Fingerprint Image Quality

Elham Tabassi
Charles L. Wilson
Craig I. Watson

NISTIR 7151
August 2004

NIST
National Institute of
Standards and Technology

AUTHENTICATED
U.S. GOVERNMENT
INFORMATION
GPO

ABSTRACT **3**

1 INTRODUCTION **3**

1.1 TERMINOLOGY 4
1.2 DATA SET AND MATCHER ALGORITHM 4

2 PREVIOUS WORK **6**

3 DEFINING FINGERPRINT IMAGE QUALITY **7**

3.1 IMAGE QUALITY AS CLASSIFICATION 11

4 MEASURING FINGERPRINT IMAGE QUALITY **12**

4.1 FEATURE EXTRACTION 13
4.1.1 MINDTCT input *13*
4.1.2 Generate image quality map *13*
4.1.4 Feature vectors *16*
4.2 PATTERN RECOGNITION (REGRESSION VS. CLASSIFICATION) 28
4.2.1 Training the neural network *34*

5 ASSESSING FINGERPRINT IMAGE QUALITY **37**

6 CONCLUSION **69**

REFERENCES **71**

APPENDIX A **72**

APPENDIX B **72**

Abstract

In this report, we propose a new definition of quality of fingerprint impressions and present detailed algorithms to measure image quality for fingerprints. We define fingerprint image quality as a predictor of matcher performance before a matcher algorithm is applied. This means presenting the matcher with *good quality* fingerprint images will result in high matcher performance, and vice versa, the matcher will perform poorly for poor quality fingerprints. We also have carried out an objective evaluation of the quality assessment of fingerprint images. Our quality measure is implemented in the C programming language and has been tested on 20 different live scan and paper fingerprints datasets collected in different operational settings. Our implementation is publicly, but export controlled, available as part of NIST's fingerprint software distribution.

1 Introduction

Studies carried out by NIST as part of its Patriot Act mandate have demonstrated that all fingerprint systems tested have widely varying performance which depends, among other factors, on fingerprint image quality [21, 22, 5, 6]. In this paper we will define a novel measure of fingerprint image quality, which can be used to estimate fingerprint match performance. We will present test results for this image quality measure for 20 different live scan and scanned paper fingerprint data sets. These datasets are explained in section 1.2. The ability to predict matcher performance has been evaluated for 14 different fingerprint systems supplied by 8 commercial fingerprint vendors. An explanation of the NIST Fingerprint SDK test plus more detailed explanation of datasets are in [6].

An ideally sensed or scanned fingerprint image has clear and distinct ridges and valleys. An automatic fingerprint recognition system can perform well on such fingerprint images. However, precise fingerprint acquisition has some peculiar and challenging aspects [1]. Often skin condition or imperfect acquisitions cause the captured fingerprint image to be far from ideal. Unclean sensor plates, non-uniform and inconsistent contact [2] can result in poor samples and feature extraction artifacts during image processing and hence increase false accept/reject rate.

An automatic fingerprint recognition system has traditionally consisted of three subsystems: fingerprint image acquisition, feature extraction, and matching. In image acquisition, a digital image of a fingerprint is captured either from the live scan of a person's finger or from an inked impression of a person's finger on paper (e.g. fingerprint cards). Feature extraction is the process of representing the captured image in some space to facilitate matching. Matching involves computing the likelihood of the fingerprint coming from subjects (persons) in the database. The performance of the whole system depends on how well each subset behaves.

It is desirable to assess "quality" of a fingerprint before any matching process. This allows poor image acquisition to be corrected before poor quality is entered into users' databases. Higher matcher performance can be achieved if a fingerprint's quality (in the sense that will be defined later in this paper) is sufficiently good and overall database integrity is improved.

This motivates us to define fingerprint image quality as a predictor of a matcher's performance**.** This means presenting the matcher with *good quality* fingerprint images will result in high matcher performance, and vice versa, the matcher will perform poorly for poor quality fingerprints. Despite some on-going and past effort in the investigation of fingerprint image quality [2,11,12,13,14,16], to our knowledge, nobody has publicly defined fingerprint image quality as a scalar to predict matcher performance. Predicting matcher performance is also valuable for biometric fusion of multiple fingerprints because the fingerprints with the best image quality can be assigned higher weight in the fusion.

The rest of this paper is organized as follows: we briefly discuss the framework and datasets used for experiments. We briefly survey some on-going and past effort in fingerprint image quality research in section 2. Section 3 defines fingerprint image quality in a broad general sense without addressing implementation details. Although it was defined and implemented for fingerprints, we believe this definition of quality can and should be applied for other biometric modalities (e.g. face, iris) if proper feature vectors are defined and measured. Section 4 covers our implementation for measuring fingerprint quality in detail. Section 5 discusses how we test our quality system. Section 6 summarizes our findings. Appendix A shows the image quality predictions for this method over all 20 data sets and all 14 fingerprint-matching systems.

1.1 Terminology

A biometric sample is an abstraction of the material input to a recognition system: it may be an image (fingerprint, face, iris), speech sequence(s) or some arbitrary composite. We define a *gallery* Γ, to represent the set of enrollees in a biometric system. It contains identically one biometric sample per subject. We use $G = |\Gamma|$ to denote the size of the enrolled population. Likewise a *probe set*, Π_Γ, serves as the set of legitimate users of the system. A *match* describes the comparison of a probe and a gallery sample of the same individual. A *non-match* arises from samples of different persons.

1.2 Data set and matcher algorithm

NIST has acquired a collection of live-scan and scanned paper fingerprints datasets collected at different operational settings.

Department of State (DOS-C) and Department of Homeland Security (DHS2-C) datasets contain live-scan plain impressions of the left and right index fingers. They were captured in an operational environment and should give results of what to expect from real time data of plain-to-plain (p2p) impression matching.

Department of Homeland Security Ten-print (DHS10) and Texas Department of Public Safety (TXDPS) were mainly inked paper segmented plain and rolled impressions and Department of Homeland Security Benefits (BEN) was mainly live-scan segmented plain and rolled impressions. These three datasets are most useful in evaluating plain to rolled (p2r) impression

matching. DHS10 and TXDPS are examples of the currently available fingerprint databases and BEN represents data captured with newer live-scan methods with some control on quality.

US-VISIT Point of Entry and Bio-Visa Application data (VISIT_POE, VISIT_POE_BVA) contain live-scan plain impressions of the left and right index fingers. They are captured in an operational environment and represent the current image data being captured in the US-VISIT program. The BVA data was captured at State Department Consular offices when applicants apply for visas and the POE are captured by the Department of Homeland Security at the points of entry into the US for international travelers. VISIT_POE compares two instances of a person that were captured at the POE and VISIT_POE_BVA compares a POE image to a BVA image.

A random sample of 6,000 subjects was selected from each of the datasets. Since DHS2-C was consolidated after testing had started, the final dataset size after consolidation was only 5,888 not 6,000. The process of consolidation has been explained in detail in [22]. The four-finger slaps in BEN, DHS10, and TXDPS were all segmented with the NIST segmentor as discussed in [10].

NIST has conducted testing of one-to-one SDK (Software Development Kit) based COTS fingerprint matching systems to evaluate the accuracy of one-to-one matching [6]. Eight commercial fingerprint vendors submitted their systems for evaluation, some more than one system. A total of fourteen SDK based matching application were tested on twenty different single finger datasets of varying difficulty. We have used similarity scores of SDK vendors to develop and test our image quality assessment system. In this report, we refer to SDK vendors by their SDK abbreviated letters. Table 1 shows list of vendor names associated with SDK letters.

SDK LETTER	VENDOR NAME
A	Name Not Released
B	Name Not Released
C	NEC
D	Cogent Systems, Inc.
E	Name Not Released
F	Cogent Systems, Inc.
G	SAGEM Morpho, Inc.
H	NEC
I	Cogent Systems, Inc.
J	SAGEM Morpho, Inc
K	Neurotechnologija Ltd.
L	Name Not Released
VTB	NIST

Table 1. List of SDK vendor names associated with SDK letters[1]

2 Previous Work

Many papers in biometric literature address the problem of assessing fingerprint image quality. Nill & Bouzas [11] propose an objective image quality based on the digital image power of normally acquires scenes. Their system is designed to assess the quality of digital images and can be applied to fingerprint as well. Bolle, et al [12] used ratio of directional area to other non-directional area as a quality measure. Shen et al [13] applied gabor filter to image sub-blocks, to identify blocks with clear repetition of ridge and valley pattern as good quality blocks. Both [12] and [13] only use the local orientation information and neglect information on global uniformity and continuity The configurations of ridges and valleys within a local neighborhood vary with the quality of input fingerprint images, so a well defined sinusoidal shaped waves of ridges and valleys may not always be observed. Global features are needed for a more precise region mask classification. They both use subjective method to test the performance of their proposed quality analysis algorithms.

Ratha and Bolle [14] proposed a method for image quality estimation from wavelet compressed fingerprint image, where the fingerprint image is WSQ compressed. Still evaluating their quality measure is a subjective matter.

[1] These tests were performed for the Department of Homeland Security in accordance with section 303 of the Border Security Act, codified at 8 U.S.C. 1732. Specific hardware and software products identified in this report were used in order to adequately support the development of technology to conduct the performance evaluations described in this document. In no case does such identification 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.

Hong et al [15] modeled the ridge and valley pattern as sine wave, and compute the amplitude, frequency as well as the variance of the sine wave to decide the quality of the fingerprint. They classify regions of fingerprint as recoverable or unrecoverable. If the percentages of recoverable blocks are smaller than a specified threshold, the input fingerprint is rejected as poor quality.

Lim and Yau [16] propose algorithms for estimating quality and validity of a fingerprint image in spatial domain. They verify the repetition of ridge and valley patterns by the ratio of the eigenvalues obtained from covariance matrix for the image block's grey-level gradient. They use the quality of the feature extracted from the fingerprint image by the automatic fingerprint identification system as the benchmark to test the performance of their proposed image quality analysis procedure. Their system is aimed to flag poor quality or invalid fingerprint images.

Yao et al [2] compute quality Q as a measure of how much reliable directional information is available in a fingerprint image. They have used ROC curves to show better performance is achieved but did not use ROC to test the performance of their proposed quality analysis algorithms.

To our knowledge, nobody has publicly defined biometric sample quality as a predictor for recognition performance, and nobody has used ranked ROCs to assess the measured quality.

3 Defining Fingerprint Image Quality

A fingerprint is a pattern of friction ridges on the surface of a fingertip. A good quality fingerprint has distinguishable patterns and features that allow the extraction of features that are useful for subsequent matching of fingerprint pairs. A minutia based automatic fingerprint-matching algorithm uses features that compares local ridge characteristics (minutia) of two fingerprints (biometric samples) $x_{g(i)}$ and $x_{p(j)}$ and produces a real valued similarity score

$$s_{ij} = F(x_{g(i)}, x_{p(j)}) \qquad \text{(eq. 1)}$$

where subscript $g(i)$ denotes i-th gallery and $p(j)$ denoted j-th probe and s_{ij} is the score of the *i*-th gallery matched against the *j*-th probe.

We call similarity scores s_{ii} of a genuine (i.e. same person) comparisons *match scores*, and similarity scores $s_{ij}, i \neq j$ of imposter (i.e. different person) comparisons *non-match scores*. So, for a gallery of size *G*, with the assumption that there is one and only one biometric sample for each subject enrolled in the system, for each probe image $x_{p(i)}$ there is one match score s_{ii} and *G*-1 non-match scores $s_{ji}, i \neq j$. Let $s_m(x_i)$ denotes match score for sample $x_{p(i)}$ and $s_n(x_{ji})$ non-match scores of $x_{p(i)}$ and $x_{g(j)}$, while $i \neq j$.

A higher similarity score is construed to indicate a higher likelihood that the samples come from the same individual. Let $M(s_m)$ denote the cumulative distribution function (CDF) of the match scores, and $N(s_n)$ the CDF of non-match scores. The *Detection Error Tradeoff Characteristic* (DET) is a plot of the false non-match rate,

$$FNMR = M(s_m) \qquad \text{(eq. 2)}$$

against the false match rate,

$$FMR = 1 - N(s_n) \qquad \text{(eq. 3)}$$

for all values of s_m and s_n. The DET, and the equivalent ROC, are the commonest statement of performance of a verification system.

We define **fingerprint image quality as a predictor of a matcher's performance**. Before advancing any further, we need to quantify matcher performance. The similarity score is the ultimate statement of expected performance: in conjunction with the underlying match and non-match distributions it yields likelihood for the samples coming from the same person or different people. The match and non-match distributions are the result of complex non-linear algorithms and are not usually random but are strongly dependent on the internal algorithm and how its parameters are set.

Figure 1 shows the histogram of vendor VTB [10,6] match and non-match scores of fingerprint impressions of 216 subjects in data set NIST- SD29 [3]. There are 216 match scores and 46440 non-match scores.

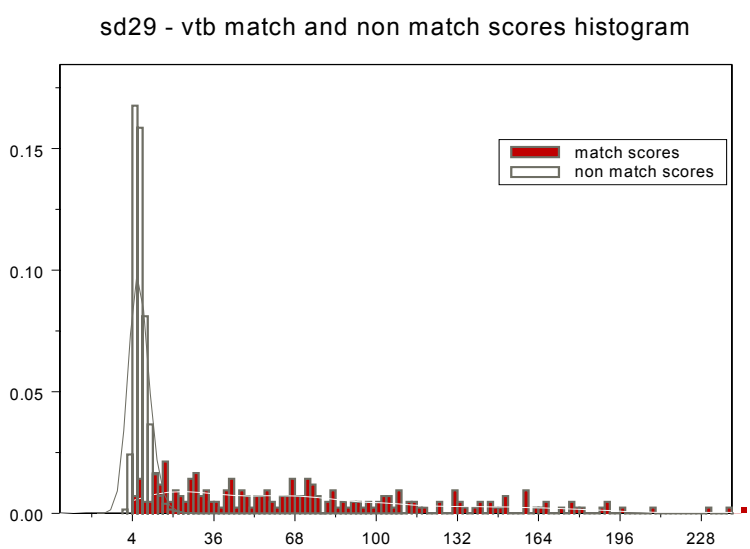


Figure 1 - SD29 VTB match and non-match scores histogram

It is common for the match distribution to be wider than non-match distribution. It is also quite typical for the two distributions to overlap. Overlapping match and non-match distributions means a given sample x_i will match falsely if its match score $s_m(x_i)$ is less than some non-match scores $s_n(x_{ji})$ $(s_m(x_i) < s_n(x_{ji})\ i \neq j)$. If the quality measure q is to be predictive of matcher performance, *good* quality fingerprints must be those with high match scores and well separated from the non-match distribution. Similarly, *poor* quality fingerprints are those with lower match scores, in particular those where their match scores are in the region of overlap with non-match scores.

Therefore, quality measure q should be indicative of the degree to which the match distribution $M(s_m)$ is separated from the non-match distribution $N(s_n)$. Specifically, we define quality q_i of biometric sample x_i to predict

$$o(x_i) = \frac{s_m(x_i) - E[s_n(x_{ji})]}{\sigma(s_n(x_{ji}))} \forall x_i \in \Gamma \text{ or } \Pi \qquad \text{(eq. 4)}$$

where *E[.]* is mathematical expectation, and $\sigma(.)$ is standard deviation, $s_m(x_i)$ is the match score, and $s_n(x_{ji})$ are the non-match scores of sample x_i, $\forall j,\ i \neq j$. Comparing a probe sample x_i with an internal gallery of *G* samples, which include one and only one sample from the same subject (person), results in a vector of *G* scores, ***s***. Only one element of vector ***s*** is x_i's match score, and the other *G*-1 are its non-match scores. *E[.]* is evaluated by computing mean of all non-match scores of probe sample x_i to all *G*-1 non-matching gallery entries. Likewise, $\sigma(.)$ is standard deviation estimated solely from the non-matching elements of ***s***. We call $o(x_i)$ normalized match score of sample x_i. Basically, we are comparing the subject's biometric sample to the claimed match sample and to other non-matching samples, and adjusting the raw score on the basis of the extra scores.

Figure 2 shows box plot of normalized match scores of SDK vendor matcher F as generated using equation (4) for right index fingers of dataset DOS-C for fingerprints that are matched correctly and those that are matched incorrectly separately. Note that fingerprints with lower values of $o(.)$ are matched falsely, where fingerprints that are matched correctly are having higher values $o(.)$.

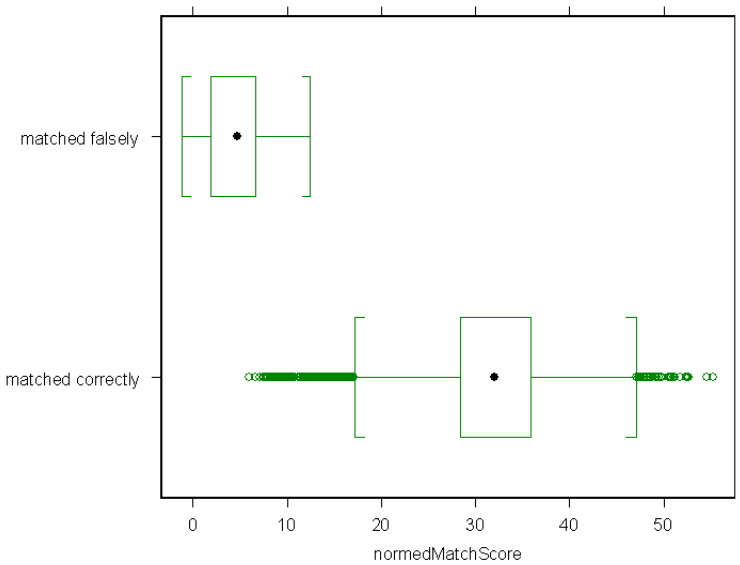


Figure 2. Box plot o(.) falsely matched and correctly matched separated. The black dot inside the box represents the median; the box shows inter-quartile-range of normalized match scores (eq. 4) of vendor F for right index fingerprints in dataset DOS-C. Whiskers are drawn are 1.5(inter-quartile-range) and data points beyond, which are considered outliers, are drawn individually.

For each fingerprint, we define its image quality as predictor of its normalized match score. Similarity scores as defined by equation 1, and so normalized match scores as defined by equation 4, are a function of both probe and gallery samples, but quality as defined here is a scalar value which is measured for each sample separately. Therefore, pair wise quality q as defined below, should be predictive of recognition performance of pair $(x_{gallery}, x_{probe})$.

$$q = H(q_{gallery}, q_{probe}) \qquad \text{(eq. 5)}$$

Extensive testing at NIST [5, 6] has shown that recognition errors are triggered by low quality samples. That is, $H(.)$ is simply a min function of the individual numbers q_{probe}, and $q_{gallery}$, and so pair wise quality is defined in equation 6. In this paper quality refers to q as defined in equation 6.

$$q = \min(q_{gallery}, q_{probe}) \qquad \text{(eq. 6)}$$

In an operational setting, if the enrolled samples are assured to have high quality, then a measurement of quality of a subject's biometric sample (probe) can be sufficient to predict its normalized match score.

We measure (scalar value) quality q_i for biometric sample x_i, by first computing a feature vector $\boldsymbol{v}_i$, which contains appropriate signal or image fidelity characteristics of x_i and then finding some (nonlinear) mapping from $\boldsymbol{v}_i$ to $o(x_i)$. Mathematically speaking:

$$\boldsymbol{v}_i = L(x_i) \qquad \text{(eq.7)}$$

$$q_i = \tilde{o}(x_i) = I(\boldsymbol{v}_i) \qquad \text{(eq. 8)}$$

Function $L(.)$ will be realized by computing characteristics and features of x_i that convey information for a matching algorithm. Applying $L(.)$ to a sample x_i results in an n-dimensional feature vector $\boldsymbol{v}_i$. For fingerprints, this includes measured clarity of ridges and valleys, size of the image, and measures of number and quality of minutiae. Section 4 has a detailed explanation of our feature vectors.

Function $I(.)$ is a mapping from the space of feature vectors $\boldsymbol{v}$ to normalized match scores $o(.)$. $\tilde{o}(x_i)$ is the predicted value for $o(x_i)$.

3.1 Image Quality as Classification

We defined biometric sample quality as a predictor of its normalized match score as defined in equation 4. Equation 8 suggests use of various regression methods to estimate the response function $o(.)$ from a vector of variables $\boldsymbol{v}$. However, as shown in detail in section 4, we tried various regression methods and failed to find a good fit primarily because: sample-specific measures (components of vector $\boldsymbol{v}$) are not linearly related to the response variable $o(.)$ and so nonlinear functions have to be found; outliers heavily influence data; and residual errors are not Gaussian for any of the regression methods tried. Plus, it is sufficient to know the level of quality (e.g. high, good, medium, low) and since quality is defined as predictor of $o(.)$ (normalized match scores) it is sufficient to know the range of $o(x_i)$ (e.g. high, good, medium, low) rather than its exact value for each sample x_i. These facts lead us to restate the problem in terms of classification. This means we define sample quality as a measure that predicts the *bin* of $o(x_i)$. Now function $I(.)$ from equation 8 is basically a classifier that maps feature vector $\boldsymbol{v}_i$ of sample x_i (eq. 7) to a quality number Q such that $o(x_i)$ falls in the k-th bin where, without loss of generality, $Q = k$. That is, the quality number q_i of sample x_i is the bin in which $o(x_i)$ will fall. Our implementation employs an artificial neural network [17] as the classifier.

The number of allowed bins, K, constitutes a coarseness parameter, against which the quality number can be traded off. For example, a high / low level ($K = 2$) is easier to achieve than the continuous case as K tends to infinity discussed above. In this paper we chose $K = 5$, that is we have five levels of quality: poor (5), fair (4), good (3), very good (2), and excellent (1). The

choice of 5 levels of image quality is a compromise. Studies at NIST [6] shows that eight to ten levels would be needed to fully characterize a matcher that is very sensitive to image quality. Matchers that are insensitive to image quality could be well characterized by three levels of image quality. We have selected five levels, which is the correct level for a medium to good quality matcher. We used irregularly spaced quantiles of $o(.)$ to define the bin boundaries for $o(.)$. For a sample, x, the quantile function is the inverse of the empirical CDF; that is,

$$\text{quantile}(p) = \min\{z \mid \text{proportion } p \text{ of } o(x) \chi z\}$$

We define $o(.)$ bin boundaries as shown in Table 2, where $W(.)$ denotes the CDF of samples that are falsely (wrongly) matched and C denotes CDF of samples that are correctly matched.

Q	QUALITY	RANGE
5	poor	$[0, W^{-1}(0.75)]$
4	fair	$(C^{-1}(0.75)], C^{-1}(0.05)]$
3	good	$(C^{-1}(0.05), C^{-1}(0.2)]$
2	very good	$(C^{-1}(0.2), C^{-1}(0.6)]$
1	excellent	$(C^{-1}(0.6), C^{-1}(1)]$

Table 2. Bin boundary for normalized match scores o(.). The boundaries were set by inspection to give useful categorization of the normalized match scores statistic.

Our bin boundary selection agrees with our definition of quality; we are labeling samples with poorest recognition rate as poor, (in this case all of them are falsely matched). Samples with fair quality are those that are borderline, some of the these samples are falsely matched, most of good quality samples are matched correctly and very good and excellent are those samples that almost all are matched correctly. In other words, excellent quality samples are those with very high FNMR (eq 2) and poor quality samples are those with high FMR (eq 3).

4 Measuring Fingerprint Image Quality

In this section we discuss our implementation of $L(.)$ and $I(.)$ of equation 7 and 8 for fingerprint images. We first apply $L(.)$ to a biometric sample x_i to get feature vector $\boldsymbol{v}_i$ and then use $\boldsymbol{v}_i$ as input into a neural network, $I(.)$. $L(.)$ is realized by computing characteristics and features of biometric sample x_i that convey information for a matching algorithm.

4.1 Feature extraction

This section explains feature extraction for fingerprints. Our proposed definition and measurement of biometric sample quality can be applied to other biometric modalities if the appropriate feature vectors are defined and computed accordingly.

It is known that fingerprint matcher algorithms commonly in use are sensitive to clarity of ridges and valleys, measures of number and quality of minutiae, and size of the image. We have used NIST Fingerprint Image Software (NFIS) [7] package to extract features, i.e. implementation of $L(.)$ of equation 7. The MINDTCT package of NFIS has a fingerprint minutia detector algorithm that accepts a fingerprint image and automatically detects minutia. It also assesses minutia quality and generates an image quality map. To locally analyze a fingerprint image, NFIS divides the image into grids of blocks. To assess the quality of each block, NFIS computes several maps (direction map, low contrast, low flow, and high curve) and summarizes the result in a quality map. All pixels in a block are assigned the same result. It should be noted that the NFIS algorithms and software parameters have been designed and set to process images scanned at 500 pixels per inch (19.69 pixels per millimeter) and quantized at 256 levels of gray. A discussion of MINDTCT parameters and how it is used in our quality assessment follows.

4.1.1 MINDTCT input

MINDTCT reads in an ANSI/NIST formatted file and searches the file structure for a grayscale fingerprint record. The application is capable of processing ANSI/NIST Type-4, Type-13, and Type-14 fingerprint image records [8].

4.1.2 Generate image quality map

MINDTCT measures quality of localized regions in the image including determining the directional flow of ridges in the image and detecting regions of low contrast, low ridge flow, and high curvature. These last three conditions represent unstable areas in the image where minutiae detection is unreliable, and together they can be used to represent levels of quality in the image. Each of these characteristics is discussed below.

4.1.2.1 ***Direction map*****:** The purpose of this map is to represent areas of the image with sufficient ridge structure. Well-formed and clearly visible ridges are essential to reliably detection of ridge endings and bifurcations. To locally analyze the fingerprint, the image is divided into grid of blocks. All pixels within a block are assigned the same value. To minimize the discontinuity in block values as one move from one block to its neighbor, windows are defined to surround blocks, and windows overlap from one block to the next. For each block in the image, the surrounding window is rotated incrementally and a Discrete Fourier Transform (DFT) is conducted at each orientation. The details are given in [7].

4.1.2.2 ***Low contrast map*****:** An image map called the low contrast map is computed where the blocks of sufficiently low contrast are flagged. This map separates the background of the image from the fingerprint, and maps out smudges and lightly inked areas of the fingerprint. Minutiae are not detected within low contrast blocks in the image. This software computes the pixel intensity distribution within the block's surrounding window. A specified percent (10%) of the

distribution's high and low tails are trimmed as possible outliers, and the width of the remaining distribution is measured. A pixel intensity threshold was derived empirically from a training sample of low and high contrast blocks extracted from real fingerprint images. Blocks that have narrow dynamic range in pixel intensity are flagged as low contrast areas.

4.1.2.3 *Low flow map*: Low flow map marks the blocks that could not initially be assigned a dominant ridge flow. Minutiae detected in low flow areas are not reliable.

4.1.2.4 High curve *map*: Minutiae detected in high curvature areas are not reliable. This is especially true of the core and delta regions of a fingerprint. A high curve map is used to marks blocks that are in high curvature areas of the fingerprint.

4.1.2.5 Quality map: As discussed, the low contrast map, low flow map, and the high curve map all point to different low quality regions of the image. The information in these maps is integrated into one general map, and contains 5 levels of quality (4 being the highest quality and 0 being the lowest). The background has a score of 0, a score of 4 means a very good region of fingerprint. The quality assigned to a specific block is determined based on its proximity to blocks flagged in the above-mentioned maps. We display quality map grayscale image with black, dark gray, medium gray, light gray, and white corresponding to scores of 0 to 4 respectively.

