2017

The Assessment of Fingerprint Quality for a More Effective Match Score in Minutiae-Based Matching Performers

Alyshia Meyers

The Assessment of Fingerprint Quality for a More Effective Match Score in Minutiae-Based Matching Performers

Alyshia Meyers

Thesis submitted
to the Eberly College of Arts and Sciences
at West Virginia University

in partial fulfillment of the requirements for the degree of

Master of Science in
Forensic & Investigative Science

Keith Morris, Ph.D., Chair

Suzanne Bell, Ph.D.

Afzel Noore, Ph.D.

Department of Forensic & Investigative Science

Morgantown, West Virginia
2017

Keywords: Fingerprints, Fingerprint Quality

ABSTRACT

The Assessment of Fingerprint Quality for a More Effective Match Score in Minutiae-Based Matching Performers

Alyshia Meyers

One of the most common types of evidence recovered from a crime scene are latent fingerprints, however these impressions are often of low quality. The quality of a latent fingerprint is described as the degree to which the ridge details can be observed. If the quality of the latent fingerprint is very clear, a minutiae-based matching algorithm with automatic extraction may detect and utilize the minutiae that are truly present in the fingerprint. However, if the impression is of poor quality, the minutiae-based matching algorithm's automatic extraction may detect fewer features and could completely miss features resulting in the return of an unrelated candidate. The aim of this research was to determine a method to improve the match score of latent fingerprints by removing the bad quality regions, where both a subjective and objective methods were utilized. The subjective method utilized the predetermined quality categories of "good," "bad" or "ugly" to assign a latent fingerprint. After classification, each impression was processed by Adobe® Photoshop® and four quality areas were serially removed. In the objective method, each latent fingerprint was assessed with NFIQ algorithm and then MINDTCT algorithm. The MINDTCT algorithm provided a quality map that was used to remove successive portions of each latent fingerprint. The resulting new images from both methods were compared to a database using the two different minutiae-based matching algorithms: AFIX Tracker® and BOZORTH3.

The results were examined utilizing the statistical methods of receiver operator characteristic (ROC) curves, area under the ROC curve (AUC), cumulative match characteristic (CMC) curve, Wilcoxon signed-rank test, Spearman's rank correlation and the comparison of the removal methods. ROC curves and the resulting AUC were able to determine that the AFIX Tracker® program is a reliable performer with high AUC values, while the BOZORTH3 minutiae-based algorithm did not perform well with low AUC scores of around 0.5. The results produced from the CMC curves showed that the subjective method produced higher rank 1 and top 10 rank identification than the objective method, contrary to what was hypothesized. The correlation scores showed the manual and automatic extraction were weakly correlated to one another. However, a very weak to no correlation between the algorithms of the BOZORTH3 and AFIX Tracker® was observed. The comparison between the subjective and objective methods of removal showed the examiner allowed for a more conservative removal of the fingerprint than the objective method. With this result in connection with the CMC curve results shows that being more conservative produces higher rank 1 and top ten rank identification scores.

Contents

1. Introduction

One of the most frequented pieces of evidence recovered from a crime scene is latent fingerprints and these may not be of the best quality. Quality of a latent fingerprint can be described as how well the ridge details in the fingerprint can be observed. There is a wide-range of factors that can affect the quality of the fingerprint and knowing these factors can greatly improve the comparison process. If the quality of the latent fingerprint is very clear, an examiner is able to identify different features. If the latent fingerprint is of a poor quality, the examiner may find few to no features for comparison. The same is true when utilizing minutiae-based matching algorithms such as AFIX Tracker® from AFIX Technologies and BOZORTH3 from the National Institute of Standards and Technology (NIST). If a latent fingerprint is of a low quality, there is a lower likelihood that AFIX Tracker® or BOZORTH3 can detect minutiae. If these algorithms do detect minutiae, it is possible that the minutiae are not true minutiae.

In recent years, there has been more attention given to estimating the quality of a latent fingerprint to assist in improving match scores [1, 2, 3, 4]. Some of these studies have created their own methods of estimating the quality, but none have fully utilized the potential of the NIST Open Source Software, NBIS [5, 6]. In this research study, we aim to determine a method of increasing the match score of the latent fingerprint by removing bad quality areas and keeping good quality areas intact by utilizing a subjective and objective method. The subjective method will consist of a single examiner assigning a quality category of "good," "bad" or "ugly" and then successively removing regions of the latent fingerprint based on criteria set forth in Section 3.2.3. The objective method will consist of the latent fingerprint being assessed by the NFIQ global quality assessment algorithm to receive a quality score. The latent fingerprint will then be examined by the MINDTCT minutiae detection algorithm to obtain the detected minutiae and a quality map. The resulting quality map will be used to successively remove different quality portions from the latent fingerprint. The study will also determine if there is a significant difference between the two minutiae-based matching algorithms of AFIX Tracker® and BOZORTH3 as well as how the match scores for the different quality portions removed are effected. To analyze the results from the subjective and objective methods, the data analysis methods of receiver operator characteristic (ROC) curves, area under the ROC curve, cumulative match characteristic (CMC) curve, Wilcoxon Signed-Rank test and Spearman's Rank-Order correlation will be utilized. With the removing of the lower quality areas from the latent fingerprints, the match scores of both minutiae-based matching algorithms will increase with the objective method achieving overall higher results than the subjective method due to the lack of bias in the objective method.

1.1 Fingerprints

Fingerprints are patterns that appear before birth, persist throughout ones lifetime and remain for some time after death that are useful when identifying an individual based on different patterns. Fingerprints are created from friction skin which is the volar skin surface of the hands, fingers, feet and toes that is characterized by alternating strips of raised ridges and furrows arranged in a pattern [7, 8]. By the third or fourth month of fetal development, friction ridges are visible and will remain generally unchanged for life [8]. There is, however, a way that the fingerprint could change through damage. If there is a cut deep enough to the dermal papillae layer (the inner layer of skin), the friction ridges will not regenerate and could be useful to identify an individual due to the patten being more peculiar [9].

1.1.1 Levels of Detail

When it comes to the identification of a fingerprint, there are three levels of features that are examined. The first level is the general pattern of the fingerprint which includes arches, loops and whorls. Figure 1.1 shows the different possible patterns.

(A) Arch (5-15%)

   (i) A type of pattern where a ridge enters one side of the fingerprint, rises in the middle and flows out the other side.

   (ii) Another type is a tented arch which is an arch, but the ridges in the middle do not flow out of the pattern [8].

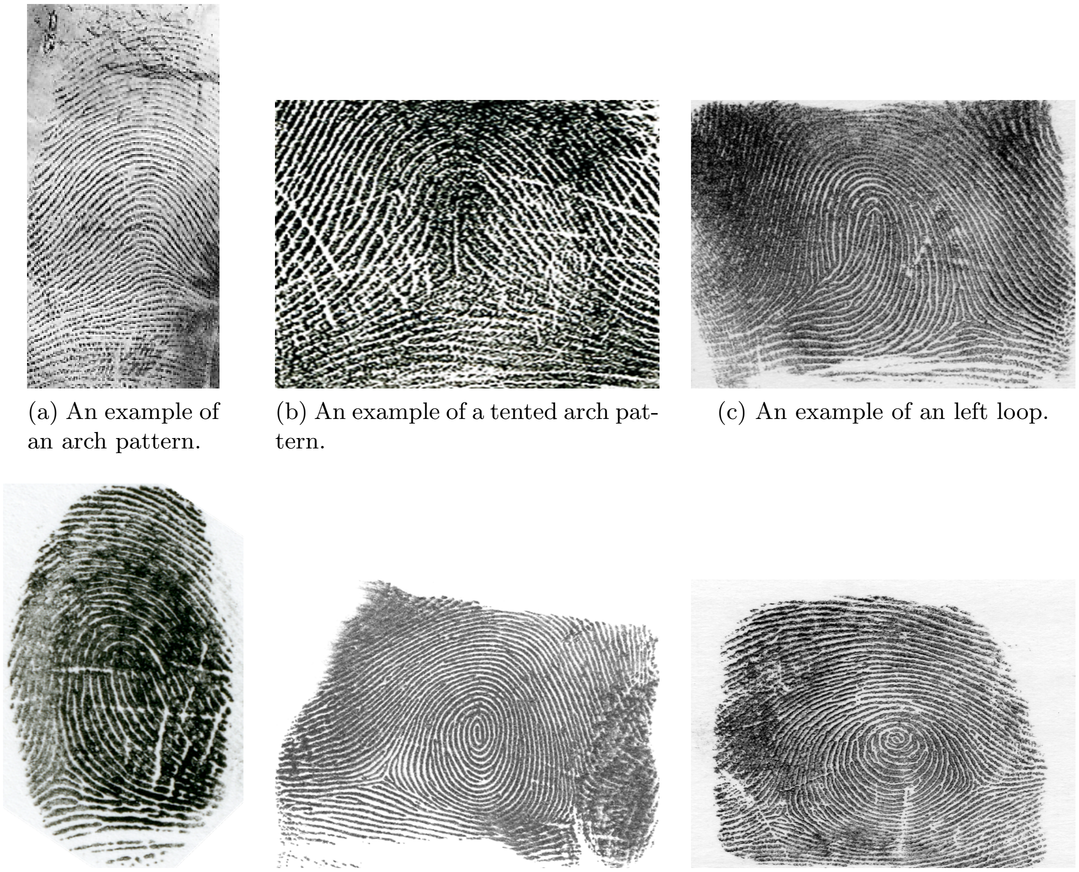
(B) Loops (60-65%)

   (i) A loop is defined as one or more ridges that enter a side of the impression, recurves and ends on or towards the same side from which it started [8].

   (ii) There are two types of loops, ulnar and radial, that are aptly named for the forearm bones which the ridges flow toward when looking at a finger. The ulnar loops flow toward the little finger, while radial loops flow toward the thumb [8]. After the fingerprint is deposited, the loops are named based on the direction that the ridges flow toward, either right or left.

(C) Whorl (30-35%)

   (i) A whorl is a pattern that contains at least two deltas and has a recurve in front of each delta. A delta can be defined as several items including an abrupt ridge



(a) An example of an arch pattern.

(b) An example of a tented arch pattern.

(c) An example of an left loop.

(d) An example of a right loop.

(e) An example of a plain whorl.

(f) An example of a central pocket loop.

Figure 1.1: These images show examples of the various fingerprint patterns that are possible including (from top left to bottom right) an arch, tented arch, left loop, right loop, plain whorl and a central pocket loop. The fingerprints missing include a double loop whorl and an accident whorl, which are rare to occur. These patterns were obtained from students in the Department of Forensic and Investigative Science at West Virgina University.

ending, a dot, a short ridge, a meeting of two ridges or a bifurcation, where one ridge splits into two.

(ii) There are four types of whorls that include plain, double loop, central pocket loop and accidental.

i. A plain whorl possesses at least two deltas and one ridge that make a complete circuit in the shape of an oval, spiral or circle.

ii. A double loop is a pattern containing two separate loops with each loop forming its own delta.

iii. A central pocket loop combines the features of a loop and whorl together by having two deltas and one ridge making a complete circuit.

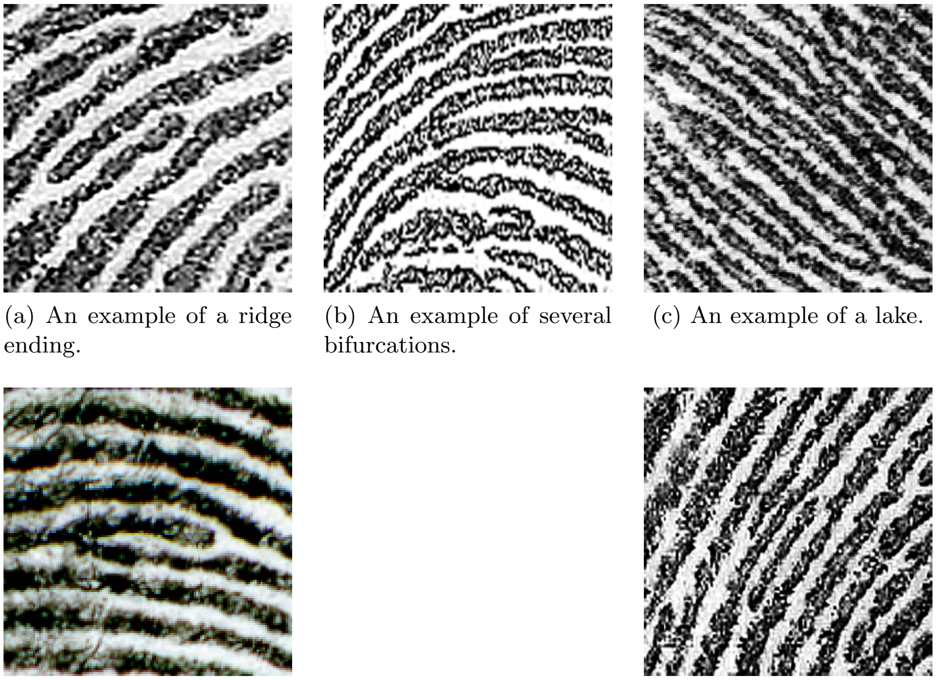
iv. The final type, accidental, can consist of two different types of patterns, have two or more deltas, possess some of the requirements of the other patterns or does not possess any characteristics of any another pattern [8].

Second level details are minutiae, ridge characteristics, points of identification or Galton characteristics. Minutia can be a point where a friction ridge terminates, beings or splits into two or more ridges [7]. Some examples of ridge characteristics include a ridge ending, bifurcation, lake/island, short ridge, spur or cross-over and examples can be seen in Figure 1.2 [10]. With a collection of these characteristics and taking into account the direction and position of the characteristics, it is possible to differentiate between two fingerprints. The clarity, or how well the small details of the 3-dimensional ridges have translated into a 2-dimensional fingerprint, of the latent fingerprint may affect this level of detail [11]. Without this level of detail, it is not possible to move onto the third level of detail.

The third level of detail includes observing the edges of the ridge and the position and shape of the pores [10]. These observation are called edgeoscopy and poreoscopy, respectively, and the use of these details are limited based on the clarity of the fingerprint [10]. The pores on the friction skin are actually an opening of a sweat or eccrine gland duct with the purpose of removing waste from the body as well as increasing friction by giving moisture to the surface of the ridges.

1.1.2 Comparison Process

The three levels of detail become important in the examination process that fingerprint examiners use called ACE-V (Analysis, Comparison, Evaluation and Verification). In the analysis phase, the examiner assesses the fingerprint impression to determine if the fingerprint is suitable for comparison [12]. If the fingerprint is not suitable for examination, the examiner will stop the analysis and the fingerprint is deemed of no value. If the fingerprint is suitable for examination, the examiner will determine the amount of variation in the characteristics that is allowed. The factors that effect the quality of the fingerprint will be discussed in further detail in Section 1.2. Levels 1, 2 and 3 details are noted for orientation purposes to be used in the comparison phase. Level 3 detail may not always



(a) An example of a ridge ending.

(b) An example of several bifurcations.

(c) An example of a lake.

(d) An example of a spur.

(e) An example of a short ridge.

Figure 1.2: This image shows examples of the various fingerprint characteristics that are possible including (from top left to bottom right) a ridge ending, bifurcation, lake/island, spur and a short ridge. These characteristics were taken from fingerprints obtained from graduate students in the Department of Forensic and Investigative Science at West Virgina University.

be present and an examiner can proceed without this level of detail. The analysis phase is completed individually for the known and exemplar fingerprints.

Next, a side-by-side comparison of the known and the exemplar fingerprints is conducted to determine if there is sufficient agreement or disagreement between the two fingerprints based upon ridge characteristics, ridge counts between minutiae and the agreement or disagreement in the clarity and quality of the fingerprints [12]. The three levels of detail are examined in this step to aid in the final determination. The examiner may choose a group of minutiae in a particular area on either fingerprint and examine the same portion of the other fingerprint to determine if the same minutiae are present. If an agreement exists, more target areas are chosen for comparison. Once the examiner believes the examination is complete, they will move into the evaluation phase to assess if there is sufficient detail to form a conclusion.

A conclusion of either (a) exclusion, (b) individualization or (c) inconclusive can be made by an examiner. An exclusion can be made if the examiner is of the opinion that there is not enough features in agreement between the fingerprints to conclude that the fingerprints did not "originated from the same source [12]." On the other hand, an individualization can be made if the examiner is of the opinion that there are sufficient features in common to conclude that the fingerprints "originated from the same source [12]." An inconclusive result can occur if the examiner is of the opinion that there is not enough features present between the two fingerprints to render an opinion.

When the analysis, comparison and evaluation phases are completed, the examiner will send the exemplar fingerprint to be verified by a separate examiner who will complete the same process [12].

1.2 Fingerprint Quality

Fingerprint quality can be defined as how well the ridge details in the fingerprint can be observed. If the quality of the fingerprint is clear and details are able to be distinguished, there is a higher likelihood that an examiner or a feature extraction algorithm extracts usable features that are truly present. However, if the quality of the fingerprint is blurry and the details are difficult to distinguish, the likelihood of an examiner or a feature extraction algorithm extracts false features is higher.

The quality of the fingerprint can be effected by a wide-range of factors including [11]:

- Type of substrate
- Type of matrix
- Development medium
- Deposition of pressure
- Pressure distortion

Different types of substrates that a latent fingerprint is deposited on can cause issues with the quality depending on the substrate. Some substrates, such as flexible surfaces or object shapes, can cause a fingerprint to be split, have creases or have an abnormal shape. Other matrices, such as dirty matrices, can prevent the entire fingerprint from not being deposited or no general detail present. On the other hand, soft substrates tend to produce good quality fingerprints due to the friction ridges sinking into the surface and creating a molded impression [11].

The matrix of a fingerprint is another factor that has the potential to effect the quality of the fingerprint. Matrices such as fluids other than sweat and fluids that are corrosive in nature can have negative effects on the quality. Fluid matrices can cause ridge breaks producing a dotted appearance while in the case of a corrosive matrix, part of the substrate may result in a take-away fingerprint leaving only a portion of the fingerprint [11]. On the other hand, mud-type matrices produce a plaster-like consistency that is likely to show Level 2 detail which could produce good quality fingerprints. Paint or blood-like matrices can provide even better details, including Level 3 details; however, an examiner must be aware that false artifacts can occur [11].

Development media is another factor that can effect the quality of a latent fingerprint and commonly includes fingerprint powder (magnetic or black), ninhydrin, and cyanoacrylate (super glue). In the use of fingerprint powder, the powder can fill the pore ducts and even disturb some of the second level details. The act of brushing the powder onto the fingerprint can alter the fingerprint due to the fragile matrix that produced the fingerprint. Another development method to be aware of is ninhydrin. Ninhydrin reacts with the amino acids that are present in sweat and tends to leave the latent fingerprint spotty [11]. Cyanoacrylate, on the other hand, can develop ridges that have fine detail. Knowing how these processes work is one key to understanding if the quality of a print is due to the exposure or if there is another type of issue effecting the print [11].

Deposition pressure, or the amount of vertical weight applied in touching a substrate, plays a large role in the quality of a fingerprint. If heavy pressure is applied, the ridges can become flattened and the valleys can become smaller causing difficultly in distinguishing minutiae. On the other hand, the light pressure can cause ridges to become narrower and the valleys may not be seen causing issues as well. Figure 1.3 shows the differences in the deposition pressure on a fingerprint [11].

Pressure distortion is described as the pressure applied on the lateral or horizontal plane [11]. This type of distortion maybe observed as having a smeared appearance of the ridges from sliding the finger in a sideways direction thus causing issues with the quality. One of the main purposes of friction skin is to help grip surfaces; however, when the pressure becomes too great, the friction skin will slip causing the ridges to slip as well. With the smeared ridges, the examiner may find it difficult to determine were a ridge should end and were another begins. In most cases, the pressure distortion will typically be in only one direction and most likely the remainder of the fingerprint is affected to some extent [11].

As discussed, these different factors can play an important role in the quality of a

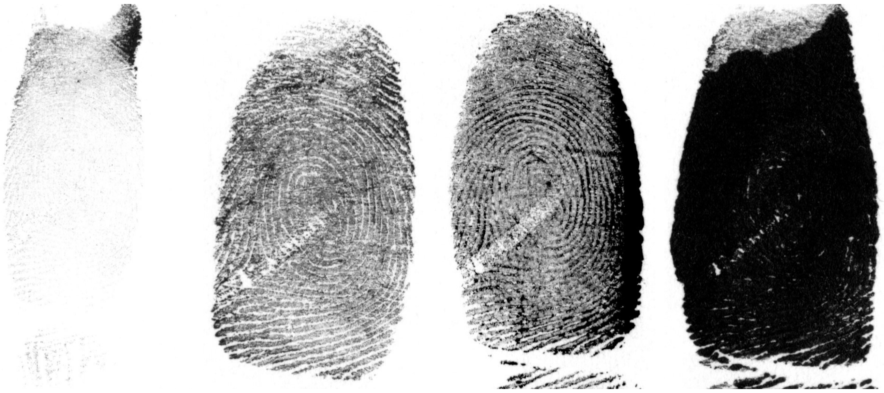


Figure 1.3: This figure shows how a fingerprint can look at different deposition pressures. From left to right, a light touch, medium touch, heavy touch and extreme deposition pressure from the author's right index finger.

fingerprint. Recognizing these factors may help in the process of removing bad quality areas or explaining why those areas cannot be removed.

1.3 National Institute of Standards and Technology Biometric Image Software (NBIS)

The National Institute of Standards and Technology (NIST) has been working with the Federal Bureau of Investigation (FBI) and the Department of Homeland Security (DHS) to develop biometric image software. NIST and the FBI having been working in tandem on the subject of fingerprints by conducting research, creating new technologies, and producing standards surrounding fingerprints, while NIST's work with DHS is equally important. Since September 11, 2001, Congress passed several new laws, such as the USA Patriot Act and the Enhanced Border Security and Visa Entry Reform Act, directing NIST to create new technologies to verify identities, perform background checks and validate names of individuals. With the work of NIST, the FBI and DHS, an open system, NBIS, was developed for conducting research. NBIS, National Institute of Standards and Technology Biometric Image Software, is not subjected to copyright and includes the open source packages of PCASYS, MINDTCT, NFIQ, AN2K7, IMGTOOLS, NFSEG and BOZORTH3 [6].

- PCASYS is an neural-network based classification system that is designed to determine the pattern of the fingerprint image as being an arch, tented arch, loop (left or right), whorl, or a scar. This allows for a more effective way of classifying a large number of fingerprints at one time and minimizing the number of candidate matches [6].

- MINDTCT is a minutiae detection algorithm that detects and records different minutiae characteristics such as ridge endings or bifurcations. This package will be discussed in more depth in Section 1.3.1.

- NFIQ is a global quality assessment algorithm that assesses the overall quality of a fingerprint on a scale of one to five, with one representing a good quality fingerprint and five representing a poor quality fingerprint. Previous research has stated that higher quality images will produce better results in a matching algorithm than a poor quality print [13, 14]. This package will be discussed in more detail in Section 1.3.2.

- AN2K7 package contains the utilities to read, write, and edit files in a format according to the ANSI/NIST-ITL 1-2007 "Data Format for the Interchange of Fingerprint, Facial, Scar Mark & Tattoo (SMT) Information"created by NIST [6, 7].

- IMGTOOLS is an all-purpose image utility that assists in the processing and converting images to the proper formats.

- NFSEG is an algorithm that segments a four-finger plain impression, called a slap, into the individual fingerprint images.

- BOZORTH3 is a fingerprint matching algorithm that takes the minutiae detected by MINDTCT from two fingerprints and determines if the fingerprints came from the same source or different sources [6]. This package will be discussed in farther detail in Section 1.3.3.

1.3.1 MINDTCT Algorithm

MINDTCT is a minutiae detecting algorithm that locates different ridge characteristics in a fingerprint. This algorithm has several steps, listed below, to ensure that the minutiae detected are true minutiae and not errors in the fingerprint.

1. Input Fingerprint File
2. Generate Image Maps
3. Binarize Image
4. Detect Minutiae
5. Remove False Minutiae
6. Count Neighboring Ridges
7. Assess Minutiae Quality
8. Output Minutiae Files

A fingerprint image, with a minimal resolution of 500 pixel per inch (ppi) and 8-bit grayscale, is read into the MINDTCT algorithm and assessed to produce five maps including: (1) a direction map, (2) a low contrast map, (3) a low flow map, (4) a high curve map and (5) a quality map. The direction map highlights the regions of the fingerprint that have clear ridge structure. If the ridges in the fingerprint are well-defined, there is a lower likelihood of false minutiae being detected later in the MINDTCT algorithm. The fingerprint image is divided into 8x8 pixel blocks and assigned a single value based on the ridge flow in that pixel block. The resulting direction maps contain an integer from zero to fifteen for each pixel block. The integer represents the particular ridge angle for that pixel block. A value of zero represents a vertical line and increases by 11.25° for each integer until the angle reaches 180°. The resulting values can then be translated to make a direction map such as in Figure 1.4 [6].

The second map, the low contrast map, illustrates regions of the fingerprint that maybe smudged or have background noise. The map detects areas of low contrast and prevents minutiae from being assigned in that region. The determination of a low contrast area is computed by evaluating the pixel intensity within the block and the surrounding blocks. If the resulting value is below a set threshold, the region is considered to have low contrast. Areas of good contrast are represented by zeros and areas of low contrast are represented by ones. The ones were converted to red and overlaid onto the original fingerprint image to show the areas of low contrast in Figure 1.5 [6].

The low flow map depicts the areas of the fingerprint that were not initially determined to have clear ridge flow, which can hinder minutiae detection. If an area is determined to have low flow and a minutia is detected, that minutia will be assigned a lower quality value due to being detected in a less reliable region. The output map illustrates the areas of low flow with zero's and one's. The values of zero represent the areas of low flow while the one's represent dominant ridge flow. For visual representation, the values of zero were converted to red and overlaid onto the original fingerprint image in Figure 1.6 [6].

The high curve map detects areas of high curvature through two measurements. The first measurement, vorticity, measures the cumulative change of ridge flow around the neighboring blocks [6]. The second measurement, curvature, calculates the largest change in the direction of ridge flow in the block and the surrounding neighbors [6]. In cases where high curve is detected, a minutia receives a lower quality score due to being detected in a less reliable region [6]. The output map is comprised of zeros and ones where the zeros are considered to not have high curve while the ones represent the high curve areas. The values of one were converted to red and overlaid onto the original fingerprint image in Figure 1.7 [6].

The final map generated, the quality map, takes into account the low contrast map, the low flow map, and the high curve map to show all the low quality regions of the fingerprint. The map contains five levels of quality (zero to four) with a single value assigned to an 8x8 pixel block. A low quality block receives a zero and a high quality block receives a four. An example of the quality map can be seen in Figure 1.8.

The next step in the MINDTCT algorithm is to binarize the fingerprint image. The grayscale fingerprint image is analyzed at every pixel to assess whether that pixel represents a ridge or a valley. If the algorithm determines the pixel represents a ridge, the pixel value is changed to zero (black). If the pixel is determined to represent a ridge, the pixel value is changed to 255 (white). This step in the MINDTCT algorithm is critical to the detection of true minutiae because it is important to maintain as much of the fingerprint's characteristics without accentuating the poor quality areas [6]. An example of a binarized image can be seen in Figure 1.9 [6].

Next, the binarized fingerprint image is scanned to identify pixel patterns that show a ridge ending or bifurcation [6]. The fingerprint image is scanned in the horizontal and vertical directions as to not miss any pattern. A list of minutiae is then created and false minutiae are removed in the following order:

1. Islands/lakes and holes
2. Minutiae in poor quality regions
3. Side minutiae, hooks, and overlaps
4. Minutiae that are too wide or too narrow

The MINDTCT algorithm then counts the neighboring ridges of each minutiae. The MINDTCT algorithm takes a single minutia and counts the ridges above and the ridges to the right of the minutia. This is conducted for each minutiae and the ridge counts are sorted based on the direction [6].

After ridge counting, the MINDTCT algorithm assesses the quality of each minutiae to ensure that if any false minutiae are still present, the minutiae will receive a lower quality score than a true minutiae in a good quality area. The MINDTCT algorithm uses two factors to produce a quality score for each minutia. First, the MINDTCT algorithm assesses the location of the minutia in the quality map to determine if the minutia is in a low or high quality region. Then, a neighborhood of 11x11 pixels is used to calculate the mean and standard deviation. The result is then used in conjugation with the location quality to produce the individual minutia quality score between 0.01 and 0.99. A value of 0.01 is a low quality score while a value of 0.99 is a high quality score [6].

