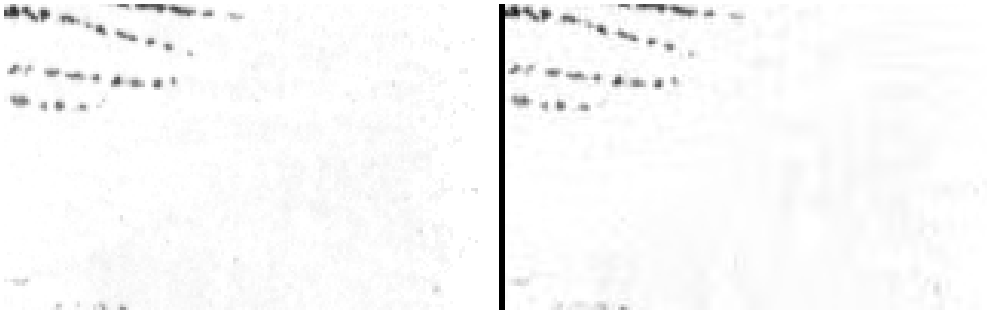


Figure 73: Effect of WSQ compression (factor 8) on artifacts in fingerprint image.

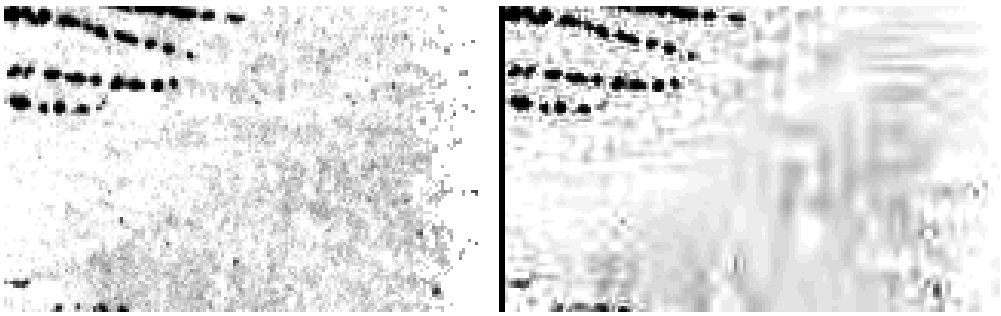


Figure 74: Effect of WSQ compression (factor 8) on artifacts in fingerprint image after setting gamma value to 0.1.

Appendix M

Software Releases

M.1 Operational Kit

The NFIQ 2.0 Operational Kit is the open-source software release of NFIQ 2.0 for operational use. It contains the actual released software source of NFIQ 2.0 – including all finally selected features and training data for NFIQ 2.0.

Furthermore, it consists of a command line tool that provides access to the NFIQ 2.0 algorithm easily. In the single program mode, the command line tool of the Operational Kit allows to compute the NFIQ 2.0 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 2.0 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 2.0 User Manual.

M.2 Research Kit

The NFIQ 2.0 Research Kit is based on a so called NFIQ 2.0 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

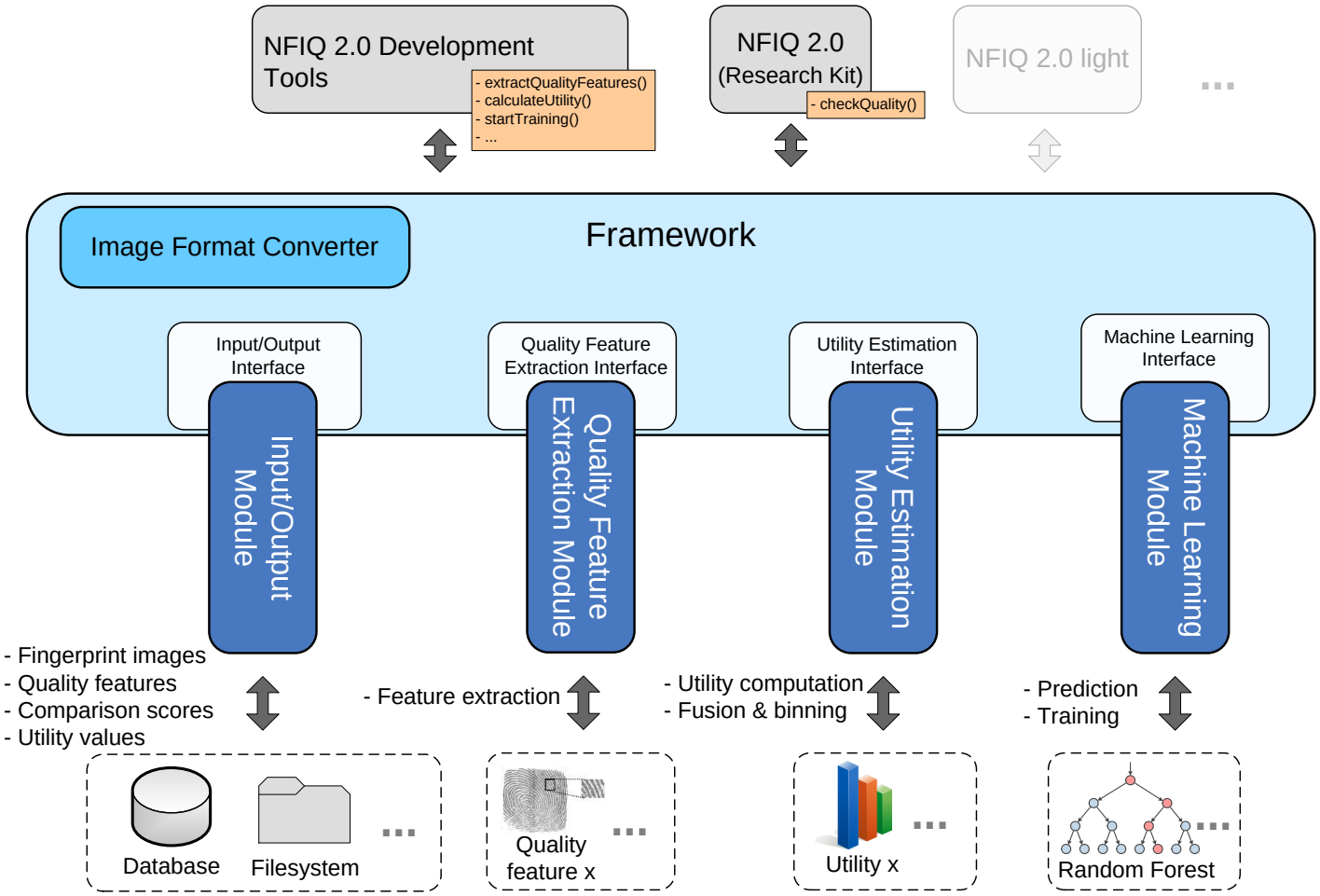


Figure 75: Architecture of the NFIQ 2.0 Framework used by the NFIQ 2.0 Research Kit.

as well as the interface functions, error handling, data types and structures are described in detail in the NFIQ 2.0 User Manual.