Figure 3 shows an example of a good quality fingerprint and its gray scale image of quality map as described above. Note that most of gray scale quality map image is white. Figure 4 shows an example of a poor quality image and its gray scale level translation. It is notable that majority gray scale quality map image is dark gray or black. Quality maps of both fingerprints are in appendix B. We have subjectively checked quality map of several fingerprint images and we are confident that quality maps generated by MINDTCT is a good representation of areas with clear ridges and valleys and areas where minutiae detection is (or is not) reliable.

For each fingerprint, we used MINDTCT to generate its quality map. Blocks with quality 0 are regarded as background. We compute the total number of blocks with quality 1 or better as the effective size of the image or foreground. Then percentage of foreground blocks with quality 1, 2, 3, and 4 are computed. We call them quality zones 1, 2, 3, and 4 respectively. Fingerprint images with higher number of quality zone 4 (equivalently smaller number of quality zone 1 and/or 2) are more desirable.

Figure 3. An example of a fingerprint subjectively assessed to be of good quality

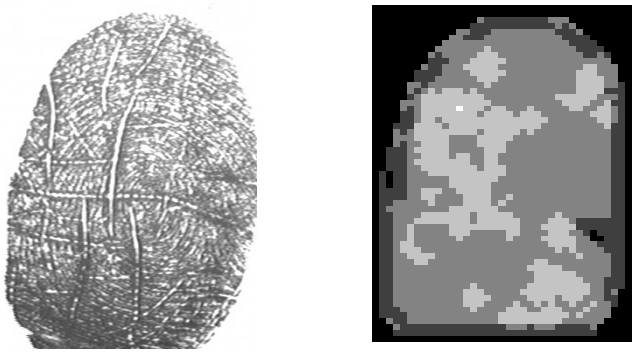


Figure 4. An example of a fingerprint subjectively assessed to be of bad quality

4.1.3 Assess Minutia Quality: NFIS computes a quality/reliability to be associated with each detected minutia point. Although NFIS performs several steps of false minutiae removal, false minutiae usually remain in the candidate list. A robust quality measure can help manage this in that false minutiae should be assigned a lower quality than true minutiae. Two factors are combined to produce a quality measure for each detected minutia point. The first is taken directly from the location of the minutia point within the quality map described above. The second factor is based on simple pixel intensity statistics (mean and standard deviation) within the immediate neighborhood of the minutia point. An area with clear ridges and valleys will have a significant contrast that will cover the full grayscale spectrum. Consequently, the mean pixel intensity of the neighborhood will be very close to 127. For similar reasons, the pixel intensities of an ideal neighborhood will have a standard deviation ≥ 64. Based on this logic and using quality map discussed in 4.1.2.5, NFIS assigns a quality value on the range 0.01 to 0.99 to each minutia. A low quality minutia value represents a minutia detected in a lower quality region of the image, whereas a high quality minutia value represents a minutia detected in a higher quality region.

For a fingerprint, NFIS detects and assesses quality of each minutia. Minutiae with quality lower than 0.5 are not reliable. We compute the number of minutiae of quality 0.5 or better, 0.6 or better, 0.75 or better, 0.8 and better, and 0.9 and better.

4.1.4 Feature vectors

We used the quality map and minutia quality assessment of NFIS and define our feature vectors, as shown in Table 3. Therefore, for each fingerprint x_i, an 11-dimensional feature vector $\boldsymbol{v}_i$ as listed in Table 3 is computed using MINDTCT of NFIS.

	NAME	DESCRIPTION
1	**foreground**	number of blocks that are quality 1 or better; i.e. foreground = $\sum_{i=1}^{4} U_i$ where U_i is number of blocks with quality i
2	**total #of minutia**	number of total minutiae found in the fingerprint
3	**min05**	number of minutiae that have quality 0.5 or better
4	**min06**	number of minutiae that have quality 0.6 or better
5	**min075**	number of minutiae that have quality 0.75 or better
6	**min08**	number of minutiae that have quality 0.8 or better
7	**min09**	number of minutiae that have quality 0.9 or better
8	**quality zone 1**	percentage of the foreground blocks of quality map with quality =1
9	**quality zone 2**	percentage of the foreground blocks of quality map with quality =2
10	**quality zone 3**	percentage of the foreground blocks of quality map with quality =3
11	**quality zone 4**	percentage of the foreground blocks of quality map with quality =4

Table 3. Feature vector description

Feature vector of the fingerprint in Figure 3 (subjectively assessed as good quality fingerprint) is computed as:
00602_01: (1642, 57, 34, 34, 34, 34, 28, 0.053, 0.167, 0.059, 0.720)

and for the fingerprint in Figure 4 (subjectively assessed as bad quality) is:
00207_01: (1690, 182, 0, 0, 0, 0, 0, 0.168, 0.572, 0.259, 0.0006)

It is apparent that the good quality fingerprint has a large quality zone 4 and higher number of minutia of reliability 0.9 or better where the poor quality fingerprint has a large quality zone 2 and no minutia with reliability 0.5 or better.

To get a better insight into the feature vectors and uncover its underlying structure we performed an explanatory data analysis (eda). Eda is a graphically oriented method of data analysis that helps to determine whether the data support the assumptions required for the classical methods of statistical inference: an outlier-free, nearly normal distribution, and serially uncorrelated observations. We have a set of 4 plots for each feature computed for right index fingers of dataset DOS-C: a histogram, a box plot, a density plot, and a normal qq-plot. A normal qq-plot (or quantile-quantile plot) consists of a plot of the ordered values of the data versus the corresponding quantiles of a standard normal distribution. If the qq-plot is fairly

linear, the data are reasonably Guassian, otherwise; they are not. Of these four plots, the histogram and density plot give us the best picture of the distribution shape, while the box plot and normal qq-plot give the clearest display of outliers. The box in each box plot represents first and third quartile of the data, black dot inside the box represents the median, data points outside [1st quartile, 3rd quartile] range are shown by black lines outside the box. Figure 5 summarizes our analysis for right index fingerprints in dataset DOS-C in statistical graphics. Table 4 shows the Spearman's rank correlation of features of right index fingerprints of dataset DOS-C and normalized match scores of vendor F as defined in equation 4.

	QUALITY ZONE1	QUALITY ZONE2	QUALITY ZONE3	QUALITY ZONE4	NORMED MATCH SCORE
FOREGROUND	0.139	0.139	0.012	0.079	0.108
TOTAL #OF MIN	0.258	-0.385	0.292	-0.315	0.076
MIN05	-0.347	-0.386	-0.510	0.563	0.391
MIN06	-0.349	-0.387	-0.513	0.566	0.391
MIN075	-0.412	-0.253	-0.521	0.555	0.313
MIN08	-0.398	-0.169	-0.455	0.483	0.246
MIN09	-0.339	-0.047	-0.303	0.327	0.131
NORMED MATCH SCORE	-0.242	-0.172	-0.255	0.289	1.0

Table 4. Features and normalized match score correlations - Dataset DOS-C – right index fingerprints and vendor F scores. Min06 and quality zone4 have the highest correlation (0.566). There are no significant correlations between normalized match score and features.

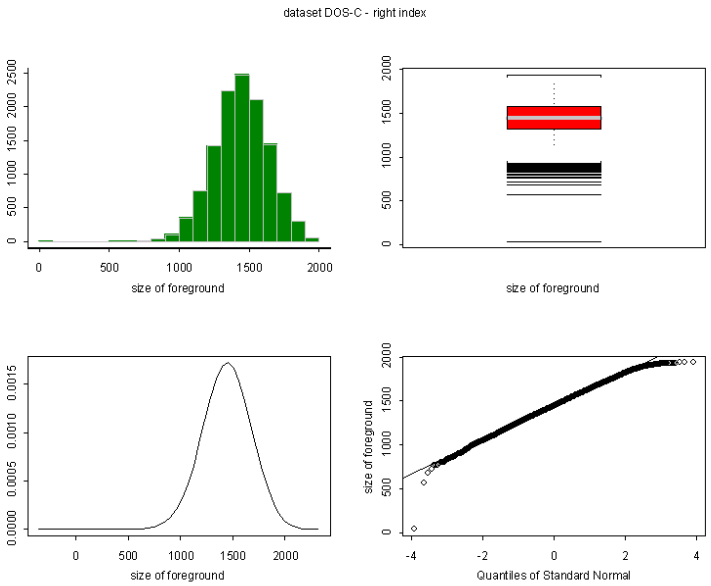


Figure 5-a. Exploratory data analysis for foreground (# of block with quality 1 or better)

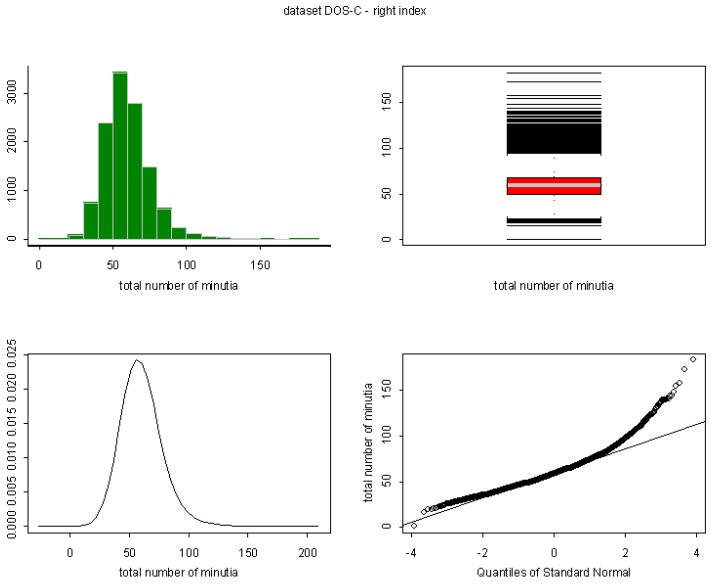


Figure 5-b. Exploratory data analysis total number of minutia

Dataset DOS-C Right Index

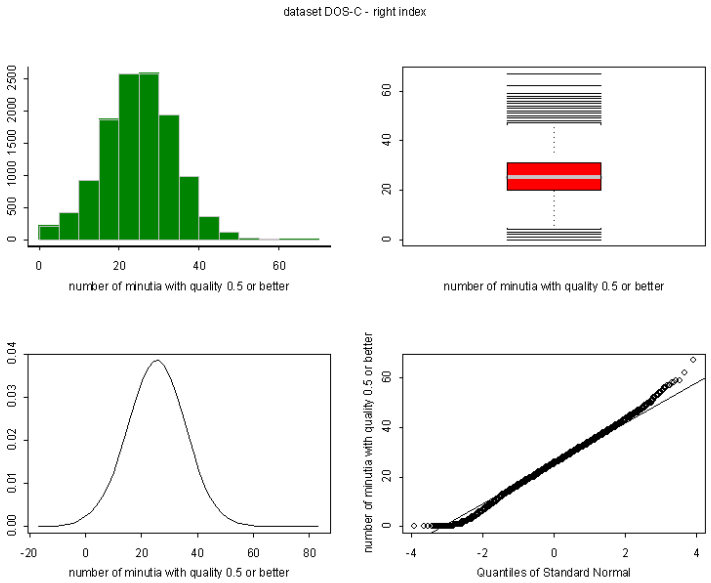


Figure 5-c. Exploratory data analysis for min05 (# of minutiae with quality 0.5 or better)

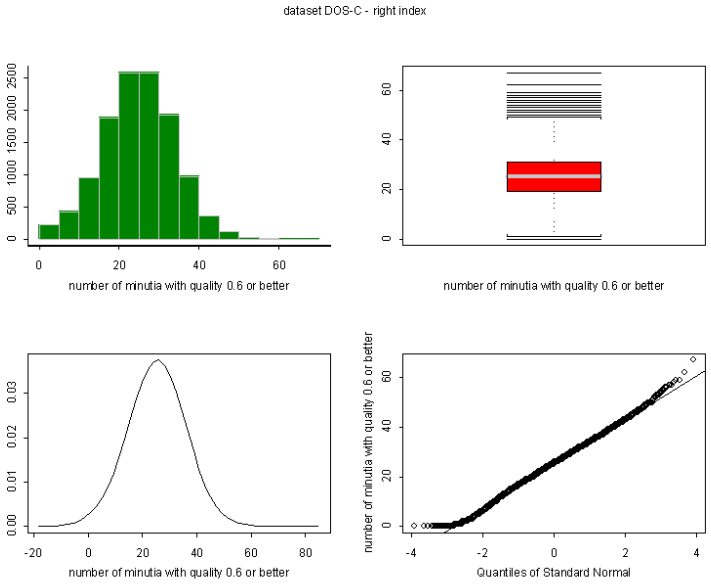


Figure 5-d. Exploratory data analysis for min06 (# of minutiae with quality 0.6 or better)

Dataset DOS-C Right Index

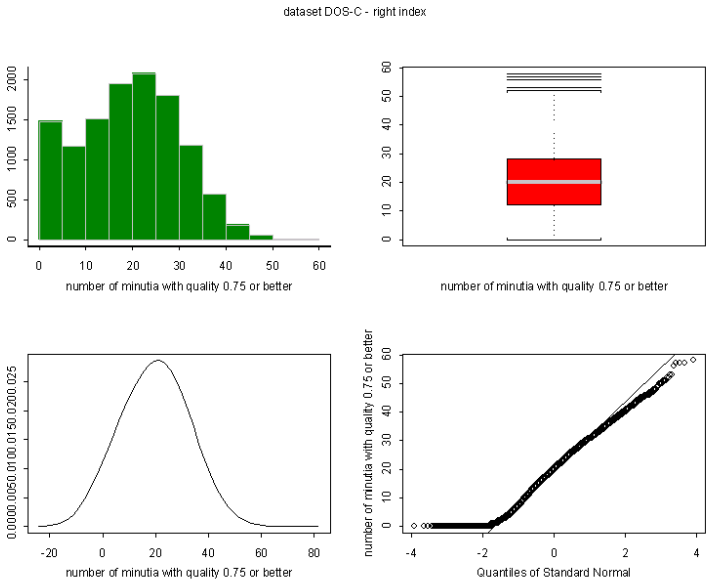


Figure 5-e. Exploratory data analysis for min075 (# of minutiae with quality 0.75 or better)

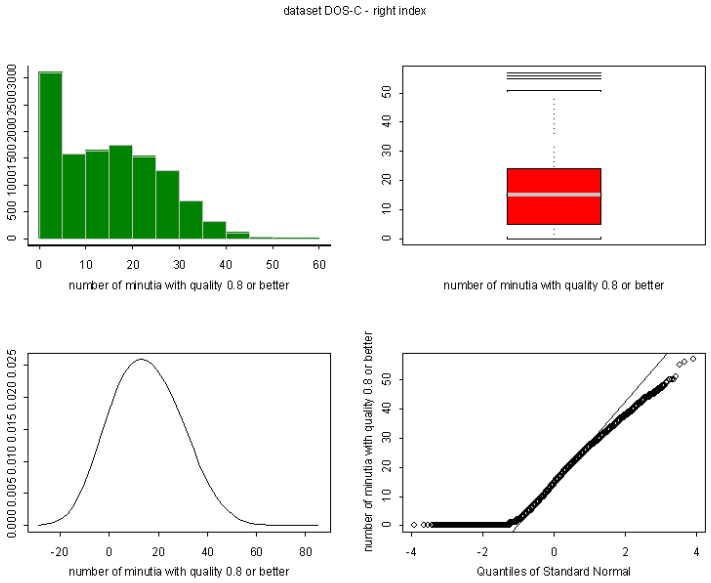


Figure 5-f. Exploratory data analysis for min08 (# of minutiae with quality 0.8 or better)

Dataset DOS-C Right Index

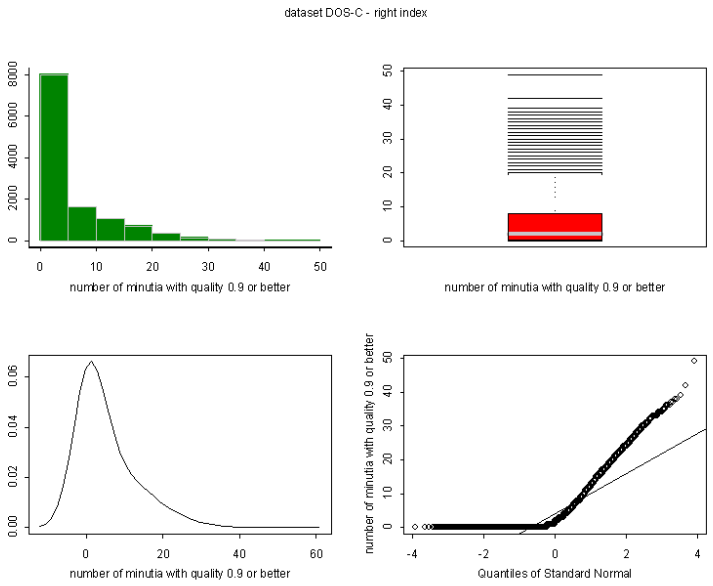


Figure 5-g. Exploratory data analysis for min09 (# of minutiae with quality 0.9 or better)

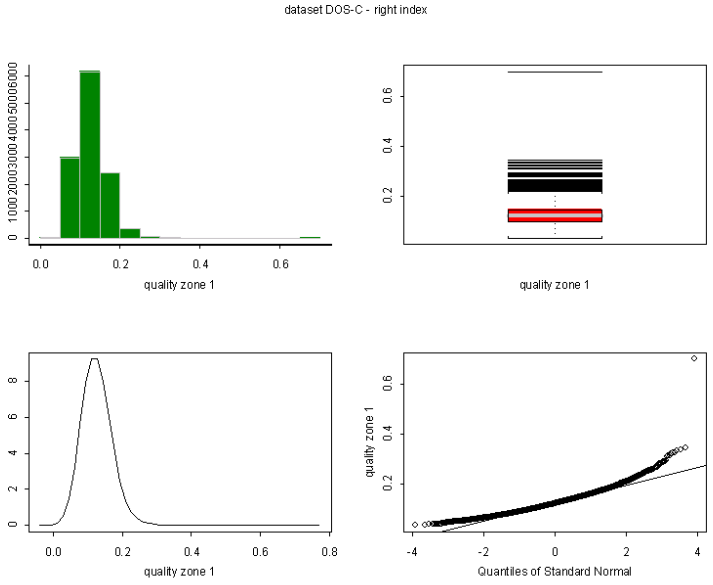


Figure 5-h. Exploratory data analysis for quality zone1 (fraction of foreground with quality 1)

Dataset DOS-C Right Index

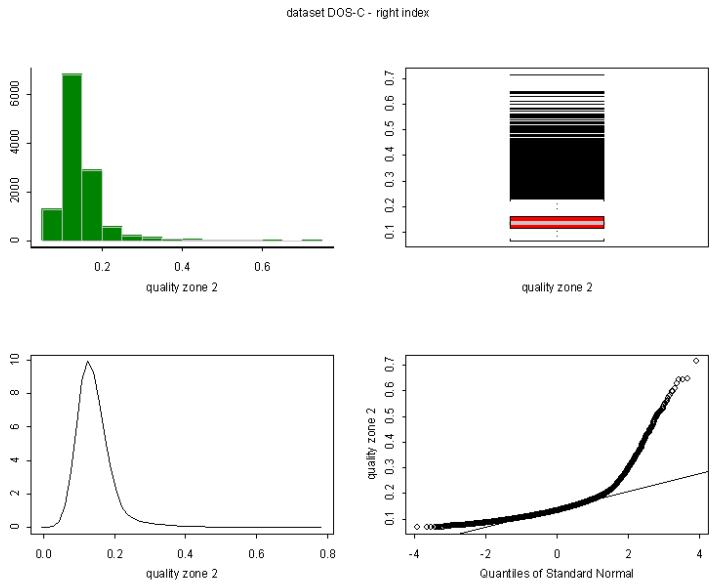


Figure 5-i. Exploratory data analysis for quality zone2 (fraction of foreground with quality 2)

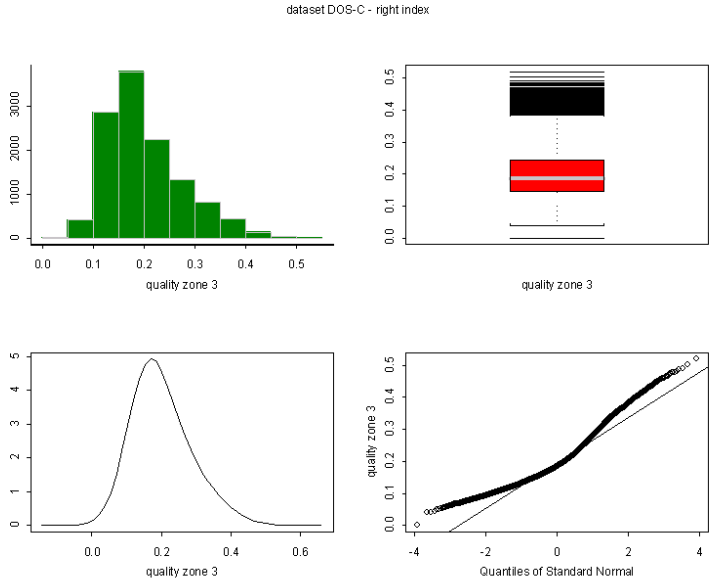


Figure 5-j. Exploratory data analysis for quality zone3 (fraction of foreground with quality 3)

Dataset DOS-C Right Index

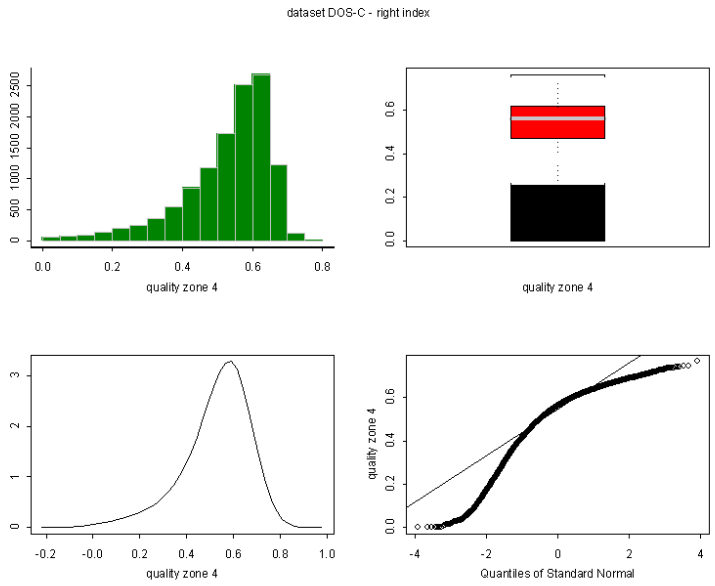


Figure 5-k. Exploratory data analysis for quality zone4 (fraction of foreground with quality 4)

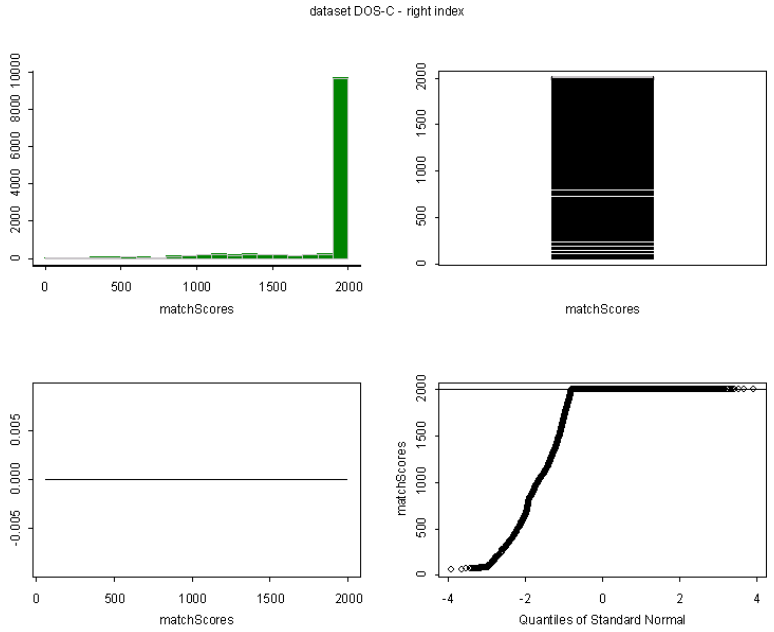


Figure 5-l. Exploratory data analysis for vendor F match scores

Dataset DOS-C Right Index, vendor F normalized match score

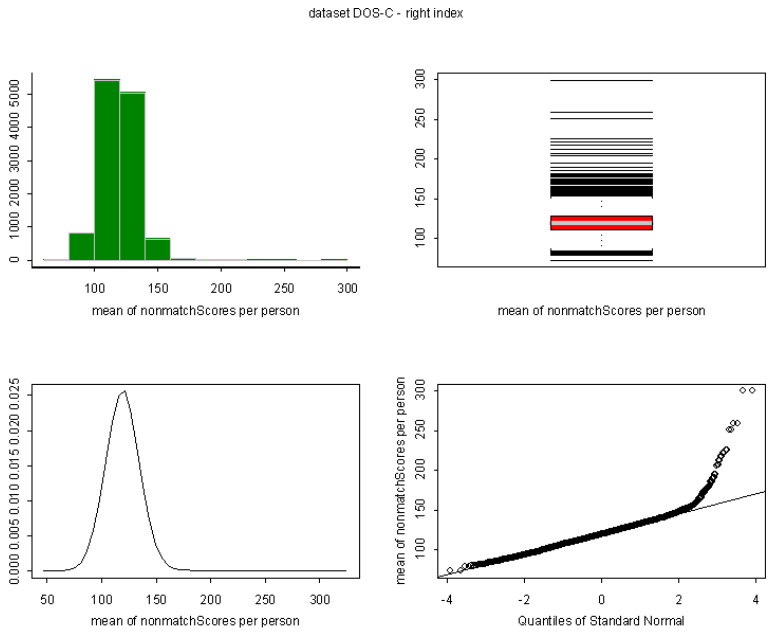


Figure 5-m. Exploratory data analysis for vendor F non-match scores - Dataset DOS-C Right Index

Note the departure of normal distribution for the tail of mean of non-match scores per person. As mentioned dataset DOS-C was consolidated and therefore non-match mean is not being artificially increased by erroneous ground truth data.

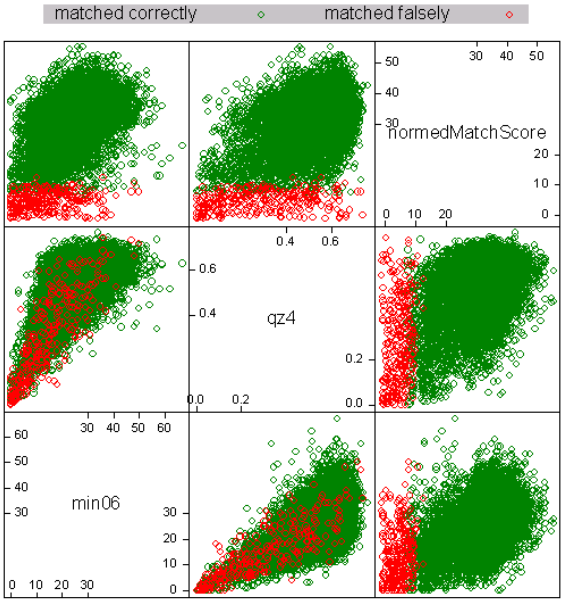
Figure 5 reveals several interesting observations:

- Outliers heavily influence data. Presence of outliers, as shown in Figures 5-a through 5-m, suggests the need for a robust statistical method. It is important to note that it will not suffice to screen data and remove outliers because the sharp decision to keep or reject an observation is wasteful. Also it is very difficult or even impossible to spot outliers in multivariate data. Rejecting outliers affects the distribution. We can do better by down-weighting dubious observations rather than by rejecting them, although we may wish to reject completely wrong observations. To compute robust measures, we can compute trimmed mean or rank correlation. The trimmed mean is the mean of the central 1-2α part of the distribution, so fraction α of the observations are removed from each end.