The final step in the MINDTCT algorithm is to produce the output files which includes the detected minutiae, the direction map, the low contrast map, the low flow map, the high curve map, the quality map and a list of attributes associated with the minutiae. The detected minutiae file contains the x- and y- coordinates, the theta angles ($\mathit{\Theta}$) and the minutia quality score for every minutiae. The remaining maps are created as previously stated. The final text file output gives the attributes for each minutia detected which includes the pixel coordinate location, direction and type [6].

1.3.2 NFIQ Global Quality Assessment Algorithm

NFIQ is a quality assessment algorithm that estimates the overall quality of a fingerprint and returns a value of one to five, with one representing a high quality fingerprint and five representing a poor quality fingerprint [6]. The quality value obtained is based on the entire image where as the quality values in the quality map produced by the MINDTCT algorithm are based on the 8x8 pixel blocks examined. The NFIQ global quality assessment algorithm is designed as a neural network which operates much like a human brain. It is designed to have an input layer that is connected to a few or many hidden layers,

Table 1.1: This table shows description for an 11-dimension feature vector that is produced for each fingerprint by the *Fing2pat* algorithm. The symbol Q represents quality from the quality map and the symbol Q_m represents minutiae quality.

	NAME	**DESCRIPTION**
1	Foreground	Number of blocks that are $Q \geq 1$; i.e. foreground $= \sum_{i=1}^{4} U_i$ where U_i is the number of blocks with quality i
2	Total minutiae	Total minutiae found in fingerprint
3	Min05	Count of minutiae with $Q_m \geq 0.5$
4	Min06	Count of minutiae with $Q_m \geq 0.6$
5	Min075	Count of minutiae with $Q_m \geq 0.75$
6	Min08	Count of minutiae with $Q_m \geq 0.8$
7	Min09	Count of minutiae with $Q_m \geq 0.9$
8	Quality zone 1	% of the Foreground Blocks with Q=1
9	Quality zone 2	% of the Foreground Blocks with Q=2
10	Quality zone 3	% of the Foreground Blocks with Q= 3
11	Quality zone 4	% of the Foreground Blocks with Q= 4

depending, and then the hidden layers are connected to an output layer. The input and output layers contain parameters or weights that allow a user to adjust the amount of impact that a factor will have on the network. These types of networks can be trained to give optimal results and lower error rates. A training set needs to ideally be an accurate representation of what the true data looks like. If this is not the case, the weights can be adjusted accordingly. Another factor with neural networks that needs to be considered is the scaling of the input values. If the input values originated from different sources, a large discrepancy may occur in the output layer results.

To train the NFIQ global quality assessment algorithm, the function, *Fing2pat*, assesses a list of fingerprint images and what the user believes the quality score should be for the fingerprint. The *Fing2pat* function produces an 11-dimension feature vector for each fingerprint based on the criteria in Table 1.1 [5]. The resulting feature vector can be used to train the NFIQ global quality assessment algorithm on a wide variety of fingerprint qualities.

1.3.3 BOZORTH3 Algorithm

The BOZORTH3 algorithm is a translation and rotation invariant minutiae-based matching algorithm. The matching algorithm utilizes the minutia feature location and orientation from the MINDTCT algorithm output file. However, the algorithm does not take into account any type of distortion that the fingerprint maybe exposed to [5].

The algorithm has three important steps:

1. Construction of an intra-fingerprint minutiae comparison table
2. Construction of an inter-fingerprint minutiae comparison table
3. Traverse the inter-fingerprint compatibility table

The matching algorithm begins by construction an intra-fingerprint minutiae comparison table. The algorithm calculates the distance between any two minutiae on the fingerprint to account for the relative translational position. If these two minutiae match two minutiae from a separate fingerprint, the distance (d) between the two minutiae in both fingerprints must remain consistent with one another, no matter how much shifting or rotational difference there is between the fingerprints. Three rotational measurements (β_j, β_k and θ_{jk}) are calculated with the goal of determining the angle between each minutiae's orientation and d. The angles ensure that when comparing two fingerprints, the angles need to be consistent to be considered a match. Angle θ_{jk} (Figure 1.10) is computed by taking the slope of the arctangent of d. The arctangent is calculated by creating a right triangle and performing trigonometric functions. Angles β_j and β_k are found by taking the orientation of each minutiae and the line created by d. The minimum and maximum of β_j and β_k are taken to create β_1 and β_2 as shown in Equation 1.1 [5].

$$\beta_1 = min(\beta_k, \beta_j) \text{ and } \beta_2 = max(\beta_k, \beta_j) \tag{1.1}$$

Each minutiae pair compared is placed into the intra-fingerprint minutiae comparison table in the format shown in Equation 1.2 [5].

$$\{d_{jk}, \beta 1, \beta 2, k, j, \varTheta_{jk}\}$$

where d is the distance between j and k, k is minutia 1 and j is minutia 2,
$\beta 1$ and $\beta 2$ are as shown in Equation 1.1 and $\varTheta_{jk}$ is the arctangent of the line d

(1.2)

The entries in the table are sorted in order of the increasing distance between the minutiae. The table is then trimmed based upon a maximum distance threshold. This comparison table is completed for every fingerprint to be compared [5].

The minutiae-based matching algorithm then constructs an inter-fingerprint minutiae comparison table to find "compatible" entries between two different fingerprints. The fingerprints being compared are subjected to three tests to determine the compatibility (Equation 1.3). The first test determines if the distances between the two minutiae in one fingerprint and the distance between two minutiae in the other are within a set tolerance, T_d. The remaining tests determine if β_1 and β_2 from each fingerprint are within the set tolerance, T_β [5].

$$\Delta_d(d(P_m), d(G_n)) < T_d$$
$$\Delta_\beta(\beta 1(P_m), \beta 1(G_n) < T_\beta$$
$$\Delta_\beta(\beta 2(P_m), \beta 2(G_n)) < T_\beta$$

where P_m is the table entry for the unknown print, G_n is the corresponding table entry for the database print, T_d and T_β are the set tolerances, and Δ_d andΔ_β are "delta" or difference functions

(1.3)

If the minutiae are within the specified tolerance, they are entered into a compatibility table in the format of Equation 1.4 [5]:

$$\{\Delta_\beta(\mathit{\Theta}(P_m), \mathit{\Theta}(G_n)), (k(P_m), j(P_m)), (k(G_n), j(G_n))\} \tag{1.4}$$

This table format utilizes a total of four minutiae, two minutiae from the unknown fingerprint (P_m) and two minutiae from a database fingerprint (G_n). The first section, Δ_β ($\mathit{\Theta}(P_m)$, $\mathit{\Theta}(G_n)$), takes into consideration the difference in the beta angle and the arctangent angle for each print (see Figure 1.10). The $k(P_m)$ and the $j(P_m)$ are location of the two minutiae in the unknown fingerprint while the $k(G_n)$ and $j(G_n)$ are the location of the two minutiae in the database fingerprint [5].

The final step of the algorithm creates a compatibility graph through associations by traversing the inter-fingerprint compatibility table and creating clusters by linking entries from the table. The clusters are then collected and combined to receive the total number of linked entries to produce a match score. The more linked entries, the greater the match score produced and vice versa. The match score roughly shows the number of matched minutiae between the two fingerprints compared and a score above 40 is considered a true match, according to NIST documentation [5].

1.4 Automated Fingerprint Identification Systems (AFIS)

In the 1960's, the number of fingerprints that were manually searched was becoming an unfeasible task due to the time it took a fingerprint examiner to search ten-print cards. NIST was able to create the aforementioned algorithms, but there was also a need to develop a device that could capture and read the fingerprints [15]. Eventually, automated fingerprint identification systems (AFIS) were created and implemented in the FBI's laboratory and throughout the United States, with manual comparisons of ten-print cards phasing out. AFIS systems have the ability to be linked to a Computerized Criminal History (CCH) that stores information about a person's criminal past or "fingerprintable" event to be recalled if a match occurs.

With the different AFIS systems available today, there are a variety of features that differ from company to company; however, there are three main components that remain the same including coders, matchers and storage databases [16]. Coders are the part of the system that digitize the fingerprint and detect the minutiae, which are then used by the matcher. The matcher compares the minutiae found to minutiae in the storage database to determine a potential match. The storage database contains the fingerprints entered into the system and there can be three databases which include ten-prints, latent fingerprints (from crime scenes) and unsolved fingerprints [16]. Examiners and law enforcement have the option to do several different combinations of searches such as [15]:

- Known ten-print against a ten-print database
- Latent fingerprint from a crime scene against the ten-print database
- Latent fingerprint from a crime scene against the latent fingerprint database
- Ten-print against the unsolved database

One of the most important items to keep in mind is that this process is not without human interaction [16]. Humans are needed to ensure the fingerprints scanned into the system are of good quality and to examine the matches supplied by the system. The AFIS can narrow down the candidates to a manageable number, but it is up to the examiner to render a final decision on whether the fingerprints match.

Quality of the fingerprints is very important to the results. If the coder cannot determine minutiae in the fingerprint or happens to determine minutiae in the fingerprint that may be false, it is possible to receive a low match score or even a false match. With fingerprints of a high quality and clearer ridge detail, the system can return a better match score. This is not always possible with the fingerprints recovered from a crime scene, which can be smudged, partials or overlaid, just for the fact that they were accidentally placed. This is an issue that cannot be prevented, but with the help of high quality exemplars and portions of the print removed, it may give a higher chance of matching.

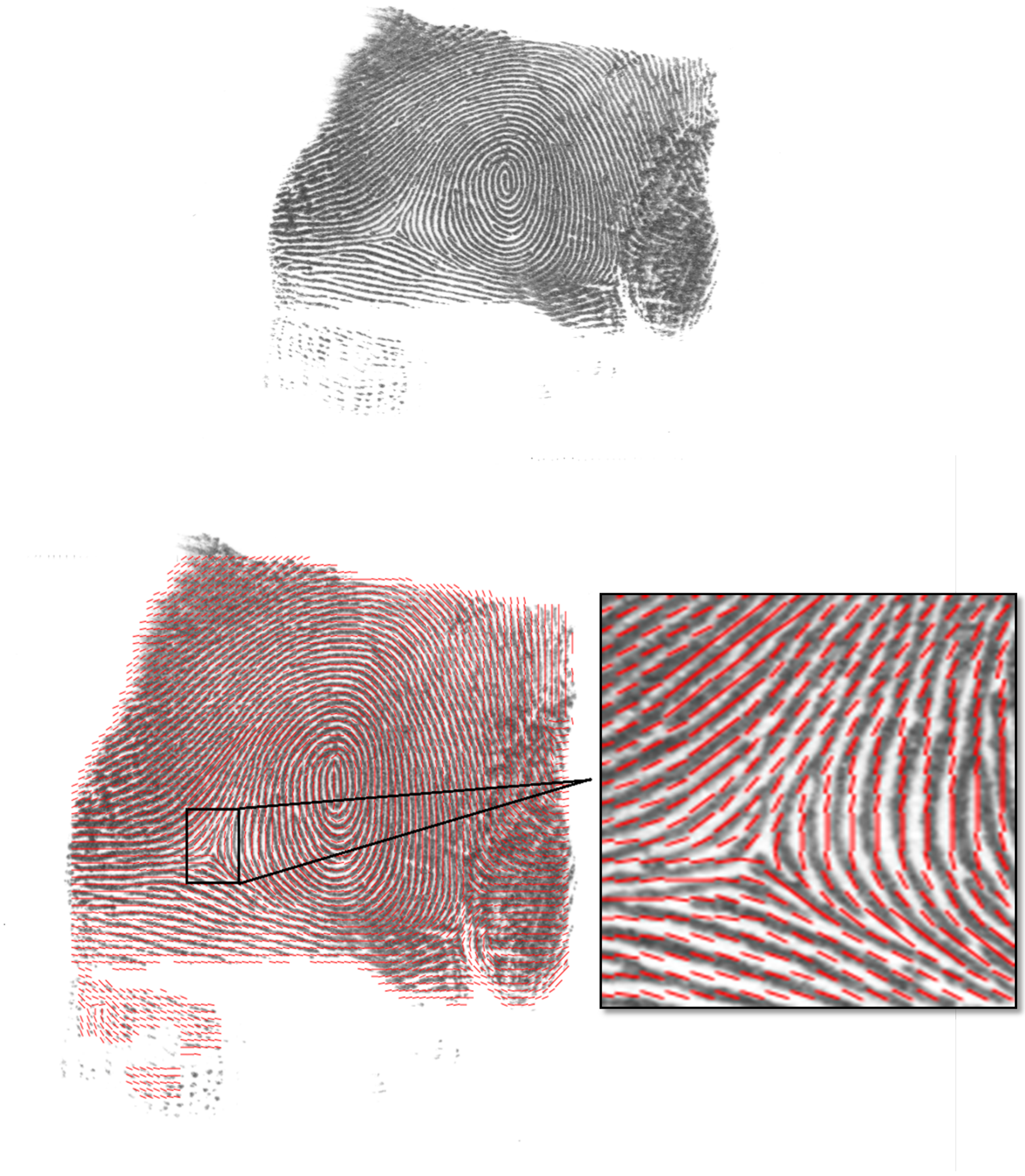


Figure 1.4: This image shows an example of the original fingerprint (top) and the direction map (bottom) that is created during the MINDTCT algorithm on an in-house fingerprint. The red on the direction map shows the angle direction that the algorithm produces along with a zoomed in portion of the fingerprint to further see the detail.

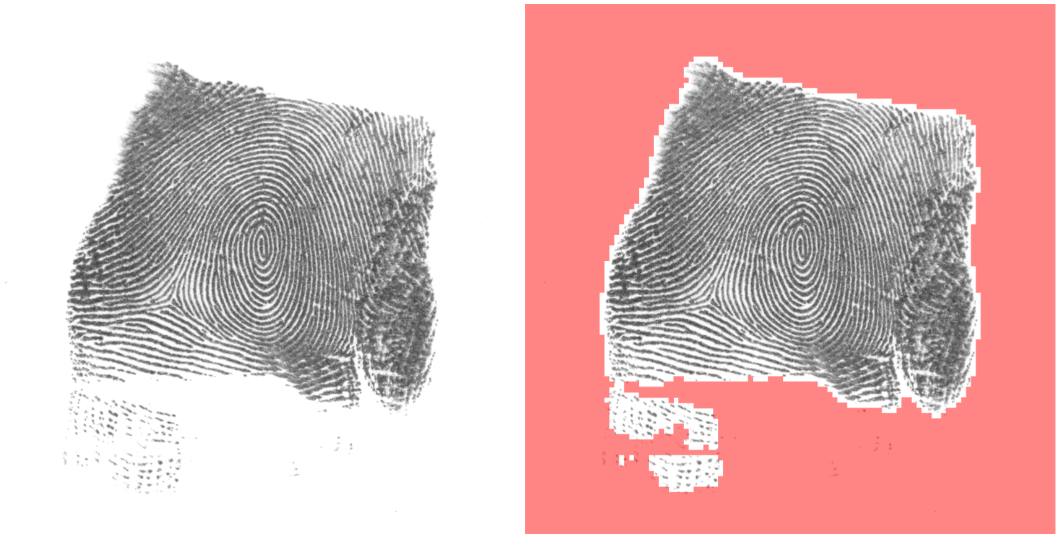


Figure 1.5: This image shows an example of original fingerprint (left) and the low contrast map (right) that is created during the MINDTCT algorithm. The red areas of the image show the low contrast portions of the in-house fingerprint.

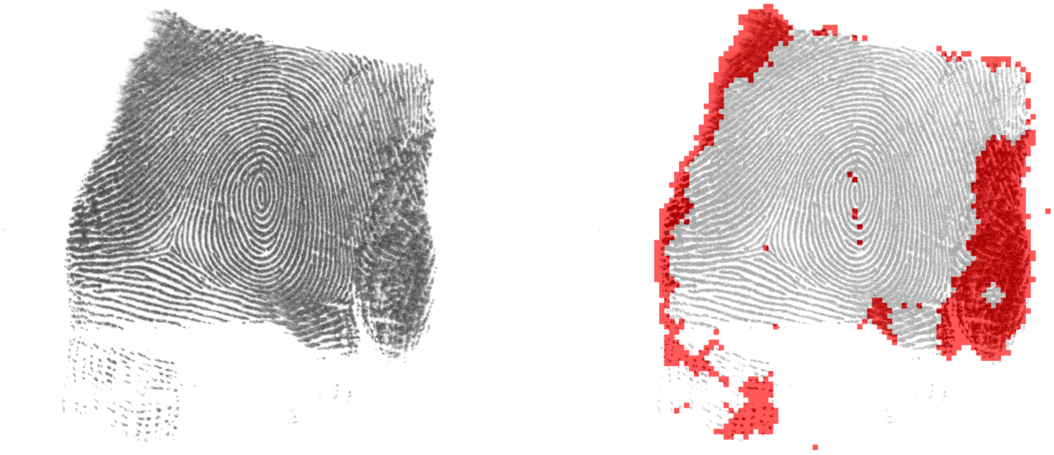


Figure 1.6: This image shows an example of the original fingerprint (left) and the low flow map (right) that is created during the MINDTCT algorithm. The red portions indicate the low flow areas in the in-house fingerprint.

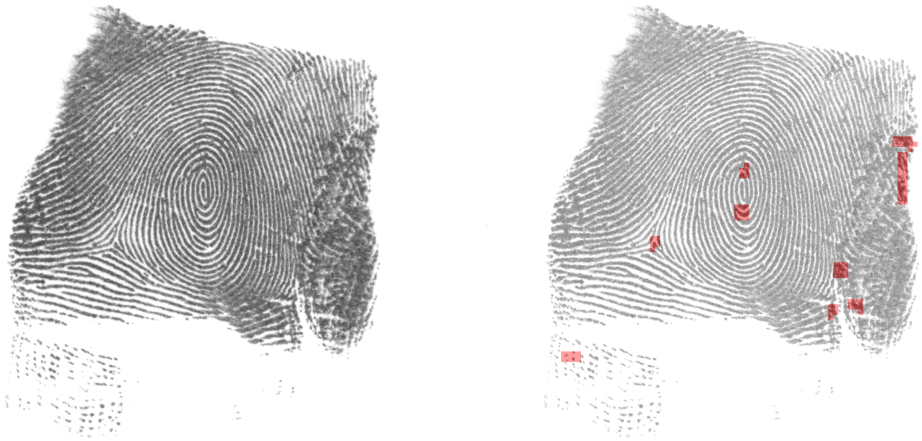
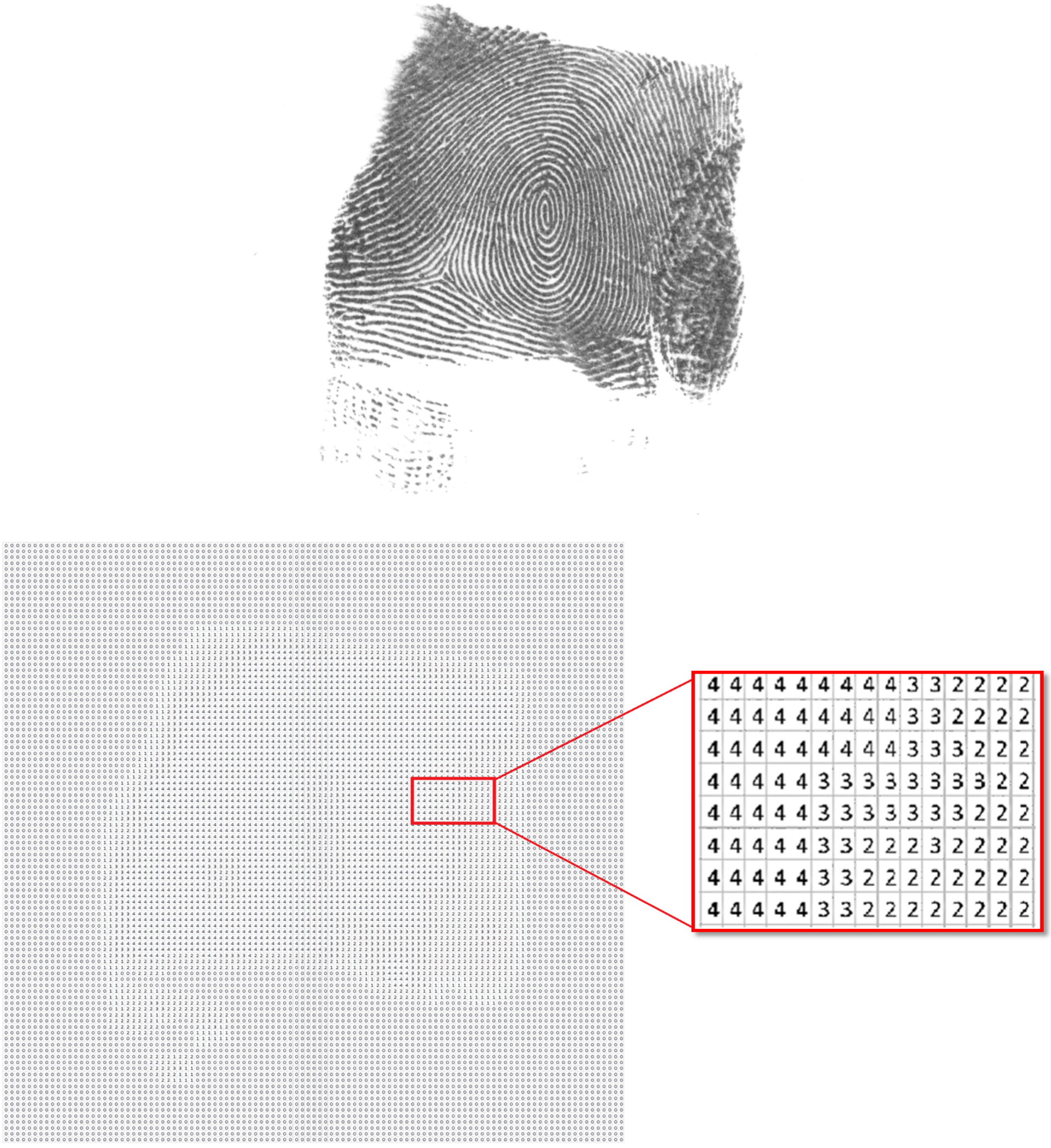


Figure 1.7: This image shows an example of the original fingerprint (left) and the high curve map (right) that is created during by the MINDTCT algorithm. The red squares indicate the high curve areas that were detected in the in-house fingerprint.



4	4	4	4	4	4	4	4	4	3	3	2	2	2	2
4	4	4	4	4	4	4	4	4	3	3	2	2	2	2
4	4	4	4	4	4	4	4	4	3	3	3	2	2	2
4	4	4	4	4	3	3	3	3	3	3	3	3	2	2
4	4	4	4	4	3	3	3	3	3	3	3	2	2	2
4	4	4	4	4	3	3	2	2	2	3	2	2	2	2
4	4	4	4	4	3	3	2	2	2	2	2	2	2	2
4	4	4	4	4	3	3	2	2	2	2	2	2	2	2

Figure 1.8: This image shows an example of the original fingerprint (top) and the quality map with a zoomed in portion of the map (bottom) that is created during the MINDTCT algorithm from a fingerprint of an in-house fingerprint. The map is composed of values of zero through four with zero representing a poor quality area and four representing a high quality area.

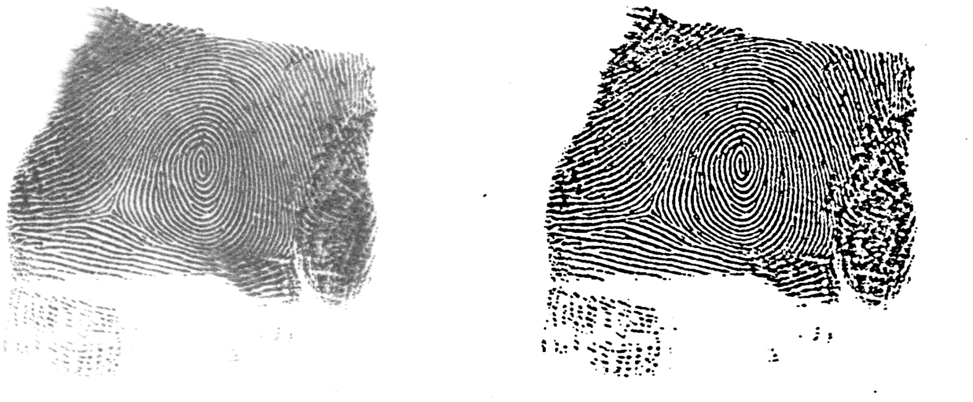


Figure 1.9: This image shows an example of the original fingerprint (left) and the binarized image (right) that is created during the MINDTCT algorithm from the in-house fingerprint.

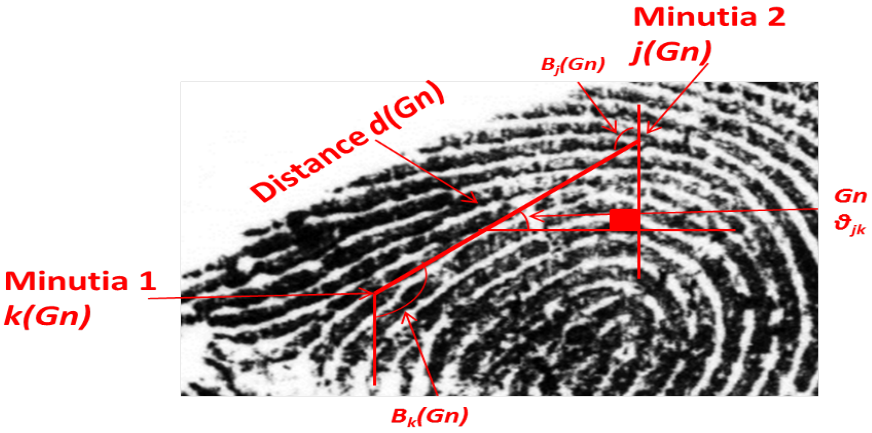


Figure 1.10: This image shows the assumed way the arctangent of the slope is found for the distance between two minutiae on a fingerprint. $k(G_n)$ and $j(G_n)$ are the minutiae on the fingerprint, d(G_n) is the distance between the minutiae, βj and βk are the minutiae angles and G_n or Θ_{jk} is the arctangent of the slope. The recreation is based on the information and images available in [5].

2. Literature Review

In recent years, the assessment of the quality of a latent fingerprint has become more apparent. Some studies, such as [1, 2], have devised their own mathematical method of determining the quality while others, such as [3], have created there own objective method to assessing the quality.

2.1 Lim *et al.*

Lim *et al.* proposed a two-part quality analysis technique to examine not only the quality of the fingerprint but also the validity of the fingerprint using local and global measures. The local analysis examined the pattern of the ridges and valleys while the global analysis examined the continuity and uniformity of the fingerprint. The local analysis utilized the orientation certainty level (ocl) to measure the energy concentration along the dominant direction of the ridges in a block [17]. The image was portioned into 32x32 non-overlapping pixel blocks and the gray level gradient found. Principal Component Analysis (PCA) was then performed on the gray level gradient. The eigenvalues were computed and the ratio of the minimum eigenvalue over the maximum eigenvalue represented the ocl.

The authors also examined the average values of the local maximum and minimum gray level values to determine a threshold value to separate ridges from valleys to ensure the fingerprint is from the current user and not the residue from the previous user. The ridge frequency was calculated for each block. Each block was then assessed using the ridge-frequency and ocl to determine whether the block was "good," "bad," "undetermined" or "blank." Table 2.1 shows how the overall quality was assigned to each block of the fingerprint image using the ocl and the ridge-frequency. The total quality score is then

Table 2.1: The resultant local block quality recreated from [1].

Ridge-Valley Structure OCL	Good	Undet.	Bad
Good	Good	Good	Undet.
Undet.	Good	Undet.	Bad
Bad	Undet.	Bad	Bad

computed using Equation 2.1 [1].

$$S_L = \frac{T_G + 0.5 * T_U}{T_G + T_U + T_B}$$

where T_G, T_U, and T_B correspond to the total number of good, undetermined and bad quality image blocks

(2.1)

The authors also performed a global analysis utilizing the continuity and uniformity of the fingerprint. The continuity examines the orientation change along the rows and columns of the block. The amount of change outside of a set threshold was accumulated and mapped into a global orientation score (S_{GO}) with a high quality score of one and a low quality score of zero. The uniformity of the ridge valley structure utilized the ratio of the ridge thickness to valley thickness for each block. The standard deviation was calculated and if there was a large deviation from the mean value, the fingerprint can be identified as a bad quality fingerprint. The score were mapped into a global score (S_{GR}) ranging from zero to one. The total quality of the image (QS) was found from Equation 2.2.

$$\text{QS} = (\alpha_1 S_L + \alpha_2 S_{GO} + \alpha_3 S_{GR}) \text{ x } \frac{min(T - T_{BL}, A_{min})}{A_{min}}$$
$$\alpha_1 + \alpha_2 + \alpha_3 = 1$$

where α_1, α_2, and α_3 are the coefficients, T_{BL} is the number of blank blocks obtained from adaptive thresholding, T is the total number of blocks or the sum of T_G, T_U, T_B and T_{BL}, and A_{min} is the required minimum number of blocks that are identified as the foreground

(2.2)

To test the proposed method, the authors utilized a feature extraction program by Jiang *et al.* for AFIS as a benchmark to test the performance of the method [4]. The number of correctly detected minutiae (c), the number of falsely detected minutiae (f), and the number of undetected minutiae (d) can be calculated after using the feature extraction program. The quality benchmark, qb, is defined in Equation 2.3.

$$qb = \frac{c}{c + d + f} \text{ x } \frac{min(Rf, Rm)}{Rm}$$

where Rm is the minimum area of the fingerprint required by AFIS and Rf is the area of the segmented fingerprint region [1]

(2.3)

The authors used 150 images for the analysis. Regression analysis showed that the proposed method was able to assign invalid and low quality values to low quality images and higher quality fingerprints received higher quality scores showing that the proposed method is able to detect quality differences in fingerprints [1].