- The distributions are not normal, and not even nearly normal. This suggests the use of nonparametric methods.

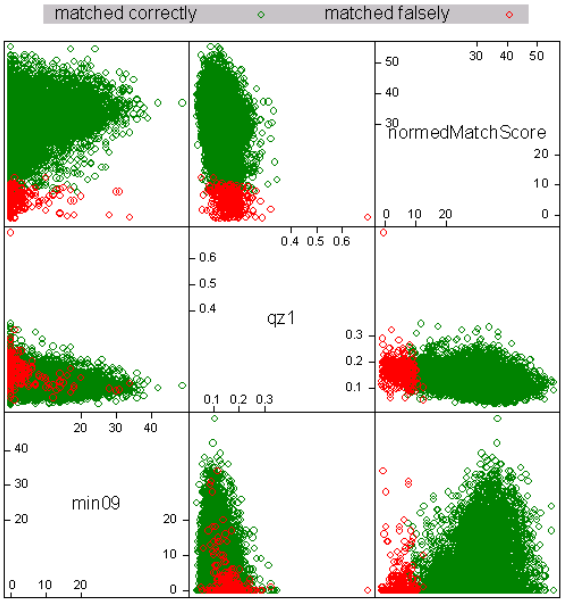
As shown by qq-plots of Figures 5a-m normality is not a valid assumption, so rank-based Spearman's ρ measures of correlation between each DOS-C right index sample's feature and its vendor F normalized match score (i.e. $o(.)$) are computed and reported in Table 4.

The correlation factors between features and normalized match scores are not very attractive. Min05, min06, and quality zone 4 with 0.391, 0.391, and 0.289 respectively are the highest. This means none of features can be a good predictor of the normalized match score all by itself. However, jointly they might be used to predict it. To investigate any association, or correlation, between any features and the normalized match score, scatter plots of each feature vs. normalized match score were plotted. Scatter plots can be used as a visual check on correlation between two variables. If two variables are strongly correlated, a scatter plot of one variable against the other will look like a straight line. A strong correlation between any feature and normalized match score ($o(.)$) means that we will be able to use that feature as a predictor of normalized match score $o(.)$. Figures 6, 7, 8, and 9 show matrices of scatter plots of components of feature vectors for DOS-C right index fingerprints and normalized match scores of vendor F. Green circles represent fingerprints that are matched correctly and red ones represent fingerprints that are not matched correctly.



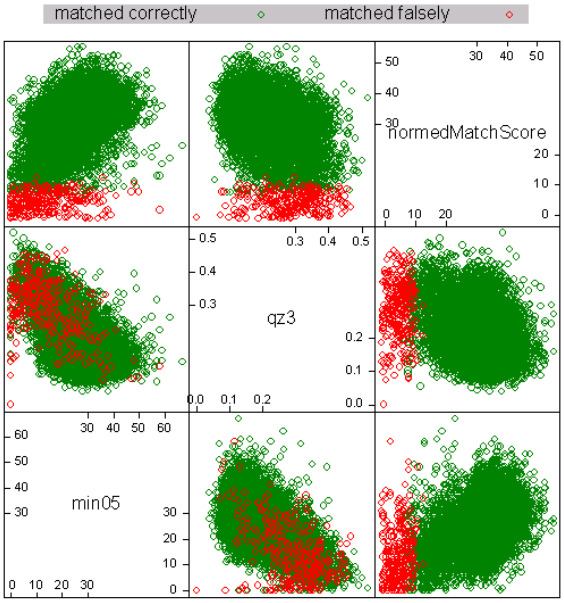
matrix plot of min06, quality zone 4, and normalized vendor F's match score (o(.))

Figure 6. Min06 and quality zone 4 are somehow correlated. Fingerprints with small values of min06 and quality zone 4 are most likely identified falsely.



matrix plot of min09, quality zone 1, and normalized vendor F's match score (o(.))

Figure 7. Scatter plots min09, quality zone1, and vendor F's normalized match score for dataset DOS-C Right Index. Fingerprints with min09 bigger than 15 are most likely identified correctly and so, by our definition, are of "good" quality.



matrix plot of min05, quality zone 3, and normalized vendor F's match score (o(.))

Figure 8. Fingerprints with small number of min05 and large values of quality zone 3 are most likely identified falsely.

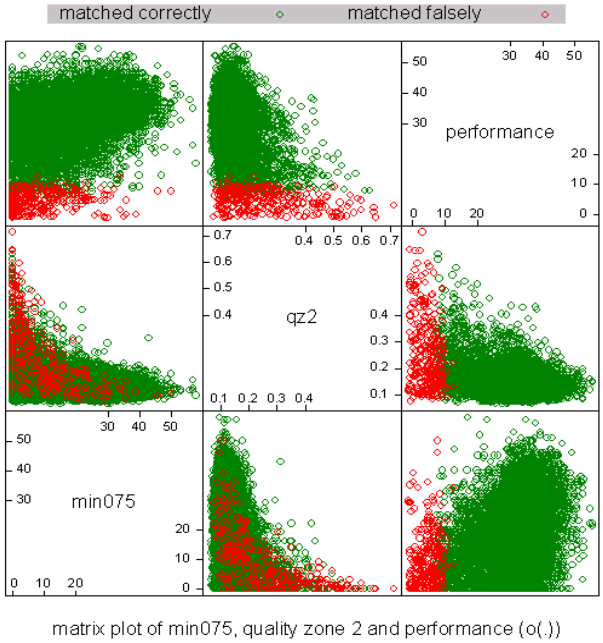


Figure 9. Fingerprints with high number of min075 (>20) and small values of quality zone 2 are most likely identified correctly and hence are of "good" quality.

Scatter plots show explicitly the consequences of the feature correlation discussed in Table 4. Figure 6 shows three sets of scatter plots between min06, quality zone4, and the normalized match scores of vendor matcher F as generated using equation 4. The plot does not reveal a clear relationship between either min06 or quality zone4 and normalized match scores. It does, however, show two weak correlations. Sufficiently large values of min06 result in green circles predicting correct match in the lower right subplot of Figure 6. Further, small values of both quality zone4 and min06 associate incorrect match (red circles) in the center left and center bottom subplots of Figure 6. This behavior is also seen in Figure 9 for features min075, and quality zone2. Figure 8 shows a weak correlation between min05 and quality zone 3. Red circles in center bottom subplot of Figure 8 indicate that sufficiently large values of quality zone 3 and small numbers of min05 result in false identification. Figure 7, for features min09 and quality zone1, shows the correlation associated with large values of the minutia variable but do not show a predictive effect when the minutia variable is combined with the quality variable. Similar results hold for the effect of other features on normalized match scores.

The observation that the quality measures generated by NFIS are capable of predicting the performance of a commercial fingerprint matching system, which uses completely different feature extraction methods, is of great importance. This suggests that an image quality measure based on the publicly available NFIS generated features can be used to predict matcher performance for one and, as we will show later in section 5, other commercial systems.

To summarize, correlation factors of none of the features is strong enough to predict the normalized match score all by itself. However, there is strong evidence that some linear or

nonlinear combination of these features can predict normalized match scores as defined in equation 4. The Next section discusses how these features are used to predict normalized match scores.

4.2 Pattern recognition (Regression vs. Classification)

In the terminology of pattern recognition, the given examples, together with their classifications, are known as the *training set*, and future cases form the *test set*. Our primary measure of success is the error (or misclassification) rate. Note that we would obtain (possibly seriously) biased estimates by re-classifying the training set, but that the error rate on a test randomly chosen from the whole population will be an unbiased estimator. To avoid the over fitting problem, we formed two disjoint subsets of fingerprint images drawn from datasets DOS-C, DHS2-C, BEN, TXDPS, and DHS10, used one subset to fit the model (train set), and used the other subset (test set) for assessment of the generalization error of the final chosen model. Data set SD29[3] was used as validation set.

Equation 8 suggests use of various regression methods. We have a vector inputs $\boldsymbol{v}$ = (feature_1, feature_2, … , feature_n), and want to predict the real valued output $o(.)$ .The linear regression model has the form

$$Y = f(x) = \beta_0 + \sum_j \beta_j x_j$$

The linear model either assumes that the regression function $E[Y \mid X]$ is linear, or that the linear model is a reasonable approximation. The β_j are unknown parameters or coefficients, and the variables x_j can come from different sources: quantitative inputs, transformations of quantitative inputs, such as log or square, and basis expansions, such as x_j^2, or interactions between variables $x_i x_j$. No matter the source of x_j, the model is linear in the parameters, β_j.

Linear regression is a natural choice when graphical evidence clearly indicates a linear relationship between response (normalized match score) and predictors (features). According to scatter plots of dataset DOS-C (Figures 5-a through 5-m) and matrix plots of Figures 6, 7, 8, and 9, such a linear relationship does not exist. We tried several regression methods but none fit the data. R^2 was never larger that 0.30, furthermore residuals did not have a standard covariance structure and residuals had a larger spread than fitted values. Therefore, though simple, the traditional linear model is not a good fit for our data because the classical linear regression techniques make several strong assumptions about the underlying data, and the feature data fail to satisfy these assumptions in several ways. For example, one or more outliers may throw off the regression line, or the data may not have a normal distribution. We know that outliers influence our data heavily and it does not have a normal distribution. To overcome the influence of outliers, we used robust regression methods; these minimize the effect of outliers while retaining the basic form of the linear method. We even replaced one or more predictors by polynomial smooth function of the predictor, converting the classical linear model into a

generalized additive model (GAM). In the regression setting, a generalized additive model has the form [23]:

$$Y = \alpha + f_1(x_1) + f_2(x_2) + \mathrm{K} + f_p(x_p) + \varepsilon$$

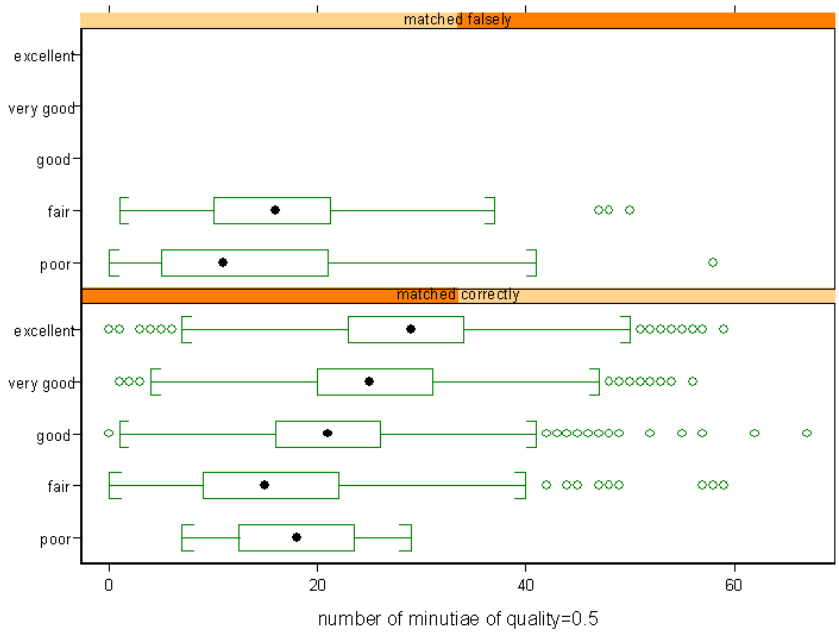
As usual, $x_1, x_2, ..., x_p$ represent predictors (features) and Y is the outcome (normalized match score); the f_i's are unspecified smooth (and so nonparametric) functions. The approach taken was to fit each function using a scatter plot smoother (e.g., a loess smoother or spline smoother), and provide an algorithm for simultaneously estimating all p functions. The model was fit using the local scoring algorithm, which iteratively fits weighted additive models by backfitting. The backfitting algorithm is a Gauss-Seidel method for fitting additive models, by iteratively smoothing partial residuals. The algorithm separates the parametric part from the nonparametric part of the fit, and fits the parametric part using weighted linear least squares within the backfitting algorithm. We tried both loess and spline smoother for the model. GAM takes care of non-linearity and non-normality of errors (residuals) but failed due to an interaction between predictors, x_j (i.e. features), which, in turn, suggests the use of projection pursuit regression [4]. This is a non-parametric regression model that includes the possibility of having interactions between the explanatory variables. Project pursuit regression applies an additive model to projected variables. That is, it is of the form:

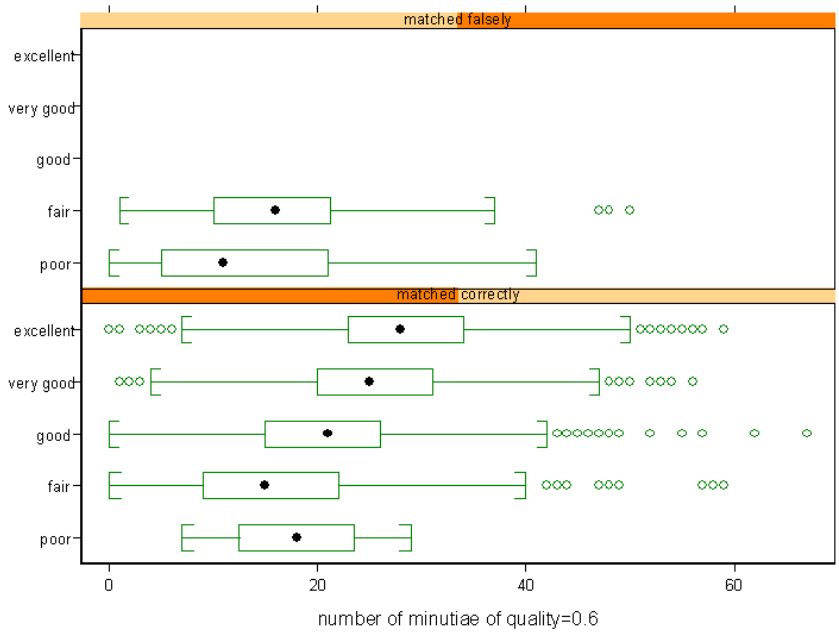
$$Y = \alpha_0 + \sum_{j=1}^{M} f_j(\alpha_j^T X) + \varepsilon$$

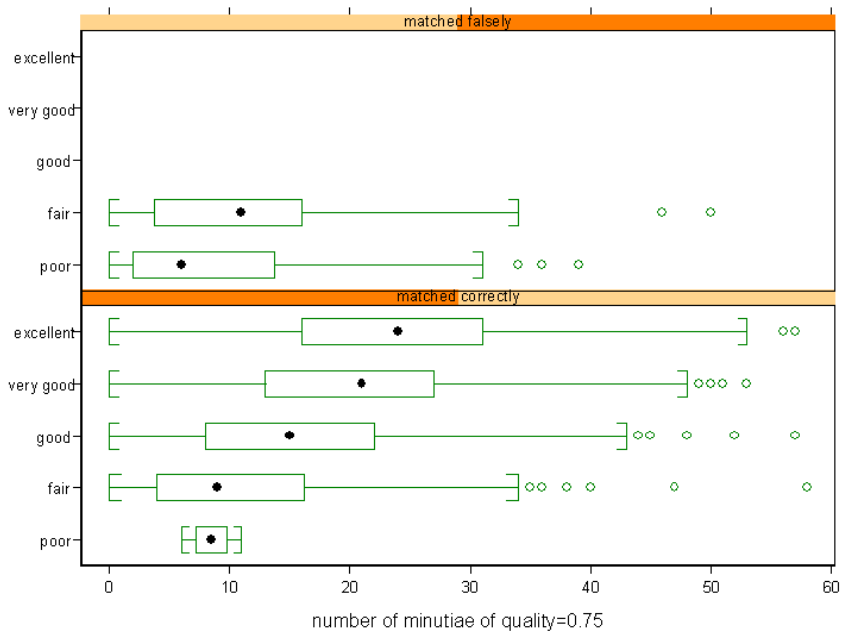
for vectors $\boldsymbol{\alpha}_j$, and a dimension M to be chosen by the user. Thus it uses an additive model on predictor variables that are formed by projecting $\boldsymbol{X}$ in M carefully chosen directions. Projection pursuit regression did not do better than any other model we discussed before.

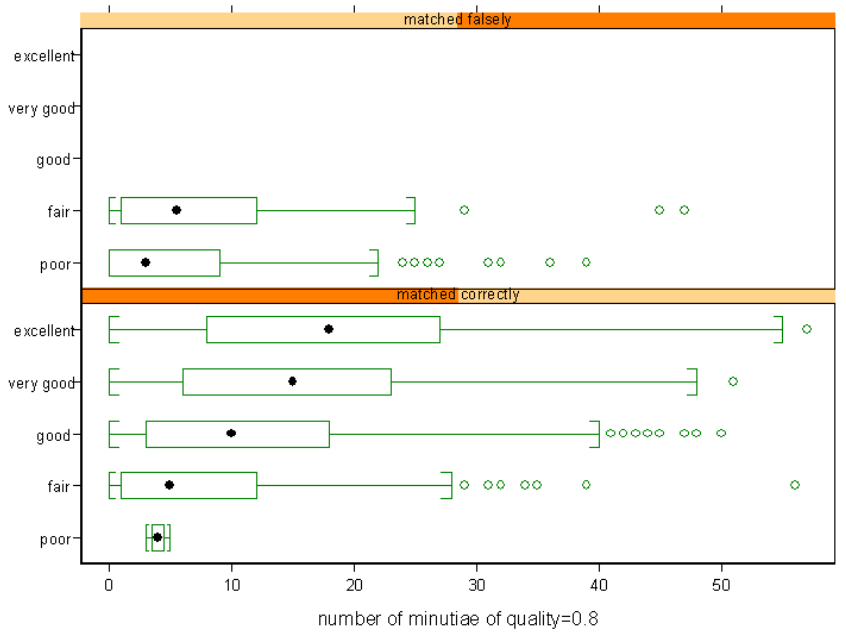
Failure of regression and additive models, plus our rationale for choosing discrete quality (section 3.1), leads us to state the problem as a classification problem. To investigate the discriminatory power of our feature vectors, for right index fingerprints in data set DOS-C, we made box plots of each feature for different levels (bins as given in Table 2) of normalized match scores of vendor F for those correctly identified and those that are falsely identified separately. These box plots are shown in Figure 10. The box in each box plot shows the middle half of the data and the black dot inside the box represents the median. The whiskers are at 1.5(inter-quartile range) and data points beyond that are considered outliers and are drawn individually. Box plots show not only the location and spread of the data but indicate skewness as well. Trellis plots of components of feature vectors of right index fingerprints of dataset DOS-C and normalized match scores of vendor F are in appendix B.

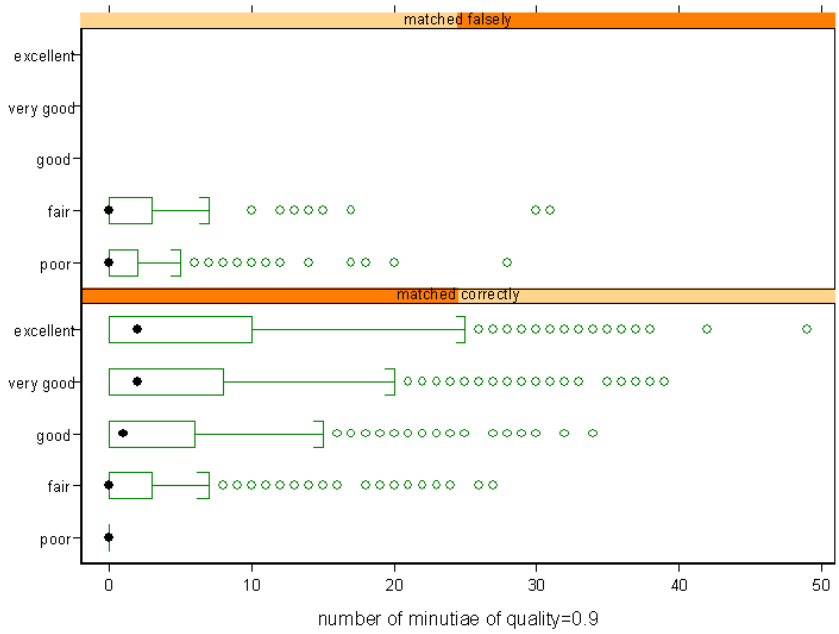
As it appears in Figure 10, the median of each feature varies over different performance bins. This implies that these features can be used to partition the space of normalized match scores *o(.)* as defined in equation 4.

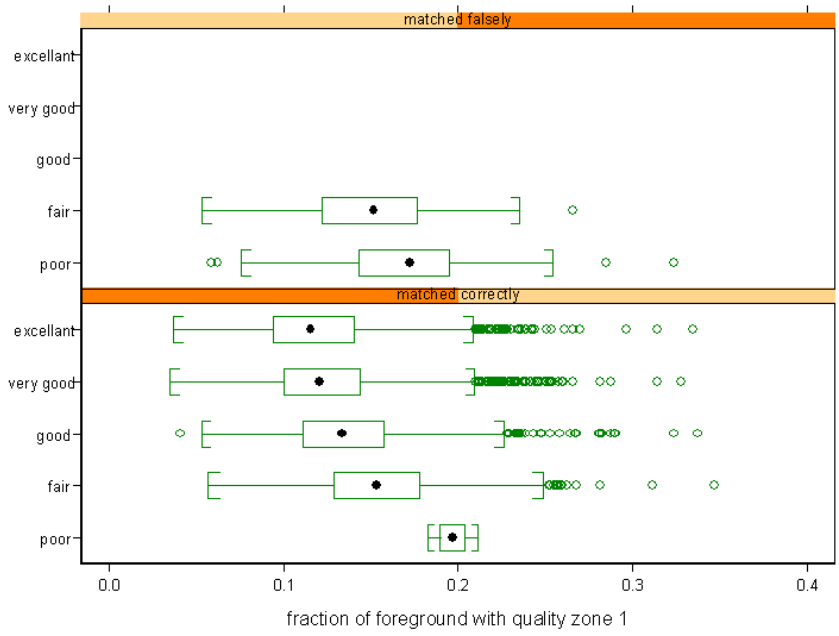

matched falsely
excellent
very good
good
fair
poor
matched correctly
excellent
very good
good
fair
poor
0
20
40
60
number of minutiae of quality=0.5

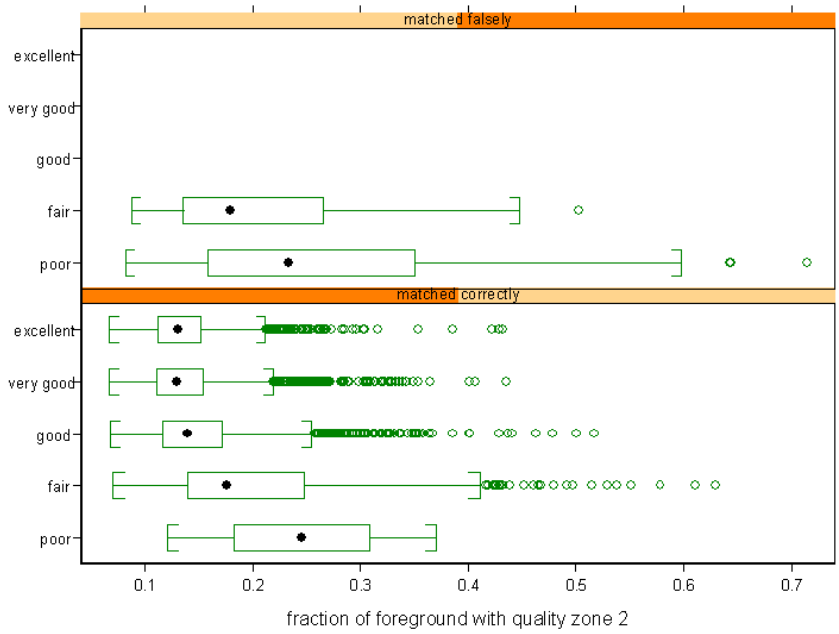

matched falsely
excellent
very good
good
fair
poor
matched correctly
excellent
very good
good
fair
poor
0
20
40
60
number of minutiae of quality=0.6

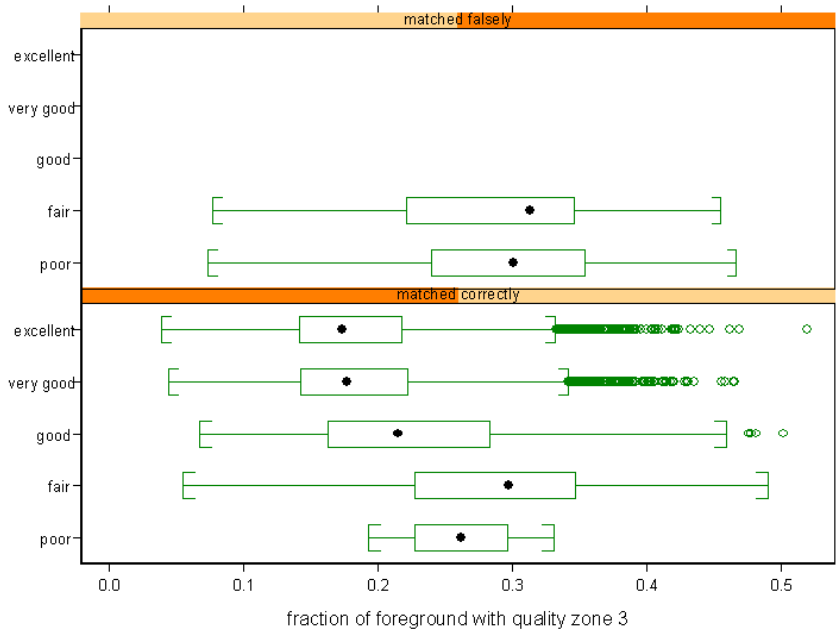

matched falsely
excellent
very good
good
fair
poor
matched correctly
excellent
very good
good
fair
poor
0
10
20
30
40
50
60
number of minutiae of quality=0.75


matched falsely
excellent
very good
good
fair
poor
matched correctly
excellent
very good
good
fair
poor
0
10
20
30
40
50
number of minutiae of quality=0.8


matched falsely
excellent
very good
good
fair
poor
matched correctly
excellent
very good
good
fair
poor
0
10
20
30
40
50
number of minutiae of quality=0.9


matched falsely
excellant
very good
good
fair
poor
matched correctly
excellant
very good
good
fair
poor
0.0
0.1
0.2
0.3
0.4
fraction of foreground with quality zone 1


matched falsely
excellent
very good
good
fair
poor
matched correctly
excellent
very good
good
fair
poor
0.1
0.2
0.3
0.4
0.5
0.6
0.7
fraction of foreground with quality zone 2


matched falsely
excellent
very good
good
fair
poor
matched correctly
excellent
very good
good
fair
poor
0.0
0.1
0.2
0.3
0.4
0.5
fraction of foreground with quality zone 3

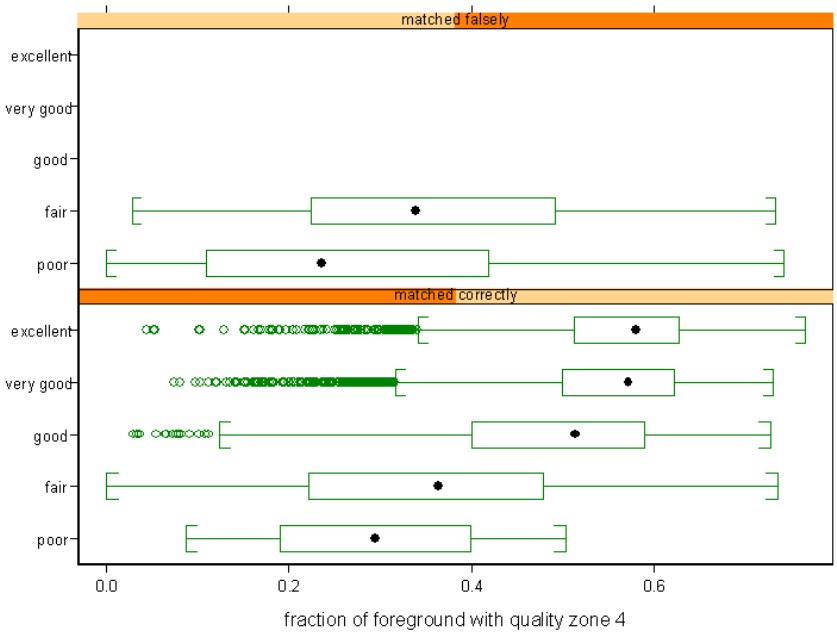


Figure 10. Box plots of each of our 9 features of dataset DOS-C Right index fingerprints for different bins of normalized match scores of vendor F. The median of each feature varies over different score bins. All samples in good, very good, and excellent bins are matched correctly.

We chose artificial neural network as the nonlinear classification method. The neural network has the capability of acting as an approximation function for an arbitrary nonlinear function and is not dependent on model based distribution functions in either the feature or the classification space.

4.2.1 Training the neural network

We used neural network implementation in NFIS package. The theory behind the machine learning techniques used in this program is discussed in [17]. It trains a 3-layer feed forward nonlinear perceptron model. The input nodes are feed-forwardly connected to the hidden nodes, and the hidden nodes to the output nodes. Training (optimization) of the weights is done using either a scaled conjugate gradient algorithm [18], or by starting out with scaled conjugate gradient and then switching to a limited memory Broyden Fletcher Goldfarb Shanno algorithm [19]. Boltzmann pruning [20], i.e. dynamic removal of connections, is performed during training.

In section 3.1 (Table 2) we explained our 5 classes of quality number. These 5 classes are our output of neural network. Input to neural network is 11-dimension feature vector discussed in section 4.1.4. We chose 22 hidden nodes. The activation functions used for hidden nodes as well as output nodes are sinusoid. Boltzmann pruning was set.

Our training set consists of plain right index, plain left index, plain right thumb, and plain left thumb fingerprints from datasets DOS-C, DHS2-C, DHS10, TXDPS, and BEN. As explained in section 1.2, these datasets contain fingerprints collected by different personnel, at different locales, and under different conditions. Therefore the training set contains various levels of quality. We have tested our system on VISIT_POE and VISIT_POE_BVA datasets and a subset of DOS-C, DHS2-C, DHS10, TXDPS, and BEN that *was* not used for training.

A full similarity matrix of the training set is needed for training the neural network. Similarity scores are needed to compute the output class of neural network (i.e. bins of normalized match scores). Since bin boundaries are based on distribution of match and non-match scores (Table 2), one may argue that training based on similarity scores of one vendor will result in "vendor dependent" weights and so the whole system may be biased and not generalize to other matchers. To avoid such a problem, we have chosen samples that are assigned the same bins by SDK vendors F, G, and VTB. Therefore, similarity scores of SDK vendors F, G, and VTB were used for our training.

Care was taken to design a training set balanced in terms of numbers of different finger positions of different quality of different datasets. Table 5 summarizes the diversity of our training set.

	RIGHT INDEX					LEFT INDEX					RIGHT THUMB					LEFT THUMB				
Quality	**1**	**2**	**3**	**4**	**5**	**1**	**2**	**3**	**4**	**5**	**1**	**2**	**3**	**4**	**5**	**1**	**2**	**3**	**4**	**5**
DOS_C	100	100	83	21	31	100	100	69	18	62										
DHS2_C	100	100	90	29	21	100	100	79	11	30										
BEN	40	40	40	40	26	40	40	40	40	40	60	60	60	60	7	60	60	60	60	15
TXDPS	40	40	40	38	35	40	40	40	36	34	60	60	60	43	14	60	60	60	31	18
DHS10	20	20	20	20	20	20	20	20	20	20	80	80	80	32	32	80	80	80	36	39
TOTAL	**300**	**300**	**273**	**148**	**133**	**300**	**300**	**248**	**125**	**186**	**200**	**200**	**200**	**135**	**53**	**200**	**200**	**200**	**127**	**72**

Table 5. Training set structure: Number of samples. Note the totals from each class are: 1000, 1000, 421, 535, and 444.

In an ideal case, we want our training set to have equal numbers of each fingers (right/left index/thumb) with equal numbers of different quality, e.g. 300 right index fingerprints of each quality 1, 2, 3, 4, and 5, and same for left index and right / left thumb. Plus it is preferred to have equal number of samples from each above-mentioned datasets. However, in reality only a small fraction of each dataset is of very poor quality, and so our training set has more samples of high and good quality than samples of low and very low quality. This is common in classification problems where large samples of rare classes may not be available. We have compensated for lower samples of poor quality in our training set by setting "class-weight"

parameter of our implementation of neural network [17]. Our implementation of neural network accepts class prior probabilities as a training parameter. In the scaled conjugate gradient method used here, both the network errors and the error signals used in the training are calculated using class weights. This ensures that the optimization is performed in a way that produces the best solution to the global problem but allows reasonable sampling of less common classes.

For each training sample, the class of neural net output is determined based on its match and non-match scores, which is computed for a pair of (probe, gallery). We know that the lower match score (and/or higher non-match mean) is due to the lower quality of the pair. However, prior to training, it is not obvious which of the (probe, gallery) fingerprint images is of the lower quality and so both probe and gallery fingerprint images are assigned the same class. If the quality of the probe and gallery are the same, samples are labeled correctly and so the neural network is trained correctly. However, if the quality of the probe and gallery are not the same, the better quality image of the pair (probe, gallery) is labeled with the wrong class. For example a probe image with very low number of good quality minutia and small quality zone 4 (indicators of low quality), scored against a gallery image with high number of good quality minutia and large quality zone 4 (indicators of good quality), will result in a low normalized match score (eq. 4). Both probe and gallery will be assigned the same class (which is low quality) and so the gallery image (the better quality image of the pair) is obviously labeled with the wrong class. We solved this problem by adjusting the "pattern-weights" parameter of our implementation of the neural network. The pattern-weights are used in the calculation of the error value that neural network attempts to minimize during training. When the training patterns (features) are sent through the network, each pattern produces an error contribution, which gets multiplied by the pattern-weight for that pattern before added to an error accumulator.

In our case, to assure that the neural network is trained properly, we must assign a lower weight to the better quality image of the pair (probe, gallery) for each subject. As discussed above, there is no knowledge of relative quality of the pair (probe, gallery) prior to training. However, we can use the predicted class of quality for each image *after* training to adjust pattern-weights and then repeat the training with adjusted pattern-weights. Therefore, we performed two rounds of training: First, we trained the neural network giving equal weights (all weights = 1) to all feature vectors. Upon completion of training, neural network predicts class of quality for each image in the training set. New pattern-weights are assigned to each image based on this predicted quality for the two images (probe, gallery) of each subject, and training is repeated using the new and adjusted pattern-weights. Pattern-weights are adjusted as explained below:

For each subject (person) in training set,
If $q_{probe} = q_{gallery}$, set probe's pattern-weight = 0.5 and gallery's pattern-weight = 0.5
If $q_{probe} < q_{gallery}$, set probe's pattern-weight = 0 and gallery's pattern-weight = 1
If $q_{probe} > q_{gallery}$, set probe's pattern-weight = 1 and gallery's pattern-weight = 0

Note that quality=5 is the poorest quality and quality=1 is the best quality.

To make sure that all feature vectors fed to the neural network are in the same range, feature vectors are normalized by vectors of global mean and global standard deviation of all feature vectors of datasets DOS-C, DHS2-C, BEN, DHS10, and TXDPS.

We have used similarity scores of vendor A → L, N, and VTB of the SDK test performed at NIST [6] to test our proposed fingerprint image quality metrics. Results of vendor F for dataset VISIT_POE have been reported in the body of this report, and results for all the other vendors for all other datasets (a total of 280 ROCs) are in appendix A. It should be noted that no subset of the VISIT_POE dataset has been used in training the neural network.

5 Assessing Fingerprint Image Quality

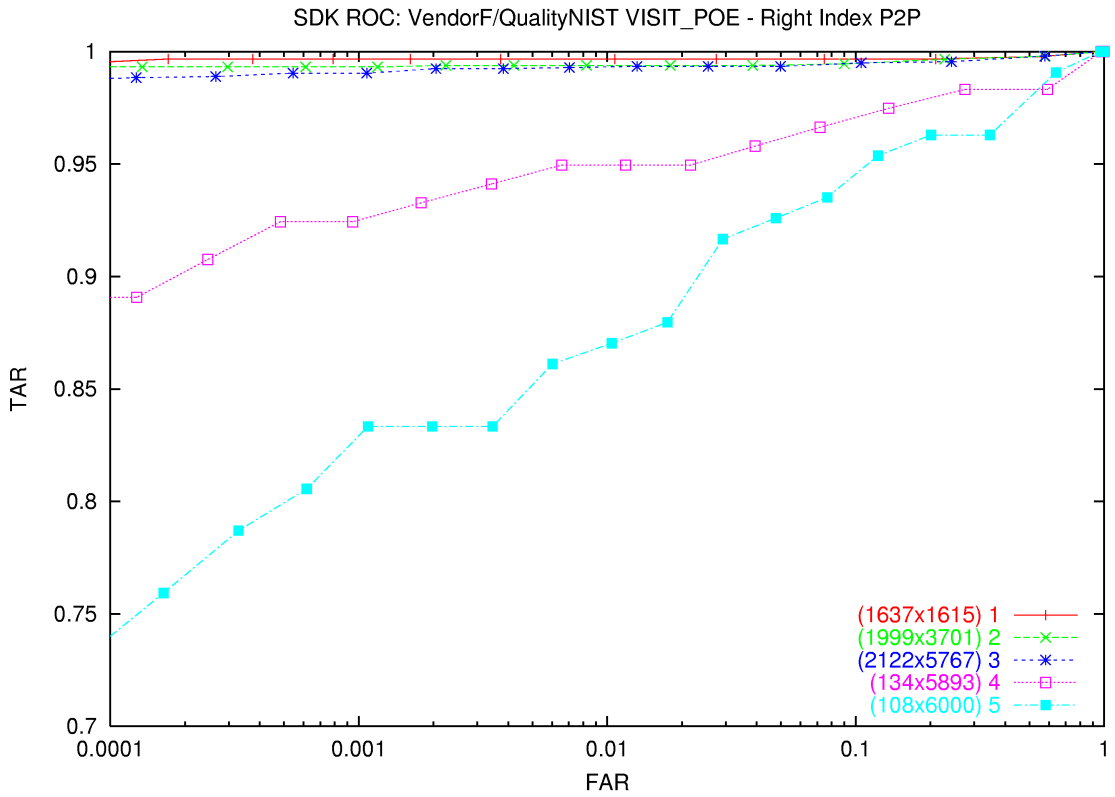
It is difficult to obtain the fingerprint image quality benchmark to test the performance of a quality analysis algorithm. The most used method to test image quality measures are based on visual (and thus subjective) assessments of images. However, with growing size of databases, manual judgment is extremely time-consuming and expensive. The present complete NIST fingerprint test archive consists of approximately 68 million fingerprints from 16 million individuals. Thus, objective measures that correlate acceptably well with perceived image quality and/or matcher performance are essential.

Our evaluation criterion is rank ROC as a function of image quality. We measure quality number q for probe and gallery fingerprint pairs of each person in our test datasets, and quality is defined by the minimum of the two numbers. Given a quality number taking on K integral values, K ROC characteristics are generated. If the k-th quality number is q_k, then the similarity scores used in the computation of the K non-match distributions $N_k(s)$ and K match distributions $M_k(s)$ are those for which the image pair has quality q_k by equations 6 and 8. The selection of number K is a design issue. At minimum K=2, that is a binary decision to accept the image as good quality or reject the image as poor quality is made. We have chosen K = 5. The choice of 5 levels of image quality is a compromise. Eight to ten levels would be needed to fully characterize a matcher that is very sensitive to image quality, such as SDK matcher A [6]. Matchers that are insensitive to image quality, such as SDK matchers H and I, could be well characterized by three levels of image quality. We have selected five levels which is the correct level for a medium to good quality matcher such as SDK matcher F.

Figures 11-a and 11-b show rank ROC performance as a function of our quality analysis for right and left index fingers of dataset VISIT_POE and similarity scores of SDK vendor F. These figures show that our proposed image quality works as a rank statistic for performance. It is worth mentioning that no subset of the VISIT_POE data has been used in training the neural network.

Each of the figures showing results, (Figures 11-24), consists of three sections. The first section of the figures contains a ranked set of ROC curves for image quality levels 1-5. To study the difference in performance for different levels of quality, it is necessary to combine results over the set of 5 ROCs (one for each class of quality). One method of combining results is to measure the variation of the TAR for each FAR. This requires readjusting the operating threshold for each probe set (i.e. different levels of quality). For many applications, this is not feasible or desirable. However, it is an appropriate technique for comparing performance of

different systems because it is not possible to set uniform thresholds across different systems. To study the effect of quality on the performance of a system, we have to set one operating threshold across all galleries and probe sets. Fixing the threshold across various levels of quality corresponds to an operational situation in which threshold is set once for an application. The operating-threshold is picked based upon a specific false accept rate (usually 1%) on an aggregated population. The second section of each figure contains a table showing operating-threshold and its corresponding FAR and TAR values for the aggregated population as well as FAR and TAR values for each image quality level computed at operating-threshold. The third section of each figure contains a plot of the values given in the second section. Specifically, it show false accept rate and true accept rate of probe sets of different quality at the fixed operating-threshold. The red square marks the false accept rate and true accept rate of the whole dataset at the fixed operating-threshold.



Vendor F – VISIT_POE – Right index threshold=350 (far,tar)=(0.0119418,0.991333)					
quality	**1(excellent)**	**2(very good)**	**3(good)**	**4(fair)**	**5(poor)**
FAR	0.0037	0.0083	0.0131	0.0216	0.0477
TAR	0.997	0.994	0.993	0.9496	0.926

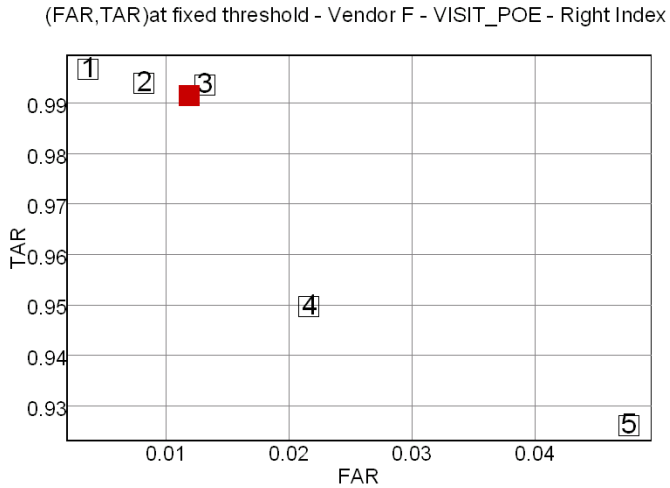
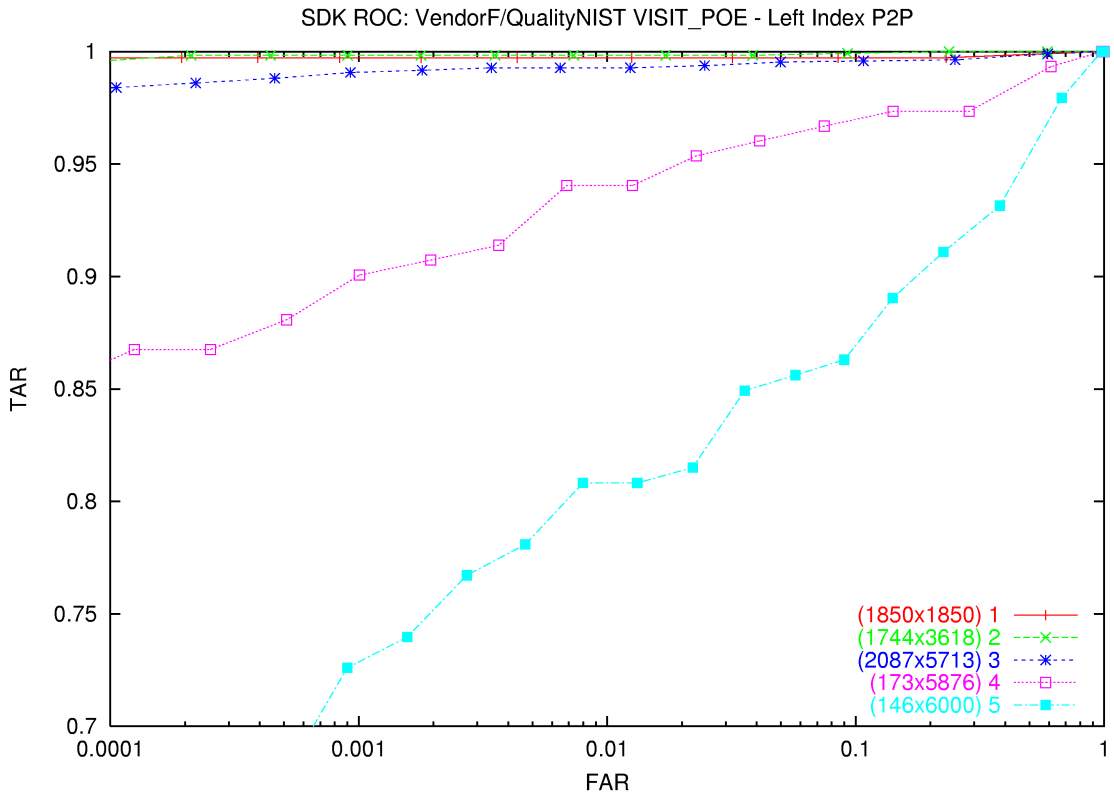


Figure 11-a. The effect of quality on ROC - Vendor F - Dataset VISIT_POE - Right index



VendorF –VISIT_POE Left Index threshold=350 (far,tar)=(0.0120981,0.989)					
quality	**1(excellent)**	**2(very good)**	**3(good)**	**4(fair)**	**5(poor)**
FAR	0.0043	0.0073	0.0124	0.0228	0.0573
TAR	0.997	0.998	0.993	0.953	0.856

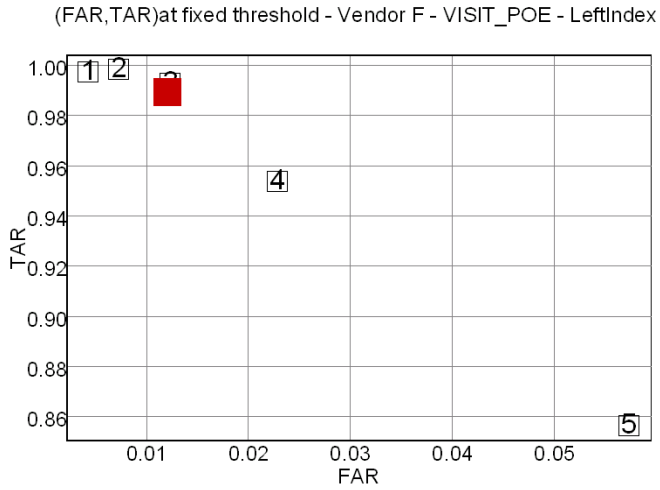


Figure 11-b. The effect of quality on ROC - Vendor F - Dataset VISIT_POE - Left index

A desirable feature of a quality measure is how well it generalizes to other matchers and other datasets. To investigate how well our system generalizes to other matchers, we have used similarity scores of SDK vendors A → L, N, and VTB to test our proposed fingerprint image quality metrics on all datasets explained in 1.2. Results of dataset VISIT_POE are shown in Figures 11 through 24, where results of all SDK vendors (14 vendors) and all datasets (20 datasets) - - a total of 280 ROCs -- are in appendix A. These ROCs show our fingerprint quality measure generalizes very well to other matchers and other datasets.

The table and plot of (false accept rate, true accept rate) pair for each quality at the same operating point, i.e. fixed threshold, follow each ROC. The threshold and (false accept rate, true accept rate) of the corresponding dataset is reported as well. The red square marks the false accept rate and true accept rate of the whole dataset at the operation-threshold specified in the table in the center section of each figure. As quality degrades, we expect lower performance, i.e. lower true accept rate and/or higher false accept rate. Figures 11 through 24 show for vendors F, C, G, I, and J the false accept rate and true accept rate of quality 1 and 2 (excellent and very good) fingerprints are almost always on the left side of the red square. It means matcher performance for probe sets with quality 1 (excellent) and quality 2 (very good) is better than the whole dataset. False accept rates and true accept rates of quality 3, 4, 5 (good, fair, and poor) are usually on the right side of the red square, which indicates lower true accept rate AND higher false accept rate than the whole dataset.

As quality degrades, true accept rate decreases for all the matchers. It is also desirable to see an increase in FAR as quality improves. Vendors F, C, G, J, and I exhibit such a behavior: as quality degrades, a lower TAR and a higher FAR is observed. Vendors A, B, N, and VTB's false accept rate decreases as quality degrades. There is no pattern for vendors D, E, H, K, and L's far as quality varies. Increasing false accept rate as quality degrades indicates vulnerability of low quality fingerprint images.

There are several interesting trends in vendor matcher performance sensitivity to image quality as defined in this paper. When all vendors are examined on quality level one data, only a few of them can get TARs over 99.6%. This suggests that there are error sources that generate approximately 0.4% errors that are independent of image quality as defined in this paper.

At the same time the FAR of different vendors at the operating-threshold ranges from 2.8% for vendor N to 0.02% for vendor I. This shows that the FAR of systems with relatively low accuracy rank in the SDK tests is not easily correlated with the TAR of the system.

In most cases, where the FAR of a vendor for quality 1 fingerprints is below 0.1%, the vendor has very uniform response to fingerprints of quality 1, 2, and 3. This demonstrates that very high accuracy systems need fewer than five levels of image quality to fully characterize their performance.

Finally, examination of the results of the SDK report shows that for the top five systems, SDK systems H, I, J, F, and G, the image quality 3 TAR on VISIT_POE data is a good overall predictor of SDK rank. The ranks of the various SDK vendor systems on different datasets are given in Table 7 taken from [6]. The ranks of the top five vendors, based on quality level 3 TAR at the fixed operating-threshold, are given in Table 8. The ranks between the DOS-C and the

DHS2-C index finger data set and the quality 2 TARs differ only in the placement of systems H and F. This difference between this ranking and the other SDK datasets is in part due to the flat to rolled matching tested by BEN, DHS10, TXDPS, OHIO, and SD29 datasets.

DOS-C/DHS2-C	BEN/DHS10/TXDPS	ALL
System I	System H	System H
System H	System I	System I
System F	System J	System J
System J	System F	System F
System G	System G	System G
System D	System D	System D
System C	System K	System C
System K	System C	System K
System L	System L	System L
System B	System VTB	System A
System VTB	System A	System VTB
System A	System B	System B
System E	System E	System E

Table 6. SDK rankings at FAR 0.01% for the different datasets taken from [6]

QUALITY3 LEFT INDEX RANK	VISIT_POE RIGHT INDEX TAR	VISIT_POE LEFT INDEX TAR
System I	0.993	0.996
System F	0.993	0.993
System H	0.993	0.995
System J	0.986	0.990
System G	0.984	0.988

Table 7. Image quality rankings for image quality 3 TAR using the left index fingerprints from US_VISIT_POE datasets for different vendors. The threshold and TAR values are taken from figure 11 for system F and from figures 17- 20 for other systems.

Distribution of qualities in the VISIT_POE dataset is shown in Table 8. Distribution of datasets DOS-C, DHS2-C, DHS10, BEN, TXDPS, and VISIT_POE_BVA are in appendix A.