2.2 Chen *et al.*

Chen *et al.* proposed a quality method to examine ridge and valley clarity using overlapping regions of grayscale distributions, the orientation flow and the overall image quality for fingerprints. Ridge-valley clarity analysis (local clarity) gives the ability to differentiate between ridges and valleys along the ridge direction. The fingerprint image was split into 32x32 pixel blocks and a 2D vector (V_1) of size 32x13 pixels was extracted. V_1 is then transformed to align to a vertical 2D vector (V_2). An average of V_2 is calculated as shown in Equation 2.4 [2].

$$V_3(i) = \frac{\sum\limits_{j=1}^{m} V_2(i,j)}{m},\ i\text{= }1...32$$

where m is the block height (13 pixels) and i is the horizontal index

(2.4)

Linear regression is applied to V_3 to find the Determine Threshold (DTI) where the regression line is positioned at the center of V_3 to classify the ridge and valley regions. A distribution of the ridge and valley grey levels is plotted along with the threshold. The overlapping regions between the ridge and valley levels is considered to be misclassified by the threshold. The clarity score can be calculated based on the area of the overlap using the following three equations:

$$\alpha = \frac{v_B}{v_T},\ \beta = \frac{R_B}{R_T},\ \text{LCS} = \frac{(\alpha + \beta)}{2}$$

where v_B is the number of bad pixels in the valley lower than the DTI, v_T the total number of pixels in the valley region, R_B is the number of bad pixels in the ridge that are lower than DTI, R_T is the total number of pixels in the ridge region, α and β are the portion of bad pixels and LCS is the local clarity score [2]

(2.5)

The LCS value of less than 0.15 represents a good quality ridge pattern, and value between 0.15 and 0.35 has intermediate quality with some noise present, a value between 0.35 and 0.55 has marginal quality with noise and anything above is a bad quality ridge pattern [2]. The global clarity score (GCS) can then be computed by Equation 2.6.

$$GCS = E(LCS(i,j)), \text{ where } E(\cdot\,) = \frac{\sum_{i=1}^{H}\sum_{j=1}^{V}(\cdot\,)}{H\cdot V}$$

LCS(i, j) is the clarity score at location (i, j), where i and j are the horizontal and vertical index of the image block and H and V are the maximum number of horizontal and vertical blocks, respectively [2]

(2.6)

The authors also examined the orientation flow of the fingerprint in local blocks. A 2D array (V_4) holds all the angles from the fingerprint. For a single block, the absolute difference of the orientation angle with the surrounding blocks is utilized. The Local Orientation Quality (LOQ) can be calculated as:

$$LOQ(i,j) = \frac{\sum_{m=-1}^{1}\sum_{n=-1}^{1}|V_4(i,j) - V_4(i-m,j-n)|}{8} \tag{2.7}$$

The local orientation quality score (LOQS) can then be defined as:

$$LOQS(i,j) = \begin{cases} 0, LOQ(i,j) \leq 8^\circ \\ \dfrac{LOQ(i,j) - 8^\circ)}{90^\circ - 8^\circ}, LOQ(i,j) > 8^\circ \end{cases} \tag{2.8}$$

The Global Orientation Quality Score (GOQS) is produced by taking the average of the LOQS values. The GCS and GOQS scores are used to calculate the Overall Image Quality (OIQ) by the following:

$$OIQ = \omega_1\cdot(1 - GCS) + \omega_2\cdot(1 - GOQS)$$
$$\omega_1 + \omega_2 = 1$$

where ω_1 and ω_2 are the weights for GCS and GOQS

(2.9)

To test the performance of the proposed method, the proposed method and a human perception score were used on a database of 115 fingerprint images. The human perception was based upon the visual inspection of the following guidelines: (1) total number of false minutiae detected by the feature extraction program in [4], (2) total number of undetected minutiae not detected by the program and (3) the area of the fingerprint. The authors evaluated the results of the methods to determine if there was a monotonic increasing relationship between the human perception and the proposed method. The results showed there was an increasing monotonic trend and concluded that the proposed method is a reliable indicator for the quality of a fingerprint [2].

2.3 Pulsfier *et al.*

Pulsfier *et al.* created an objective computer based system, using three different softwares, to identify the different regions of a fingerprint to determine the quality [3]. The first software of Universal Latent Workstation (ULW) has the capability to create an image clarity map for the fingerprint. The map contains the colors of black, red, yellow and green to indicate the regions of the background, debatable ridge flow, debatable minutiae and definitive minutiae, respectively. The map is overlaid on top of the fingerprint with the fingerprint is still visible in the background [3]. The grayscale levels, brightness and contrast were adjusted to leave just the map colors visible. The image was saved in grayscale and opened in the GNU Image Manipulation Program (GIMP) software to convert the clarity map into color [18]. The map colors included black, blue, yellow and white which represent the background, debatable ridge flow, debatable minutiae and definitive minutiae, respectively. The image was saved for a pixel counting algorithm in the final step. The authors wrote a program in Mathematica® to produce the percentage of all the individual colors except black [19]. The quality of the fingerprint was determined to be the percentage of definite minutiae (white) in the fingerprint. The authors tested this procedure with different development procedures, such as ninhydrin and cyanoacrylate, and the quality scores varied for these fingerprints methods [3]. Between the original fingerprint and those processed with development procedures, the results fluctuated and no trend could be seen. Only fingerprints that have been photographed can be used at this point in time to reduce any error.

3. Technical Note

3.1 Introduction

Fingerprint quality has been the subject of research for the last several years. During this time, researchers have developed their own algorithms for assessing the quality of fingerprints such as [3], [2], [20], and [21]. On the other side, other researchers have looked into using already available software fingerprint quality such as with [3].

However, no research has taken full advantage of National Institute of Standards and Technology (NIST) Image Software (NBIS) to compare subjective and objective methods of assessing quality. NBIS provides fingerprint capabilities including detecting minutiae, creating a local quality map and producing a global quality score for a fingerprint image. The software has the capability to provide an objective quality to an entire fingerprint as a whole or provide a quality for a fingerprint image for partial involvement of removing poor quality areas without a user interacting. The objective method can then be compared to a subjective method, which is an analysis performing a similar method of assigning a quality to an image and removing poor quality portions of the fingerprint. The goal of this paper is to perform and test the steps necessary in implementing the software properly and creating the necessary criteria to ensure consistency in both the subjective and objective methods of successive quality removal.

3.2 Methods

3.2.1 National Institute of Standards and Technology Biometric Image Software (NBIS)

National Institute of Standards and Technology Biometric Image Software (NBIS) is an open source software that contains the seven packages of PCASYS, MINDTCT, NFIQ, AN2K7, IMGTOOLS, NFSEG and BOZORTH3. The packages of importance for the research project are MINDTCT, NFIQ and BOZORTH3. MINDTCT is a minutiae detection algorithm that assesses each fingerprint and produces several files including a direction map, a low contrast map, a low flow map, a high curve map, a quality map and a list of minutiae locations and quality. The quality map produces a single score for every 8x8 pixel block of the fingerprint image on a scale of zero to four with a value of zero corresponding to a poor quality region and a value of four corresponding to a good quality region. The quality scores can be utilized to objectively remove potions of a fingerprint at a later stage. NFIQ is a global quality assessment algorithm that assesses the entire image and returns a score on a scale of one to five with one representing a good quality image and five representing a low quality image. The final algorithm, BOZORTH3, is a minutiae-based matching algorithm that assesses the location of the minutiae against the minutiae files from the entire gallery of fingerprint images. The software packages were downloaded based on the instructions provided by NIST [5, 6].

3.2.2 Image Requirements

NBIS requires certain specifications for fingerprint images in order to be accepted by the different algorithms. These specifications were tested on a set of 45 latent fingerprint test images of varying qualities. Fingerprint images were then utilized to test MINDTCT, the minutiae detection algorithm, and NFIQ, the global quality assessment algorithm.

3.2.2.1 Dimension of Images

Although not required, image dimensions should be a factor of eight due to the algorithm assessing the image with an 8x8 pixel kernel. Each image was opened individually into Adobe® Photoshop® to observe the image dimensions. The pixel dimensions were calculated to determine if the height and width, in pixels, were a factor of eight. If they were not, the amount of pixels needed to be added to the height, width or both was determined. The *canvas size*, located in the *resize* menu under the *image* tab, was selected and the dropdown menu for the height and width were selected to set the units to pixels. In the anchor section of the *canvas size* dialog box, the left middle square is chosen to allow for pixels to be added to the right side and/or the bottom of the image. The dimensions are changed in the boxes for the height and width to reflect the new dimensions, if necessary. The resulting fingerprint images were saved to indicate the image was padded.

To automate this process, R® via RStudio® was used to create a script to pad each image without user interference. The padding script (Appendix A.1) allows each fingerprint to be read into the program and individually checks the image dimensions. If the image does not have dimensions that are a factor of eight, the script calculates the number of pixels needed for the right and/or the bottom of the image. The resulting script was tested with the 45 images and found to produce the same results as manual padding in Adobe® Photoshop®.

3.2.2.2 Resolution

Another factor considered was the resolution, in pixels per inch (ppi), of the fingerprint images. According to NIST documentation, 500 ppi is the standard typically used by AFIS or other capture devices. However, the American National Standard for Information Systems - Data Format for the Interchange of Fingerprint, Facial, and Other Biometric Information (ANSI) recommends using a resolution of 1000 ppi due to more detail in the ridges and the structure being present [7]. Based on this recommendation, a resolution of 1000 ppi was used for the fingerprint images. The 45 test fingerprint images were checked with Adobe® Photoshop® to determine if the fingerprint images were the correct resolution of 1000 ppi. The images were checked by opening the *image size* dialog box located under the *image* tab. The resolution was changed to 1000 ppi, under the *resize* menu in the *image* tab, and the constrain proportions box was unchecked to ensure the image dimensions are not altered.

To automate this process in R® via RStudio®, the padding script had an additional statement added to the reading out of the image to ensure the resolution of 1000 ppi (Appendix A.1).

3.2.2.3 Color Mode

The final factor tested was the color mode of the images. The NBIS algorithms are designed to accept only 256 levels of 8-bit grayscale images otherwise the fingerprint image is rejected. Three different methods were tested to determine which program would provide the easiest process to ensure the fingerprint images were grayscale before use in the algorithms. The first program tested was Adobe® Photoshop®. In the *mode* menu under the *image* tab, grayscale was selected and the image was saved.

The fingerprint images were also processed with the R® padding script to determine if the script could be used to save the fingerprint images out as grayscale instead of using another program after. In the output of the image, the channel of the fingerprint was set to gray for the image when saved.

The final program tested was ImageJ, which is an open source image processing program. Each image was opened in the program and the *image tab* is selected. The *type sub-tab* was chosen and 8-bit selected to allow the fingerprint image to be grayscale. The steps for this process was automated with a macro as seen in Appendix A.2.

3.2.3 Determination of Category Criteria

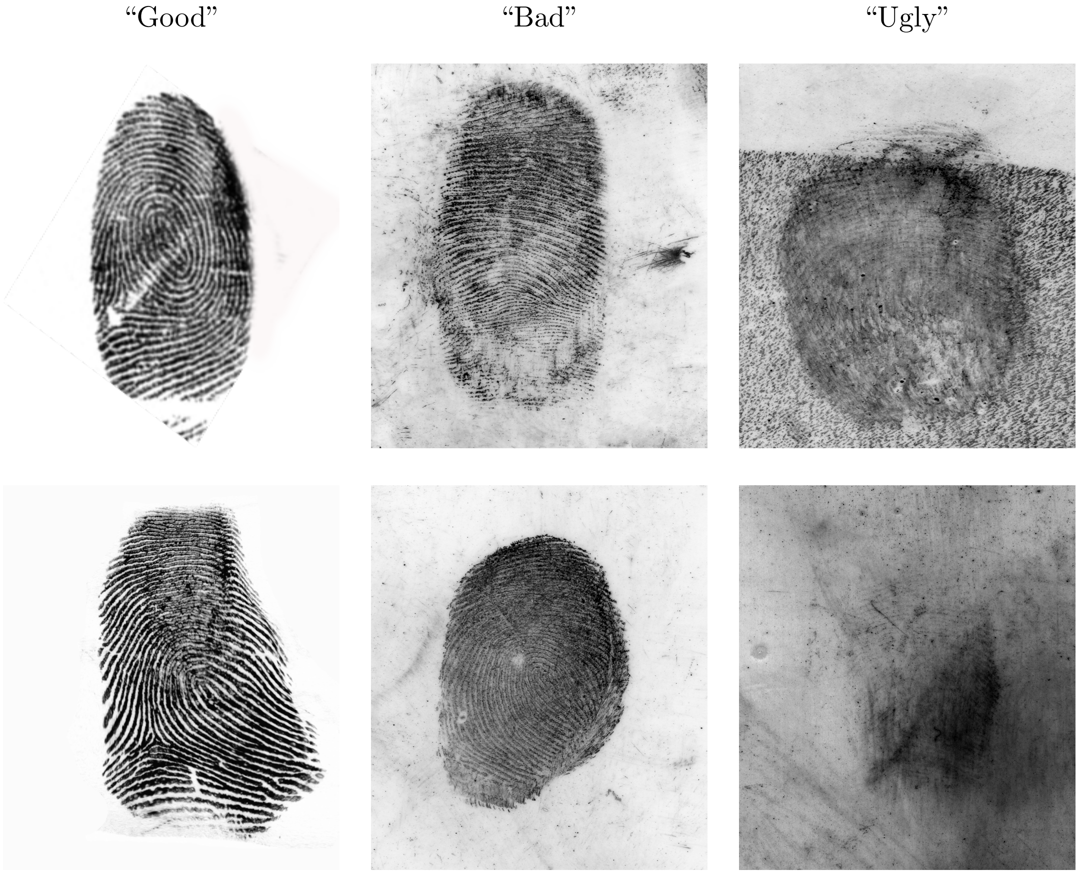
In the subjective portion of the research, each fingerprint was placed into a qualitative category of "good," "bad," or "ugly." The following criteria were developed based upon viewing images and observing their characteristics:

1. "Good" - Images that have clear and visible ridge detail and may have some interfering background.

2. "Bad" - Images that have interfering background and have some ridge detail that is smudged or blurry.

3. "Ugly" - Images that have some to no ridge detail or are completely unreadable.

Figure 3.1 provides visual examples of each category. The categories were tested on 45 test images.



“Good” “Bad” “Ugly”

Figure 3.1: Examples of fingerprints (from left to right) that correspond to the categories of “good,” “bad,” and “ugly.”

3.2.4 Determination of Removal Criteria

To achieve the objective portion of the research, the MINDTCT algorithm was utilized. The 45 testing images with the correct size, dimension, and color mode were assessed by the MINDTCT algorithm to obtain the quality map. The quality map contains values on the scale of zero to four with zero being a poor quality region and four being a high quality region. The map produces a single quality score for an 8x8 pixel block. An R® script was written to expand each single value to the entire 8x8 pixel block in the fingerprint image (Appendix A.3). The resulting expanded map would be the same size as the image.

A second script was written to successively remove the quality map areas from the original image (Appendix A.4). Each value of zero in the quality map are changed to the white pixel value of 255 in the original image. This image is saved and denoted as "Area 0." Next, the values of one were changed to white pixels on the Area 0 image and saved as "Area 1." This process was continued for the remaining values with the quality value of four resulting in a blank, white image. The R® removal script was tested on the 45 test images and assessed with the NFIQ global quality assessment algorithm to obtain quality scores.

The resulting images from the removal code were compared to the corresponding original image to create subjective removal criteria. The original image was opened and visually compared to the Area 0 image. The Area 0 images showed the background removed from around the fingerprint. In some cases, some regions of the background were left behind and contained straight lines or black spots. For Area 1 images, the entire background was removed with only the fingerprint remaining. Area 2 images appeared to have areas that were either too dark or too light to show ridge detail removed. In the final case of Area 3 images, the fingerprint images showed only the regions where the ridges were either smudged or blurry being "whited" out. With these observations, the following criteria were formed for the subjective removal:

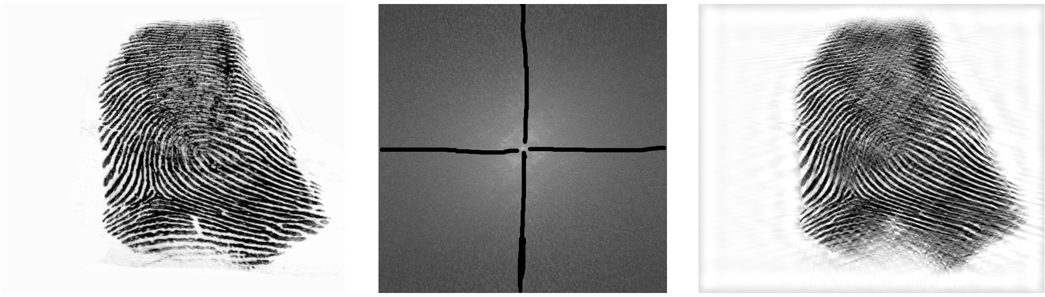
1. Area 1 - The true background of the image with the exception of any shape that maybe mistaken as ridge formations such as lines or spots.

2. Area 2 - The remaining background of the image and any region that does not show ridge detail. Also, parts of the fingerprint that may not show details along the outer edge of the fingerprint where the ridges may not have made contact.

3. Area 3 - Any regions of the fingerprint too dark or too faint to determine any ridge detail.

4. Area 4 - Any regions of the fingerprint where the ridges are smudged and/or blurry as well as any regions where the ridges are spotty and cannot be traced.

The criteria were tested with six images from the test set and assessed with the NFIQ global quality assessment algorithm.



(a) The original fingerprint image.

(b) The power spectrum of the original spectrum with sections blacked out .

(c) The resulting fingerprint image inverting the power spectrum.

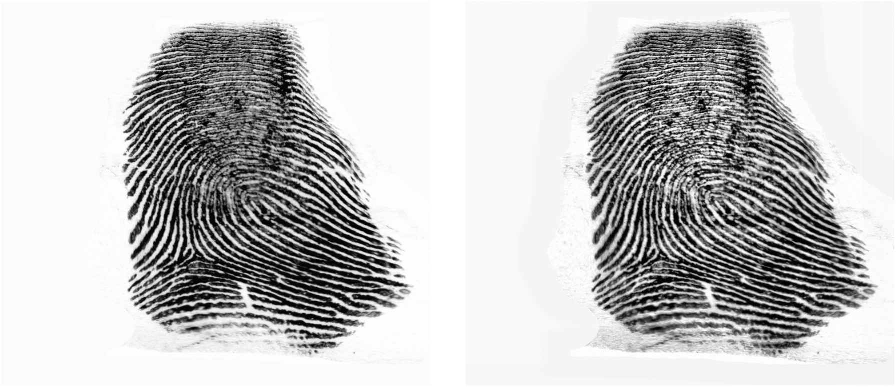
Figure 3.2: These images give visualization to the steps of Fast Fourier Transform (FFT). The original image can be seen in A. In B, the power spectrum of the original image and the blacking out of different frequencies in the power spectrum. Image C shows the resulting image after performing the inverse FFT with the power spectrum.

3.2.5 Image Pre-Processing Tests

A question proposed was whether or not pre-processing a fingerprint image can improve or hurt the NFIQ global quality score of the images. To test this question, six images with a variety of quality scores were processed in ImageJ with the Fast Fourier Transform (FFT) function. FFT produces a power spectrum that can have sections "blacked out" to remove different frequencies in the image. The images for testing had the horizontal and vertical lines removed, such as in the case of Figure 3.2.B. The inverse FFT of power spectrum is taken to produce an image with some frequencies removed (Figure 3.2.C) and assessed by the NFIQ global quality assessment algorithm to receive a quality score.

The same six images were also pre-processed in ImageJ using an enhanced local contrast algorithm. This algorithm utilizes the method of contrast limited adaptive histogram equalization (CLAHE) to enhanced the image based upon a specified blocksize and not the entire image at one time [22]. The parameters used for this method included a blocksize of 127, histogram bins of 256, a max slope of three, and performed slow. The *blocksize* corresponds to the size of the area around a pixel that is to be equalized. The *histogram bins* relates to the number of histograms bins to be used for the equalization. In the case of the *max slope*, the slope limits the range of histogram values when the pixels are remapped. A value of one will result in the original image whereas a high number could result in a high contrast image. The function of *slow* allows the function to perform more accurately [22].

Figure 3.3 shows an example of before and after CLAHE was applied to an image. The final images were assessed with the NFIQ global quality assessment algorithm to determine if the quality scores changed.



(a) The original fingerprint image.

(b) The fingerprint image after having CLAHE performed on the image.

Figure 3.3: These images give visualization of the before and after of the CLAHE preprocessing. The original image can be seen in A. In B, the result after having CLAHE performed on the image.

3.2.6 Training of the NFIQ Algorithm

The NFIQ global quality assessment algorithm is a global assessment of fingerprint images. The algorithm had previously been trained by the scientists at NIST; however, documentation provided stated that the algorithm was not trained with a high amount of low quality images. To ensure that lower quality images are not misclassified by the algorithm, the quality algorithm was trained with a new set of fingerprint images. The fingerprint images used for the training and testing of the algorithm were fingerprint images from the Universal Latent Workstation (ULW) from the Federal Bureau of Investigation. In this program, images were in the predefined quality categories of good (84 images), bad (86 images) and ugly (85 images). All of the fingerprint images were reclassified into the following categories based on the associated criteria:

1. Good Images (NFIQ Quality Equivalent of One) - Images considered to have clear and good ridge detail with limiting background interference.

2. Okay Images (NFIQ Quality Equivalent of Two) - Images considered to have visible ridge detail and have an interfering background.

3. Bad Images (NFIQ Quality Equivalent of Three) - Images considered to have blurry ridge detail but the ridges can be traced.

4. Ugly Images (NFIQ Quality Equivalent of Four) - Images considered to have some ridge detail that can be traced.

Table 3.1: This table shows description for an 11-dimension feature vector that is produced for each fingerprint by the *Fing2pat* algorithm. The symbol Q represents quality from the quality map and the symbol Q_m represents minutiae quality.

	NAME	**DESCRIPTION**
1	Foreground	Number of blocks that are $Q \geq 1$; i.e. foreground $= \sum_{i=1}^{4} U_i$ where U_i is the number of blocks with quality i
2	Total minutiae	Total minutiae found in fingerprint
3	Min05	Count of minutiae with $Q_m \geq 0.5$
4	Min06	Count of minutiae with $Q_m \geq 0.6$
5	Min075	Count of minutiae with $Q_m \geq 0.75$
6	Min08	Count of minutiae with $Q_m \geq 0.8$
7	Min09	Count of minutiae with $Q_m \geq 0.9$
8	Quality zone 1	% of the Foreground Blocks with Q=1
9	Quality zone 2	% of the Foreground Blocks with Q=2
10	Quality zone 3	% of the Foreground Blocks with Q= 3
11	Quality zone 4	% of the Foreground Blocks with Q= 4

5. Unreadable Images (NFIQ Quality Equivalent of Five) - Images considered to not have any useful information present.

With the reclassification, the quality spread was 42 good images, 40 okay images, 71 bad images, 51 ugly images, and 26 unreadable images. The NFIQ global quality assessment algorithm was trained with 205 images and tested with 25 images. The test set contained five images selected from each quality category. A list file was created with the pathway to each fingerprint image and the corresponding quality score. Good images were assigned values of one and each category increased by a value of one with the unreadable images assigned the values of five. The list file is assessed by the *fing2pat* function to create an 11-dimension vector for each image. Table 3.1 shows how each value is calculated. The resulting *fing2pat* file is normalized with global values, provided by NIST, with the function *znormpat*. The steps are repeated for the test images.

The normalized file locations for the testing and training were updated in the NFIQ spec file. This spec file contains all the information that the algorithm requires to execute properly. To further test the algorithm in an attempt to lower the misclassification rate, the parameters of the prior and pattern weights were adjusted to various values. The *prior weights* control the amount of impact that the patterns have on the algorithm whereas *pattern weights* allow for the adjustment of the priors to account for unnatural frequencies. Table 3.2 shows the different value combinations attempted. The resulting lowest misclassification was kept and used to train the algorithm.

Table 3.2: The following table presents the different prior and pattern weights that were attempted. Various combinations of the priors and pattern weights were utilized to find a combination to lower the misclassification rate of the NFIQ global quality assessment algorithm.

Quality Score	Prior Weights										
1	0.2	0.1	0.25	0.10	0.15	0.10	0.30	0.30	0.35	0.15	0.15
2	0.2	0.1	0.25	0.10	0.15	0.10	0.25	0.30	0.10	0.15	0.15
3	0.2	0.6	0.20	0.35	0.40	0.20	0.10	0.10	0.25	0.25	0.25
4	0.2	0.1	0.10	0.10	0.15	0.10	0.25	0.20	0.10	0.20	0.25
5	0.2	0.1	0.20	0.35	0.15	0.50	0.10	0.10	0.20	0.2 5	0.10
1	0.17	0.15	0.16	0.15	0.16	0.16	0.15	0.15	0.15	0.15	0.145
2	0.17	0.17	0.17	0.17	0.17	0.16	0.18	0.19	0.20	0.19	0.19
3	0.23	0.25	0.26	0.27	0.27	0.27	0.27	0.27	0.27	0.26	0.27
4	0.23	0.25	0.24	0.23	0.22	0.23	0.23	0.22	0.21	0.23	0.215
5	0.20	0.18	0.18	0.18	0.18	0.18	0.17	0.17	0.17	0.17	0.17

Quality Score	Pattern Weights					
1	0.5	0.25	0.25	0.25	0.30	0.30
2	0.65	0.30	0.45	0.45	0.45	0.45
3	1.00	1.00	0.75	0.50	0.25	0.10
4	0.65	0.30	0.45	0.45	0.45	0.45
5	1.00	1.00	0.75	0.50	0.25	0.10

3.2.7 BioCop Database Formation

Fingerprints (both latent fingerprints and ten-print cards) were obtained from the BioCop 2008 database. The database consists of:

1. 1000 ppi rolled ten-print cards from 972 participants
2. Crossmatch (live scanner) ten-print cards from 1,135 individuals

   For each individual, there may be more than one ten-print card
3. Latents fingerprints with various types of development methods from a variety of substrates from 1,135 individuals

Each rolled ten-print card folder and Crossmatch folder contained EBTS files and images of each individual finger from the ten-print card. The latent fingerprint folders contained a variety of images including the fingers, the joints, and parts of the palm. For the purpose of this research, only the fingers were utilized. The individual fingers from the Crossmatch and the rolled sets were processed using the padding (Appendix A.1) and grayscale (Appendix A.2) codes for future use.

3.2.7.1 AFIX Tracker® Database

Ten-print cards were created with the individual fingers from each participant and coded to be the same dimensions (three inches by eight inches) as the rolled section of a standard FBI fingerprint card [23]. The R® script can be seen in see Appendix A.5. The R®-created ten-print cards were uploaded to AFIX Tracker,® and each participant was given a unique letter combination. The letter combinations and its corresponding participant were tracked in a text file. Each fingerprint card was automatically extracted by the program when uploaded. A total of 1,148 participants and a total of about 3,234 ten-print cards were uploaded to create the database.

3.2.7.2 BOZORTH3 Database

The individual fingers that were previously padded and converted to grayscale were placed into one directory. A list of all the images along with their path location were created using an R® script (Appendix A.6).

3.3 Results

3.3.1 Image Requirements

3.3.1.1 Color Mode

When the color mode was tested with Adobe® Photoshop,® the fingerprint images presented to be in grayscale based on the *mode* menu under the *Image* tab. Each image was

Table 3.3: This following table presents the category breakdown for the 45 test images. The fingerprints in bold represent the fingerprint images chosen to be used for pre-processing test.