	Quality	#of sample in test set	%of total
RIGHT INDEX	1 (excellent)	1637	27.28
	2 (very good)	1999	33.32
	3 (good)	2122	35.37
	4 (fair)	134	2.23
	5 (poor)	108	1.8
LEFT INDEX	1 (excellent)	1850	30.83
	2 (very good)	1744	29.07
	3 (good)	2087	34.78
	4 (fair)	173	2.88
	5 (poor)	146	2.43

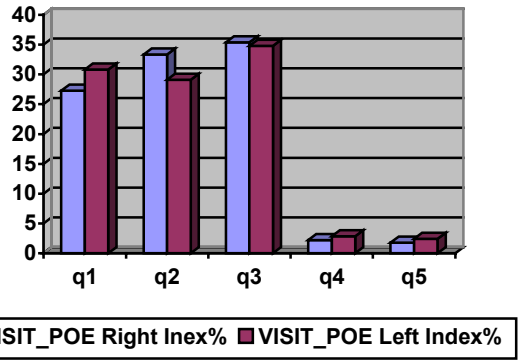
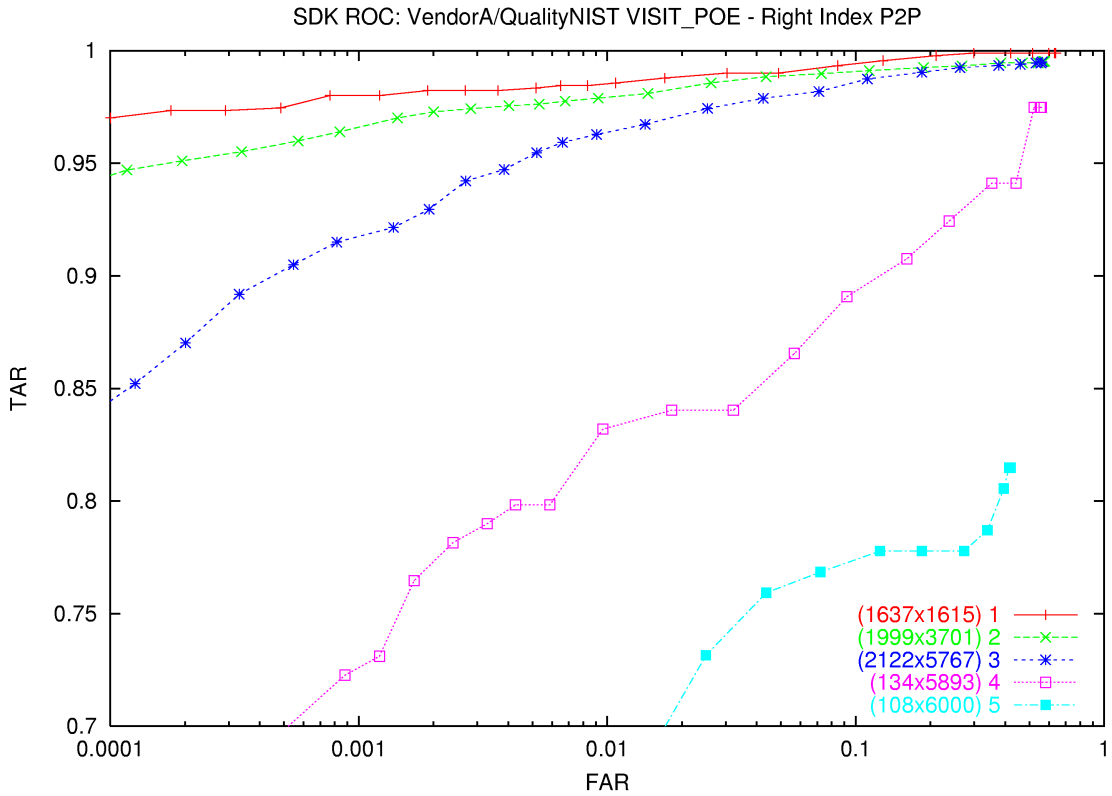


Table 8. Percentage of various quality levels for dataset VISIT_POE



Vendor A– VISIT_POE – Right index threshold=0.24 (far,tar)=(0.0144008,0.962833)					
Quality	**1(excellent)**	**2(very good)**	**3(good)**	**4(fair)**	**5(poor)**
FAR	0.0170	0.0146	0.0142	0.0096	0.0081
TAR	0.988	0.981	0.967	0.832	0.620

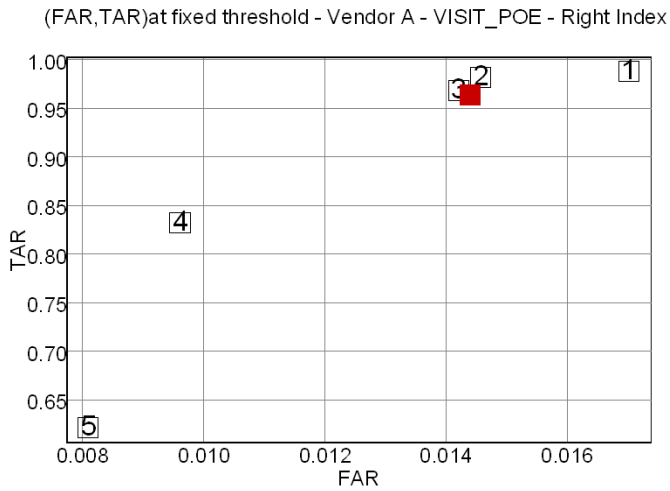
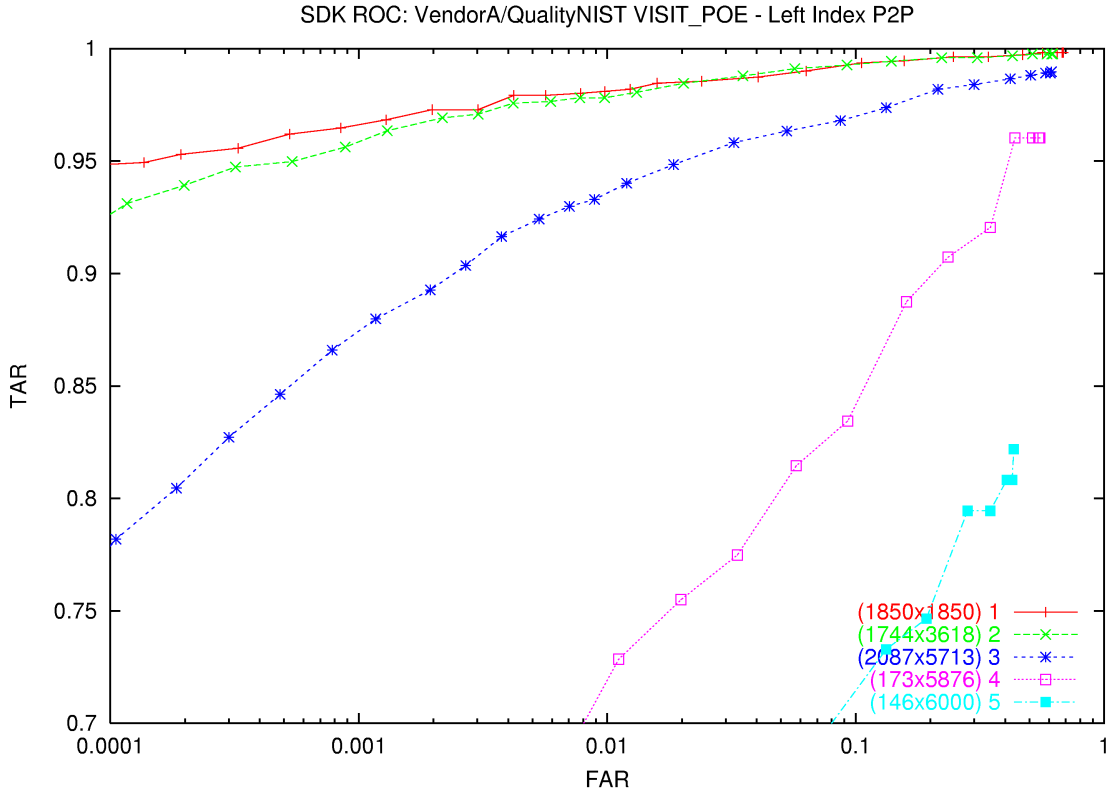


Figure 12-a. The effect of quality on ROC - Vendor A - Dataset VISIT_POE - Right index



Vendor A – VISIT_POE – Left index threshold=0.26 (far,tar)=(0.0126554,0.9375)					
Quality	**1(excellent)**	**2(very good)**	**3(good)**	**4(fair)**	**5(poor)**
FAR	0.0159	0.013	0.012	0.0071	0.0059
TAR	0.985	0.981	0.940	0.689	0.534

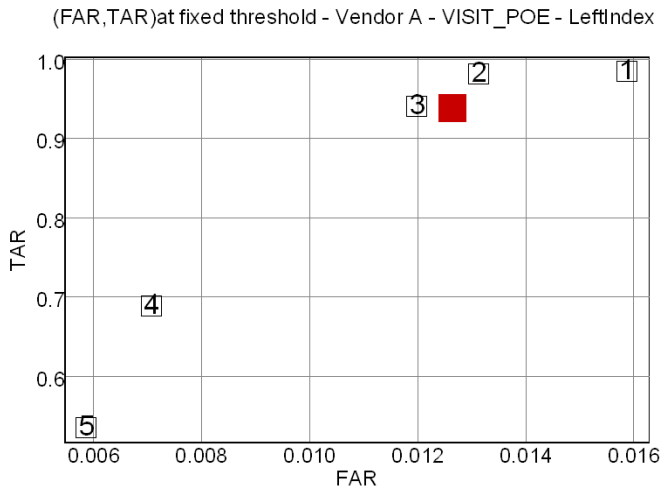
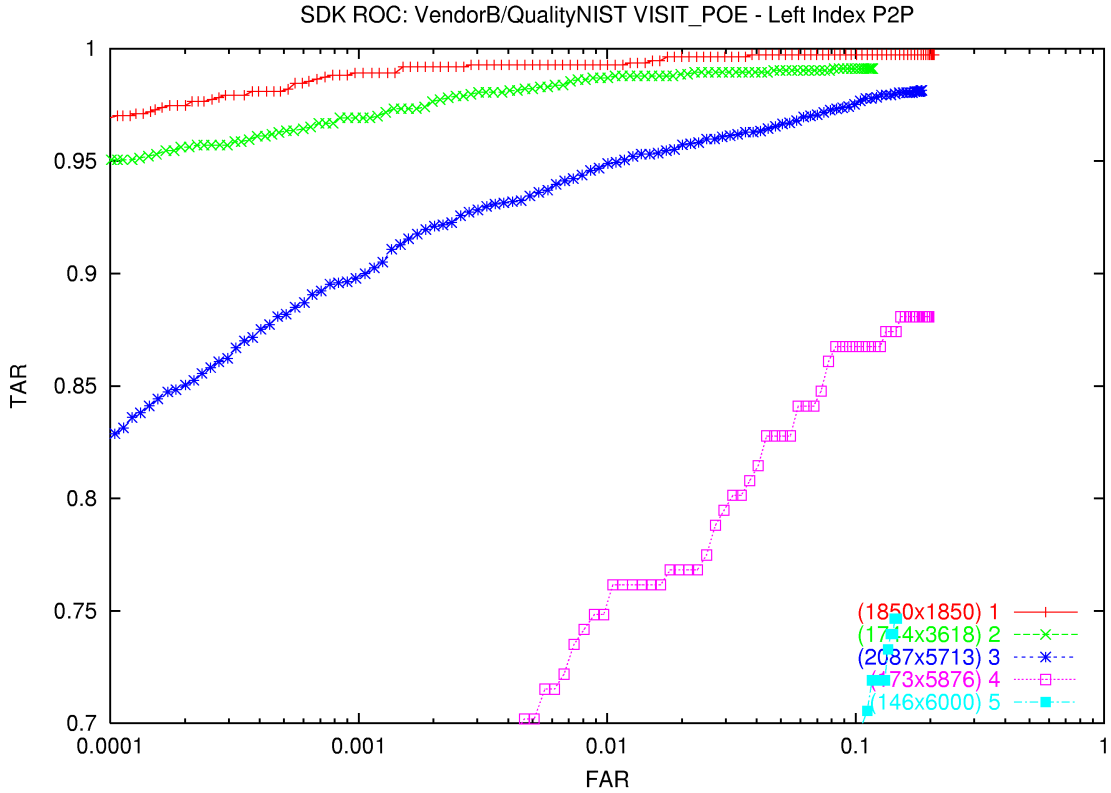


Figure 12-b. The effect of quality on ROC - Vendor A - Dataset VISIT_POE - Left index



Vendor B – VISIT_POE – Left index threshold=124 (far,tar)=(0.0102684,0.941667)					
Quality	**1(excellent)**	**2(very good)**	**3(good)**	**4(fair)**	**5(poor)**
FAR	0.0187	0.0088	0.0100	0.0067	0.0030

TAR	0.996	0.987	0.949	0.722	0.493

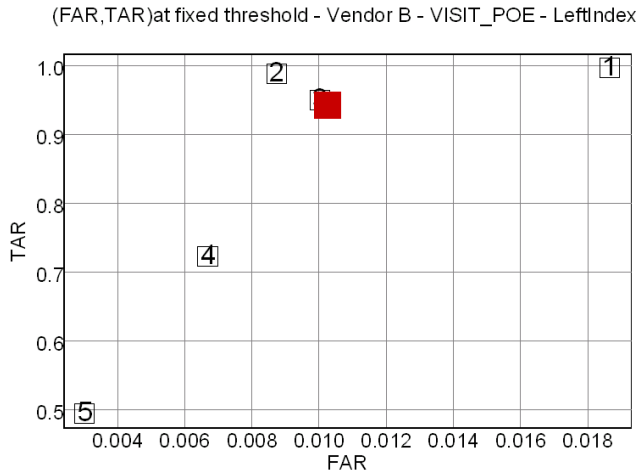
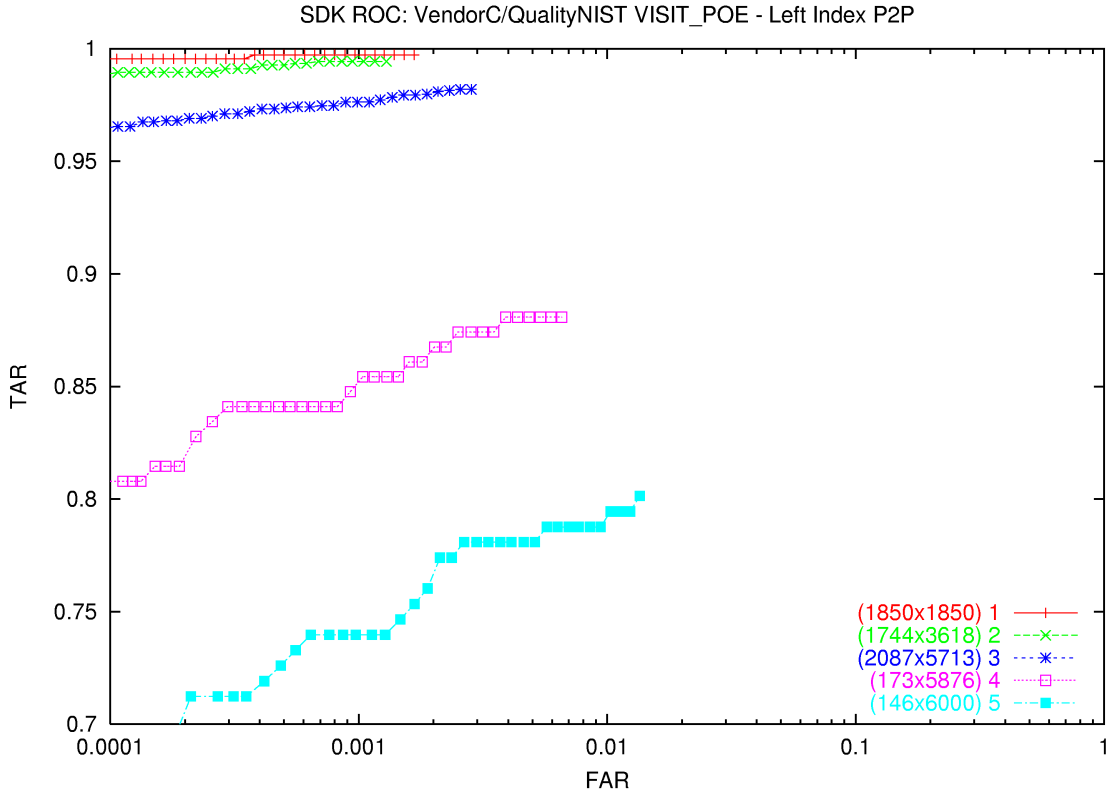


Figure 13-a. The effect of quality on ROC - Vendor B - Dataset VISIT_POE Left index



Vendor C – VISIT_POE – Left index threshold=110 (far,tar)=(0.001,0.975)					
Quality	**1(excellent)**	**2(very good)**	**3(good)**	**4(fair)**	**5(poor)**
FAR	0.00066	0.00045	0.00099	0.00225	0.00511

TAR	0.997	0.993	0.976	0.867	0.781

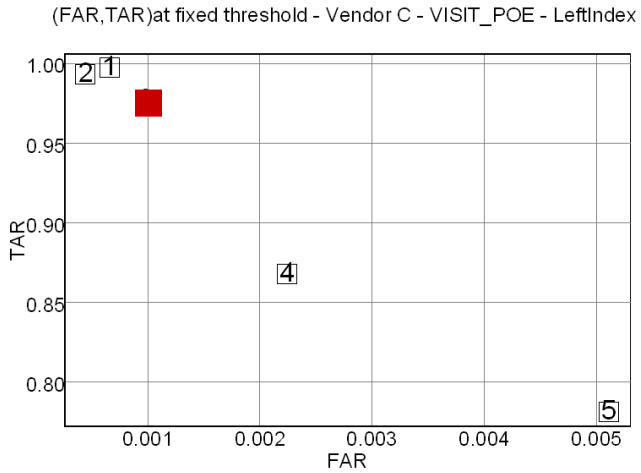
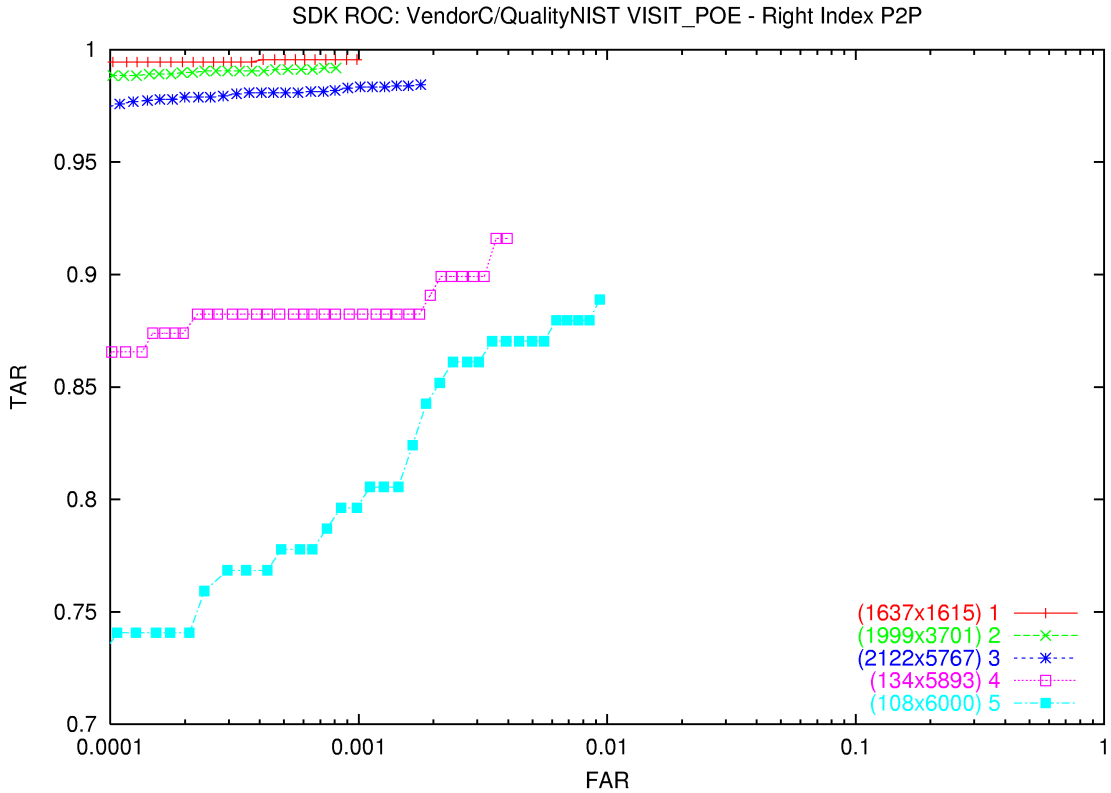


Figure 14-a. The effect of quality on ROC - Vendor C - Dataset VISIT_POE - Left index



Vendor C – VISIT_POE – Right index threshold=50 (far,tar)=(0.001,0.982)					
Quality	**1(excellent)**	**2(very good)**	**3(good)**	**4(fair)**	**5(poor)**
FAR	0.00067	0.00052	0.00114	0.00261	0.00624

TAR	0.995	0.991	0.983	0.899	0.880

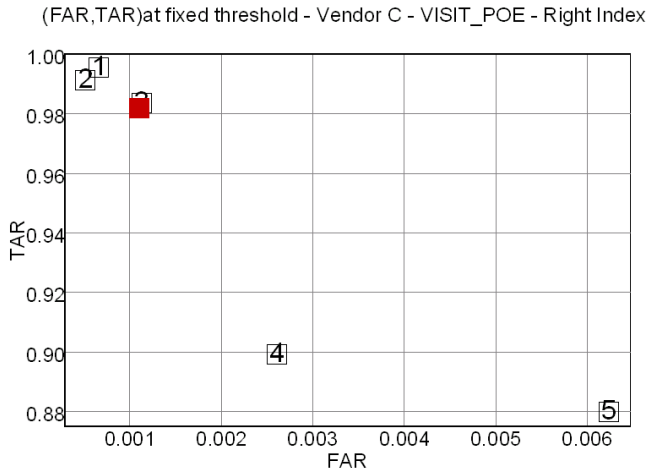
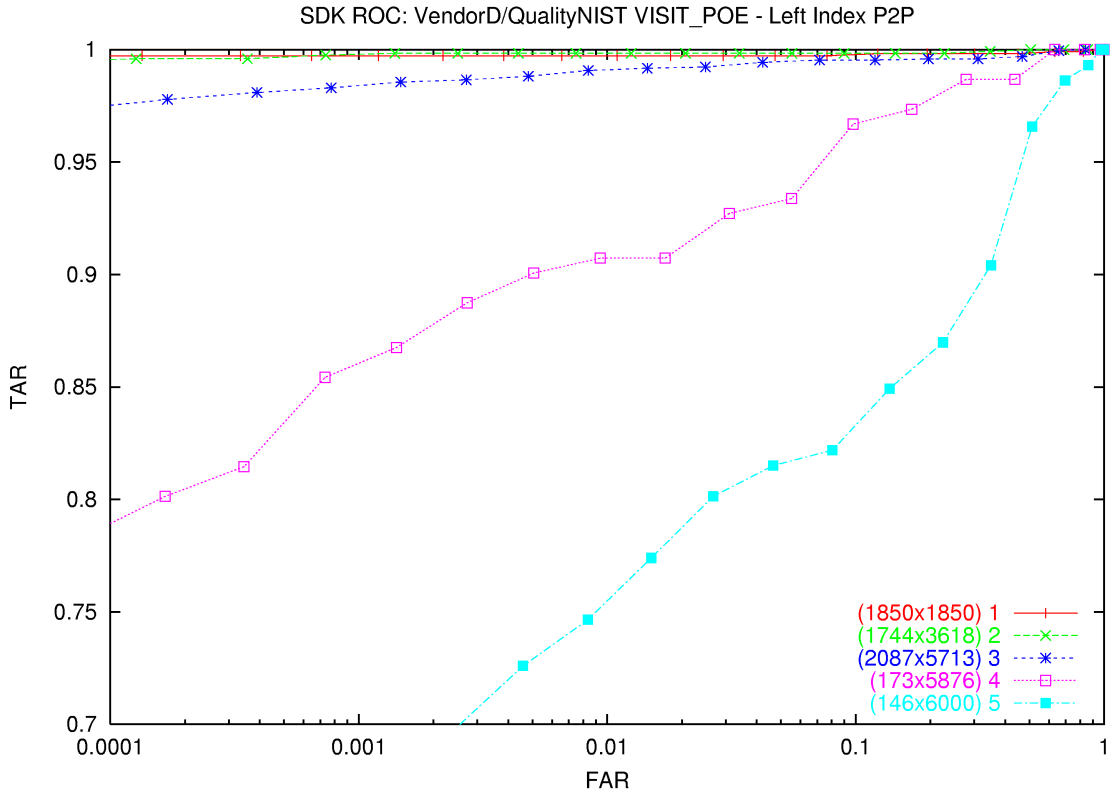


Figure 14-b. The effect of quality on ROC - Vendor C - Dataset VISIT_POE -Right index



Vendor D – VISIT_POE – Left index threshold=600 (far,tar)=(0.016,0.984)					
Quality	**1(excellent)**	**2(very good)**	**3(good)**	**4(fair)**	**5(poor)**
FAR	0.0179	0.0206	0.0145	0.0094	0.0150
TAR	0.997	0.998	0.992	0.907	0.774

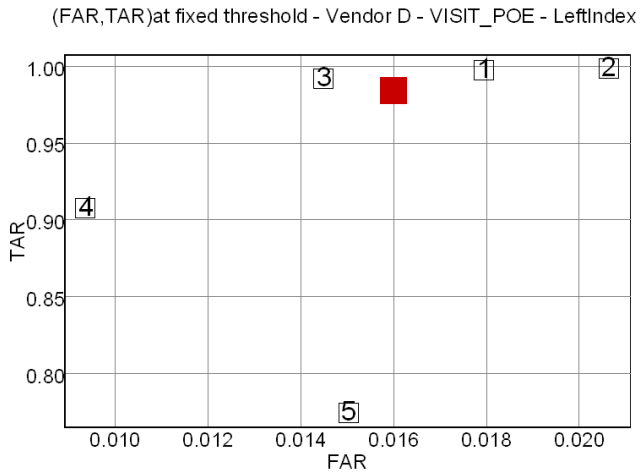
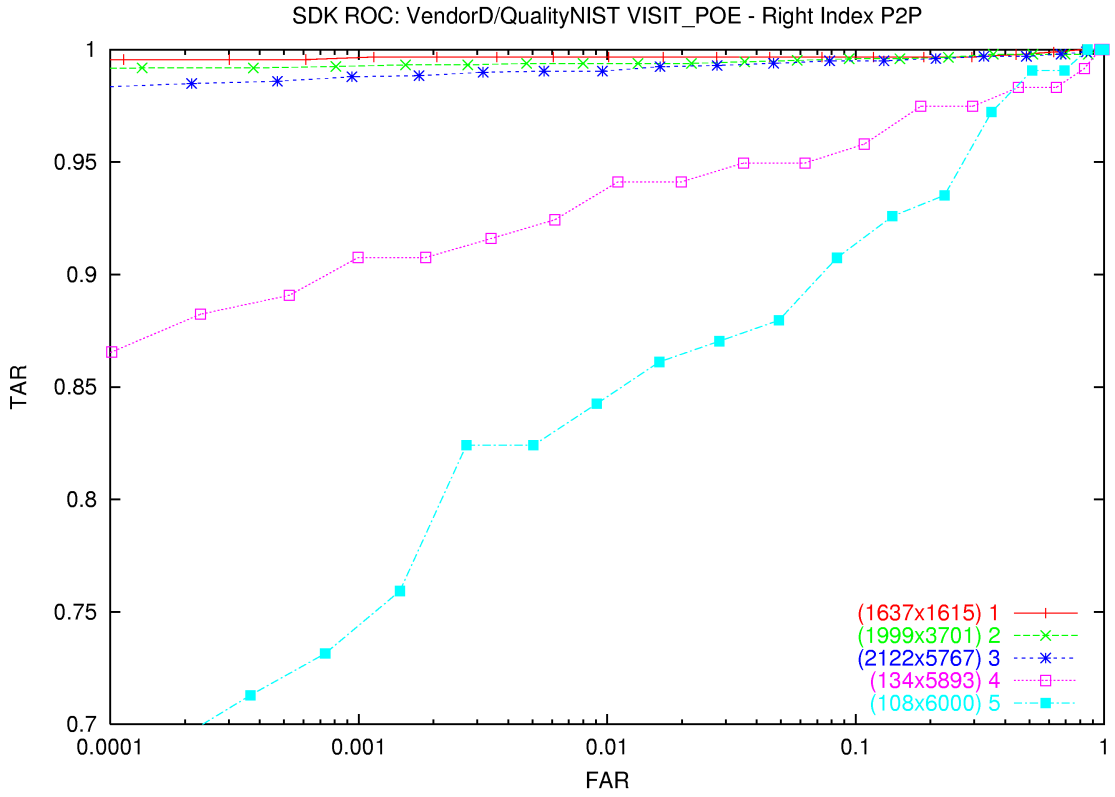


Figure 15-a. The effect of quality on ROC - Vendor D - Dataset VISIT_POE - Left index



Vendor D – VISIT_POE – Right index threshold=650 (far,tar)=(0.01,0.986)					
Quality	**1(excellent)**	**2(very good)**	**3(good)**	**4(fair)**	**5(poor)**
FAR	0.010	0.013	0.0096	0.006	0.0091
TAR	0.997	0.994	0.990	0.924	0.8426

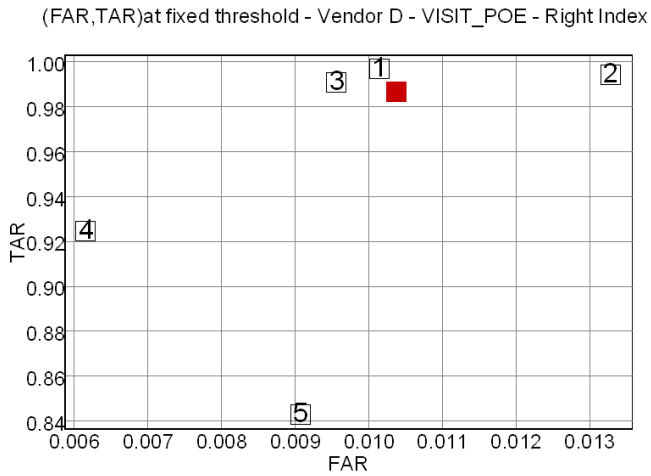
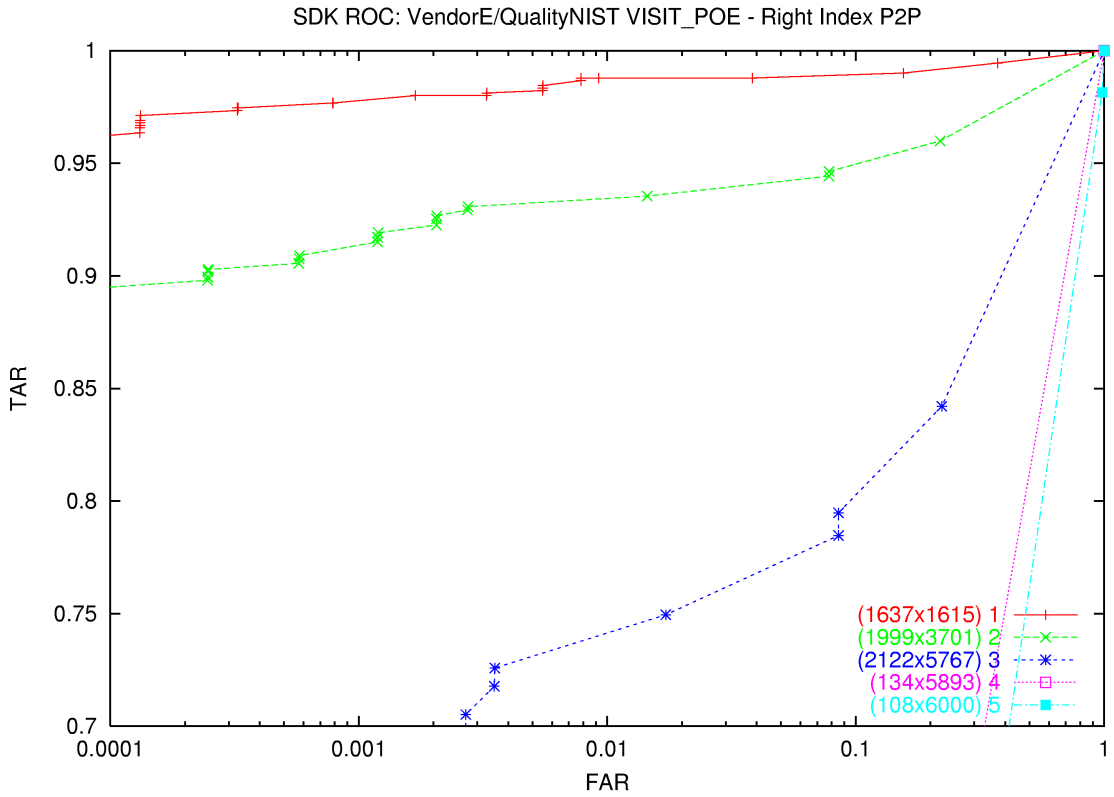


Figure 15-b. The effect of quality on ROC - Vendor D - Dataset VISIT_POE - Right index



Vendor E – VISIT_POE – Right index threshold=60 (far,tar)=(0.018,0.828)					
Quality	**1(excellent)**	**2(very good)**	**3(good)**	**4(fair)**	**5(poor)**
FAR	0.0384	0.0145	0.0172	0.0206	0.0059
TAR	0.988	0.935	0.749	0.403	0.194

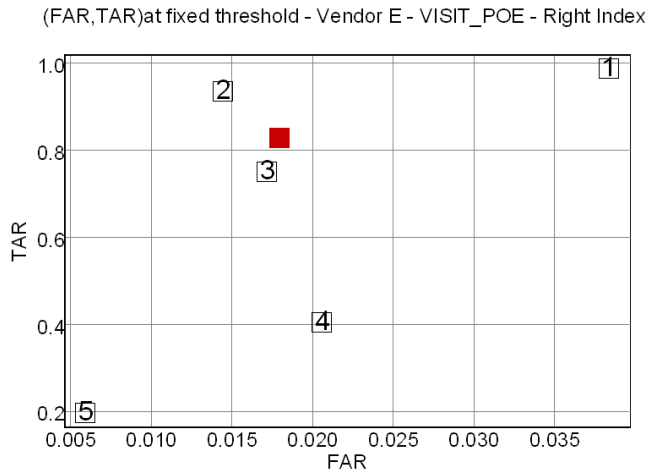
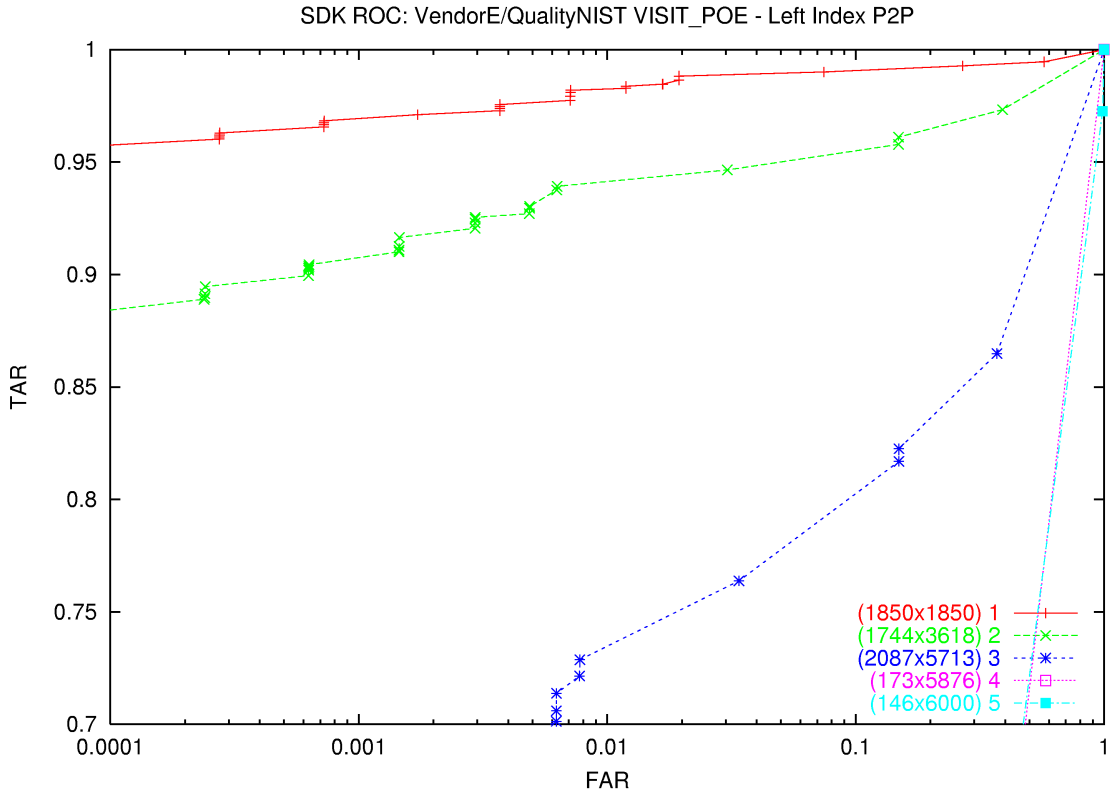


Figure 16-a. The effect of quality on ROC - Vendor E - Dataset VISIT_POE - Right index



Vendor E – VISIT_POE – Left index threshold=60 (far,tar)=(0.037,0.829)					
Quality	**1(excellent)**	**2(very good)**	**3(good)**	**4(fair)**	**5(poor)**
FAR	0.075	0.030	0.034	0.031	0.01
TAR	0.990	0.946	0.764	0.298	0.164

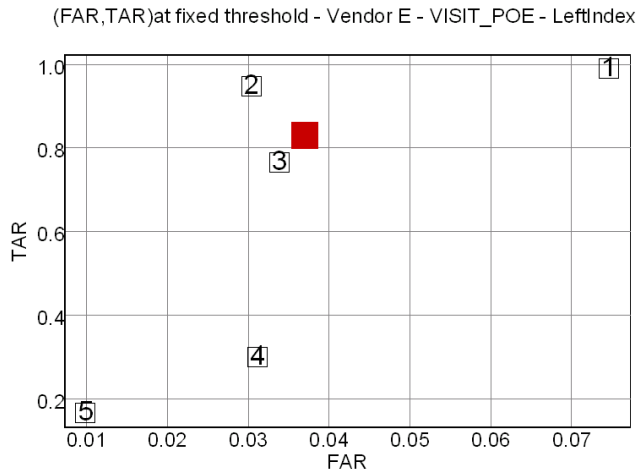
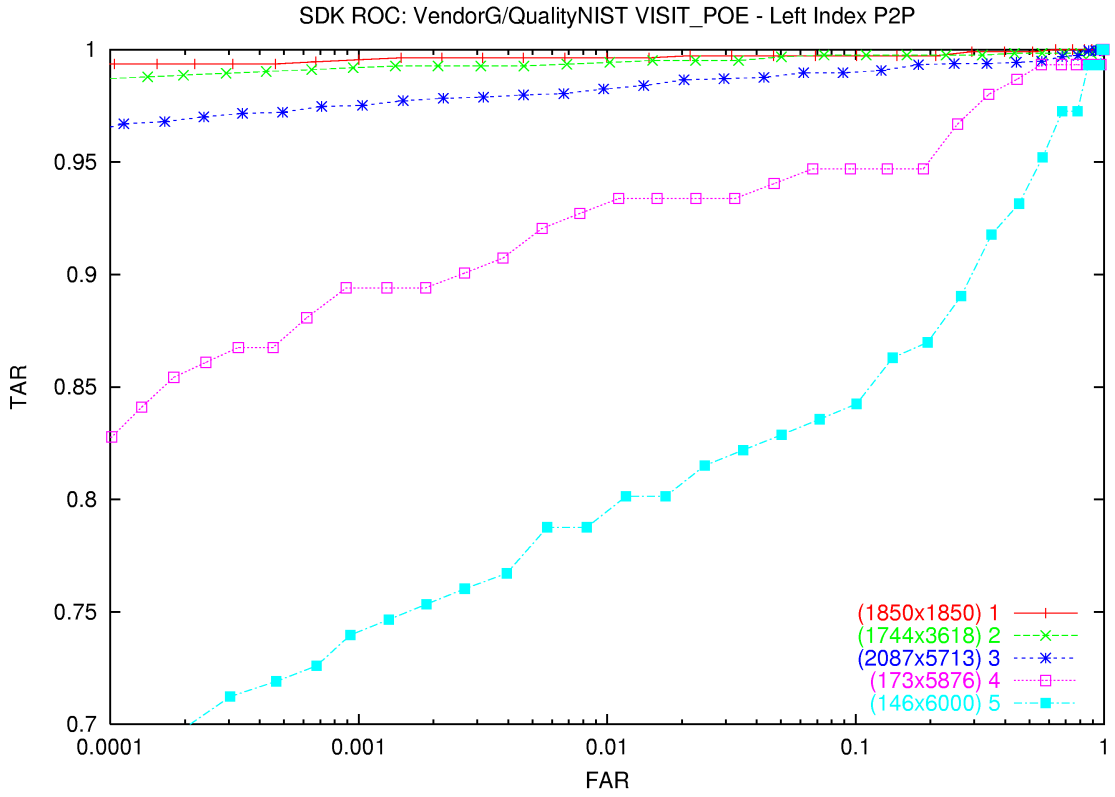


Figure 16-b. The effect of quality on ROC - Vendor E - Dataset VISIT_POE - Left index



Vendor G – VISIT_POE – Left index threshold=2700 (far,tar)=(0.012,0.980)					
Quality	**1(excellent)**	**2(very good)**	**3(good)**	**4(fair)**	**5(poor)**
FAR	0.0099	0.0102	0.0140	0.0159	0.0172
TAR	0.996	0.994	0.984	0.934	0.801

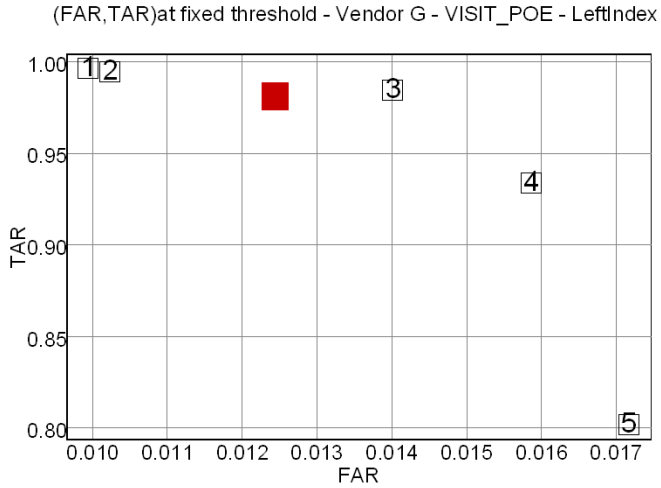
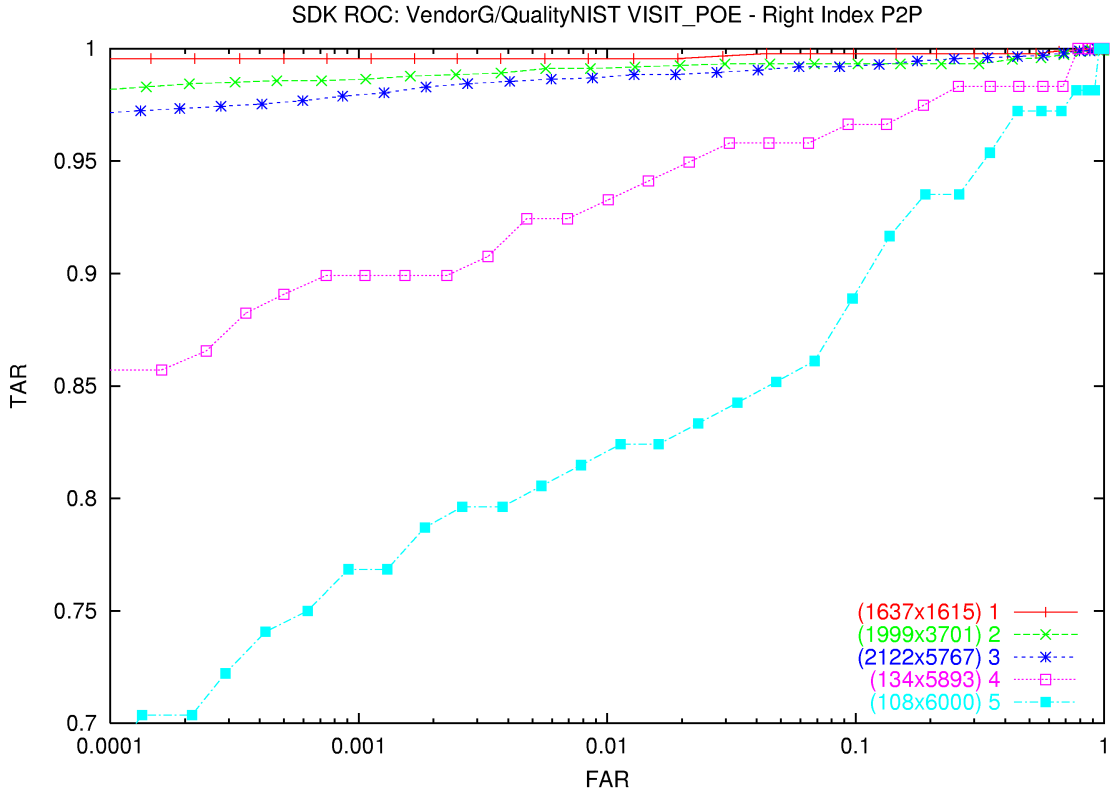


Figure 17-a. The effect of quality on ROC - Vendor G - Dataset VISIT_POE - Left index



Vendor G – VISIT_POE – Right index threshold=2700 (far,tar)=(0.011,0.985)					
Quality	**1(excellent)**	**2(very good)**	**3(good)**	**4(fair)**	**5(poor)**
FAR	0.0084	0.0086	0.0128	0.0147	0.0161
TAR	0.996	0.991	0.988	0.941	0.824

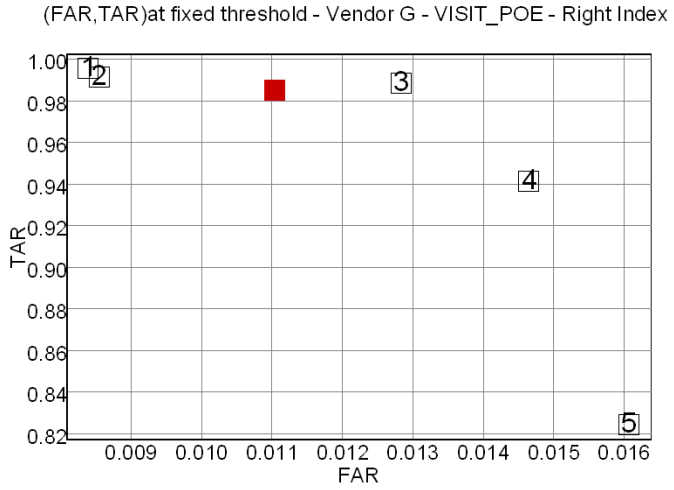
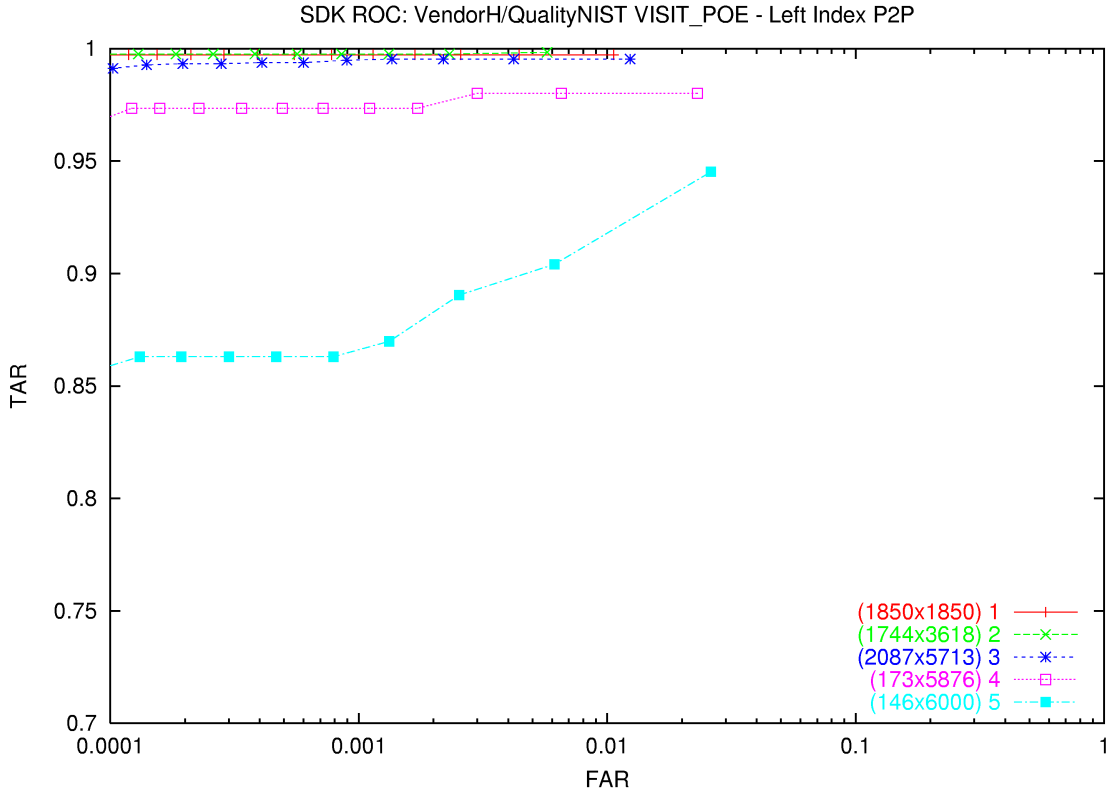


Figure 17-b. The effect of quality on ROC - Vendor G - Dataset VISIT_POE - Right index



Vendor H – VISIT_POE – Left index threshold=40 (far,tar)=(0.001,0.991)					
Quality	**1(excellent)**	**2(very good)**	**3(good)**	**4(fair)**	**5(poor)**
FAR	0.0017	0.0008	0.0014	0.0017	0.0013
TAR	0.997	0.998	0.995	0.974	0.870

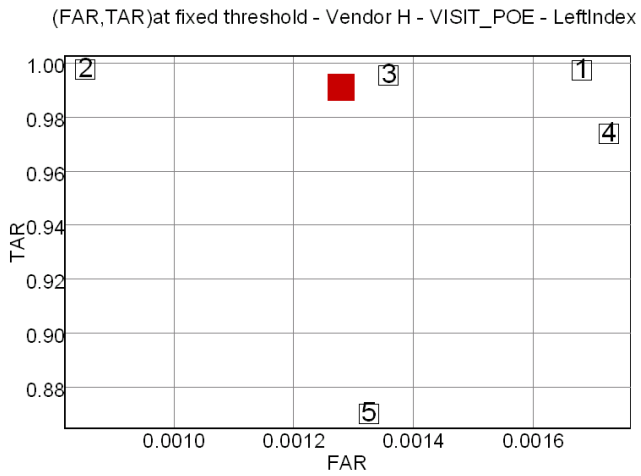
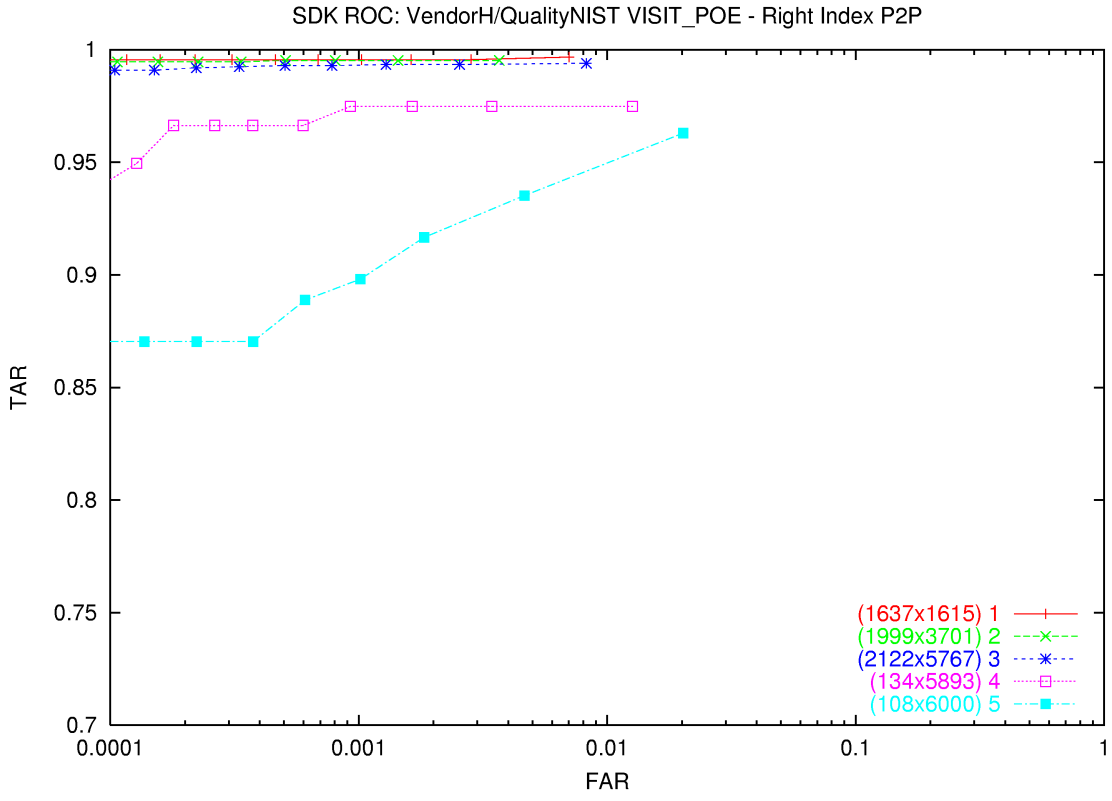


Figure 18-a. The effect of quality on ROC - Vendor H - Dataset VISIT_POE - Left index



Vendor H – VISIT_POE – Right index threshold=30 (far,tar)=(0.001,0.992)					
Quality	**1(excellent)**	**2(very good)**	**3(good)**	**4(fair)**	**5(poor)**
FAR	0.0016	0.00080	0.00129	0.00164	0.00183
TAR	0.996	0.995	0.993	0.975	0.917

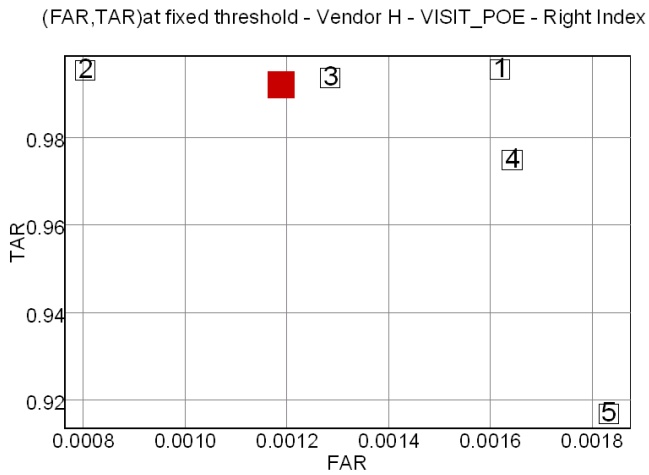
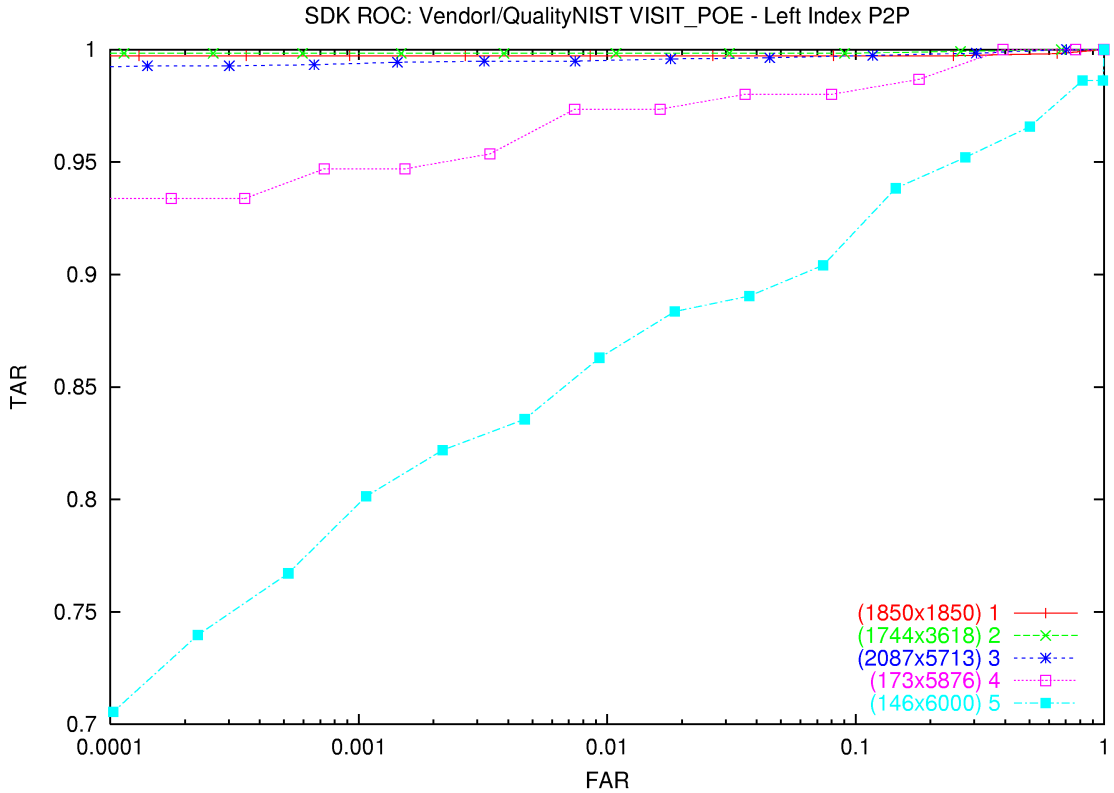