Good	Bad			Ugly
A	C	1h	4b	2c
B	E	2a	4c	2e
D	**G**	2d	4e	2f
F	1b	3a	4h	4a
G1	1c	3b	5c	5a
1a	1d	3c	5d	5b
4d	1e	3d	5f	5e
4f	1f	3e	5g	
4g	**1g**	3f	5h	
		6a		

tested with the NFIQ global quality assessment algorithm to determine if the image would be accepted and produce a quality score. In some cases, the fingerprint images would be accepted and a global quality score was obtained. In other cases, the fingerprint images would cause an error stating that the image was not 8-bit. The images that caused the error were opened in Adobe® Photoshop® again to examine if there could have been an error with converting the image. However, when the fingerprint images were checked in Adobe® Photoshop,® they showed to be 8-bit grayscale images. The images were tested again with the algorithm and an error was still received. It is undetermined as to why these images caused an error when they were processed in the same manner as the other images that received a score. The Adobe® Photoshop® method was removed from the potential list of methods.

The fingerprint images processed with the R padding script were tested with the NFIQ global quality assessment algorithm. All the images produced an error stating that the image was not 8-bit. The images were checked in ImageJ and showed to be RGB images instead of grayscale images. With these results, the option of using the R® script to convert to 8-bit grayscale will not be used.

The images processed with ImageJ were tested and able to obtain an NFIQ global quality score. This program was chosen to ensure that all images used in this research would be grayscale.

3.3.2 Determination of Category Criteria

The categorization of the 45 test images produced a breakdown of nine good images, 29 bad images, and seven ugly images. Table 3.3 shows the category breakdown.

3.3.3 Determination of Removal Criteria

The resulting NFIQ values for the images and their counterparts showed surprising results. Table 3.4 shows the fingerprint name and the areas removed with the corresponding NFIQ value. Area 4 images presented the expected results of NFIQ values of five due to the images being completely white. Some fingerprint images did not produce the expected NFIQ value when observing the quality scores for the different areas removed of the fingerprints (Table 3.5). In the case of fingerprints F and G1, the subjective image received a quality score of five whereas in the subjective they were classified as good quality images. The same was seen in the ugly fingerprints of 2c, 2e, and 5e where the fingerprints were categorized as ugly but received a quality score of one. When visualizing some of these fingerprints, it was apparent that the algorithm may be detecting sections of the background in the images and believing that it is part of the fingerprint factoring into the quality score. Several subjective images were categorized as bad quality fingerprints, 17 fingerprints as seen in Table 3.5, were classified with the value of one. With many of the bad quality images being classified with a value of five, the algorithm may be misclassifying these fingerprints based upon a low number of ugly quality fingerprints used in training the algorithm previously used.

The quality score results, when observing the successive decrease in area, were mixed. In cases where the image quality score stayed consistent in the decreasing removal, the quality values from the original images varied. In the case of fingerprints A and B, the images being with a quality score of one and remained that value until reaching Area 4. In the other case, latent fingerprints G1, 1e, 1f and others began with a quality score of five and remained a score of five.

Some of the results were surprising when looking at the different areas. In the latent fingerprints gave a value of one for Area 0 and the next area removed produced a quality score of five (Table 3.6). This was an extreme change in the quality score and the fingerprint images were examined. The fingerprint images did not show a significant difference from one area to another which would cause such an extreme shift in quality score. It is unclear as to why the NFIQ global quality assessment algorithm would make an extreme change in the quality value. With the variety of results obtained from the fingerprint images, the NFIQ global quality assessment algorithm would be trained with a larger sample size than was previously used by NIST.

For the six images that had quality areas manually removed, the results were consistent with the objective method (Table 3.7). Latent D gave an original quality score of four and the successive removal produced values of four with the exception of Area 3 with a score of three. This was the same results seen in the objective removal of areas. The other fingerprints seen the same results as latent fingerprint D with being similar to those in the objective removal of areas. This shows a consistency in the removal of samples, but more testing needs to be performed with more samples to show how similar or different the two methods are from one another.

Table 3.4: The resulting NFIQ global quality assessment scores for fingerprints images that have had areas objectively removed using the R$^{®}$ script.

Name	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**G1**	**1a**	**1b**	**1c**	**1d**	**1e**	**1f**	**1g**
Original	1	1	1	4	3	5	1	5	3	5	5	5	5	5	1
Area 0	1	1	1	4	3	5	5	5	3	5	5	5	5	5	5
Area 1	1	1	5	4	3	5	1	5	3	5	5	5	5	5	1
Area 2	1	1	5	3	4	5	1	5	4	5	1	1	5	5	5
Area 3	1	1	5	3	5	5	5	5	5	5	5	5	5	5	5
Area 4	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
Name	**1h**	**2a**	**2b**	**2c**	**2d**	**2e**	**2f**	**3a**	**3b**	**3c**	**3d**	**3e**	**3f**	**4a**	**4b**
Original	1	1	4	1	5	1	1	1	1	5	1	1	5	5	5
Area 0	1	5	5	5	5	5	1	1	5	5	1	5	5	5	5
Area 1	5	1	1	1	1	1	5	1	1	1	4	5	5	5	5
Area 2	5	1	1	5	5	5	5	5	5	5	5	5	5	5	5
Area 3	5	5	5	5	5	5	5	5	1	5	5	5	5	5	5
Area 4	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
Name	**4c**	**4d**	**4e**	**4f**	**4g**	**4h**	**5a**	**5b**	**5c**	**5d**	**5e**	**5f**	**5g**	**5h**	**6a**
Original	5	2	5	1	1	5	5	5	5	5	1	5	1	5	5
Area 0	5	2	5	2	3	5	5	5	1	5	1	5	2	1	1
Area 1	5	2	5	5	3	5	5	5	5	5	1	5	1	5	3
Area 2	5	3	5	3	1	4	5	5	5	5	5	5	5	4	5
Area 3	5	5	5	5	2	5	5	5	5	5	5	5	5	5	5
Area 4	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5

Table 3.5: A comparison of the subjective category versus the objective quality obtained from the NFIQ global quality assessment algorithm for the original images.

Objective Subjective	1	2	3	4	5
Good	A, B, 4f, 4g	4d	1a	D	F, G1
Bad	C, G, 1g, 1h 2a, 3a, 3b, 3d 3e, 5g	-	E	2b	1b, 1c, 1d, 1e, 1f, 2d 3c, 3f, 4b, 4c, 4e, 4h 5c, 5d, 5f, 5h, 6a
Ugly	2c, 2e, 2f, 5e	-	-	-	4a, 5a, 5b

Table 3.6: A comparison of the subjective category versus the objective quality obtained from the NFIQ global quality assessment algorithm for the original images.

Area 0 Original	1	2	3	4	5
1	A, B, C, 1h, 2f, 3a, 3d, 5e	4f, 5g	4g	D	G, 1g, 2a, 2c 2e, 3b, 3e
2	-	4d	-	-	-
3	-	-	E, 1a	-	-
4	-	-	-	D, 2b	-
5	5c, 5h, 6a	-	-	-	F, G1, 1b, 1c, 1e, 1f 3c, 3f, 4a, 4b, 4c, 4e 4h, 5a, 5b, 5d, 5f

Table 3.7: The resulting NFIQ global quality assessment scores for fingerprint images that have had areas subjectively removed based on the set forth criteria.

Name	**D**	**F**	**G**	**1g**	**4d**	**4g**
Original	4	5	1	1	2	1
Area 0	4	5	1	5	2	1
Area 1	4	5	1	2	2	3
Area 2	4	5	4	5	2	1
Area 3	3	5	5	5	5	2

Table 3.8: A comparison of the NFIQ global quality assessment scores produced for the images pre-processed with the ImageJ methods.

Name	D	F	G	1g	4d	4g
Original	4	5	1	1	2	1
FFT	5	5	3	1	4	4
CLAHE	4	5	5	5	3	3

3.3.4 Image Pre-Processing Tests

The fingerprint images pre-processed with the first ImageJ method of FFT showed that the quality score either decreased (fingerprints D, G, 4d, and 4g) or stayed the same quality score (fingerprints of F and 1g) (Table 3.8). With these results, it showed that the pre-processing method of FFT masking does not improve the quality of the fingerprint image, but tends to hurt the outcome. More fingerprint images should be tested to determine if this pattern holds true.

The results for the second method of CLAHE showed mixed results (Table 3.8). The fingerprints of D and F produced the same quality score as their original fingerprint, quality scores four and five, respectively. The fingerprint images of G and 1g, however, showed to have an extreme decrease. Both cases produced an original quality score of one whereas the CLAHE image produced a score of five. The observations of these fingerprint images do not show any large differences between the images. In the cases of latent fingerprints D, 4d and 4g, the quality scores were higher in the CLAHE than in the FFT method. These were not extreme changes, just an increase in a quality value of one or two. With these results, neither method showed to have better results than the other. If either method would be used, more samples would need to be tested to show if either method assisted in improving the quality score. However, at this time, the pre-processing methods do not enhance the quality scores with the NFIQ global quality algorithm.

3.3.5 Training of the NFIQ Algorithm

In the training of the algorithm, the different patterns attempted produced a confusion matrix as seen in Table 3.9. Images that are classified on the diagonal (from the top left to bottom right) and the cell above and below the diagonal were considered correctly classified due to the possibility that the fingerprint image may fit better in one of the those categories. The values found outside the specified cells were considered misclassified. The misclassification rate was calculated by summing the total number of misclassified images and dividing by the total number of images multiplied by 100. The goal was to find the lowest misclassification rate possible. The following table (Table 3.10) shows the lowest misclassification rates (under 18%) with the corresponding prior and pattern combination. The remaining misclassification rates and confusion matrices can be seen in Appendix B.1. The lowest misclassification rate found was 13.91% with the pattern weight distribution

Table 3.9: An example of a confusion matrix produced from training the NFIQ algorithm.

Actual **Experimental**	1	2	3	4	5
1	13	3	23	0	3
2	3	13	22	1	1
3	0	0	71	0	0
4	1	4	37	5	4
5	0	0	15	0	11

of 1 = 0.3, 2 = 0.45, 3 = 0.1, 4 = 0.45, and 5 = 0.1 and the prior distribution of 1 = 0.16, 2 = 0.16, 3 = 0.27, 4 = 0.23 and 5 = 0.18. This combination was kept to retrain the algorithm for future use.

Table 3.10: The following table presents the lowest misclassification rates (under 18%) along with the pattern weights and prior weights. The lowest misclassification rate was kept to train the NFIQ global quality assessment algorithm.

	Pattern	Prior	Pattern	Prior	Pattern	Prior	Pattern	Prior	Pattern	Prior
1	0.5	0.15	0.25	0.15	0.25	0.16	0.25	0.15	0.25	0.145
2	0.65	0.18	0.45	0.19	0.45	0.17	0.45	0.19	0.45	0.19
3	1	0.27	0.75	0.27	0.5	0.27	0.5	0.27	0.5	0.27
4	0.65	0.23	0.45	0.22	0.45	0.22	0.45	0.22	0.45	0.215
5	1	0.17	0.75	0.17	0.5	0.18	0.5	0.17	0.5	0.17
	17.82%		16.08%		17.82%		16.95%		17.82%	
1	0.3	0.15	0.3	0.15	0.3	0.15	0.3	0.2	0.3	0.3
2	0.45	0.18	0.45	0.2	0.45	0.19	0.45	0.2	0.45	0.25
3	0.25	0.27	0.25	0.27	0.25	0.26	0.1	0.2	0.1	0.1
4	0.45	0.23	0.45	0.21	0.45	0.23	0.45	0.2	0.45	0.25
5	0.25	0.17	0.25	0.17	0.25	0.17	0.1	0.2	0.1	0.1
	16.52%		17.82%		15.21%		16.95%		15.65%	
1	0.3	0.15	0.3	0.17	0.3	0.15	0.3	0.16	0.3	0.15
2	0.45	0.15	0.45	0.17	0.45	0.17	0.45	0.17	0.45	0.17
3	0.1	0.25	0.1	0.23	0.1	0.25	0.1	0.26	0.1	0.27
4	0.45	0.25	0.45	0.23	0.45	0.25	0.45	0.24	0.45	0.23
5	0.1	0.2	0.1	0.2	0.1	0.18	0.1	0.18	0.1	0.18
	17.39%		16.52%		16.52%		14.78%		15.21%	
1	0.3	0.16	0.3	0.16	0.3	0.15	0.3	0.15	0.3	0.15
2	0.45	0.17	0.45	0.16	0.45	0.18	0.45	0.19	0.45	0.2
3	0.1	0.27	0.1	0.27	0.1	0.27	0.1	0.27	0.1	0.27
4	0.45	0.22	0.45	0.23	0.45	0.23	0.45	0.22	0.45	0.21
5	0.1	0.18	0.1	0.18	0.1	0.17	0.1	0.17	0.1	0.17
	15.65%		**13.91%**		14.78%		16.08%		14.78%	
1	0.3	0.15	0.3	0.145						
2	0.45	0.19	0.45	0.19						
3	0.1	0.26	0.1	0.27						
4	0.45	0.23	0.45	0.215						
5	0.1	0.17	0.1	0.17						
	16.95%		16.52%							

4. The Assessment of Fingerprint Quality

4.1 Introduction

One of the most frequented pieces of evidence recovered from a crime scene is latent fingerprints and these may not be of the best quality. Quality of a latent fingerprint can be described as how well the ridge details in the fingerprint can be observed. There is a wide-range of factors that can affect the quality of the fingerprint and knowing these factors can greatly improve the comparison process. If the quality of the latent fingerprint is very clear, an examiner is able to identify different features. If the latent fingerprint is of a poorer quality, the examiner may find few to no features for comparison. The same is true when utilizing minutiae-based matching algorithms such as AFIX Tracker® of AFIX Technologies and BOZORTH3 from the National Institute of Standards and Technology (NIST). If a latent fingerprint is of a low quality, there is a lower likelihood that AFIX Tracker® or BOZORTH3 can detect minutiae. If these algorithms do detect minutiae, it is possible that the minutiae are not true minutiae.

In recent years, there has been more attention given to estimating the quality of a latent fingerprint to assist in improving the match scores [1, 2, 3, 4]. Some of these studies have created their own methods of estimating the quality, but none have fully utilized the potential of the NIST Open Source Software, NBIS [5, 6]. In this research study, we aim to determine a method of increasing the match score of the latent fingerprint by removing bad quality areas and keeping good quality areas intact by utilizing a subjective and objective method. The subjective method will consist of a single examiner assigning a quality category of “good,” “bad” or “ugly” and then successively removing regions of the latent fingerprint based on criteria set forth in Section 3.2.3. The objective method consisted of the latent fingerprint being assessed by the NFIQ quality algorithm (from NIST) to receive a quality score. The latent fingerprint were examined by the MINDTCT algorithm (from NIST) to obtain the detected minutiae and a quality map. The resulting quality map was used to successively remove different quality portions from the latent fingerprint. The study determined if there is a significant difference between the two minutiae-based matching algorithms of AFIX Tracker® and BOZORTH3 as well as how the match scores for the different quality portions removed are effected. To analyze the results from the subjective and objective methods, the data analysis methods of receiver operator characteristic (ROC) curves, area under the ROC curve, cumulative match characteristic (CMC) curves, Wilcoxon signed-rank test and Spearman's rank correlation was utilized. With the removing of the lower quality areas from the latent fingerprints, the match scores in AFIX Tracker® matching algorithm showed an increase with the subjective method achieving overall higher results than the objective method.

4.2 Methods

4.2.1 Selection of Images

For this research, 500 latents fingerprints were randomly selected from the BioCop 2008 database latent print folders using R® via RStudio® (Appendix C.1). The BioCop 2008 database was created and maintained by West Virgina University. The resulting fingerprints were a variety of finger numbers, substrates, development methods and quality. The selected latent fingerprints were processed with the R® padding code and the ImageJ grayscale code (Appendices A.1 and A.2). A total of 50 fingerprints have been processed with both the subjective and objective methods.

4.2.2 Subjective Method

Each of the randomly selected fingerprint images were opened in Adobe® Photoshop® and assessed. If the fingerprint was processed with cyanoacrylate, the colors were inverted to allow the fingerprint to be black and the background white. If the fingerprint image contained multiple fingerprints, each of the fingerprints were examined and the fingerprint with the lowest quality was kept. In cases where two samples of the same name and finger were selected, the two images were evaluated to determine if the fingerprint images were of different qualities. If they were of different qualities, both images were kept; however, if they were of the same quality only one image was kept. All images were cropped if an excessive amount of background or if multiple fingerprints were present.

Each fingerprint was assigned a qualitative quality category of "good," "bad," or "ugly" based on the criteria set forth in Section 3.2.3. The areas of zero through four were then successively removed based upon the criteria in Section 3.2.4. Each resulting image was saved to a folder respective to the area removed. After the 50 images, were completed, each folder was processed with the R® padding code and the ImageJ grayscale code (Appendices A.1 and A.2).

4.2.3 Objective Method

Each randomly selected image was assessed with the trained NFIQ global quality assessment algorithm to obtain a global quality score of one through five with a quality score of one representing a good quality image and five representing an ugly quality image. Each image was then assessed with the MINDTCT minutiae detection algorithm to obtain the quality map. The resulting quality map was read into R® via RStudio® to expand the quality map to the size of the fingerprint image with the previously written R® expansion code (Appendix A.3). The expanded map was then used to successively remove the different quality values of zero through four with zero representing a low quality area and 4 representing a high quality area in the original image using the R® removal code (Appendix A.4). The resulting images were processed with the R® padding code and the ImageJ grayscale code (Appendices A.1 and A.2).

4.2.4 AFIX Tracker®

The original images and the removal images were uploaded to the AFIX Tracker® Crime Scene section two times. The first set of images were automatically extracted by the program while the second set of images were manually extracted by one user. The marked images were searched against the previously created database to return the maximum number of results (100). The resulting list was searched to determine if the correct result was presented. If the correct individual and finger were identified, the status of the correct image was changed to identified while the remaining images were marked as non-identified.

After the fingerprints were processed with AFIX Tracker,® the datum was extracted using R® via RStudio® with the AFIS code (Appendix C.2). The AFIS code pulls the information on the fingers, the search jobs, the match results, the number of minutiae marked, the status of the identified or non-identified in the match results, the database information and latent cases. For the purposes of this research, the latent cases and the finger files were individually pulled and saved along with the coded output to allow for the number of minutiae marked for each latent fingerprint to be extracted and saved using the R® code in Appendix C.2.

4.2.5 BOZORTH3

The original fingerprints and the removal images for both the subjective and objective methods were assessed with the MINDTCT minutiae detection algorithm to retrieve the minutiae files. Each fingerprint's minutiae file was searched against the gallery list of fingerprints created in Section 3.2.7.2. The search result returns a text file with a match score for each fingerprint in the gallery list. The text file was read into R® via RStudio® to append the gallery names to the match scores. The updated results file was sorted based on the match score in decreasing order. A rank column (in the increasing order of one to 32,303), a status column of "No," the area of removal for the particular image, and a method column of either subjective or objective were appended to the list. The gallery name column was searched for the sample name and finger number. If the match score was greater than zero, the corresponding status cell was changed to "Yes," else the status was left as "No." The match score and the rank of the match were extracted and placed into a results spreadsheet. This process was completed for all fingerprints searched.

4.2.6 Method Comparison

To test how the subjective and objective methods of removal compare to one another, the images from the corresponding area in the subjective and objective methods were compared. Each image of the corresponding area in subjective and objective were opened in R® via RStudio®. Each image was thresholded with the package "imager" [24]. Pixels values below 0.9 were changed to black and pixels above 0.9 were changed to white. The subjective image black pixels were then converted to gray.

For the union of the subjective and objective images, if statements were utilized. If a pixel value in the subjective image was gray (0.5) and the corresponding pixel location value in the objective image was black (zero), the resulting pixel in the new image would have a pixel color of black (zero). If the subjective image had a gray pixel color (0.5) and the objective image had a white pixel color (one) in the corresponding position, the new image would have a pixel color of dark gray (0.333). If the subjective image had a white pixel color (one) and the objective image had a black pixel color (zero), the new image would have a light gray pixel color (0.662). If both images had a pixel color of white, the new image would have a pixel color of white. This process was completed for every pixel in the images. The color black in the new image represented the regions where the both removal methods did not remove regions of the fingerprint. The color of light gray illustrates the regions were the objective method did not remove portions of the image, and the dark gray represents the regions were the subjective method did not remove portions of the image. The white pixels show regions where the image has been removed by both methods. An example can be seen in Figure 4.1.

In the intersection of the subjective and objective methods, only portions that both methods did not mark out were kept. If the subjective image had a pixel color of gray (0.5) and the objective image had a pixel color of black (zero), the new image would have a pixel value of zero. All other combinations would result in a pixel value of one. An example can be seen in Figure 4.2.

4.2.7 Data Analysis Methods

After the match scores and the ranks have been collected for the latent fingerprints from the subjective and objective methods, the program R® via RStudio® was utilized to analyze the data. The statistical methods of receiver operator characteristics (ROC) curves, area under the ROC curve (AUC), cumulative match characteristic (CMC) curve, Wilcoxon Signed-Rank test and Spearman's Rank-Order correlation were used to assess the effect of the removal methods and minutiae-based matching algorithms on the results obtained.

4.2.7.1 Receiver Operator Characteristics (ROC) Curves

ROC curves are often used in the medical field to assess the validity of laboratory tests [25, 26, 27]. A ROC curve is a plot of a system's true positive rate against the system's false positive rate where the true positive rate represents the percentage of time that the

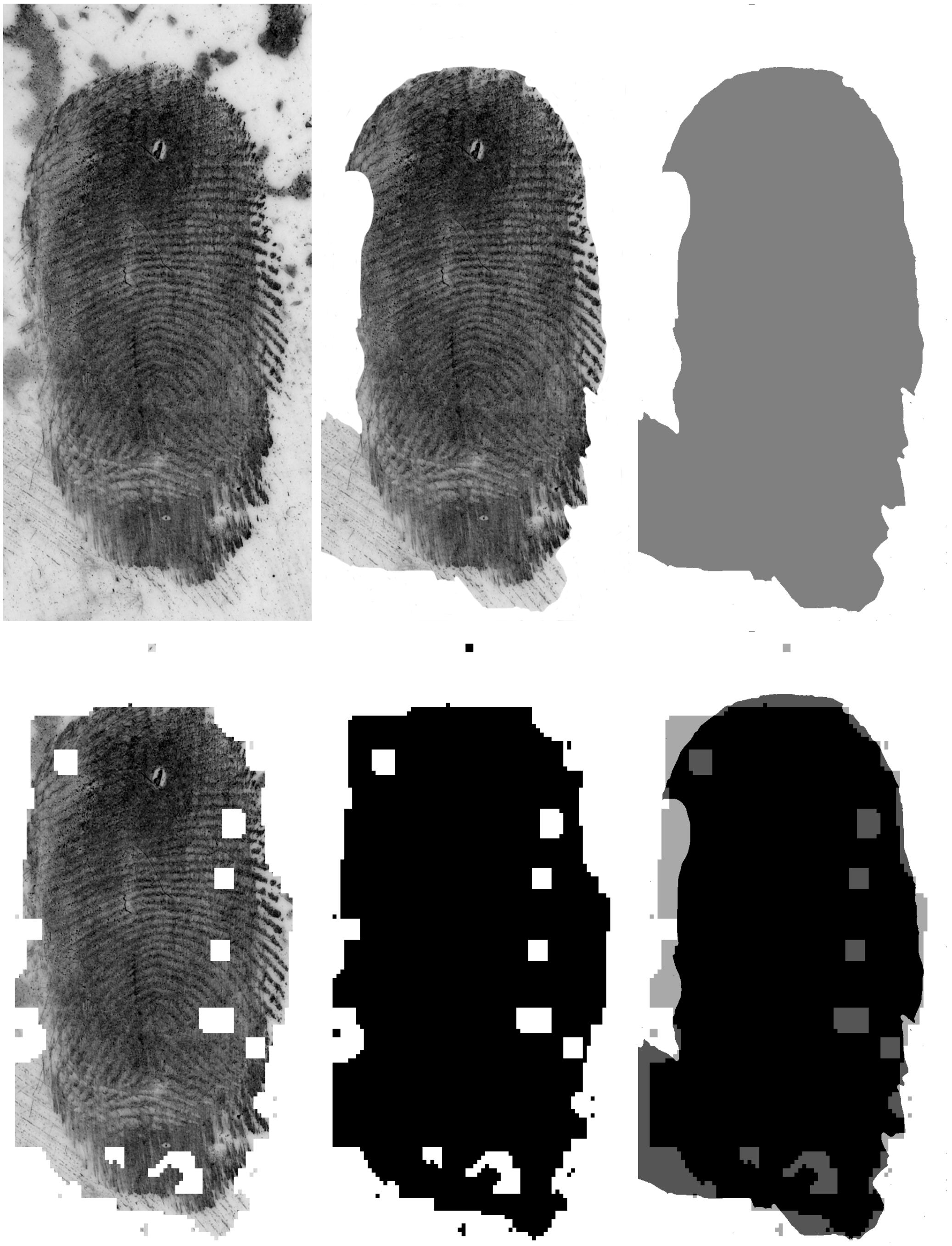


Figure 4.1: These images represents visual of the image thresholding and the union of the two methods. From top left to bottom right, the original image, the subjective image with regions of the image removed, the threshold of the subjective image, the objective image with regions of the image removed, the thresholding of the objective method and the union of the thresholded images. The black represents the regions were both images kept the fingerprint, the dark gray represents the portions of the image were the subjective image was only left and the light gray represents the portions of the image that the objective left.

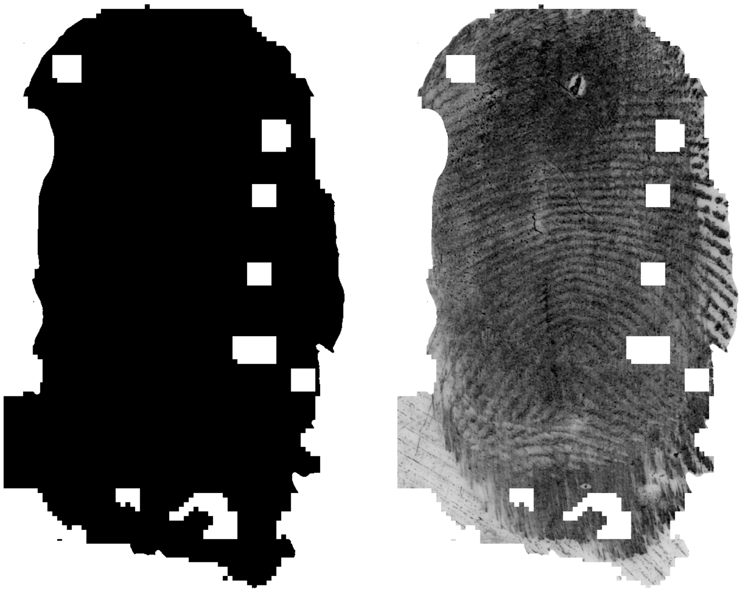


Figure 4.2: These images represent a visualization of the image thresholding and the intersection of the subjective and objective methods. On the left is the intersection of the subjective and objective images where both methods left the image behind. The image on the right is the overlay of the intersection image and the original image.

true positive rate is really a positive while the false positive rate represents the percentage of time that the result is reported as a positive but is really a negative result. A threshold can be chosen in order to improve the true positive rate or the false positive rate, but changing the value of one can effect the other. In the case of this research, the match scores, or the value given by the matching algorithm and the rank scores, were used to assess the performance of the automatic and manual extraction, the subjective versus objective method, and the comparison of AFIX Tracker® versus BOZORTH3.

4.2.7.2 Area Under the ROC Curve (AUC)

The AUC can provide a quantitative representation of how well a test performed. It is known as the area between the ROC curve and the x-axis of the plot resulting in values between zero and one. A score of one indicates the performance of the test is more accurate with the correct identification. If the score is 0.5, the chance of discrimination is just a chance, while a score below 0.5 signifies that there were more incorrect than correct identification. The value itself indicates the probability that a randomly selected known match is higher than a randomly selected known non-match. The AUC was calculated from the ROC curves computed from the match scores and ranks of the matching algorithms, as well as to assess the performance of the automatic and manual extraction, the subjective versus objective method, and the comparison of AFIX Tracker® versus BOZORTH3.

4.2.7.3 Cumulative Match Characteristic (CMC) Curve

A cumulative match characteristic (CMC) curve presents the identification rates over the entire gallery of fingerprints. This curve allows one to determine what percentage of the identifications are of a certain rank. For instance, to see the identification rate at rank 1, the number of rank one scores is taken and divided by the number of samples tested against the database. As the ranks increase, the curve will approach 100%. A more accurate system will begin high and reach 100% quickly [28].

4.2.7.4 Wilcoxon Signed-Rank Test

The Wilcoxon signed-rank test is a nonparametric statistical procedure for comparing two samples that are related [29]. To used this test, three assumptions need to be meet [29]:

1. Data is paired and comes from the same population
2. Each pair is chosen randomly and independently
3. Data is measured on an ordinal scale

In the case of this research, the rank scores from each area removed were compared to the original image rank. This will allow one to see if there is a difference in the ranks between the original image and the areas removed. If a significant difference is observed, the two images are considered different from one another in terms of how they perform in the matching algorithms. The rank scores from the original samples and the area removed for the corresponding sample had the absolute difference found and completed with the remaining pairs in the dataset. The pairs were reordered and assigned a new rank based on the absolute difference, excluding the differences of zero. The positive and negative ranks were summed individually and the smaller of the two summations was kept [29]. A confidence level of 95% was chosen to determine if there truly was a significant difference and not a difference due to chance. To accomplish this, the z-score was calculated based on a normal distribution utilizing Equation 4.1 [30].

$$\text{z}=\frac{T-\mu_T}{S_T}$$

where μ_T is the mean and T is the T statisic and S_T is the standard deviation

(4.1)

Based on the z-score and the confidence level, it can be determined whether the original images and the different areas removed are significantly different from one another.

4.2.7.5 Spearman's Rank-Order Correlation

Another nonparametric method of Spearman's Rank-Order correlation or Spearman's rho (ρ) was used to compare the relationship between the automatic and manual extraction methods to determine if they are comparable to one another based on the rank scores of the algorithms, as well as if there was a significant difference between the two matching algorithms. The Spearman's Rank-Order coefficient is found using Equation 4.2 [29].

$$r_s = 1 - \frac{6\Sigma D_i^2}{n(n^2-1)}$$

where r_s is the Spearman's rank order coefficient, n is the number of rank pairs and D_i is the difference betwen a ranked pair

(4.2)

Next, a confidence level of 95% was chosen and a z-score was computed by taking the square of the number of samples minus 1. This value is then multiplied by the correlation coefficient. Based on the z-score and the confidence level, it can be determined if the two methods are significantly different from one another [30].

4.3 Results

4.3.1 Subjective and Objective Method Quality Outcome

In the subjective portion, the 50 fingerprints were placed into the qualitative categories of “good,” “bad,” and “ugly.” The categories can be broken down as ten “good” images, 26 “bad” images and fourteen “ugly” images. The objective portion produced 17 quality scores of one, four quality scores of two, three quality scores of three, two quality scores of four and 24 quality scores of five. When comparing the qualitative categories to the NFIQ global quality assessment scores, each qualitative category was spread over each NFIQ global quality assessment score (Table 4.1).

For the NFIQ global quality assessment score of five, the six “good” and 12 “bad” images were classified into this category. When looking at the images classified as a five, five of the six images were developed with black powder and spots of the black powder can be observed in the background possibly effecting the quality score. In the case of the final fingerprint image, the fingerprint was developed with ninhydrin which can cause the fingerprints to have a spotty ridge detail which can lead to the NFIQ global quality assessment algorithm having difficulty detecting minutiae or even false minutiae.

In the qualitative category of “bad,” the 21 of the 26 fingerprints received a quality score of either one (nine images) or five (twelve images). This shows that the images were placed into the incorrect qualitative category or that the NFIQ global quality assessment algorithm is detecting different regions in the background which leads to a poor quality score. There was no clear pattern seen in looking at the development type of black powder versus ninhydrin; however, the cyanoacrylate images both received a quality score value of one which was surprising considering the these images were not of a high quality when visualized.

In the “ugly” images, the 13 images were split between the quality score of one (seven images) and the quality score of five (six images). It was expected that these images would all receive the quality scores of three or higher. These results tend to lead to the conclusion that the images were originally placed into the wrong qualitative category or the NFIQ global quality assessment algorithm needs further training.

The images were broken down based on the development type to include 15 ninhydrin fingerprints, 28 black powder fingerprints and seven cyanoacrylate fingerprints. The quality breakdown can be seen in Tables 4.2, 4.3, and 4.4.

No clear patterns were seen when observing the development type breakdown based on quality. The majority of the fingerprints were classified to into either the quality score of one or the quality score of five. Surprisingly, the cyanoacrylate fingerprints were mainly classified into “good” quality fingerprints (five images). The cyanoacrylate fingerprints were deposited onto a dark surface with a ruler and scale providing the possibility that the algorithm is detecting these portions and mistaking them as part of the fingerprint.

The different substrates were also examined to determine if they had an impact. The different substrate breakdown includes paper (15 samples), glass (three samples), ceramic

Table 4.1: The breakdown of the subjective method qualitative categories and the corresponding objective method quality score values.

Sub \ Ob	1	2	3	4	5
Good	1908866-f01	1080361-f08 1806751-f03	158696-f01	-	1210471-f02 1219247-f01 1393289-f03 1461826-f01 1721997-f08 1908866-f09
Bad	1219242-f02 1337722-f06 1400445-f03 1603003-f01 1613857-f06 1737457-f01 1768313-f02 1896537-f07 1989280-f06	1170602-f01 1721997-f06	1466760-f02 1707928-f08	103930-f08	1015855-f01 1019761-f06 1113890-f02 1113890-f06 1210471-f04 1474079-f01 1556126-f06 1677430-f03 1697412-f01 1781429-f10 1859045-f08 1963950-f04
Ugly	1360786-f01 1556527-f08 1697142-f09 1781429-f07 1781429-f08 1878605-f01 1879242-f01	-	-	1340957-f02	1400445-f01 1473993-f06 1878605-f01 1900746-f07 1991381-f04 1991415-f03

Table 4.2: The breakdown of the subjective method qualitative categories and objective method quality score values for the development method of ninhydrin.

Sub \ Ob	1	2	3	4	5
Good	-	1860751-f03	-	-	1721997-f08
Bad	1337727-f06 1603003-f01	1170602-f01 1721997-f06	1466760-f02	1030930-f08	1015855-f01 1113890-f02 1113890-f06 1556126-f06
Ugly	1360786-f01 1556527-f08	-	-	-	1991415-f03

Table 4.3: The breakdown of the subjective method qualitative categories and objective method quality scores for the development method of black powder.

Sub Ob	1	2	3	4	5
Good	-	1080361-f08	1581696-f01	-	1292471-f01,1210471-f01,, 1393289-f03,1461826-f01 1908866-f09
Bad	1219247-f02 1613857-f06 1768313-f02 1896537-f07 1989280-f06	-	1707928-f08 -	- -	1019761-f06, 1210471-f04 1474079-f01, 1677430-f03 1697142-f01, 1781429-f10 1859045-f08, 1963950-f04
Ugly	1781429-f08 1878605-f02 1879242-f01	-	-	-	1400445-f01, 1878605-f01 1900746-f07 1991381-f04

Table 4.4: The breakdown of the subjective method qualitative categories and objective method quality score values for the development method of cyanoacrylate.

Sub Ob	1	2	3	4	5
Good	1908866-f01	-	-	-	-
Bad	1400445-f03, 1737547-f01	-	-	-	-
Ugly	1697142-f09, 1781429-f07	-	-	1340957-f02	1473993-f06

Table 4.5: The breakdown of the subjective method qualitative categories and objective method quality score values for the substrate of ceramic plates.

Sub \ Ob	1	2	3	4	5
Good	-	1080361-f08	1581696-f01	-	1210471-f02, 1219247-f01 1461826-f01, 1908866-f09
Bad	1219247-f02 1613857-f06 1768313-f02 1896537-f07 1989280-f06	-	1707928-f08	-	1210471-f04, 1474079-f01 1677430-f03, 1697142-f01 1781429-f10 1859045-f08 1963950-f04
Ugly	1781429-f08 1878605-f01	-	-	-	1400445-f01, 1878605-f01 1900746-f07, 1991381-f04

plate (25 samples), CD (four samples) and bag (three samples). The paper samples provided the same results as the ninhydrin fingerprints (Table 4.2). For the glass samples, the sample of 1393289-f03 was classified as a "good" image with a quality score of five, while the other sample of 1019761-f06 was classified as a "bad" image and received a quality score of five. The final sample was classified as an "ugly" image but received a quality score of one. With the low amount of samples, no true conclusion can be made for the samples deposited on glass.

The substrate of the ceramic plate contained the highest number of samples (25). Of the 25 samples, 15 samples were classified with a quality score of five. The "bad" images received NFIQ global quality assessment score of either one or five, and only one image received a quality score of three. The ceramic plate breakdown can be seen in Table 4.5.

In the samples deposited on the substrate of the CD, only four samples were used. Sample 1908866-f01 was classified as a "good" fingerprint with a quality score of one, while the sample of 1400445-f03 was classified as a "bad" fingerprint with a quality score of one. The final two images, 1340957-f02 and 1473993-f06, were both classified as "ugly" images and received the quality scores of four and five, respectively. Three of the four samples had the quality score and the respective qualitative category agree. A larger sample size would be needed to determine if this trend could hold true for this substrate.

In the final substrate types, bags, three samples were obtained. All three samples were classified with a quality score of one; however, one sample was classified as a "bad" quality image whereas the remaining two images were classified as "ugly" quality images. More samples of this substrate are needed to determine if a relationship can be found.

With these results, it may indicate that more weight is being placed on the quality scores of one and five instead of producing the expected quality scores.

Table 4.6: The average minutiae count for the automatic and manual extraction for the subjective and objective methods.

	Subjective		Objective		
Area	Auto	Manual	Auto	Manual	Area
Original	11.04	14.60	6.08	12.4	Original
1	13.6	15.28	1.21	9.16	0
2	14.2	14.72	2.14	7.96	1
3	11.98	12.40	2.03	1.5	2
4	8.70	8.44	1.76	0.16	3
Average	11.88	13.15	6.08	6.23	Average

4.3.2 AFIX Tracker®

4.3.2.1 Number of Minutiae: Manual Extraction vs. Automatic Extraction

The two separate uploads allowed for the comparison of the automatic extraction by the AFIX Tracker® software versus the manual extraction by one user. The subjective images had an average of 11.88 minutiae in the automatic extraction while the objective images had an average of 6.08 minutiae extracted. Interestingly, the minutiae counts do not start to decrease until after Area 2 in the subjective method and Area 1 in the objective method. It was expected that the minutiae count would decrease with the decrease in the area size (Table 4.6).

In the manual extraction, both the subjective and objective methods showed an increase in the number of minutiae with the subjective method having an average of 13.15 minutiae and the objective method having an average of 6.23 minutiae marked. The Area 1 images had the highest average of 15.28 minutiae. This may be due to the AFIX Tracker® system producing better contrast in the images with more of the image removed. In the objective images, Area 0 produced the highest average of 12.4 minutiae marked and continued to decrease (Table 4.6).

When examining the individual areas of the automatic extraction compared to the manual extraction in the subjective method, the original images presented to have a higher amount of minutiae extracted in the manual extraction in comparison to the automatically extracted minutiae (Figure 4.3). This is to be expected no regions of the fingerprint removed.

In Area 1, the manual extraction showed to have more minutiae extracted than in the automatic extraction. The amount of minutiae extracted by both methods were mainly below 20 minutiae. The figure can be seen in Appendix D.1.

The Area 2 images showed the manual extraction to have a higher number of minutiae extracted in comparison to the the automatic extraction. These samples extracted about 20 minutiae and below. The samples that had the automatic extraction be higher showed to have 20 or more minutiae extracted (Appendix D.1). The Area 3 scatter plot showed

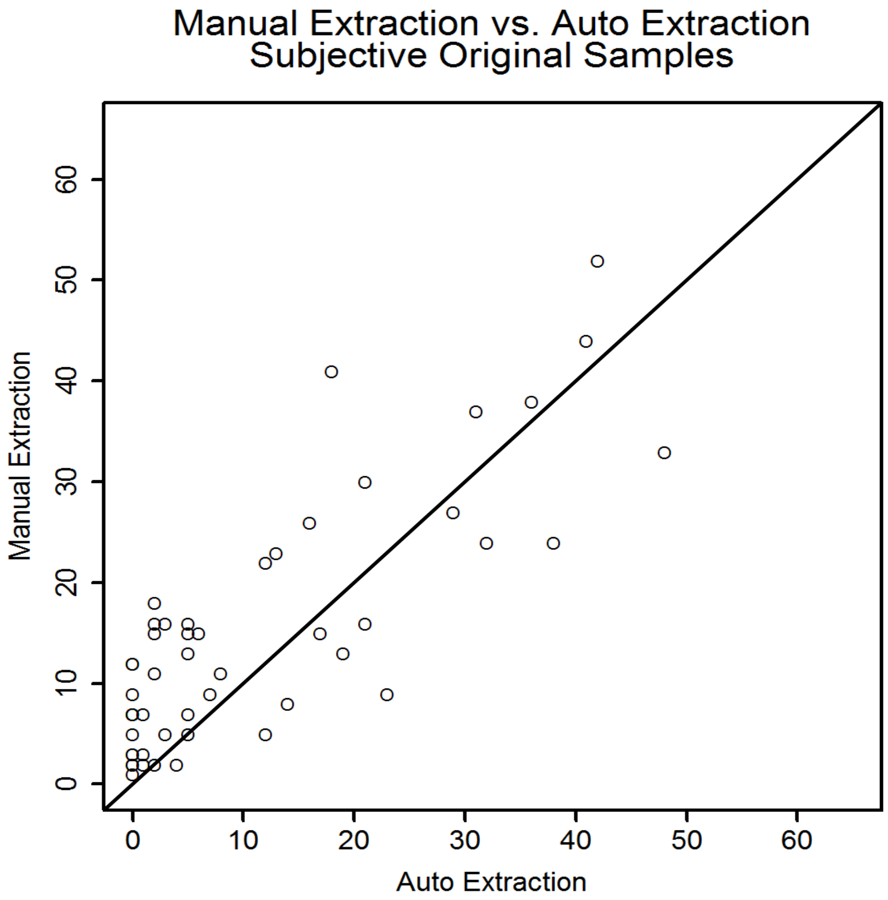


Figure 4.3: This figure presents the scatter plot for the number of minutiae extracted using automatic extraction and manual extraction for the subjective original samples.

the same trend as the Area 2 minutiae extraction.

Area 4 samples showed to have the number of minutiae extracted for both methods shift toward the bottom left corner of zero. Even with this shift, the manual extraction did produce more minutiae than in the corresponding automatic extraction (Appendix D.1). With this result, the question becomes whether or not the minutiae that the algorithm is finding true minutiae or false minutiae.

For the objective method, the original images showed to have more minutiae extracted by the algorithm than by the user. There were also samples where there no minutiae were found by the algorithm; however, the user was able to detect minutiae. The Area 0 images produced a similar scatter plot. It was seen that as the area decreases, the amount of minutiae extracted by the algorithm decreases; however, some images were able to have minutiae manually extracted. The decrease may be due to the amount of area that has been removed from the fingerprint in the method (Appendix D.1).

Overall, the subjective method produced more extracted minutiae than in the objective method. This could be explained by a number of reasons. First, there is a decrease in the amount of surface area available for the extraction. It may be a case were the subjective method is leaving more of the fingerprint behind to be marked than in the objective method. Another explanation could be that the system is extracting false minutiae which can cause issues when it comes to the matching algorithm.

4.3.2.2 Number of Minutiae: Development Methods

The development methods were examined to determine if the type of development could influence the number of minutiae found. When examining the number of minutiae for the original image in the subjective method, black powder images had an average of 14.75 minutiae extracted, while the manual extraction provided a higher of 18.75 minutiae. In looking at the cyanoacrylate for the automatic extraction, the highest average number of minutiae was 6.57 while the manual extraction had an average of 9.71. The ninhydrin development was similar to the cyanoacrylate with the automatic extraction providing an average of 6.20 minutiae and the manual extraction having an average of 10.13 minutiae. The low number of minutiae for the development methods of cyanoacrylate and ninhydrin can be explained by the poor quality of several images. The box plot for the subjective original images can be seen in Appendix D.2.

For the objective method, the results were different. In the case of the automatic extraction the black powder fingerprints had an average of 14.36 minutiae. The average number of minutiae for the remaining two methods was less than half of the black powder average. The cyanoacrylate fingerprints had an average number of minutiae of 4.71 while the ninhydrin fingerprints had an average number of minutiae of 6.73. The manual extraction provided a similar average for the black powder fingerprints with an average of 15.82 minutiae. The cyanoacrylate fingerprints provided twice the average amount of minutiae in the automatic extraction with an average of 8.00 minutiae. The ninhydrin fingerprints were slightly higher than the automatic extraction with an average of 8.07 minutiae. The box plot for the objective original images can be seen in Appendix D.2.

In the Area 1 images, the average minutiae counts for all three development methods were higher in the subjective method. The black powder fingerprints had an average minutiae count of 17.71 while the cyanoacrylate had an average minutiae count of 9.86. The ninhydrin images had the lowest average minutiae count of 7.66. The manual extraction provided an increase in the average minutiae score by a value of one. The black powder images had an average minutiae count of 19.36. The cyanoacrylate had an average minutiae count of 10.86 while the ninhydrin had an average minutiae count of 9.73. The box plot for the subjective Area 1 images can be seen in Appendix D.2.

The objective images for Area 0 had a lower minutiae count than the subjective method. In the automatic extraction, the black powder fingerprints had an average minutiae count of 14.96 while the manual extraction had an average minutiae count of 13.57. The cyanoacrylate fingerprints produced an average of 6.14 minutiae in the automatic extraction while the manual extraction produced an average of 2.57. The final development method of ninhydrin provided an average number of minutiae of 3.07 for automatic extraction while the manual extraction had an average number of minutiae of 4.00. The box plot for the objective Area 0 images can be seen in Appendix D.2.

The Area 2 images for the subjective method had a mixture of results. The black powder fingerprint average minutiae count was 17.43 for the automatic extraction while the minutiae extraction average minutiae was 18.43. The cyanoacrylate fingerprint images in the automatic extraction produced a higher automatic extraction number of 13.00 while the Area 1 had an average of 9.86. The manual extraction was lower than Area 1 with an average minutiae count of 9.86. After the Area 2 average minutiae counts, the remaining areas average minutiae counts decreased which is to be expected with the decrease in area available to mark.

The Area 2 images for the objective method all had a decrease in the minutiae count, which is to be expected. All the average minutiae counts can be seen in Table 4.7 as well as the box plots in Appendix D.2. Overall, it appears that the black powder development showed to have the most minutiae extracted in both methods. With having a higher amount of minutiae, it is likely more matches can be found by the matching algorithms. What can also be seen is that the subjective images produced higher minutiae counts than the objective image which may reflect the amount of fingerprint available for marking.

4.3.2.3 Number of Minutiae: Substrate Type

The substrate types were also examined to determine if there is an effect on the number of minutiae. In the case of paper, the subjective method had an average of 6.75 minutiae for the automatic extraction and 8.59 average minutiae for the manual extraction. Area 2 in the automatic extraction showed to have the highest amount of minutiae, which is interesting due to the decrease in the area of the fingerprint. In the manual extraction, the original image had the highest average number of minutiae at 8.07 (Table 4.8). The box plot can be seen in Appendix D.3.

Table 4.7: The average number of minutiae for the subjective and objective method for the automatic extraction and manual extraction.

Ninhydrin					
	Subjective		Objective		
Area	Auto	Manual	Auto	Manual	Area
Original	6.20	10.13	6.73	8.07	Original
1	7.66	9.73	3.07	4.00	0
2	8.47	10.07	2.33	3.07	1
3	7.07	8.07	0	0	2
4	4.33	4.93	0	0	3
Average	6.75	9.54	2.43	3.03	Average

Black Powder					
	Subjective		Objective		
Area	Auto	Manual	Auto	Manual	Area
Original	14.75	18.75	14.36	15.82	Original
1	17.71	19.36	14.96	13.57	0
2	17.43	18.43	13.21	11.96	1
3	14.86	15.96	1.28	2.67	2
4	11.53	11.57	0	0.29	3
Average	15.26	16.81	8.76	8.86	Average

Cyanoacrylate					
	Subjective		Objective		
Area	Auto	Manual	Auto	Manual	Area
Original	6.57	9.71	4.71	8.00	Original
1	9.86	10.86	6.14	2.57	0
2	13.00	9.86	5.14	2.43	1
3	11.00	7.43	0	0	2
4	6.71	3.43	0	0	3
Average	9.42	8.26	3.20	2.60	Average

For the substrate of glass, the average number of minutiae in the subjective method was 11.33 for the automatic extraction and 12.53 for the manual extraction. The highest number of minutiae differed between the automatic and manual extraction with the automatic extraction having the highest average in the original image with 14.00 minutiae and Area 1 in the manual extraction with 13.67 minutiae. The objective method produced half the average number of minutiae for both methods (automatic with 5.33 and manual with 6.33). The original images showed to have the highest average in both the automatic and manual extraction with 11.67 and 12.00, respectively (Table 4.8). The box plot can be seen in Appendix D.3.

In the substrate of the ceramic plate, the subjective method had a high average number of minutiae for both the automatic extraction and the manual extraction with 15.73 and 17.33, respectively. Interestingly, the original images did not produce the highest amount of minutiae for either the automatic extraction or the manual extraction. Area 1 in the automatic extraction and manual extraction had the highest average number of minutiae with 18.24 and 19.96, respectively. In the objective method, the average number of minutiae automatically extracted was 9.13 while the manual extraction provided an average of 9.17 showing similar extraction results. Area 1 in the automatic extraction had the highest average number of minutiae with 15.76 while the original image in the manual extraction had the highest average number of minutiae at 16.28 (Table 4.8). The box plot can be seen in Appendix D.3.

The substrate of the CD showed to have an average of 13.60 and 9.25 for the automatic and manual extraction, respectively, for the subjective method while in the objective method, the average number of minutiae automatically extracted was 5.55 and manual extraction was 3.45. For the subjective method, the automatic extraction with the highest number of average minutiae was Area 2 with 18.25 minutiae, and in the manual extraction the Area 1 images had the highest average number of minutiae at 12.75. The objective method showed something different. The Area 1 images in the automatic extraction had the highest average at 10.50 while the original images in the manual extraction had an average number of minutiae at 9.50 (Table 4.8). The box plot can be seen in Appendix D.3.

The final substrate of the bag showed to have the lowest averages overall. This may be in part to both the low number of samples deposited on the substrate as well as the quality being low. The average number of minutiae for the automatic extraction in the objective method was 3.87 minutiae with the highest number of average minutiae produced by the Area 2 images with six minutiae. In the manual extraction, the average number of minutiae found was 6.93 with Areas 1 and 2 having the highest number of minutiae at 8.33. In the objective method, the average number of minutiae was 0.67 for the automatic extraction with Area 1 having the highest average number of minutiae at 0.33. The manual extraction showed to have more minutiae with the average being 1.47, and the original images had an average of six minutiae detected (Table 4.8). The box plots for all of the substrates and areas can be seen in Appendix D.3

Overall, the ceramic plate showed to have the highest number of minutiae due to the development of black powder being utilized which showed to have the largest amount of minutiae extracted. More samples are needed for the substrates of the bag, glass, and CD to make any conclusions. In comparing the automatic extraction to the manual extraction, the subjective automatic extraction produced more minutiae, in all averages, with the exception of the ceramic plates. The objective manual extraction appeared to produce more minutiae than the automatic extraction, except in the case of the CD. Overall, one extraction method did not out perform the other in any one area. One take away that can be made is that the automatic extraction, in the AFIX Tracker® system, does have the tendency to mark false minutiae in a fingerprint. The use of manual extraction allows a user to mark the true minutiae in a fingerprint, which may help in the process of improving the chance of receiving a true match.

4.3.2.4 Match Scores

The match scores received were normalized based upon the top match score for each search allowing the match scores to range from zero to one. In the subjective method match scores, it appears that the manual extraction provided more match scores than with the automatic extraction. However, the match scores that were found in the automatic extraction gave higher match scores of about one. The original image manual extraction match scores were above 0.4 but the automatic extraction had match scores of zeros. In the case of the automatic extraction, when the manual extraction had a match score of 1, the automatic extraction match scores were above 0.4.

In the case of the Area 1 subjective images, a large amount of the manual extraction match scores had a corresponding automatic extraction match score of zero. In some instances, the manual extraction with a match score of one had an automatic extraction match score of one. Interestingly, match scores were found on the extremes of the normalized values (zero and one) of the automatic extraction with no match scores falling in between.

The subjective Area 2 provided match scores of zero for the automatic extraction while the manual extraction provided match scores of zero and one. Match scores for the automatic extraction showed to be higher than 0.8 with a corresponding match score of one. A single sample was seen to have a similar match score for the manual extraction and automatic extraction; however, the manual extraction score was slightly higher.The Area 3 match scores showed to be on the extremes of the plot with a large amount of automatic extractions having a match score of zero. Area 4 match scores showed to have several samples with a match scores of zero for both extraction methods. Several automatic extraction match scores produced a value of one with a manual extraction score ranging between zero and one.

In the objective methods for the original samples, the match scores were spread along the left vertical axis and the top horizontal axis. Interestingly, no samples were found in the middle of the plot. A large amount of samples were found at the match score of zero for both the manual extraction and the automatic extraction as well as at the match score

Table 4.8: The automatic extraction and manual extraction of the subjective and objective methods based on the substrate type. This table provides the average number of minutiae from the both methods as well as for each area removed.

		Paper			
	Subjective			Objective	
Area	Auto	Manual	Auto	Manual	Area
Original	6.20	10.13	6.73	3.03	Original
1	7.60	9.73	3.07	8.07	0
2	8.47	10.07	2.33	3.07	1
3	7.07	8.07	0	0	2
4	4.33	4.93	0	0	3
Average	6.75	8.59	2.43	3.03	Average
		Glass			
	Subjective			Objective	
Area	Auto	Manual	Auto	Manual	Area
Original	14.00	13.67	11.67	12.00	Original
1	13.33	14.33	8.33	9.67	0
2	12.00	13.67	8.67	8.33	1
3	10.00	11.33	0	1.67	2
4	7.33	9.67	0	0	3
Average	11.33	12.53	5.73	6.33	Average
		Ceramic Plate			
	Subjective			Objective	
Area	Auto	Manual	Auto	Manual	Area
Original	14.84	19.36	14.68	16.28	Original
1	18.24	19.96	15.76	14.04	0
2	18.08	19.00	13.76	12.40	1
3	15.44	16.52	1.44	2.80	2
4	12.04	11.80	0	0.32	3
Average	15.73	17.33	9.13	9.17	Average
		CD			
	Subjective			Objective	
Area	Auto	Manual	Auto	Manual	Area
Original	11.00	11.00	8.25	9.50	Original
1	13.75	12.75	10.50	4.00	0
2	18.25	11.00	9.00	3.75	1
3	14.00	8.00	0	0	2
4	11.00	3.50	0	0	3
Average	13.60	9.25	5.55	3.75	Average
		Bag			
	Subjective			Objective	
Area	Auto	Manual	Auto	Manual	Area
Original	0.67	8.00	0	6.00	Original
1	4.67	8.33	0.33	0.67	O
2	6.00	8.33	0	0.67	1
3	7.00	6.67	0	0	2
4	1.00	3.33	0	0	3
Average	3.87	6.93	0.67	1.47	Average

of one for both extraction methods.

The Area 0 match scores showed a similar trend to the original samples being on the extremes of the plot. The manual extraction match scores were higher than 0.5 while the automatic extraction produced match score of 0.8 and higher. The remaining samples showed to have match scores of zero for both the manual extraction and the automatic extraction.

The Area 1 match scores showed that both extraction match scores either had a match score of one or zero. A trend in the automatic extraction showed that the scores will be either a match score or one of zero with no scores in between.

The Area 2 samples showed that the automatic extraction had only match scores of zero. The manual extraction did produce match scores of zero and above 0.75. The objective samples, from the automatic extraction, provided no minutiae for the algorithm to use. However, the manual extraction method does produce some minutiae to search with.

The Area 3 samples showed to have only two points available on the scatter plot. The automatic extraction showed to produce no match scores while the manual extraction produced match scores of either zero or one. These match scores are expected with the low amount of image area remaining from the removal process with several of the images left blank. All the individual area scatter plots can be seen in Appendix D.4.

Overall, the match scores showed a clear cutoff in the areas that returned a match score. In the case of the subjective samples, Areas 3 and 4 should not be utilized due to a low number to no match scores being produced. The objective method cutoff can be seen in the Areas of 2 and 3 since not many match scores were not seen. In the case of the automatic extraction versus the manual extraction, a case can be made that the manual extraction, overall, produces higher match scores and in some cases the on;y match score.