Figure 18-b. The effect of quality on ROC - Vendor H - Dataset VISIT_POE - Right index



Vendor I – VISIT_POE – Left index threshold=400 (far,tar)=(0.001,0.992)					
Quality	**1(excellent)**	**2(very good)**	**3(good)**	**4(fair)**	**5(poor)**
FAR	0.00058	0.00101	0.00191	0.00383	0.01129
TAR	0.998	0.997	0.996	0.972	0.880

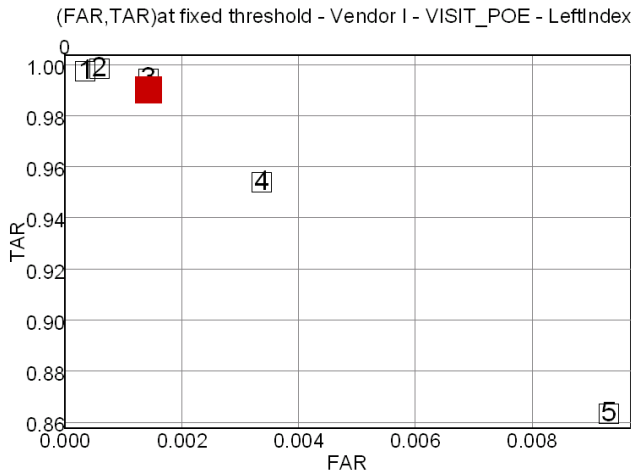
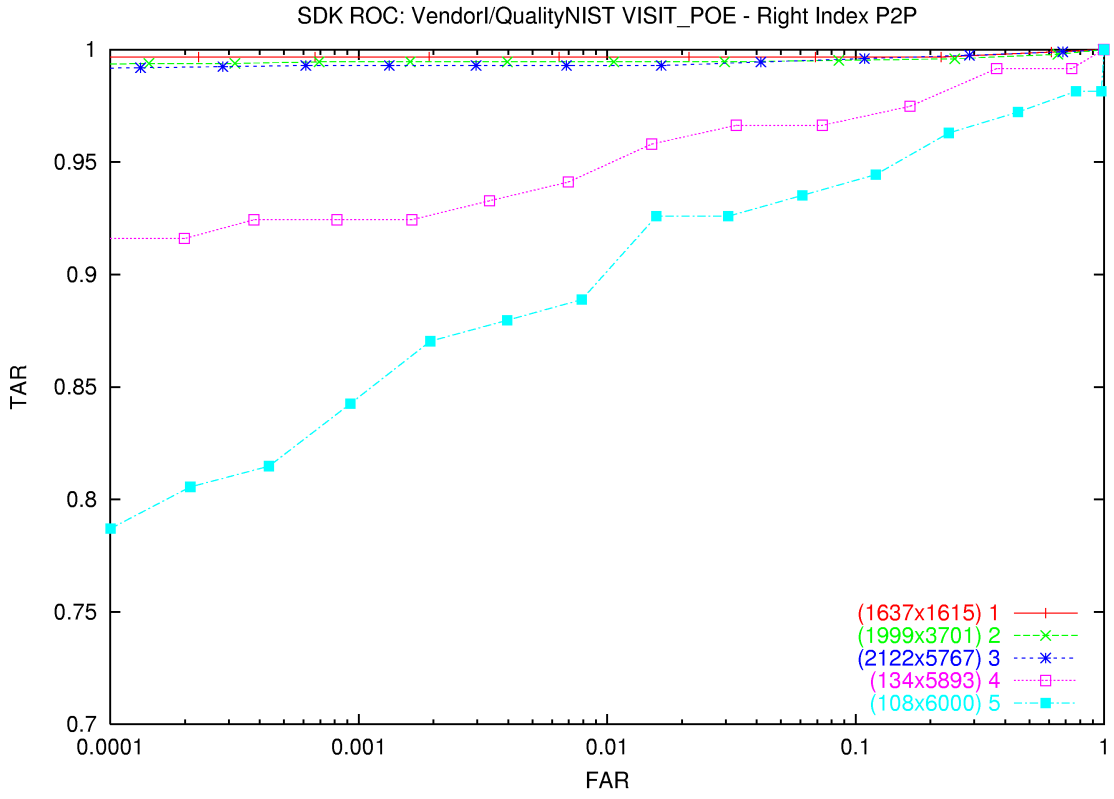


Figure 19-a. The effect of quality on ROC - Vendor I - Dataset VISIT_POE - Left index



Vendor I – VISIT_POE – Right index threshold=450 (far,tar)=(0.001,0.99)					
Quality	**1(excellent)**	**2(very good)**	**3(good)**	**4(fair)**	**5(poor)**
FAR	0.00022	0.00069	0.00133	0.00336	0.00789
TAR	0.997	0.995	0.993	0.933	0.889

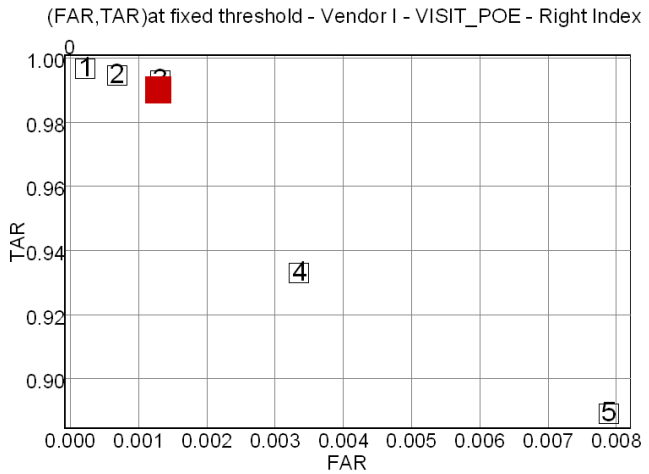
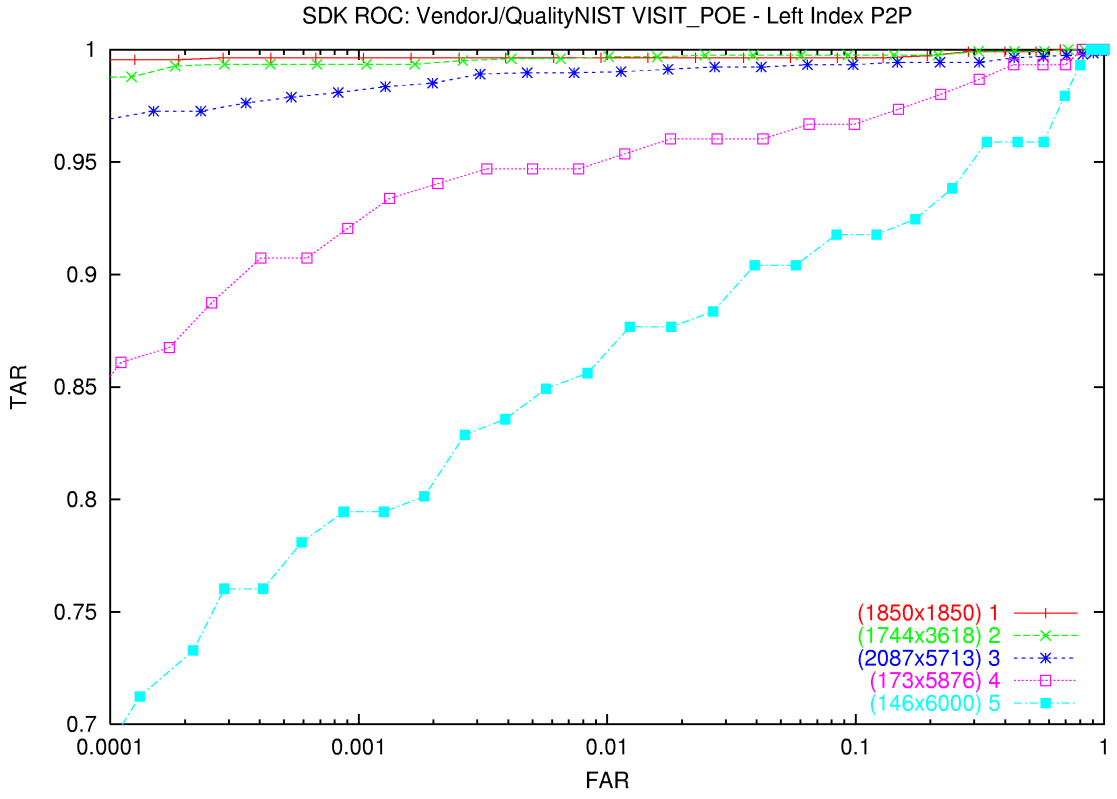


Figure 19-b. The effect of quality on ROC - Vendor I - Dataset VISIT_POE - Right index



Vendor J – VISIT_POE – Left index threshold=2600 (far,tar)=(0.01,0.987)					
Quality	**1(excellent)**	**2(very good)**	**3(good)**	**4(fair)**	**5(poor)**
FAR	0.0094	0.0101	0.0114	0.0118	0.0181
TAR	0.996	0.997	0.990	0.954	0.877

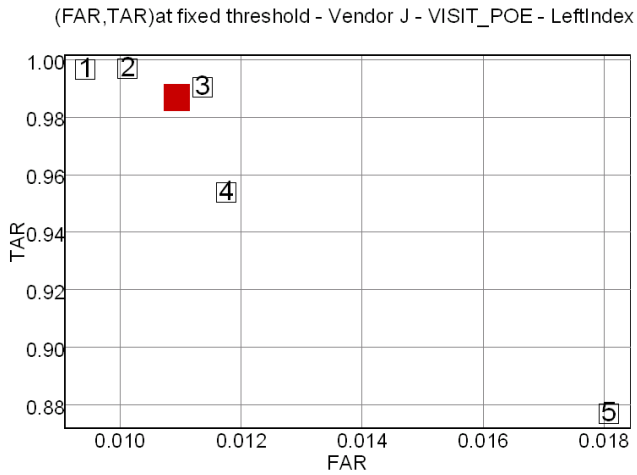
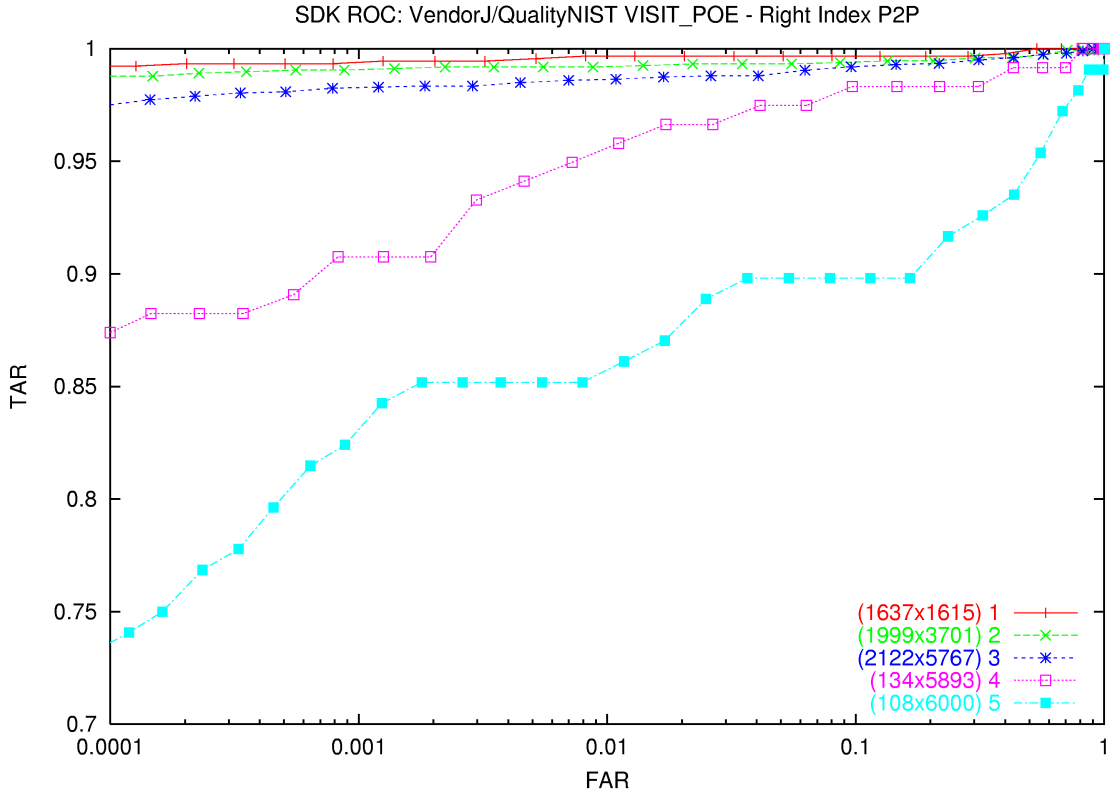


Figure 20-a. The effect of quality on ROC - Vendor J - Dataset VISIT_POE - Left index



Vendor J – VISIT_POE – Right index threshold=2600 (far,tar)=(0.01,0.987)					
Quality	**1(excellent)**	**2(very good)**	**3(good)**	**4(fair)**	**5(poor)**
FAR	0.0082	0.0088	0.0108	0.0111	0.0170
TAR	0.997	0.992	0.986	0.958	0.870

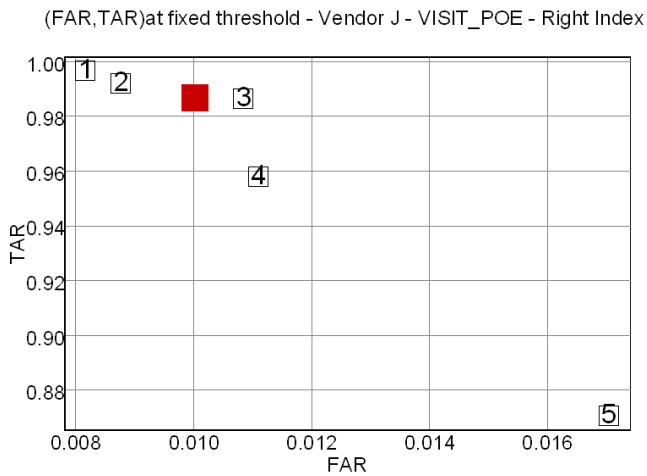
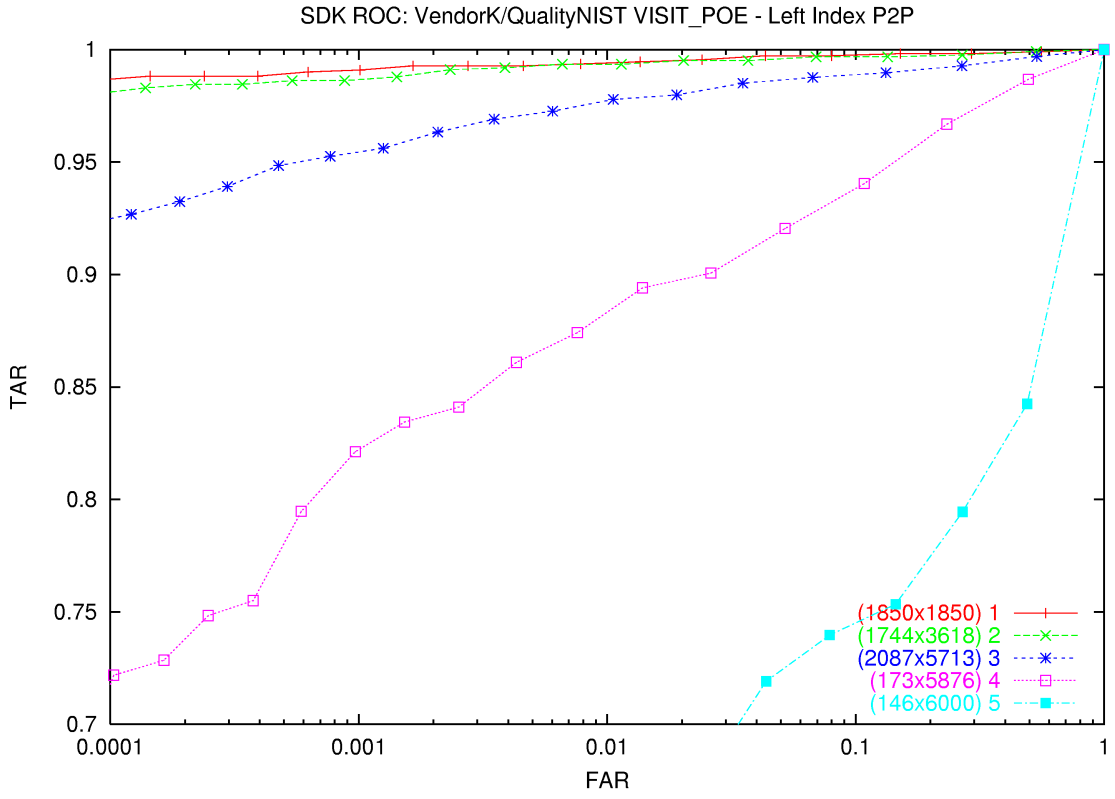


Figure 20-b. The effect of quality on ROC - Vendor J - Dataset VISIT_POE - Right index



Vendor K – VISIT_POE – Left index threshold=14 (far,tar)=(0.011,0.969)					
Quality	**1(excellent)**	**2(very good)**	**3(good)**	**4(fair)**	**5(poor)**
FAR	0.0136	0.0114	0.0106	0.0076	0.0147
TAR	0.995	0.994	0.978	0.874	0.651

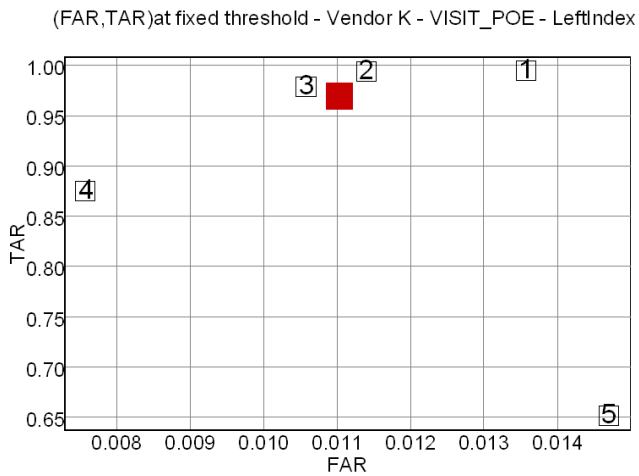
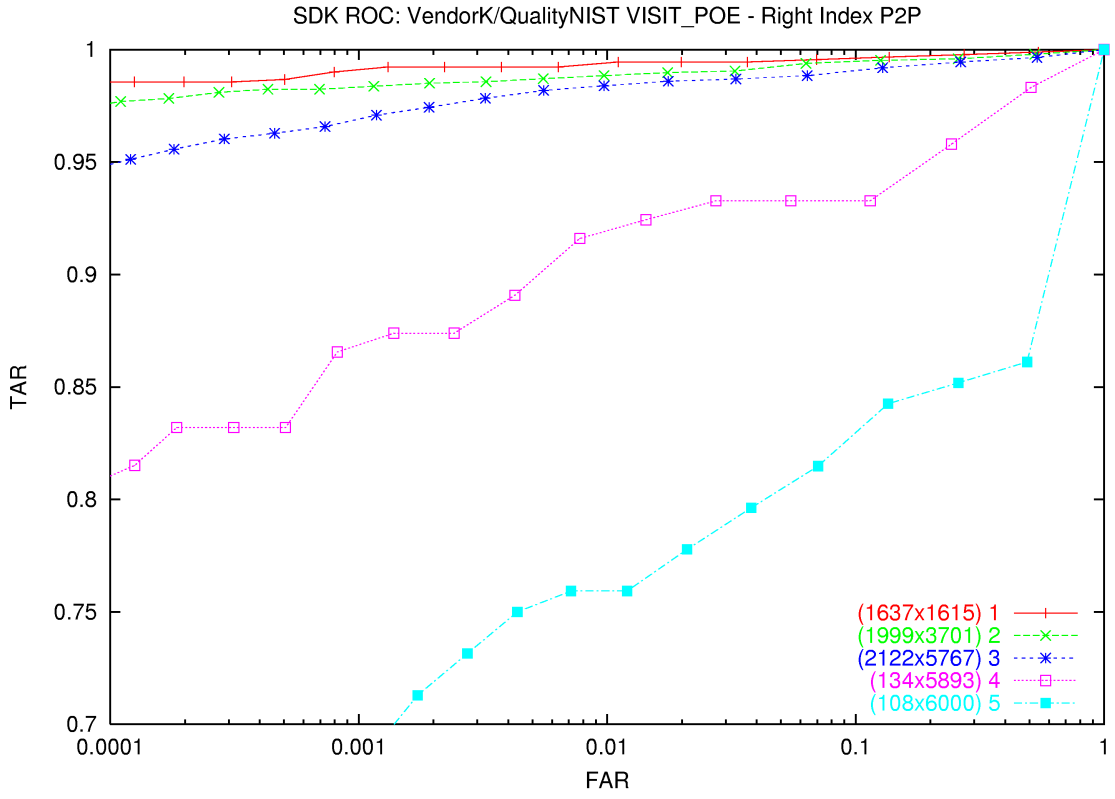


Figure 21-a. The effect of quality on ROC - Vendor K - Dataset VISIT_POE - Left index



Vendor K – VISIT_POE – Right index threshold=12 (far,tar)=(0.018,0.980)					
quality	**1(excellent)**	**2(very good)**	**3(good)**	**4(fair)**	**5(poor)**
FAR	0.0199	0.0175	0.0176	0.0143	0.0210
TAR	0.994	0.990	0.986	0.924	0.778

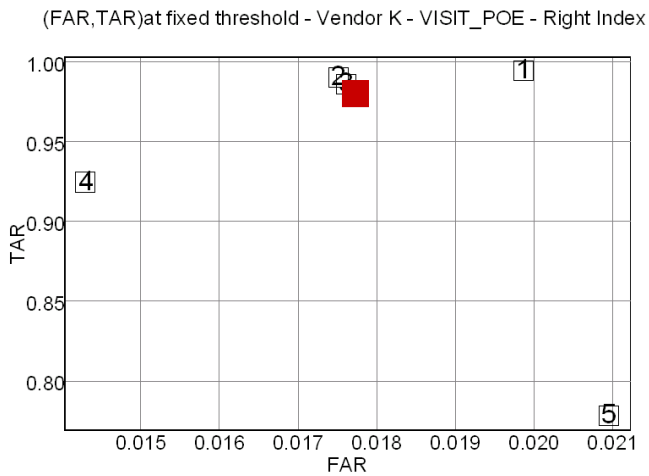
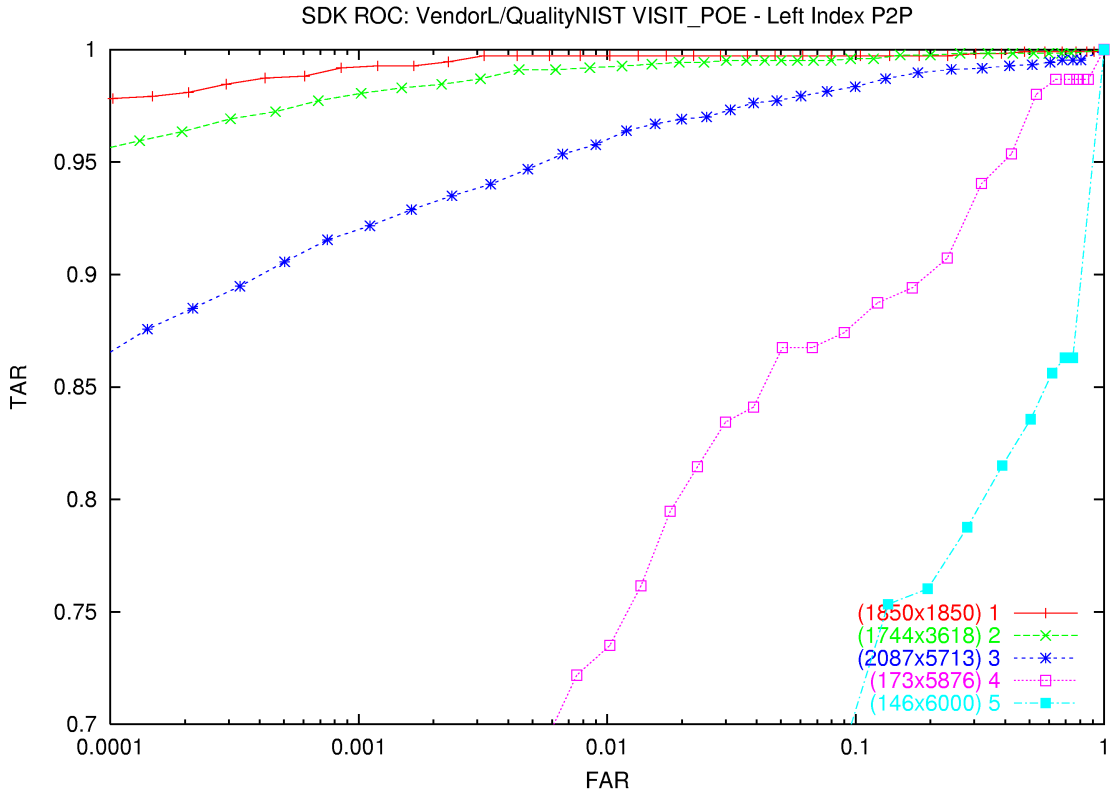


Figure 21-b. The effect of quality on ROC - Vendor K - Dataset VISIT_POE - Right index



Vendor L – VISIT_POE – Left index threshold=2100 (far,tar)=(0.01,0.952)					
Quality	**1(excellent)**	**2(very good)**	**3(good)**	**4(fair)**	**5(poor)**
FAR	0.0102	0.0151	0.0090	0.0075	0.0065
TAR	0.997	0.994	0.958	0.722	0.514