4.3.2.5 Match Scores: Development Type

The match scores were observed, based on the development type, to determine if a particular development type produced higher match scores than the other. In the original images, the automatic extraction showed to have match scores of either zero or one in all three development methods. However, the manual extraction showed to be different for the black powder and the ninhydrin. The black powder samples showed to have a range of match scores from zero to one whereas the ninhydrin samples had a range of match scores between zero and 0.30 with a stray sample at the match score of one (Appendix D.5).

In Area 1, the automatic extraction showed to be similar for the cyanoacrylate and the ninhydrin samples in the original images. However, the black powder images showed to have a range of match score between zero and 0.6 with a few outlining samples. The manual extraction showed to produce match scores for the entire range of zero to one for the black powder fingerprints. The ninhydrin fingerprints showed to have match scores ranging between zero and 0.2 with a few samples outside of this range and the cyanoacrylate having match scores of zero (Appendix D.5).

The Area 2 images in the automatic extraction show match scores for the black powder ranging from zero to 0.4 with a few samples outside of this range. The ninhydrin images showed to have lower match scores from zero to 0.2 with a few samples outside of that range and the cyanoacrylate samples had a match score of zero. The manual extraction showed to have a higher range of match scores for the black powder fingerprints ranging from zero to one. The cyanoacrylate showed to have the same results as in the automatic extraction whereas the ninhydrin images showed to have a higher number of match scores between zero and 0.25 (Appendix D.5).

In the Area 3 images, both the automatic and manual extraction methods showed to not produce a range of match scores with a score of zero for the development methods of cyanoacrylate and ninhydrin. The black powder images showed to have a range of match scores between zero and 0.6 in the automatic extraction while the manual extraction match scores ranged from zero to one (Appendix D.5).

The Area 4 images showed not have a range of match scores for the automatic extraction, however each method did produce a match score of one. The manual extraction produced a range of match scores for the black powder fingerprints between zero and one whereas the other two development methods did not have a range in match scores. The ninhydrin fingerprints did produce two match scores at 0.8 and 1. The low number of match scores for the increase in area number is to be expected due to the surface area available is decreasing as well as removing the number of minutiae available. Overall, the manual extraction showed to have more match scores present for the black powder images whereas the other two development methods produced little to no match scores. The box plots for the match scores can be seen in Appendix D.5.

In the objective method, the automatic extraction did not provide a large number of match scores in the original images. Each of the development types did produce one match score of one, which may be outliers. In the manual extraction, the black powder development showed to have a large range of match scores from zero to one while the cyanoacrylate and the ninhydrin did produce at least one match score of one.

Area 0 provided match scores for the automatic extraction for the black powder development ranging from zero to 0.5. The other two development types did produce a match score of one. In the manual extraction, the black powder development have match scores ranging from zero to one while the ninhydrin development showed to have two match scores at 0.5 and one. The cyanoacrylate did not produce any match scores above zero.

The Area 1 automatic extraction did not provide a range of match scores but did have match scores of one. The high match score is good to see however, there appears to only one match score for each development method. The manual extraction produced a range of match score for the black powder development between zero and 0.85 while the remaining two development methods did not have a range of match scores.

In Areas 2 and 3, neither extraction method produced no match scores for any of the development methods, with the exception of two match scores in Area Two for the black powder development in the manual extraction. The black powder development provided a large number of match scores. This may be due to the large amount of black powder development samples. The box plots for the match scores can be seen in Appendix D.5.

Overall, the black powder development method showed to produce more match scores than the other two development types which is consistent with the average number of minutiae found and the number of samples. In the case of the different area, more evidence is provided to show that the subjective method areas of 3 and 4 should not be used, even with the black powder fingerprints producing match scores. This assessment can not be made for all development types. A similar case can be made for the objective images in the case of the manual extraction. The original, Area 0 and Area 1 were able to obtain match scores for at least the black powder development. If taken as a whole, only the original images performed well for the matching algorithm. Interestingly, the automatic extraction showed to have difficulty producing match scores for ninhydrin and cyanoacrylate fingerprints. An explanation could be that the automatic extraction produced false minutiae on the images causing an incorrect match to occur. This may also stem from the fact that these images were of a lower quality.

4.3.2.6 Match Scores: Substrate Type

The substrate types were examined to determine if the substrate type could have an effect on the match scores. The automatic extraction for the original images in the subjective method provided ranges in match scores between zero and 0.5 for the CD and the glass surfaces. The ceramic plate and the paper showed to have match scores of one but there was no range in the match scores. In the manual extraction, the ceramic plate showed a range of match scores from zero to one. The substrate of the CD showed a range of match scores between zero and 0.5 while the paper showed to have a range from zero to 0.25. The glass surfaces showed to have match scores between 0.5 and one. This result is interesting due to the match scores mainly clustered with lower match scores and not starting at the value of 0.5.

In the Area 1 automatic extraction, the substrates of the CD, ceramic plate and glass provided a range of match scores between zero and 0.45, while the substrate of paper provided two match scores of 0.8 and one. In the manual extraction, the glass and ceramic plate substrates provided the same pattern as in the original. The paper substrate gave a range of match scores between zero and 0.2 while the substrates of the bags and the CDs did not have any match scores.

Area 2 automatic extraction showed to have a large dropout of the ceramic plate match scores with only three match scores present in comparison to the Area 1 ceramic plates. The CDs and the glass substrates showed to have that same range in the match scores of zero to 0.5. The paper substrate showed to have a match score range of zero to 0.2. In the manual extraction, the CDs had the same range in match scores while the ceramic plates had match scores ranging from zero to one. This is interesting and provides evidence that the manual extraction helps improve match scores. The glass substrate showed to have the same results as in the original images while the paper substrate showed to have a match score range of zero to 0.21.

For the Area 3 images, the CD and glass substrates had similar match score spreads of between zero and 0.5. The ceramic plate substrate spread was higher with match scores between zero and 0.55. The bag and paper substrates did not produce match score spreads. The bag samples were a small sample size which may lead to the non-existent match scores. In the manual extraction, the glass and ceramic plate substrates showed to have a similar pattern as in the previous areas. The CD substrate showed to have a spread of match scores from zero to about 0.15.

In the final subjective area of four, the CD substrate showed to have a match score spread of zero to 0.5 while the glass substrate had a spread of zero to 0.3. Interestingly, the ceramic plate did not produce a match score spread while in the manual extraction, the match score spread was between zero and one. In the case of the glass substrate, the match score spread was between zero and 0.5. This is a change considering the other areas for the manual extraction produced match score spread above 0.5. In looking at both methods, the decrease in the spread of match scores could be due to the decrease in the surface area of the fingerprint available to extraction minutiae.

The objective method for the original images showed interesting results. The automatic extraction showed no spread in the match score for the ceramic plate while the manual extraction had a spread in match scores of zero to one. The glass substrate in the automatic extraction showed to have a spread of zero to 0.5 while in the manual extraction, the spread was from 0.5 to one. The bag and paper substrates both showed no spread in the match scores.

The Area 0 images also showed to have no spread in the match scores in the ceramic plate for the automatic extraction while the manual extraction had a match score spread of zero to one. The glass substrate in the automatic extraction had a spread in the match scores from zero to 0.5 while the manual extraction provided a spread in the match score between 0.4 and 0.85. The paper and bag substrates did not provide a spread in the match scores.

The Area 1 match scores showed that the substrates of the CD and the glass had match score spreads in the automatic extraction between zero and 0.5 while the remaining substrates did not produce match score spreads. In the manual extraction, the glass substrate showed a different spread than the Area 0 with a match score spread of zero to 0.5. The ceramic plate produced a large spread in the match score between zero and 0.7. The remaining substrates did not produce match score spreads.

Areas Two and Three showed no match score spreads which is consistent with the amount of the fingerprint image present after the area removal. Overall, the glass substrate produced consistent match score spreads. In the case of the bag and the paper, more samples are needed to determine if a pattern can be found. The results from the glass show that even with a small number of samples, higher match score can still be obtained with a range in the quality of the image. This also gives evidence to using only certain areas removed.

4.3.2.7 ROC Curves

The AFIX Tracker® data was used to produce ROC curves to examine the performance of the subjective and objective methods overall and for the corresponding areas removed. Table 4.9 shows the match score AUC, equal error rates (EER) and the position of the EER

For the match scores in the subjective method, all of the AUCs' were above the value of 0.880 showing that the AFIX Tracker® algorithm is able to produce a high number of correct identifications in all of the area removals.

The equal error rate allows one to determine where the false positive rate and and the false negative rates intersect. The point at which the lines intersect gives the performance of the system. The lower the EER, the better the system performed. In the subjective method, the EER for all of the samples was 0.2 with a range in the match scores. Area 3 showed to have the EER at the lowest match score of 36368 whereas Area 2 had the highest match score at 52748. Ideally, one would like to see that match scores as high as possible with the lowest EER possible. In the case of the subjective method, Area 2 performed the best with the highest match score of 52748 and was reflected with the highest AUC score of 0.922 (Table 4.9). This is consistent with the Area 2 images having the highest number of minutiae extracted.

In the objective method, the EER rates ranged with the lowest EER of 0.03 (Area 2) and the remaining areas having an EER of 0.2. The highest match score was for Area 1 at 45879 with an EER of 0.2. With the combination of a low EER and the highest match score, it appeared that Area 1 performed the best in the AFIX Tracker® matching algorithm with the lowest EER and the highest AUC value.

When breaking down the match scores for the automatic extraction, the subjective areas produced AUC values above 0.900 and the resulting EER were below 0.2 (Table 4.10). The lowest EER was 0.1 for Area 2 with at a match score of 60145. In the objective method, the match scores were above 0.900. The highest AUC was from Area 1 at 0.965 with an EER of 0.1 at the match score of 67542. With the combination of a low EER and the highest match score, it appears that Area 1 performed the best in the AFIX Tracker® matching algorithm with the lowest EER and the highest AUC value.

The p-values were calculated by row to compare the subjective method to the objective method to determine if there was a significant difference in the areas removed and their performance in match algorithm. The p-value for subjective Area 1 versus objective Area 0 and subjective Area 3 versus objective Area 2 had significant differences with the p-values of 0.0368 and 0.0000751, respectively (Table 4.10). The significant difference can show that the two areas removed are truly different from one another in their performance in the algorithm.

An interesting result found was the p-value for the subjective Area 1 versus objective Area 0 showing a significant difference in the performance of the algorithm. Both the ROC curves produced the same AUC score but showed to be significantly different from one another in their performance. When plotted against one another, the ROC curves do

Table 4.9: The AUC values along with the equal error rates (EER) at the particular match score for the subjective and objective methods for manual extraction and automatic extraction.

Subjective				Objective			
Area	AUC	EER	Position	Area	AUC	EER	Position
Original	0.908	0.2	49577	Original	0.916	0.2	42180
1	0.887	0.2	44822	0	0.887	0.2	40595
2	0.922	0.2	52748	1	0.921	0.2	45879
3	0.883	0.2	36368	2	0.987	0.03	22102
4	0.898	0.2	39538	3	-	-	-

Table 4.10: The AUC values along with the equal error rates (EER) at the particular match score for the subjective and objective methods the automatic extraction. The p-value for each row was also calculated.

Subjective				Objective				p-Value
Area	AUC	EER	Position	Area	AUC	EER	Position	
Original	0.927	0.2	58560	Original	0.913	0.2	40067	0.829
1	0.938	0.2	54861	0	0.938	0.2	53276	0.0368
2	0.952	0.1	60145	1	0.965	0.1	67542	0.292
3	0.935	0.2	54333	2	0.9645	0.09	8364	0.0000751
4	0.912	0.2	47464	3	-	-	-	-

not lie directly on top of one another with the exception of the beginning of the curves and the end of the curves (Figure 4.4). Even with having the same AUC , the data ultimately performed differently to create a significant difference between the areas removed.

In the manual extraction, the AUC scores were lower than the automatic extraction, in both the subjective and objective methods, the original images having an AUC of 0.901 and Area 1 having the lowest AUC of 0.854. All areas, with the exception of Area 1, had an EER value of 0.2. When observing the p-values, a significant difference can be seen in subjective Area 3 versus objective Area 2 with a p-value of 0.0000751 (Table 4.11). This was an expected due to the match scores in the previous sections.

For the rank scores in all of the samples for the subjective and objective methods, the AUC scores were all the value of 0.90 with EER values of below 0.2 (Table 4.12). When looking at the automatic extraction, the subjective methods showed to have AUC values of above 0.90 and EER values of below 0.20. The objective method showed to have similar AUC values (Table 4.13). No significant differences were seen in the automatic extraction.

The manual extraction AUCs' showed to have higher AUC values than in the automatic extraction for all areas. The EER that was the lowest (0.09) for Area 2 and had the highest AUC of 0.964. In the objective method, the highest AUC was in the Area 2 of 0.971 with the lowest AUC belonged of 0.5. The only area that showed to be significantly different

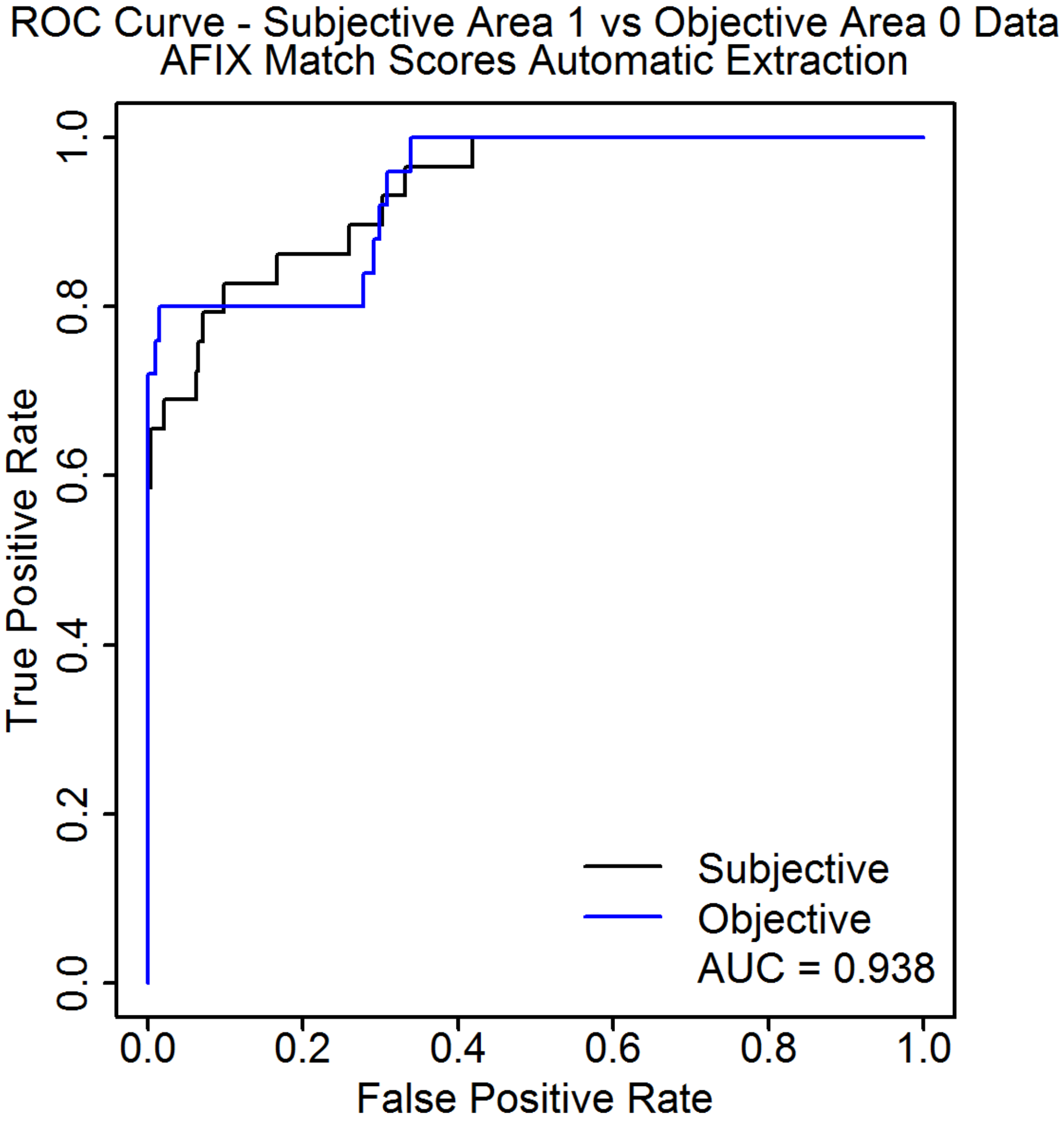


Figure 4.4: The ROC curve comparison for the subjective Area 1 vs objective Area 0 for the AFIX Tracker® automatic extraction.

Table 4.11: The AUC values along with the equal error rates (EER) at the particular match score for the subjective and objective methods for manual extraction. The p-value for each row was also calculated

Subjective				Objective				p-Value
Area	AUC	EER	Position	Area	AUC	EER	Position	
Original	0.901	0.2	38481	Original	0.919	0.1	42708	0.602
1	0.854	0.3	38481	0	0.852	0.3	37425	0.957
2	0.908	0.2	43237	1	0.901	0.2	35840	0.639
3	0.858	0.2	32141	2	0.999	0.008	21045	0.0000751
4	0.889	0.2	36896	3	-	-	-	-

Table 4.12: The AUC values along with the equal error rates (EER) at the particular rank for the subjective and objective methods for manual extraction and automatic extraction.

Subjective				Objective			
Area	AUC	EER	Position	Area	AUC	EER	Position
Original	0.937	0.1	11.5	Original	0.926	0.1	11.2
1	0.926	0.1	13	0	0.949	0.1	10.4
2	0.946	0.1	10.1	1	0.964	0.05	5.68
3	0.921	0.1	14.6	2	0.969	0.05	3.3
4	0.921	0.2	12.2	3	-	-	-

Table 4.13: The AUC values along with the equal error rates (EER) at the particular rank for the subjective and objective methods for automatic extraction. The p-value for each row was also calculated. No true positive match scores were not able to be obtained by the objective Area 2 images.

Subjective				Objective				p-Value
Area	AUC	EER	Position	Area	AUC	EER	Position	
Original	0.923	0.08	6.95	Original	0.89	0.1	13.3	0.584
1	0.935	0.1	10.6	0	0.985	0.03	4.25	0.102
2	0.916	0.1	10.4	1	0.986	0.03	4.25	0.196
3	0.952	0.1	7.26	2	-	-	-	-
4	0.915	0.2	15.3	3	-	-	-	-

Table 4.14: The AUC values along with the equal error rates (EER) at the particular rank for the subjective and objective methods for manual extraction. The p-value for each row was also calculated

Subjective				Objective				p-Value
Area	AUC	EER	Position	Area	AUC	EER	Position	
Original	0.946	0.1	13.4	Original	0.948	0.1	10.6	0.939
1	0.921	0.1	14.1	0	0.923	0.1	14.2	0.953
2	0.964	0.09	9.48	1	0.946	0.08	8.22	0.594
3	0.903	0.2	18.4	2	0.971	0.05	2.67	0.0345
4	0.927	0.1	8.37	3	-	-	-	-

was the subjective Area 3 versus objective Area 2 with a p-value of 0.0345, consistent with the match scores (Table 4.14).

Overall, the AUC scores provided insight into the methods. When based solely on the match scores, the automatic extraction for both methods had higher AUC's than the manual extraction, which is surprising due to the difference in the amount of minutiae and the match scores. In looking at the individual areas for the match scores in the automatic extraction, Area 2 in the subjective method provided the highest AUC and the lowest EER while the objective counter part of Area 1 showed the same results. The manual extraction showed that subjective Area 2 provided the highest AUC and the lowest EER while the objective original showed to have the highest AUC and lowest EER. However, Area 1 did provide an AUC above 0.90 showing an overall good performance.

In looking solely at the ranks, the objective method received higher AUC scores (with the exception of the original samples in the automatic extraction and Area 1 in the manual extraction) showed higher AUCs than the subjective method; however, no clear method for automatic and manual extraction was better with the ranks. When breaking down the areas, the manual extraction for the subjective Area 2 and objective Area 1 were the highest in AUC values and the lowest in the EER value. This provides more evidence that these are the lowest areas to be assessed by the algorithm and are also performing higher in the matching algorithms.

4.3.2.8 Wilcoxon Signed-Rank Test

The Wilcoxon Signed-Rank test compares how related two samples are. This method can help show if the area removed is similar to the original or if the removal is having an effect on the ranks. In the subjective method, it appeared that the original image versus each area in both the automatic extraction and the manual extraction showed to have no significant difference on the rank scores. This helps to show that the subjective method is able to provide similar results when decreasing in the area removal (Table 4.15).

In the case of the objective method, the only significant difference was found in both the automatic and manual extraction for the original image versus Area 2. This shows a

Table 4.15: The Wilcoxon Signed-Rank test p-values for the original images versus the areas removed for the subjective method.

Subjective Area	Automatic P-Value	Manual P-Value
Original vs. Area 1	0.125	0.0585
Original vs. Area 2	0.25	0.4375
Original vs. Area 3	0.3125	0.331
Original vs. Area 4	1	0.421

Table 4.16: The Wilcoxon Signed-Rank test p-values for the original images versus the areas removed for the objective method.

Objective Area	Automatic P-Value	Manual P-Value
Original vs. Area 0	0.75	0.922
Original vs. Area 1	0.375	0.0600
Original vs. Area 2	0.0312	0.00011
Original vs. Area 3	-	-

significant difference in the ranks which may be caused by a decrease in the number of fingerprints that are blank causing higher ranks (Table 4.16).

4.3.2.9 CMC Curves

In the CMC curves for the manual extraction subjective method, it can be seen that 40% of all of the samples were within the first 100 ranks for the AFIX Tracker system. The low rate of identification could be due to several factors including the quality of the ten-print cards and as well as the quality of the latent fingerprints. The low quality in some of the ten-print cards can affect the possibility of providing a high match score. The same can be said for a low quality latent fingerprint. If the latent fingerprint is of a lower quality, there is a lower likelihood of minutiae being detected which can translate into either a low match score or no match at all.

The original images showed to have a classification rate of 14.6% for the rank one identification. The top ten ranks provided an identification rate of 35%. In the case of Area 1, the rank 1 identification rank was lower than the original at 12.93% and the top ten identification rate was the same as the original images. Area 2 had the highest rank 1 identification rate of 16.37% and the top ten identification rate was higher than the previous two areas at 37.6%. Area 3 had a rank 1 identification rate of 10.3% and the top ten identification rate of 27%. Area 4 had the lowest identification rate for rank 1 at 7.96% and the lowest top ten identification rate of 22%. This low rate could be explained by the lower amount of the fingerprint available for marking and matching. The CMC curve can be seen in Figure 4.5.

The CMC curve for the automatic extraction for the subjective method showed to have about 30% of all of the samples being within the top 100 ranks for the AFIX Tracker system. The automatic extraction showed to be higher than the manual extraction; however, this is still a low identification rate for the system. Again, this low rate can be due to the low quality in the ten-print cards as well as the quality of the latent fingerprints.

The original images showed to have an identification rate of 9.5% for rank 1 identification. The top ten rank identification rate was 27%. The Area 1 rank 1 identification rate was 9.89% while the top ten identification rate of 28% which is higher than the original images. The Area 2 rank 1 had an identification rate of 9.27% and a top ten identification rate of 25%. Area 3 had a low rank 1 identification rate of 7.76% and a top ten identification rate of 21%. Area 4 had a higher rank 1 identification than Area 3 of 8.23% and a lower top ten identification rate of 18%. The CMC curve can be seen in Figure 4.6.

In the comparison of the manual extraction versus the automatic extraction, the manual extraction returned a higher rank 1 identification rate than the automatic extraction by about 10%. In observing the different areas, the manual extraction Area 2 had the highest rank 1 identification rate while the automatic extraction Area 1 had the highest rank 1 identification. Overall, the manual extraction appears to produce a higher rank 1 identification and a higher top ten identification rate.

The objective method for the manual extraction showed to have about 40% of all samples being within the top 100 ranks for the AFIX Tracker system. Area 3 showed to not produce a curve on the plot which can be explained with several of the images being white and not able to produce a result.

The original samples showed to have a rank 1 identification rate of 12.93% and a top ten identification rate of 33%. Area 0 had a rank 1 identification rate of 12.2% with a top ten identification rate of 34%. Area 1 showed a decline in the rank 1 identification rate at 10.7% and a lower top ten identification rate of 27%. Area 2 had the lowest rank 1 identification rate of 2.5%. No top ten rank identification rate could be made due to the results producing rank 1 and rank 2 identifications. This could be due to a small number of samples able to produce rank scores as well as a lower amount of the fingerprint available for marking. The CMC curve can be seen in Figure 4.7.

In the objective method with the automatic extraction, about 32% of all of the sample were within the top 100 ranks for the AFIX tracker system. Areas 2 and 3 did not produce a curve on the plot which could be explained with several of the images being white and not able to produce a result.

The original samples produced a rank 1 identification rate of 6.89% and a top ten identification rate of 19%. Area 0 produced a rank 1 identification rate of 8.88% and a top five identification rate of 27%. Area 1 produced a rank 1 identification rate of 12.5% and a top five identification rate of 32%. The CMC curve can be seen in Figure 4.8.

The automatic extraction, overall, showed to produce lower amount of ranks for the samples. They do, however, show to have a higher rank 1 identification rate.

In comparison of the manual extraction versus the automatic extraction, the manual

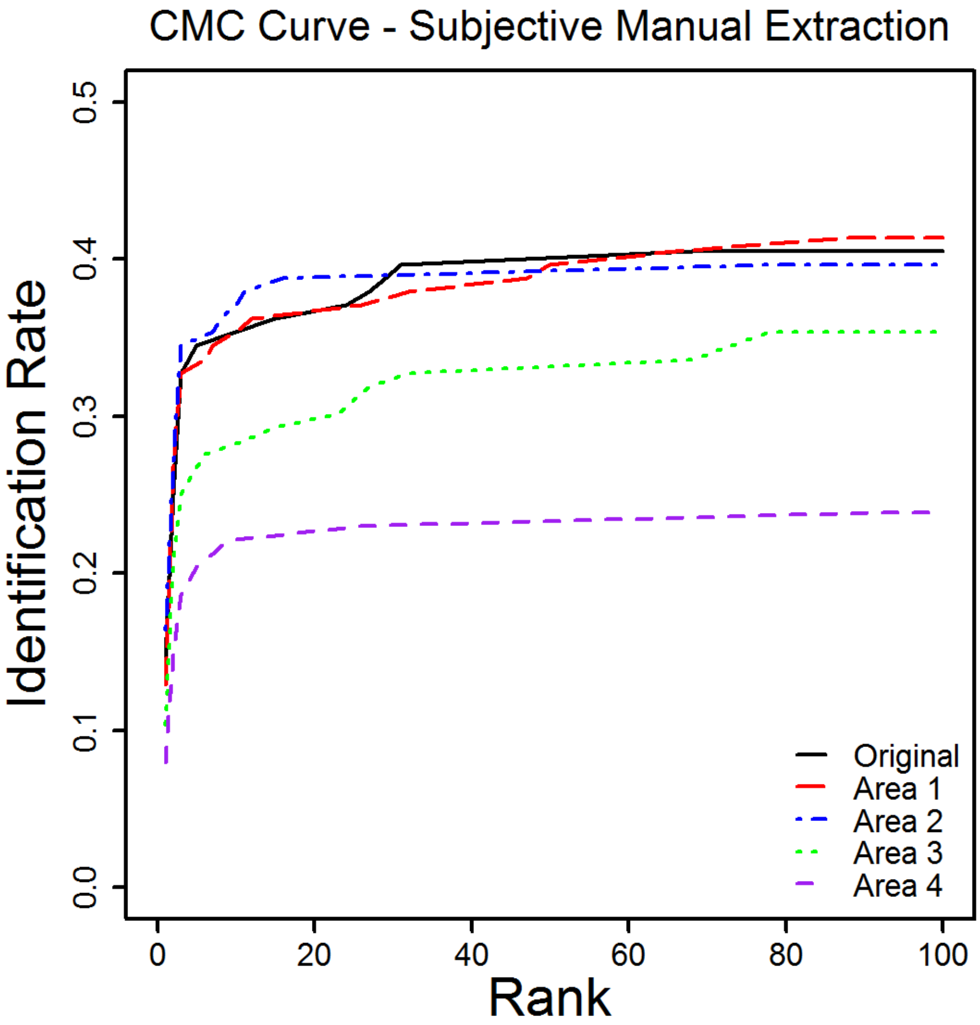


Figure 4.5: The CMC curve for the subjective removal in the manual extraction of minutiae for the AFIX Tracker® system.

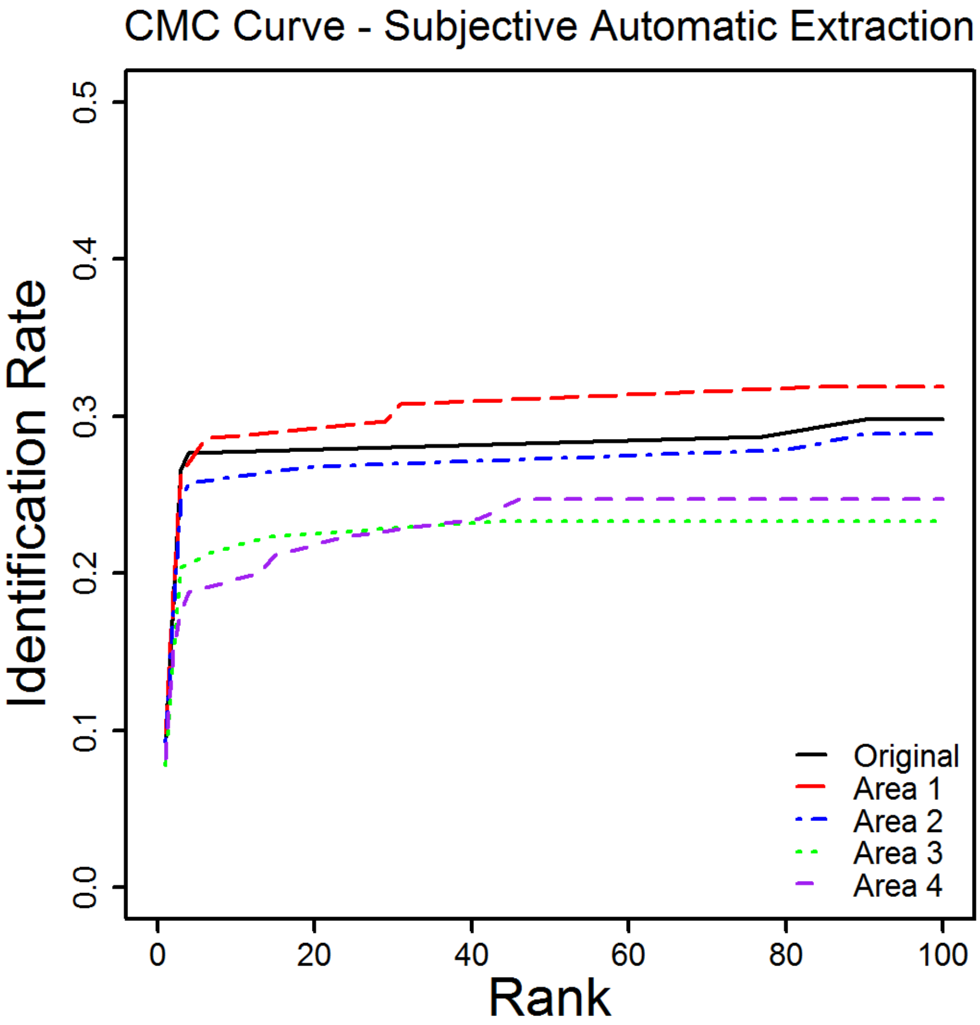


Figure 4.6: The CMC curve for the subjective removal in the automatic extraction of minutiae for the AFIX Tracker® system.

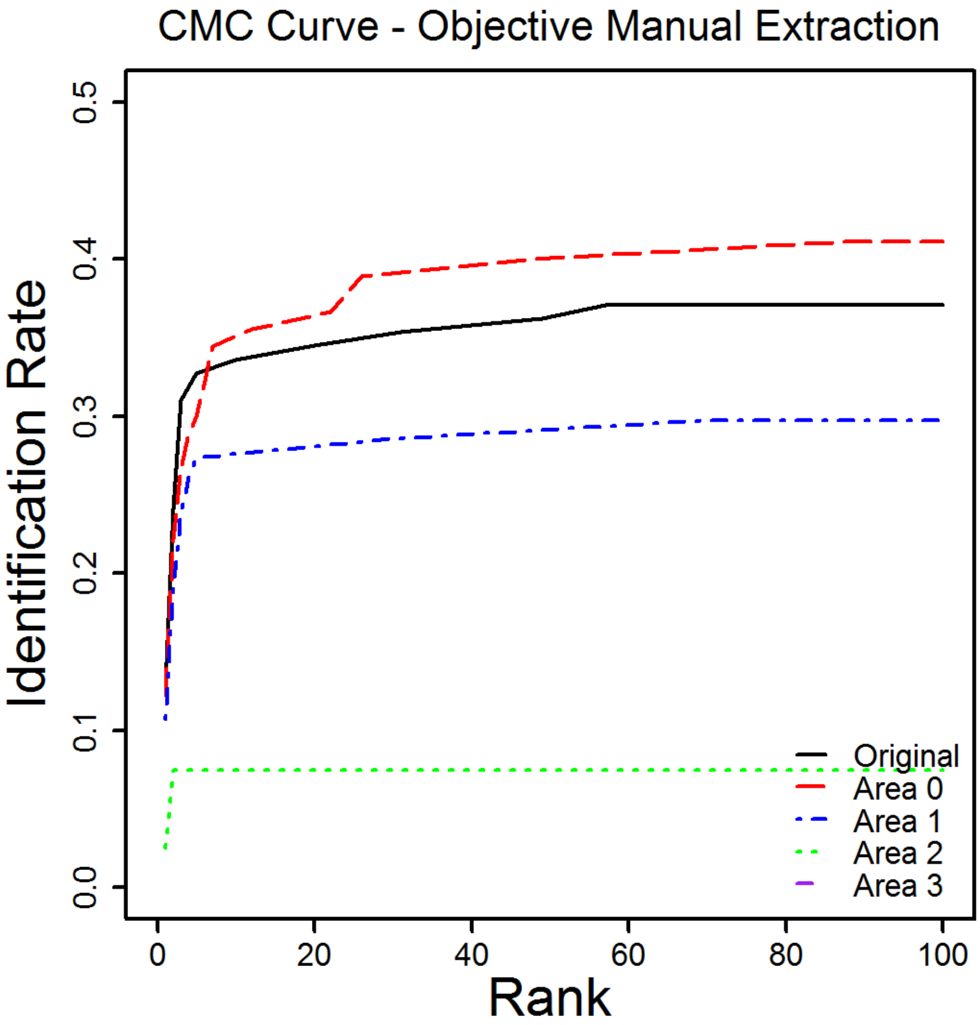


Figure 4.7: The CMC curve for the objective removal in the manual extraction of minutiae for the AFIX Tracker® system.

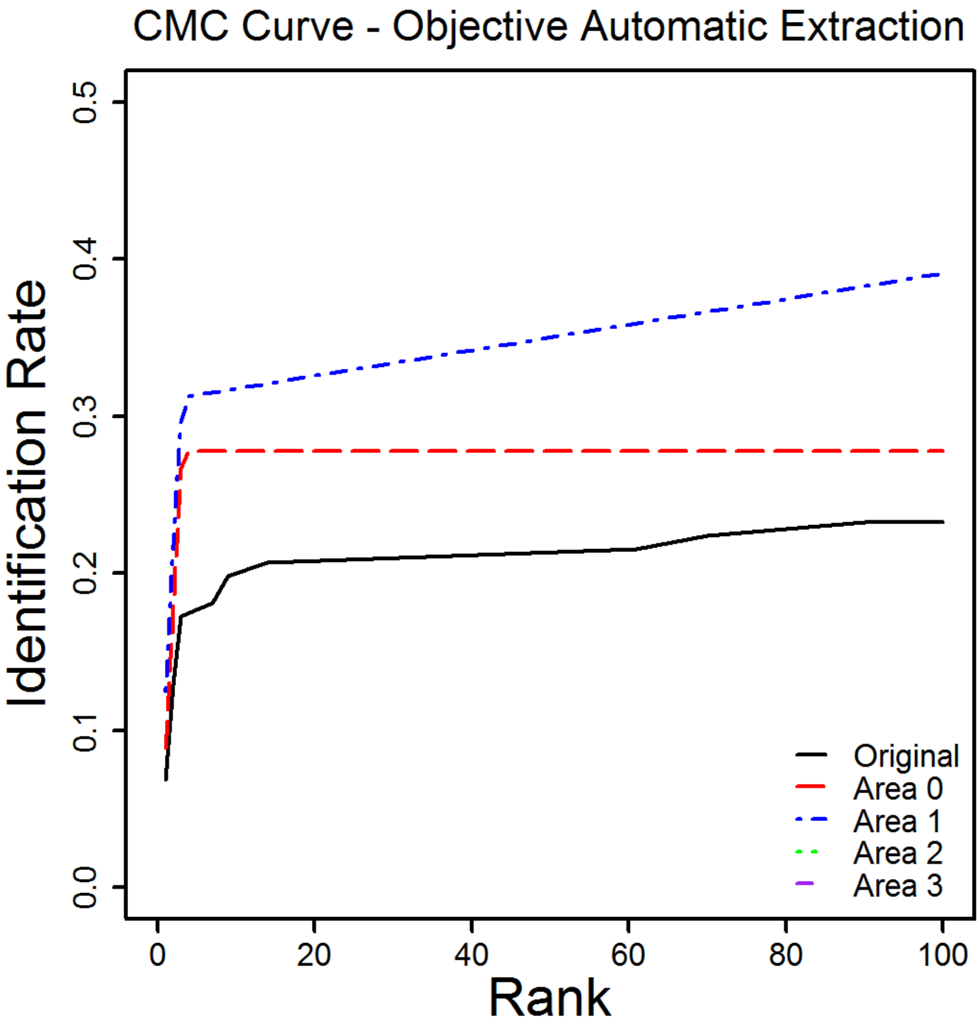


Figure 4.8: The CMC curve for the objective removal in the automatic extraction of minutiae for the AFIX Tracker® system.

extraction showed to have higher rank 1 identification rates than in the automatic extraction and an overall higher identification rate return. When looking at the different areas removed, the manual extraction had the highest rank 1 identification rate while Area 1 in the automatic extraction had the highest rank 1 identification rate.

When comparing the subjective to the objective method, the subjective manual extraction method showed to have the highest rank 1 identification rate overall. This result is surprising due to the thought that the objective method would produce higher results with the limit of human interaction.

A question raised in comparing the CMC curves to the ROC curves results was why are these two results different from one another? The ROC curves provide a good verification of the system with high AUC values and low EER values whereas the CMC curves are showing low identification rates of the system. DeCann and Ross provide some insight into this problem. The ROC curves are used to observe the probability that a genuine and impostor scores are correctly classified while the CMC curves observe the probability that the matching algorithm returns the correct identification within the ranks [31]. An explanation for the low identification rate could be due to the increase in the database size. With a higher number of samples in the database, there is a higher chance that a true match will not be the top match [31]. In the case of the smaller database with high quality ten-prints, there is a higher chance that the true match will be correctly identified. For the database used in this research, the database was large with a range in the quality of the ten-print cards. This database was searched with a range in qualities of the latent fingerprints ultimately effecting the outcome of the identification rate for this project.

4.3.3 BOZORTH3

The BOZORTH3 normalized datum was plotted with rank versus the match score. For all of the results, a large number of samples produced no match score but did have a rank present. This is due to all of the samples in the gallery were returning a match score and having a rank. A large majority of the match scores fell between the values of 0.2 and 0.5. There were a few match score above with the highest match score of 0.8. The ranks for the fingerprints ranged from zero to about 32,000 due to the number of samples in the database. For the subjective samples, the match scores were below 0.5. However, the objective samples were shifted to a lower match score and a large amount of samples that had a match score of zero. Visually, it appears that subjective method produces higher rank scores in the method. The original subjective samples showed to have the a large amount of match scores below the rank score of 10,000 and the match score were shifted higher. The Area 1 sample match scores were shifted to between 0.3 and 0.6. The same was seen in the remaining area removal.

In the objective original samples, the large majority of the samples were ranked 10,000 and above with a match score of 0.2 to 0.35. For the Area 0 samples, a smaller number of samples had a match score of zero. The majority of the samples were ranked 10,000 and above with match scores between 0.25 and 0.35. The same was seen in Area 1. In the

Table 4.17: The AUC values along with the equal error rates (EER) at the crossing of the match score and the rate for the subjective and objective methods for the match scores.

Subjective				Objective				
Area	AUC	EER	Position	Area	AUC	EER	Position	p-Value
Original	0.571	0.5	5.39	Original	0.566	0.5	5.6	0.88
1	0.605	0.4	5.6	0	0.585	0.5	5.39	0.493
2	0.412	0.5	5.6	1	0.618	0.5	5.6	0.310
3	0.62	0.5	5.18	2	0.878	0.2	4.13	¡2.2 x 10 -16
4	0.723	0.4	4.76	3	0.958	0.1	3.5	¡2.2 x 10 -16

Area Two samples, a large number of received a match score of zero while the majority of the remaining samples were ranked 10,000 and above. The highest match score produced was 0.5 with a high rank. The four samples all received match scores of above 0.20 and rank scores of 8,000 and higher. With the high ranks produced, this algorithm is not ideal to use.

4.3.3.1 ROC Curves

ROC curves were produced for the BOZORTH3 match scores and the rank scores. AUC scores for the results were not very high with the AUC scores ranging from 0.571 for the original image to 0.723 for Area 4 in the subjective method. The EER for all the methods were between 0.4 and 0.5. This is a high error rate and ideally error rate should be as low as possible. The objective method showed to have higher AUC rates between 0.585 for the original and 0.949 for Area 0. The EER rates ranged from 0.1 to 0.5 with Area 3 having the lowest EER of 0.1. The wide range of EER was interesting to see especially with the high AUC value for Area 0. With the low AUC scores and higher error rates, BOZORTH3 is not the best matching algorithm to use (Table 4.17). The p-values showed a significant difference in the subjective Area 3 versus objective Area 2 and subjective Area 4 versus objective Area 3 with the p-values for both being less than 2.2 x 10-16. This result could be due to the images having a lower amount of surface area to mark and a difference in the areas removed.

The rank scores showed similar AUCs for the subjective method and the EERs were all in the range of 0.4 to 0.5 which shows that the method does not perform well overall. In the objective method, the results were similar to the subjective (Table 4.18). The p-values showed a significant difference in subjective Area 3 versus objective Area 2 with a p-value of 0.00229.

4.3.3.2 Wilcoxon Signed-Rank Test

The Wilcoxon signed-rank test was performed for the individual areas against the original image to determine if there is a significant different in the rank scores. For the subjective

Table 4.18: The AUC values along with the equal error rates (EER) at the crossing of the rank score and the rate for the subjective and objective methods for the rank scores.

Subjective				Objective				
Area	AUC	EER	Position	Area	AUC	EER	Position	p-Value
Original	0.423	0.6	14203	Original	0.557	0.5	14622	0.588
1	0.416	0.6	13726	0	0.576	0.5	15462	0.827
2	0.545	0.5	15042	1	0.612	0.4	13783	0.710
3	0.554	0.5	15252	2	0.708	0.3	10845	0.00229
4	0.631	0.4	13573	3	0.679	0.4	13153	0.480

Table 4.19: The Wilcoxon Signed-Rank test p-values for the original images versus the areas removed for the subjective and objective method.

Area	Subjective p-Value	Objective p-Value	Area
Original vs. Area 1	0.341	3.51e-07	Original vs. Area 0
Original vs. Area 2	0.955	0.0017	Original vs. Area 1
Original vs. Area 3	0.957	0.1443	Original vs. Area 2
Original vs. Area 4	0.007	9.207e-05	Original vs. Area 3

method, the p-values were not significantly different except for Area 4 which showed a significant difference with a p-value of 0.007. This shows a significant difference in the ranks which may be due to the decrease in the area leading to blank images.

In the objective samples, all of the ares, with the exception of Area 2, showed to have significant differences between the ranks which could be explained by the areas removed.

4.3.3.3 CMC Curves

In the CMC curves for the subjective method, it can be seen that 80% of all of the samples were within the top 35000 ranks for the BOZORTH3 algorithm. The low rate of identification could be due to several factors including the quality of the ten-print cards, the quality of the latent fingerprints and the way the algorithm performs.

The original samples did not produce any rank 1 or top ten identification scores. The ranks for matches do not occur until rank 48 and was an extreme low rate of 0.6%. Area 1 did not produce a rank score until rank 206 with a rate of 0.6%. Area 2 produced the highest rank of 17 with a rate of 0.6%. Area 3 and 4 produced ranks at 144 and 261, respectively, with a rate of 0.6%. The results of using this matching algorithm do not produce a high identification and very low rank scores. These results are in line with those produced by the ROC curves. The CMC curves can be seen in Figure 4.9.

In the CMC curves for the objective method, it can be seen that 97% of all of the samples were within the top 33000 ranks for the BOZORTH3 algorithm. The low rate of

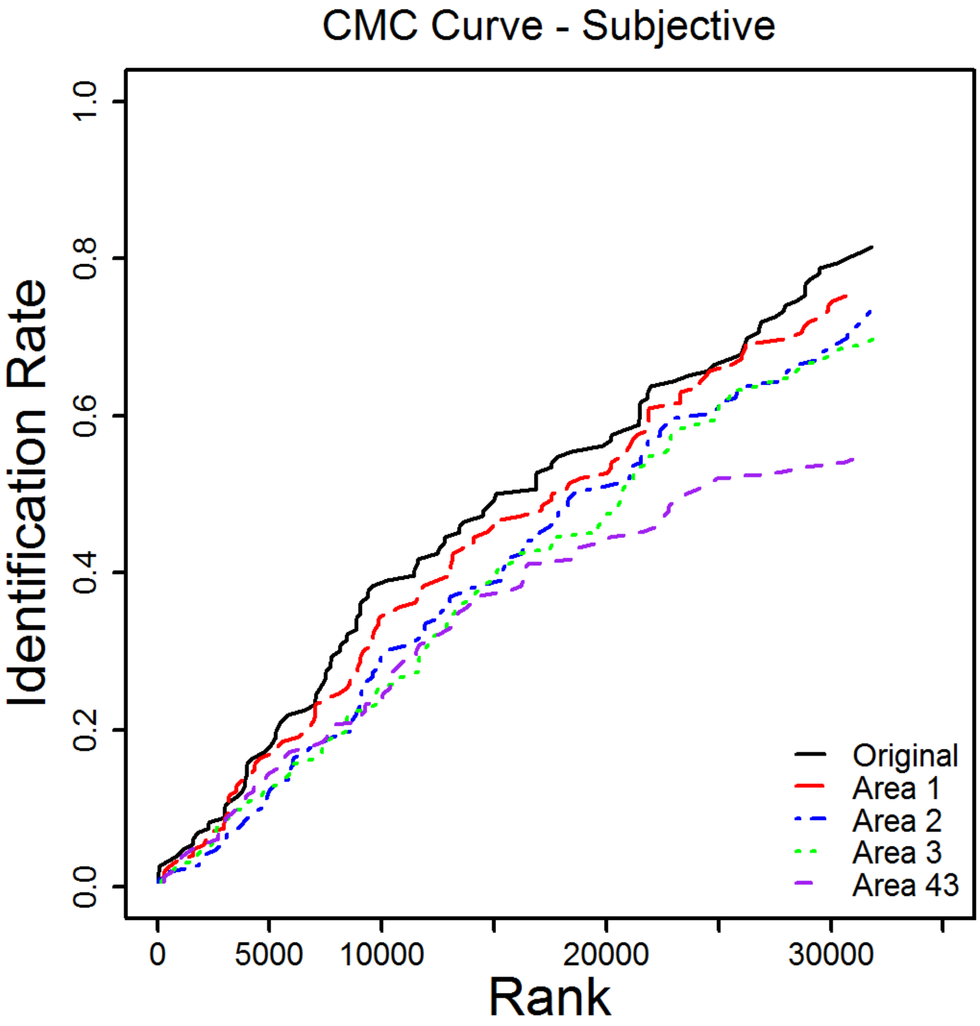


Figure 4.9: The CMC curve for the subjective method from the BOZORTH3 algorithm.

Table 4.20: The correlation scores for the automatic extraction versus manual extraction for the subjective and objective areas.

Subjective		Objective	
Area	Correlation	Area	Correlation
Overall	0.5383	Overall	0.6147
Original	0.5402	Original	0.574
Area 1	0.3865	Area 1	0.5182
Area 2	0.7023	Area 2	0.509
Area 3	0.5856	Area 3	-
Area 4	0.6740	Area 4	-

Table 4.21: The correlation scores for the BOZORTH3 algorithm and the automatic extraction for AFIX Tracker® for the subjective and objective areas.

Subjective		Objective	
Area	Correlation	Area	Correlation
Overall	0.1299	Overall	0.1939
Original	0.1637	Original	0.1119
Area 1	0.0824	Area 0	0.0315
Area 2	0.0738	Area 1	0.030
Area 3	0.1647	Area 2	-
Area 4	0.1963	Area 3	-

identification could be due to several factors including the quality of the ten-print cards, the quality of the latent fingerprints and the way the algorithm performs.

The original samples did not produce any rank 1 or top ten identification scores. The ranks for matches do not occur until rank 106 and is an extreme low rate of 0.6%. Area 0 did not produce a rank score until rank 362 with a rate of 0.6%. Area 1 produced the highest rank of 31 with a rate of 0.6%. Area 2 and 3 produced ranks at 479 and 209, respectively, with a rate of 0.6%. The results of using this matching algorithm do not produce a high identification and very low rank scores. These results are in line with those produced by the ROC curves. The CMC curves can be seen in Figure 4.10.

4.3.4 Correlation Scores

The Spearman's Rank-Order correlation scores were computed to determine if a relationship exists between the two algorithms and the two extraction methods. Correlations were computed for the automatic extraction versus manual extraction, BOZORTH3 versus automatic extraction and the BOZORTH3 versus manual extraction.

In the BOZORTH3 versus the automatic extraction, the overall correlation score for the subjective method was 0.129 for a very weak correlation between the two methods. A

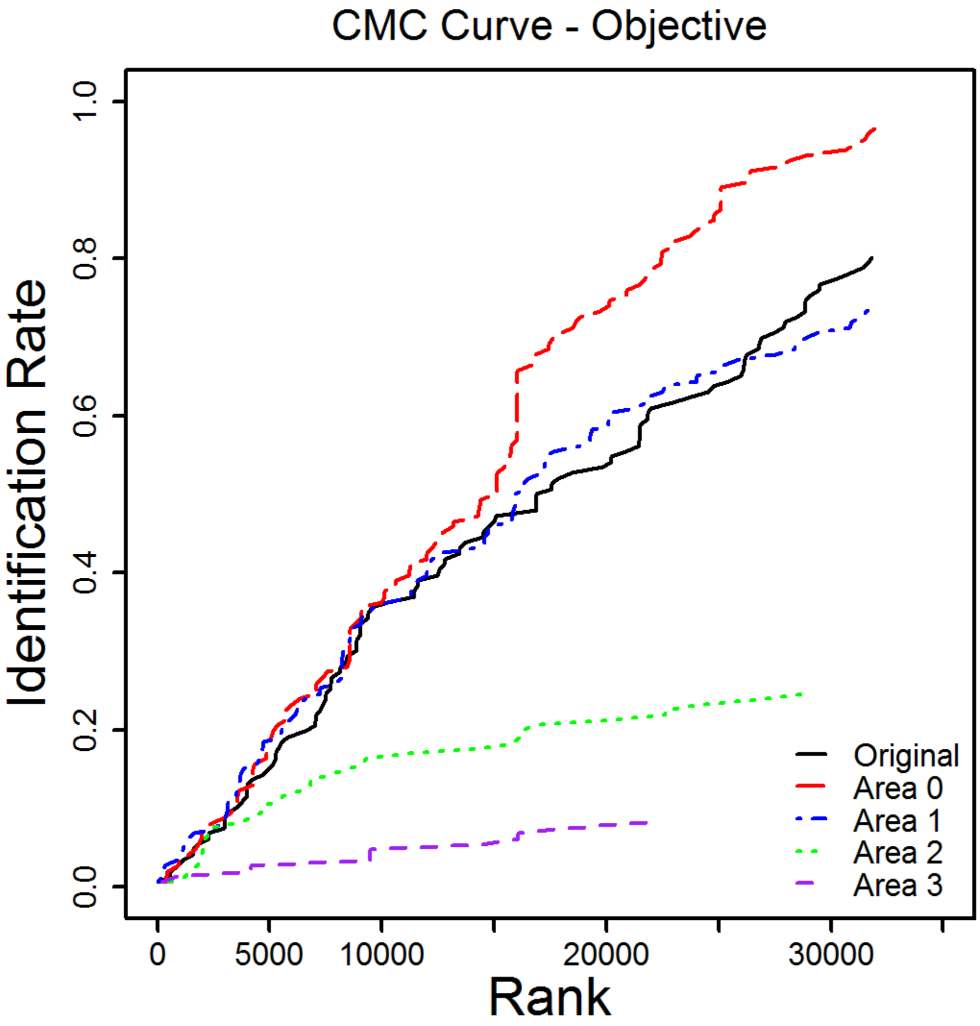


Figure 4.10: The CMC curve for the objective method from the BOZORTH3 algorithm.

Table 4.22: The correlation scores for the BOZORTH3 algorithm and the manual extraction for AFIX Tracker® for the subjective and objective areas.

Subjective		Objective	
Area	Correlation	Area	Correlation
Overall	0.1862	Overall	0.3027
Original	0.0946	Original	0.1111
Area 1	0.2733	Area 0	-0.0069
Area 2	-0.0069	Area 1	0.2018
Area 3	0.3235	Area 2	0.2605
Area 4	0.1974	Area 3	-

similar weak correlation value was seen with the objective method and remaining areas showing the two algorithms do not have a strong relationship to one another.

A similar trend was seen in the manual extraction where there were not strong correlation scores between the two algorithms. This gives evidence that the two algorithms do not have a relationship to one another.

Interestingly, the correlation scores for the manual extraction versus automatic extraction for both of the methods were above 0.5 giving a moderate relationship between the methods. This result can be due to the use of the same algorithm after the minutiae extraction.

4.3.5 Comparison of Subjective vs. Objective Area Removal

For the comparison of the subjective and objective methods of removal, the pixels counts for the subjective method, the objective method and the intersection image. The intersection value is divided by the amount of subjective gray pixels. The same was completed for the objective method. The resulting values were then plotted against one another. The amount of images where the subjective images contained more gray pixels than the objective images occurred in 181 images and 19 times in the cases were the objective images contained more pixels than the subjective method.

In Figure 4.11, the points located on the dashed lines represent images were the subjective and objective method were similar in area removal. The points that lie above the dashed line represent images were the objective method retained more pixels than the subjective method. Those that lie below the dashed line represent the images were the subjective method retained more of the fingerprint images.

In the case of objective Area 0 versus the subjective Area 1, it appeared that seven images had more area kept by the objective images than the subjective images. These images were developed with ninhydrin (three images) and black powder (four images). Two of the ninhydrin fingerprints appeared to have a larger difference in the removal between the methods. In the objective Area 1 versus subjective Area 2, eleven images (three ninhydrin and eight black powder) had more of the image removed in the subjective method than the objective method. In objective Area 2 versus subjective Area 3, it appears that only one samples (ninhydrin) has more of the image remaining in the objective image than the subjective image. It appears that the images are now being shifted down and to the right of the plot representing the objective method removing a higher amount than in the subjective method. In the final area comparison of objective Area 3 versus subjective Area 4, one image (ninhydrin) showed to have an equal amount the image removed by both methods of removal. The remaining images were shifted toward the bottom right showing that the objective image removed a higher amount of the image and the subjective kept more of the image.

When looking at the entire plot and examining each panel, it appears that with each removal, the subjective method is more conservative in leaving portions of the fingerprint. Overall, this gives visualization to how the different methods are removing portions of a fingerprint. With this data, there is a high divergence between the two methods and is not ideal. The two methods should ideally be very similar in the removal of the areas.

4.4 Conclusions

The main goal of this research was to determine if removing the bad quality areas from a fingerprint will help increase the match scores of the images. Overall, this was proven in the research with the subjective manual extraction Area 2 providing an increase in the match scores.

The sample size for this project was low with 50 samples of varying development methods, substrates, and quality. More samples are needed overall to add to the data analysis.

In the assessment of the fingerprint quality, the NFIQ global quality assessment algorithm scores did not correspond to the qualitative quality category assigned by the user. The NFIQ scores were mainly categorized as a value of one (high quality) or a value of five (low quality). In further examination, each NFIQ category was spread between the different qualitative categories showing that the algorithm is not accurately reflecting the visualization by the user. It may be the case that either the categories assigned by the user are not reflective of the sample or that the NFIQ algorithm needs to be further trained to truly reflect what the user is visualizing. Further research should be conducted to investigate both possibilities.