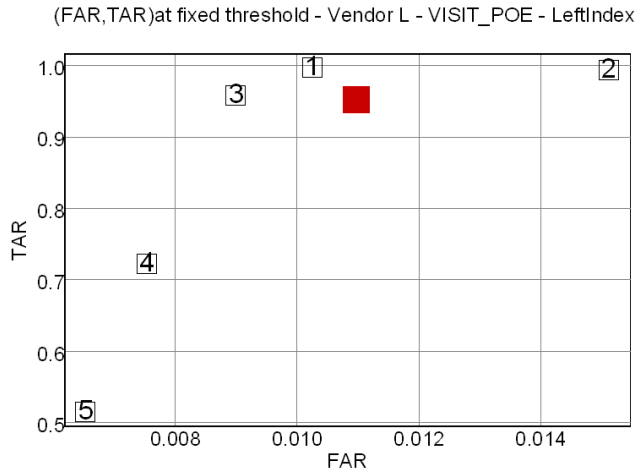
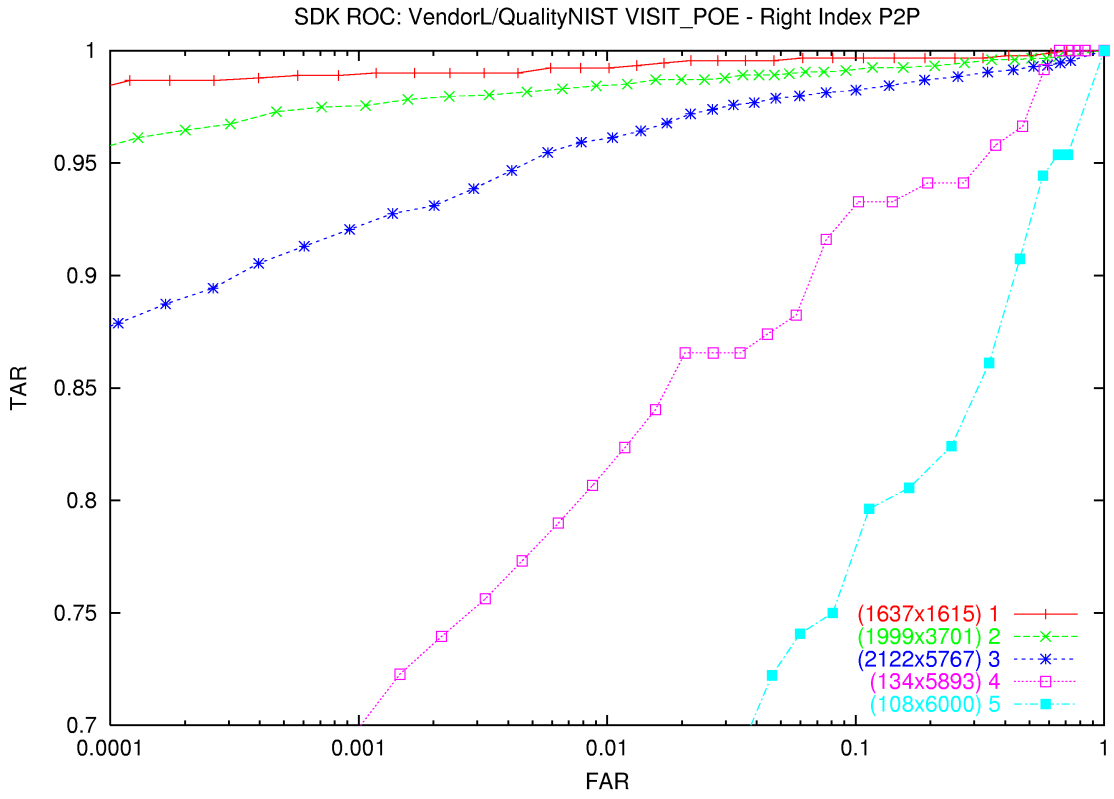


Figure 22-a. The effect of quality on ROC - Vendor L - Dataset VISIT_POE - Left index



Vendor L – VISIT_POE – Right index threshold=2000 (far,tar)=(0.012,0.962)					
Quality	**1(excellent)**	**2(very good)**	**3(good)**	**4(fair)**	**5(poor)**
FAR	0.0101	0.0157	0.0104	0.0087	0.0073
TAR	0.992	0.987	0.961	0.807	0.602

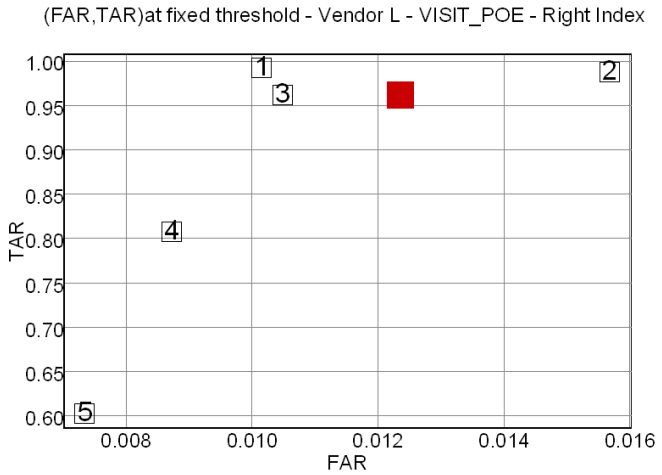
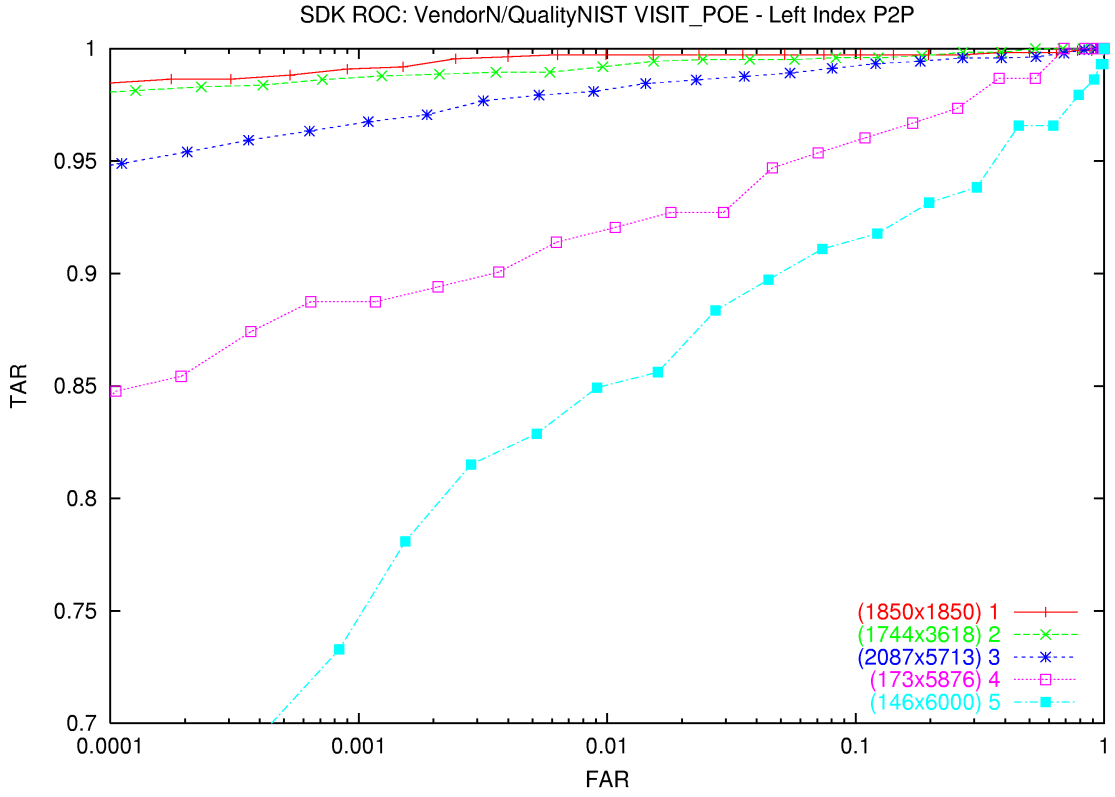


Figure 22-b. The effect of quality on ROC - Vendor L - Dataset VISIT_POE - Right index



Vendor N – VISIT_POE – Left index threshold=36 (far,tar)=(0.01,0.979)					
Quality	**1(excellent)**	**2(very good)**	**3(good)**	**4(fair)**	**5(poor)**
FAR	0.0234	0.0096	0.0088	0.0062	0.0028
TAR	0.997	0.992	0.981	0.914	0.815

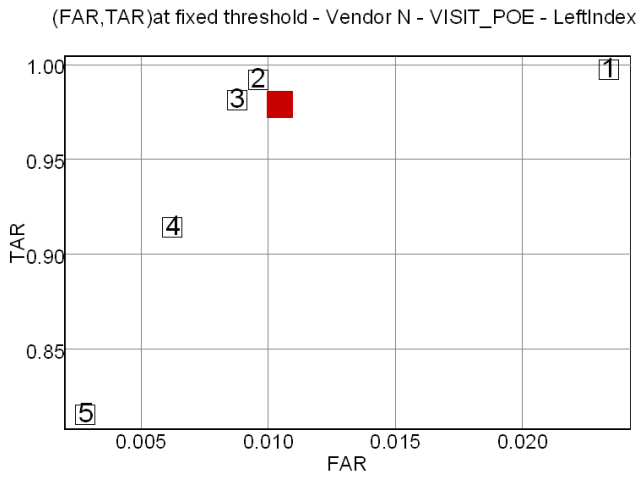
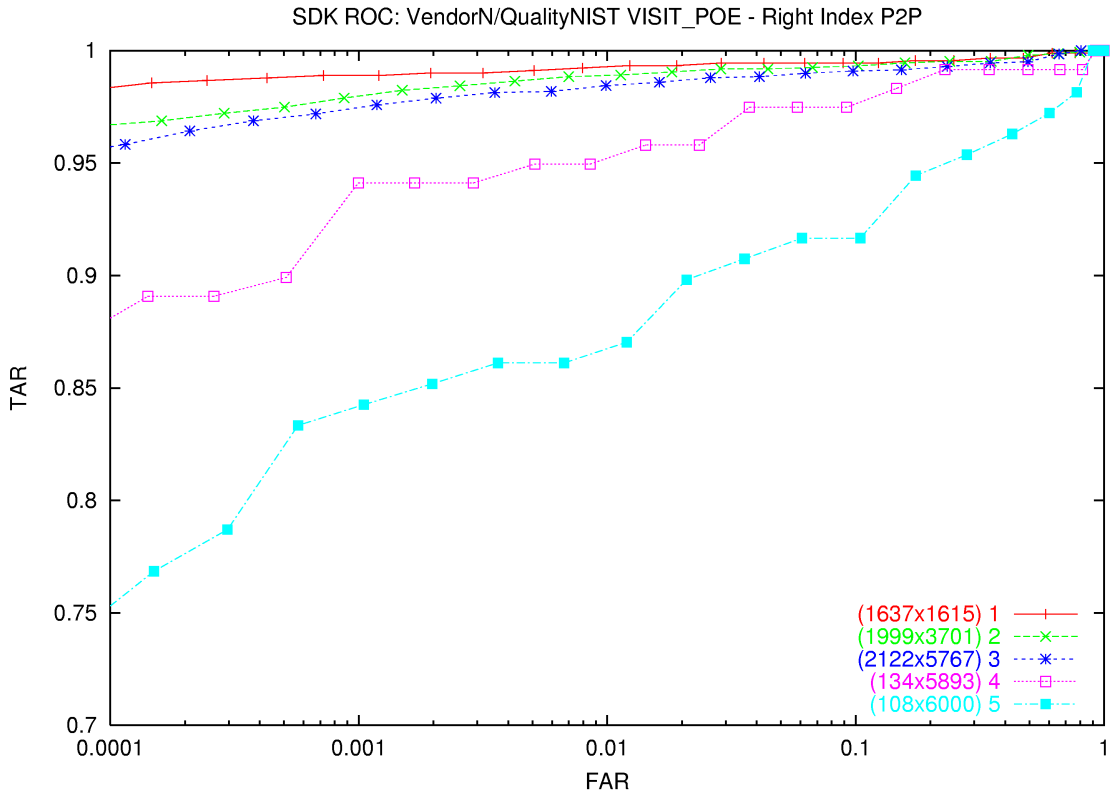


Figure 23-a. The effect of quality on ROC - Vendor N - Dataset VISIT_POE - Left index



Vendor N – VISIT_POE – Right index threshold=34 (far,tar)=(0.01,0.984)					
Quality	**1(excellent)**	**2(very good)**	**3(good)**	**4(fair)**	**5(poor)**
FAR	0.0287	0.0113	0.0099	0.0086	0.0036
TAR	0.994	0.989	0.984	0.949	0.861

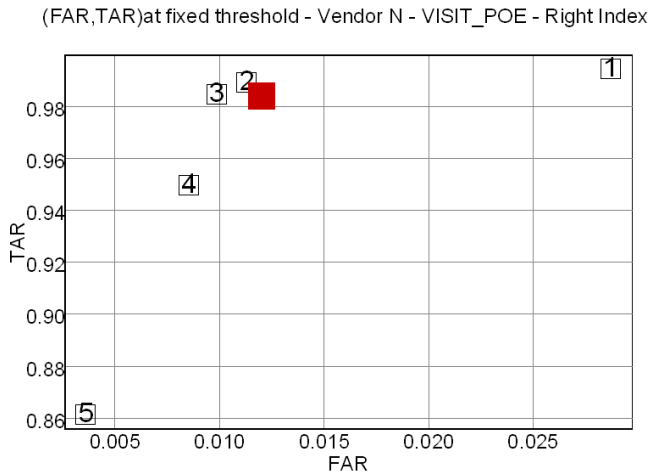
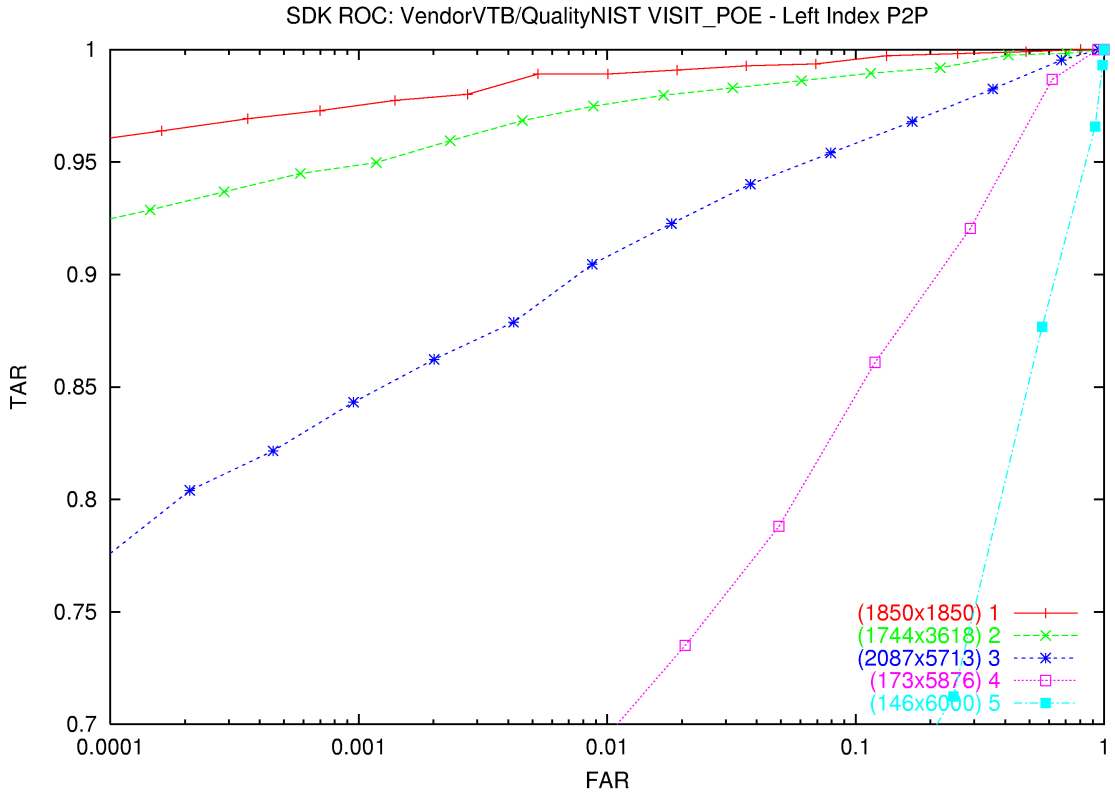


Figure 23-b. The effect of quality on ROC - Vendor N - Dataset VISIT_POE - Right index



Vendor VTB – VISIT_POE – Left index threshold=18 (far,tar)=(0.011,0.916)					
Quality	**1(excellent)**	**2(very good)**	**3(good)**	**4(fair)**	**5(poor)**
FAR	0.0191	0.0168	0.0087	0.0040	0.0028
TAR	0.991	0.980	0.905	0.623	0.397

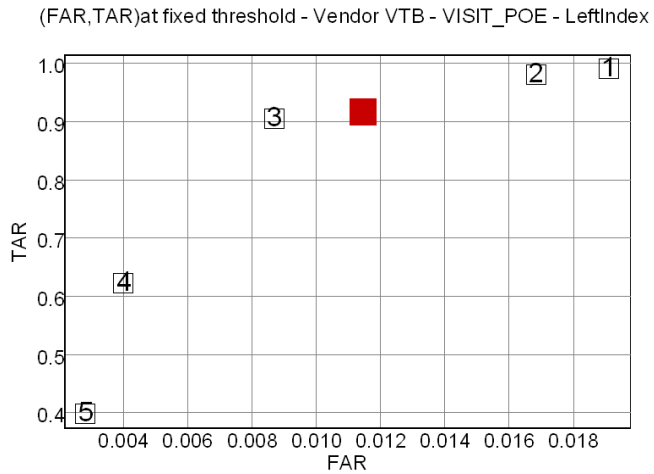
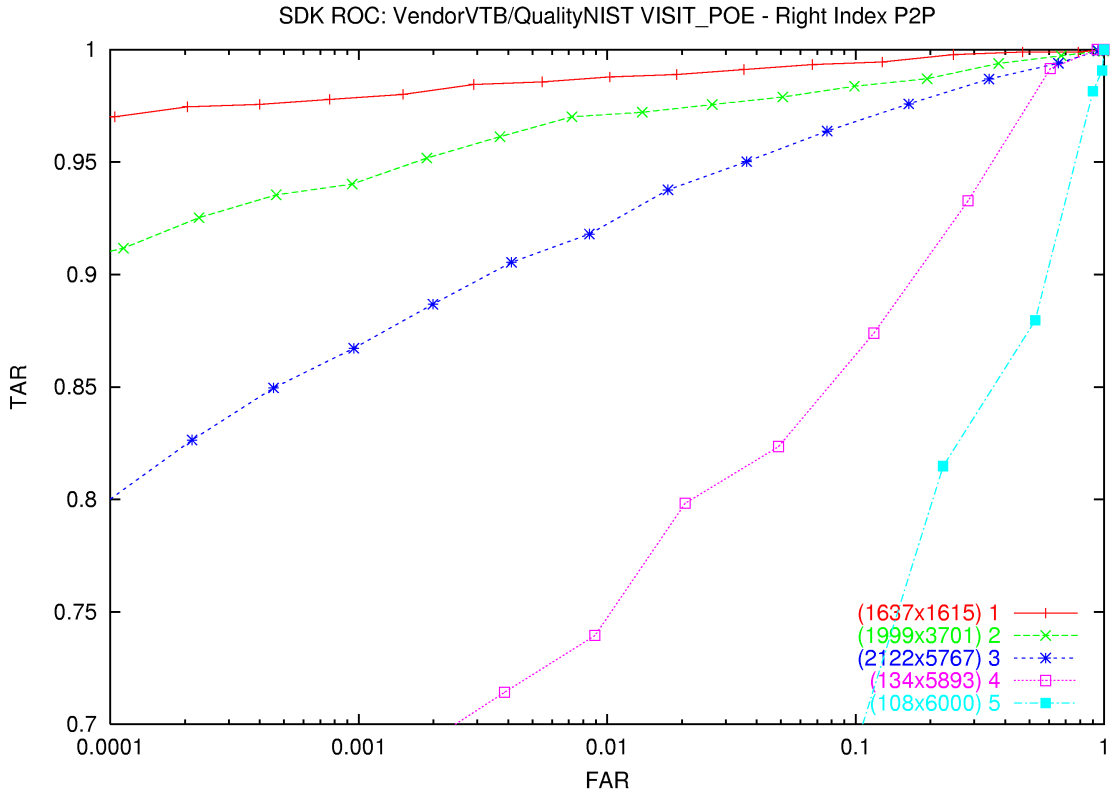


Figure 24-a. The effect of quality on ROC - Vendor VTB - Dataset VISIT_POE - Left index



Vendor VTB – VISIT_POE – Right index threshold=18 (far,tar)=(0.01,0.93)					
Quality	**1(excellent)**	**2(very good)**	**3(good)**	**4(fair)**	**5(poor)**
FAR	0.019	0.0138	0.0085	0.0039	0.0020
TAR	0.989	0.972	0.918	0.714	0.435

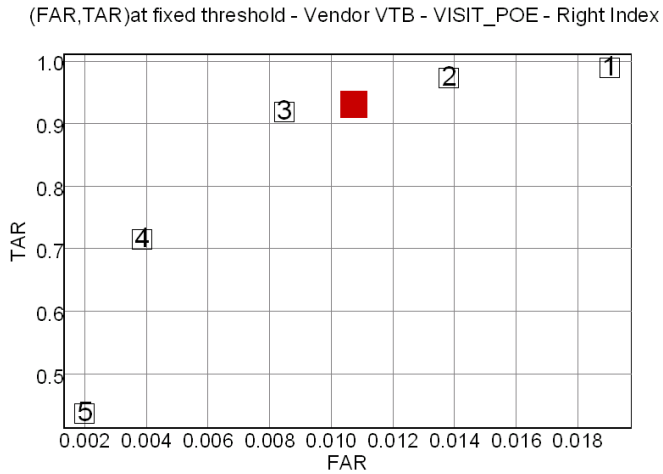


Figure 24-b. The effect of quality on ROC - Vendor VTB - Dataset VISIT_POE - Right index

6 Conclusion

In this report we developed a method to assess quality of a fingerprint that can forecast matcher performance. This includes an objective method of evaluating quality of fingerprints. Our definition of quality can be applied to other biometric modalities and upon proper feature extraction can be used to assess quality of any mode of biometric samples.

This paper explains the method used to generate quality determining feature vectors from the NIST NFIS fingerprint software. The method used to train an artificial neural network that generates image quality values is then presented. These image quality values were then tested on 300 different combinations of fingerprint images data and fingerprint matcher systems and found to predict matcher performance for all systems and datasets. The test results presented in the body of the report for US-VISIT POE data show that the method is highly accurate even for matcher and data combinations that were not used in the neural network training.

Automatically and consistently determining quality of a given biometric sample for identification and/or verification is a problem with far reaching applications. If one can determine low quality biometric samples, this information can be used to improve the acquisition of new data. This same quality measure can also be used to selectively improve archival biometric gallery by replacing poor quality biometric samples with better quality samples. Weights for multimodal biometric fusion can be selected to allow better quality biometric samples to dominate the fusion. All of these applications require that the quality of

the biometric sample be determined prior to identification or verification. The image quality measure presented here meets this requirement. Most of these applications also require that quality of the biometric sample be computed in real time during data acquisition. The method presented here is fast enough to meet this speed requirement.

References

1. A.Jain, L. Hong, S. Pantanki, R. Bolle. Identity authentication using fingerprints, In Proceedings of the IEEE, Vol. 85, No. 9, pp. 1365-1388. 1997
2. M. Y. Yao, S. Pantanki, N. Haas, N. Ratha, R. Bolle. Quantifying Quality: A case study in fingerprints
3. C. Watson. NIST Special Database 29 Plain and Rolled Images from Paired Fingerprint Cards, NIST, November 2001
4. Friedman, J. H. and Stuetzle, W. (1981). Projection pursuit regression. Journal of the American Statistical Association **76**, 817-823.
5. R. A. Hicklin, H. Korves, B. Ulery, M. Zoepfl, M. Bone, P. Grother, R. Micheals, S. Otto, C. Watson.Fingerprint Vendor Technology Evaluation (FpVTE) 2003. NISTIR 7123
6. C. Watson, A. Wilson, K. Marshall, M. Indovina, R. Snelick. Studies of one-to-one fingerprint matching with vendor SDK matchers. NISTIR 7119
7. M.D. Garris, C.I. Watson, R.M. McCabe, C.L. Wilson. User's guide to NIST fingerprint image software. NISTIR 6813
8. R.M. McCabe. Data format for the interchange of fingerprint, facial, scar mark & tattoo (SMT) information. American national standard ANSI/NIST-ITL 1-2000, July 2000
9. Rev. 12-84. U.S. Department of Justice, Federal Bureau of Investigation. The science of fingerprints.
10. C.L. Wilson, C.I. Watson, M.D. Garris, A. Hicklin. Studies of fingerprint matching using the NIST verification test bed. NISTIR 7020
11. N. Nill and B.H. Bouzas. Objective image quality measure derived from digital image power spectra. Optical Enginering, April 1992. Vol. 31 no. 4 pp.813-825
12. Bolle et al. System and methods for determining the quality of fingerprint images. United States patent number US596356, 1999.
13. L.L. Shen, A. Kot and W.M Koo. Quality measures of fingerprint images. 3rd international conference AVBPA 2001, p182-271, June 2001.
14. N.K. Ratha and R.M. Bolle. Fingerprint image quality estimation. IBM computer science research report RC21622, 1999.
15. L. Hong, Y. Wan and A.K. Jain. Fingerprint image enhancement: algorithm and performance evaluation. IEEE transaction on pattern analysis and machine intelligence, vol. 20, no.8, August 1998.
16. E. Lim, X. Jiang, W. Yau. Fingerprint quality and validity analysis. IEEE ICIP 2002
17. C.L. Wilson, J.L. Blue, O.M. Omidvar. Training Dynamics and Neural Network Performance. Neural Networks, vol. 10, No. 5, pp. 907-923, 1997.
18. J.L. Blue and P.J. Grother. Training feed forward networks using conjugate gradients. NISTIR 4776 and in Conference on Character Recognition and Digitizer technologies, vol. 1661, pp. 179-190, San Jose, February 1992.
19. D. Liu and J. Nocedal. On the limited memory BFGS method for large scale optimization, Imathematical programming B, vol. 45, 503-528, 1989.

20. O.M. Omidvar and C. L. Wilson. Information content in neural net optimization, NISTIR 4766, and in Journal of connection science, 6:91-103, 1993.
21. S. Wood, C.L. Wilson. Studies of plain to rolled fingerprint matching using NISY Agorithm Test Bed (ATB). NISTIR 7112.
22. C. L. Wilson, M.D. Garris, C.I. Watson. Matching performance for the US-VISIT IDENT system using flat fingerprints. NISTIR 7110
23. Hastie, T. and Tibshirani, R. (1990). Generalized Additive Models. Chapman and Hall, London.

Appendix A

ROCs
Data set quality distributions
This appendix is in a separate file.

Appendix B

Trellis plots of feature vector components and performance
This appendix is in a separate file.