Once the areas of the fingerprints were removed subjectively and objectively, the number of minutiae were found with using AFIX Tracker® through manual extraction and automatic extraction. Overall, the manual extraction showed to produced more minutiae than AFIX Tracker® automatic extraction which shows that the user interaction can still be used to help increase the number of minutiae available for searching. Automatic extraction can also lead to the extraction method marking false minutiae in the fingerprint which can cause an issue later with the matching algorithm. The development method of black powder produced a higher number of minutiae to search which lead to higher match scores seen. The substrate of glass produced interesting results with high match scores

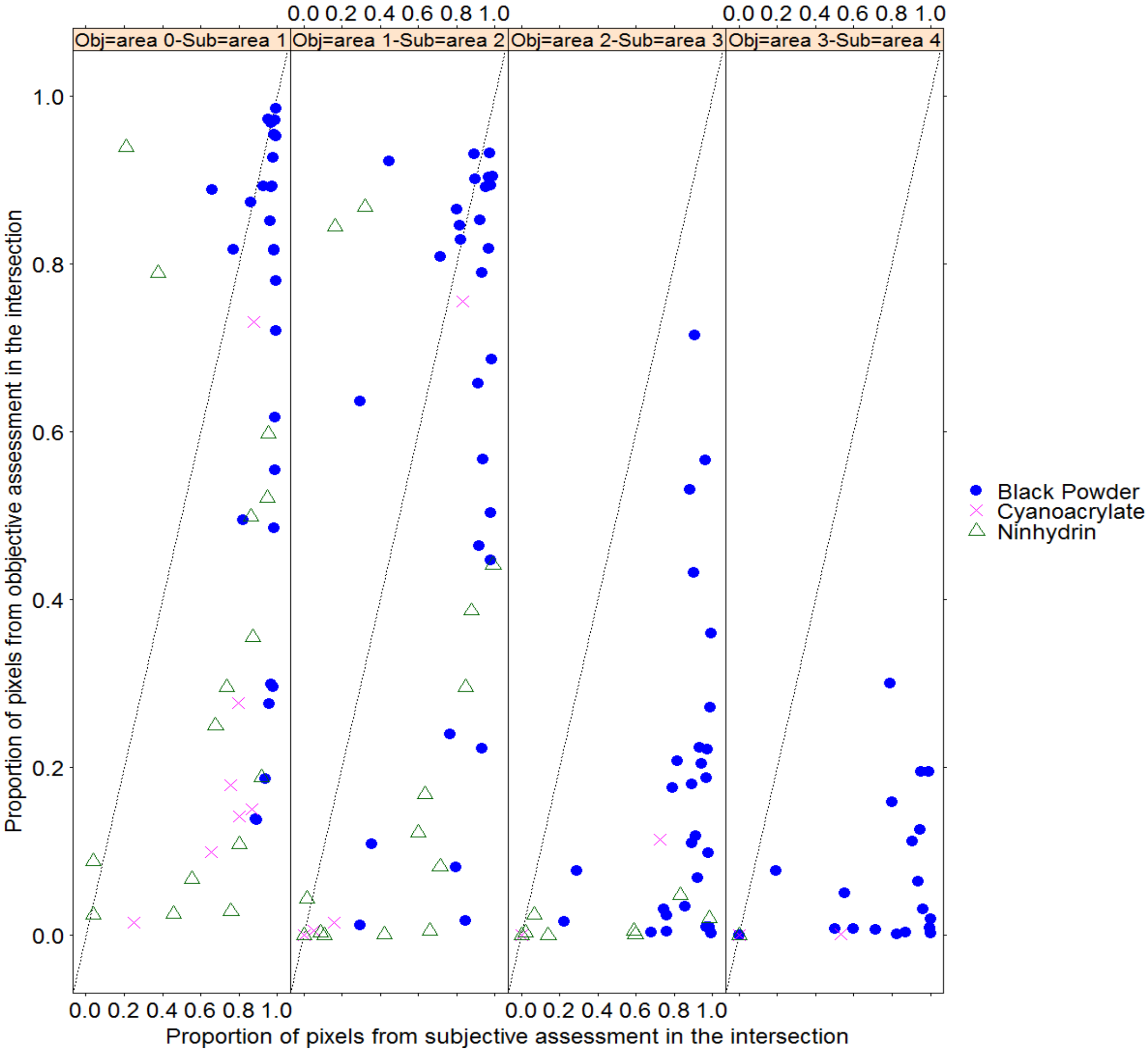


Figure 4.11: This figure presents the proportion of the objective pixels to the proportion of the subjective pixels. The panels from left to right represent objective Area 0 versus subjective Area 1, objective Area 1 versus subjective Area 2, objective Area 2 versus subjective Area 3 and objective Area 3 versus subjective Area 4. The blue circles represent the black powder development, the pink circles represent the cyanoacrylate development and the green circles represent the ninhydrin development.

for a range of the quality in the images. A question raised is whether or not more glass samples would produce the same trend or create a large range in match scores. Another question raised was would an increase in the number of samples for each development method show the same trend in the black powder producing more minutiae and higher match scores in comparison to the other two development methods. The same question can be raised for the case of the substrates.

ROC curves were produced to examine the performance of both the algorithms and how the different areas removed perform. The AUC scores for the subjective and objective methods showed to be above 0.85 providing a good performed of those areas in the AFIX Tracker® algorithm. The p-values provided evidence that neither method outperformed the other; however, some area removals were different from one another which could be due to the difference in the fingerprint available to mark. Further investigation into these results could be made in the future. In the BOZORTH3 ROC curves, the overall AUC scores were very low producing values below 0.7. In the case of the two matching algorithms, the BOZORTH3 algorithm is not recommended to be used since the AUC scores obtained were close to the value of 0.5 when the ideal AUC is close to one.

The rank scores were utilized to determine if a significant difference in the ranks for the different areas removed were seen in comparison to the original images using the Wilcoxon Signed-Rank Test. The subjective areas showed to have no significant differences in the difference area comparisons for both the automatic extraction and the manual extraction. However, the objective method showed to have significant difference in the original versus Area 2 in both extraction methods providing evidence that the original and the respective area providing differences in the ranks. This result leads to the questions of why these two areas are truly providing differences in the ranks. Could this be due to the amount of the fingerprint removed or even the amount of minutiae available to search? Further research should be conducted into why this result has occurred.

The CMC curves showed an interesting result that subjective method in the AFIX Tracker® system provided higher rank 1 and top 10 rank identification rates than in the objective method. It was hypothesized that the objective method would help improve the match scores, however, it does not appear to be the case at this current time. This result lends to show that a user should still be determining the areas of removal due to higher rank identification rates. Further more, the manual extraction for the subjective method showed to have higher identification rates in comparison to the automatic extraction. This provides evidence that the extraction of minutiae by the user is allowing for higher identification rates in comparison to the automated process showing that human interaction is still necessary to help increase the chance of producing a match by the system. Interestingly, would an increase in the number of samples increase or decrease the identification rate of the CMC curves or stay consistent based upon the quality of the samples?

The CMC curves for the BOZORTH3 algorithm showed to not produce ranks until outside of the top 10 identifications. Most identifications were found ranks 100 and above. This provides more evidence that the BOZORTH3 algorithm should not be utilized.

The correlation scores for the manual extraction versus automatic extraction provided a mild correlation between the two methods. This mild correlation could be due to using the same algorithm to obtain the rank scores. BOZORTH3 was compared to each of the extraction method and weak to no correlation was seen between the methods. This provides evidence that the two minutiae-based matching algorithms are performing differently from one another.

The removal of the different quality areas from the subjective and objective methods were compared to determine if the two removal methods were similar in the area removal. The results showed that the user is more conservative in the removal than in the objective method. When pairing this result with the CMC curve results, the conservation of the areas removed still provides enough area for minutiae to be found. The objective method removes more of the fingerprint reduces the amount of area to extract minutiae from.

The research project was able to produce a usable protocol that has shown an increase in the rank 1 and rank 10 identification rates. If researchers and/or fingerprint examiners were to utilize the areas removed, the ideal method would be to use the subjective area removal and the manual extraction of the minutiae. Of the areas removed, Area 2 had an increase in the rank 1 identification. Researchers and/or fingerprint examiners should also expose the original images to compare the identification matches are similar.

In the future, the NFIQ quality algorithm should be examined more closely to determine if any other parameters can be changed to help improve the quality score to reflect the visual assessment and create stricter qualitative categories In terms of the matching algorithms, the BOZORTH3 algorithm should not be utilized due to the poor performance with the images in this research. AFIX Tracker® is a viable option to be used with the higher AUC values obtained. The subjective method of removal does help improve the rank identification of a sample better than the objective method. The objective method should be explored farther to determine if modifying the NIST algorithms could improve the objective method.

A. Technical Note Codes

A.1 R® Padding Code

```
library(graphics)
library(png) #needed to read in the png image
library(tools)
library(EBImage)


rot<-function(m) t(m)[,nrow(m):1] #rotates the image
# =======================================================

###directories for input and output files
file.names =
dir("Z:/Alyshia_Meyers/NBIS_Files/Fingerprint_Images/original_images/",
pattern=".png")

#begins loop for padding
for(i in 1:length(file.names)){
  #fname <- "d"
  fname <- file.names[i]
  inDir <-
"Z:/Alyshia_Meyers/NBIS_Files/Fingerprint_Images/original_images/"
  outDir <- "Z:/Alyshia_Meyers/NBIS_Files/Processed_Files/"
  outDir.pad <-
"Z:/Alyshia_Meyers/NBIS_Files/Fingerprint_Images/original_images/Padded/"



  subDir <- paste0(outDir,file_path_sans_ext(fname))#removes the .png
extensions
  cat(subDir,"\n")

  if (file.exists(subDir)){
    setwd(file.path(subDir))
  } else {
    dir.create(file.path(subDir))
    setwd(file.path(subDir))
  }

  # ======================================
  ##padding the original image
  inDir.img <- paste0(subDir, ".png")
  f.print<-readPNG(inDir.img) #reads in png in the grayscale format
  #values between 0 and 1, white is 1 and black is 0, gray is in between
img

  #dim(f.print)

  n.rows.img=nrow(f.print)
  n.cols.img=ncol(f.print)
  # n.rows.img
  # n.cols.img

  row.mod <- n.rows.img %% 8
  # row.mod
  col.mod <- n.cols.img %% 8
  # col.mod

  if (row.mod == 0){n.pad.row <- 0
  }else{
    n.pad.row <- 8 - row.mod
    vals <- n.pad.row*n.cols.img
    out <- rep.int(1, vals)
    out.mat <- matrix(out,nrow=n.pad.row)
    #out.mat <- data.frame(out.mat)
    f.print <- rbind(f.print, out.mat)
  }
  n.rows.img=nrow(f.print)
  #n.rows.img

  if (col.mod == 0){n.pad.col <- 0
  }else{
    n.pad.col <- 8 - col.mod
    vals <- n.pad.col*n.rows.img
    out <- rep.int(1, vals)
    out.mat <- matrix(out,ncol=n.pad.col)
    #out.mat <- data.frame(out.mat)
    f.print <- cbind(f.print,out.mat)
  }
  #n.cols.img=ncol(f.print)
  #n.cols.img

  f.print <- data.matrix(f.print) #needs to be created into a matrix to
write as an image file
  f.print <-rot(f.print)
  f.print <-rot(f.print)
  f.print <-rot(f.print)
  #f.print <-rot(f.print)

  image(f.print)
  n.rows.img=nrow(f.print)
  n.cols.img=ncol(f.print)
  n.rows.img
  n.cols.img

  f.print <- apply(f.print, 2, rev)

  fnameout<-paste0(outDir.pad, file_path_sans_ext(fname), "_pad.png")
  f.print<- channel(f.print, "gray")
  writeImage(f.print, fnameout, dpi = c(1000,1000))


  #dim(f.print)

 }#ends loop for padding
  #ends padding
```

A.2 ImageJ Grayscale Macro

```
run(”8-bit)
close()
```

A.3 R® Expansion Code

```
library(graphics)
library(png) #needed to read in the png image
library(tools)
library(EBImage)


rot<-function(m) t(m)[,nrow(m):1]

###directories for input and output files
file.names = dir("Z:/Alyshia_Meyers/NBIS_Files/mindtct/", pattern=".qm")

#begins loop for padding
for(i in 1:length(file.names)){
  #fname <- "d"
  fname <- file.names[i]
  inDir <-
"Z:/Alyshia_Meyers/NBIS_Files/Fingerprint_Images/original_images/"
  outDir <- "Z:/Alyshia_Meyers/NBIS_Files/Processed_Files/"
  outDir.pad <-
"Z:/Alyshia_Meyers/NBIS_Files/Fingerprint_Images/original_images/Padded/"

  inDir.qm <- "Z:/Alyshia_Meyers/NBIS_Files/mindtct/"
  inDir.pad <-
"Z:/Alyshia_Meyers/NBIS_Files/Fingerprint_Images/original_images/Padded/"


########## reads in quality map
inName.qm <- paste0(inDir.qm,fname)
test <- as.matrix(read.table(inName.qm, header=FALSE, sep = "",
as.is=TRUE))#readsin the quality map
test <- data.frame(test)


n.col <- ncol(test)#number of columns
n.row <- nrow(test)#number of rows

i.col <- n.col * 8
i.row <- n.row * 8

img <- matrix(, i.row, i.col)

test.row <- 1
test.col <- 1

for (i in 1:i.row){
  for (j in 1:i.col){ #determine if the numer of row are a factor of 8

    if (i %% 8 == 0){
      img[i,j] <- test[test.row, test.col] #if the rows are not a factor
of 8, adds the number of pixels necessary
      if(j==i.col){test.row <- test.row + 1}
    }
    if (j %% 8 == 0){
      img[i,j] <- test[test.row, test.col] # performs the same test with
the columns
      if (test.col %% n.col == 0){test.col <- 1}else{test.col <-
test.col + 1}
    }
    else
    {
      #        cat(test[test.row, test.col],"\n")
      img[i,j] <- test[test.row, test.col] #puts the image together
    }

  } # end for j
  #cat(i/8, "\n")
} # end for i

#img.out <- data.frame(img)
img[i.row, i.col]<-img[i.row, i.col-1] #this is a cheat!!!

#writes out the expanded quality map
fnameout <- paste0(outDir, file_path_sans_ext(fname)) #writes out the
image
outExp.qm <-
paste0(fnameout,"/",file_path_sans_ext(fname),"_pad_qm_out.csv")
write.csv(img, outExp.qm, row.names=FALSE)
}
#############finishes quality map expansion
```

A.4 R® Objective Successive Removal Code

```
library(graphics)
library(png) #needed to read in the png image
library(tools)
library(EBImage)


rot<-function(m) t(m)[,nrow(m):1] #rotates the image
# =======================================================

###directories for input and output files
file.names = dir("Z:/Alyshia_Meyers/NBIS_Files/mindtct/",
pattern=".png")

#begins loop for padding
for(i in 1:length(file.names)){
  #fname <- "d"
  fname <- file.names[i]
  inDir <-
"Z:/Alyshia_Meyers/NBIS_Files/Fingerprint_Images/original_images/"
  outDir <- "Z:/Alyshia_Meyers/NBIS_Files/Processed_Files/"
  outDir.pad <-
"Z:/Alyshia_Meyers/NBIS_Files/Fingerprint_Images/original_images/Padded/"

  inDir.qm <- "Z:/Alyshia_Meyers/NBIS_Files/mindtct/"
  inDir.pad <-
"Z:/Alyshia_Meyers/NBIS_Files/Fingerprint_Images/original_images/Padded/"


##############Removes low quality successively

#read in from in inDir
fnameout <- paste0(inDir.pad, file_path_sans_ext(fname))
inDir.img <- paste0(fnameout, "_pad.png") #reading in the padded image
f.print<-readPNG(inDir.img) #reads in png in the grayscale format
#values between 0 and 1, white is 1 and black is 0, gray is in between
img

qm<-img

n.rows.qm=nrow(qm)
n.cols.qm=ncol(qm)

#removes 0s in quality map
for (i in 1:n.rows.qm) {
  for (j in 1:n.cols.qm){
    if (as.numeric(qm[i,j])== 0) {f.print[i,j]<- 1.0} #finds the values
of 0 in the quality map and
  }#closes second loop     #replaces them with the pixel value of 1 in
the original image
}#closes first loop
```

```
f.print.out<-rot(f.print)
f.print.out <-rot(f.print.out)
f.print.out <-rot(f.print.out)
f.print.out <- apply(f.print.out, 2, rev)

fnameout <- paste0(outDir, file_path_sans_ext(fname))
out.fname <-
paste0(fnameout,"/",file_path_sans_ext(fname),"_pad_zero.png")
f.print.out<- channel(f.print.out, "gray")
writeImage(f.print.out, out.fname, dpi = c(1000,1000))

#removes 1s in quality map
for (i in 1:n.rows.qm) {
  for (j in 1:n.cols.qm){ #finds the values of 1 in the quality map and
    if (as.numeric(qm[i,j])== 1) {f.print[i,j]<- 1.0}#replaces them with
the pixel value of 1 in the area 0 image
  }#closes second loop
}#closes first loop

f.print.out<-rot(f.print)
f.print.out <-rot(f.print.out)
f.print.out <-rot(f.print.out)
#f.print <-rot(f.print)
f.print.out <- apply(f.print.out, 2, rev)


fnameout <- paste0(outDir, file_path_sans_ext(fname))
out.fname <-
paste0(fnameout,"/",file_path_sans_ext(fname),"_pad_one.png")
f.print.out<- channel(f.print.out, "gray")
writeImage(f.print.out, out.fname, dpi = c(1000,1000))

#removes 2s in quality map
for (i in 1:n.rows.qm) {
  for (j in 1:n.cols.qm){ #finds the values of 2 in the quality map and
    if (as.numeric(qm[i,j])== 2) {f.print[i,j]<- 1.0} #replaces them
with the pixel value of 1 in the area 1 image
  }#closes second loop
}#closes first loop

f.print.out<-rot(f.print)
f.print.out <-rot(f.print.out)
f.print.out <-rot(f.print.out)
#f.print <-rot(f.print)
f.print.out <- apply(f.print.out, 2, rev)

fnameout <- paste0(outDir, file_path_sans_ext(fname))
out.fname <-
paste0(fnameout,"/",file_path_sans_ext(fname),"_pad_two.png")
f.print.out<- channel(f.print.out, "gray")
writeImage(f.print.out, out.fname, dpi = c(1000,1000))
```

```
#removes 3s in quality map
for (i in 1:n.rows.qm) {
  for (j in 1:n.cols.qm){ #finds the values of 3 in the quality map and
    if (as.numeric(qm[i,j])== 3) {f.print[i,j]<- 1.0} #replaces them
with the pixel value of 1 in the area 2 image
  }#closes second loop
}#closes first loop

f.print.out<-rot(f.print)
f.print.out <-rot(f.print.out)
f.print.out <-rot(f.print.out)
#f.print <-rot(f.print)
f.print.out <- apply(f.print.out, 2, rev)

fnameout <- paste0(outDir, file_path_sans_ext(fname))
out.fname <-
paste0(fnameout,"/",file_path_sans_ext(fname),"_pad_three.png")
f.print.out<- channel(f.print.out, "gray")
writeImage(f.print.out, out.fname, dpi = c(1000,1000))


#removes 4s in quality map
for (i in 1:n.rows.qm) {
  for (j in 1:n.cols.qm){ #finds the values of 4 in the quality map and
    if (as.numeric(qm[i,j])== 4) {f.print[i,j]<- 1.0} #replaces them
with the pixel value of 1 in the area 3 image
  }#closes second loop
}#closes first loop

f.print.out<-rot(f.print)
f.print.out <-rot(f.print.out)
f.print.out <-rot(f.print.out)
#f.print <-rot(f.print)
f.print.out <- apply(f.print.out, 2, rev)

fnameout <- paste0(outDir, file_path_sans_ext(fname))
out.fname <-
paste0(fnameout,"/",file_path_sans_ext(fname),"_pad_four.png")
f.print.out<- channel(f.print.out, "gray")
writeImage(f.print.out, out.fname, dpi = c(1000,1000))

}
```

A.5 R® Ten-print Card Creation Code

```
library(png)
library(imager)
library(tools)

inDir<-"C:/Users/Alyshia/Documents/crossmatch tenprint images
(grayscale and padded)/"
outDir<- "C:/Users/Alyshia/Documents/remaining ten print cards"

ID_Numbers <-list.files(path=inDir) # pulling all of the images
ID_Numbers <- substr(ID_Numbers, 1,12) #narrowing the name down
ID_Numbers <- unique(ID_Numbers) #giving the unique numbers

Fingers <- c("__f01", "__f02", "__f03", "__f04", "__f05", "__f06",
             "__f07", "__f08", "__f09", "__f10")

fname <- c()

for(i in 1:length(ID_Numbers)){
  png(file = paste0(outDir, "/",ID_Numbers[i], ".png"), width=8,
height=3, units="in", res = 1000)
  par(mfrow=c(2,5), mar=c(0,0,0,0), mai=c(0,0,0,0), omi=c(0,0,0,0))
  for (j in 1:length(Fingers)){
    fname[i] <- paste0(inDir, ID_Numbers[i], Fingers[j], ".png") #
creating the fingerprint cards
    test_new <- load.image(fname[i])
    plot(test_new, axes=FALSE, frame.plot=FALSE, ylab="", xlab="")
   }
  dev.off()
  cat(ID_Numbers[i], "\n")
}
```

A.6 R® List Path Code

```
library(tools)


# ===============================
#gallery files
files<-("F:/research_gallery")

x<-list.files(path=files, full.names=TRUE, recursive = FALSE)
#pulls the list of files and gives the entire path
x<-data.matrix(x) # creates it as a data frame

y<-x[,1] #only keeps the first column

write.table(y, file="C:/NBIS/gallery_minutiae_path.txt",
quote=FALSE, sep=" ", row.names = FALSE, col.names = FALSE)
#writes out the list in a table
```

B. Technical Note Remaining Data

B.1 Confusion Matrices: Pattern Weights of 1 = 0.5, 2 = 0.65, 3 = 1, 4 = 0.65, and 5 = 1

Table B.1: Pattern Weights of 1 = 0.5, 2 = 0.65, 3 = 1, 4 = 0.65, and 5 = 1

Priors	Actual Experimental	1	2	3	4	5
0.2	1	23	6	5	1	7
0.2	2	7	18	7	3	5
0.2	3	4	8	35	8	16
0.2	4	3	5	18	19	6
0.2	5	1	0	1	2	22
	Misclassified = 51	22.17 %				
Priors	Actual Experimental	1	2	3	4	5
0.1	1	13	3	23	0	3
0.1	2	3	13	22	1	1
0.6	3	0	0	71	0	0
0.1	4	1	4	37	5	4
0.1	5	0	0	15	0	11
	Misclassified = 48	20.86 %				

Table B.2: Pattern Weights of 1 = 0.5, 2 = 0.65, 3 = 1, 4 = 0.65, and 5 = 1

Priors	Actual Experimental	1	2	3	4	5
0.25	1	26	5	5	0	6
0.25	2	8	22	5	0	5
0.2	3	6	11	35	0	19
0.1	4	3	13	24	0	11
0.2	5	1	0	4	0	21
	Misclassified = 62	26.95 %				
Priors	Actual Experimental	1	2	3	4	5
0.1	1	9	3	21	0	9
0.1	2	3	8	18	0	11
0.35	3	1	0	54	0	16
0.1	4	1	3	29	0	18
0.35	5	0	0	4	0	22
	Misclassified = 66	28.69 %				
Priors	Actual Experimental	1	2	3	4	5
0.15	1	16	5	17	0	4
0.15	2	3	11	18	2	6
0.4	3	0	1	67	1	2
0.15	4	2	2	34	7	6
0.15	5	0	0	9	0	17
	Misclassified = 44	19.13 %				
Priors	Actual Experimental	1	2	3	4	5
0.1	1	14	3	8	0	17
0.1	2	2	7	10	0	21
0.2	3	3	0	30	0	38
0.1	4	2	1	8	0	40
0.5	5	0	0	0	0	26
	Misclassified = 90	39.13 %				
Priors	Actual Experimental	1	2	3	4	5
0.3	1	31	4	2	1	4
0.25	2	9	21	1	9	0
0.1	3	12	10	15	20	14
0.25	4	4	5	4	34	4
0.1	5	1	0	3	5	17
	Misclassified = 55	23.91 %				

Table B.3: Pattern Weights of 1 = 0.5, 2 = 0.65, 3 = 1, 4 = 0.65, and 5 = 1

Priors	Actual Experimental	1	2	3	4	5
0.3	1	27	9	2	1	3
0.3	2	6	29	0	5	0
0.1	3	10	16	20	14	11
0.2	4	5	11	7	25	3
0.1	5	1	2	1	4	18
	Misclassified = 52	22.60 %				
Priors	Actual Experimental	1	2	3	4	5
0.35	1	31	0	7	0	4
0.1	2	17	4	11	1	7
0.25	3	8	0	48	0	15
0.1	4	3	2	29	2	15
0.2	5	2	0	5	0	19
	Misclassified = 54	23.47 %				
Priors	Actual Experimental	1	2	3	4	5
0.15	1	22	3	11	1	5
0.15	2	5	14	15	3	3
0.25	3	3	2	44	3	19
0.2	4	2	3	24	11	11
0.25	5	0	0	4	0	22
	Misclassified = 54	23.47 %				
Priors	Actual Experimental	1	2	3	4	5
0.15	1	19	6	9	1	7
0.15	2	3	15	13	4	5
0.25	3	2	1	48	6	14
0.25	4	1	3	20	21	6
0.2	5	0	0	3	2	21
	Misclassified = 49	21.30 %				
Priors	Actual Experimental	1	2	3	4	5
0.17	1	21	7	8	1	5
0.17	2	7	17	9	3	4
0.23	3	6	1	42	7	15
0.23	4	2	4	15	21	9
0.2	5	0	0	2	3	21
	Misclassified = 50	21.73 %				

Table B.4: Pattern Weights of 1 = 0.5, 2 = 0.65, 3 = 1, 4 = 0.65, and 5 = 1

Priors	Actual Experimental	1	2	3	4	5
0.15	1	20	4	10	2	6
0.17	2	4	16	11	4	5
0.25	3	4	2	45	8	12
0.25	4	1	2	20	21	7
0.18	5	0	0	5	3	18
	Misclassified = 51	22.17 %				
Priors	Actual Experimental	1	2	3	4	5
0.16	1	23	4	9	1	5
0.17	2	4	17	10	4	5
0.26	3	4	3	47	6	11
0.24	4	2	2	23	17	7
0.18	5	0	0	6	2	1
	Misclassified =49	21.30%				
Priors	Actual Experimental	1	2	3	4	5
0.15	1	19	6	11	1	5
0.17	2	4	15	11	5	5
0.27	3	4	2	49	6	10
0.23	4	2	0	24	20	5
0.18	5	0	0	7	2	17
	Misclassified = 50	21.73 %				
Priors	Actual Experimental	1	2	3	4	5
0.16	1	22	4	11	1	4
0.17	2	4	17	9	4	6
0.27	3	2	3	50	5	11
0.22	4	2	2	24	15	8
0.18	5	0	0	6	2	18
	Misclassified = 49	21.30 %				
Priors	Actual Experimental	1	2	3	4	5
0.16	1	21	4	11	1	5
0.16	2	2	13	15	5	5
0.27	3	2	2	49	6	12
0.23	4	2	2	22	19	6
0.18	5	0	0	6	2	18
	Misclassified = 51	22.17%				

Table B.5: Pattern Weights of 1 = 0.5, 2 = 0.65, 3 = 1, 4 = 0.65, and 5 = 1

Priors	Actual Experimental	1	2	3	4	5
0.15	1	19	10	8	1	4
0.18	2	2	17	14	4	3
0.27	3	2	2	53	5	9
0.23	4	1	1	22	22	5
0.17	5	0	0	8	2	16
	Misclassified =41	17.82 %				
Priors	Actual Experimental	1	2	3	4	5
0.15	1	23	4	9	1	5
0.19	2	4	17	13	2	4
0.27	3	2	3	53	3	10
0.22	4	1	3	25	15	7
0.17	5	0	0	7	2	17
	Misclassified = 44	19.13%				
Priors	Actual Experimental	1	2	3	4	5
0.15	1	17	4	15	1	5
0.2	2	5	19	9	3	4
0.27	3	1	2	51	6	11
0.21	4	1	4	26	15	5
0.17	5	0	0	8	2	16
	Misclassified = 53	23.04%				
Priors	Actual Experimental	1	2	3	4	5
0.15	1	21	7	9	1	4
0.19	2	6	17	8	4	5
0.26	3	4	3	50	5	9
0.23	4	1	4	23	16	7
0.17	5	0	0	6	2	18
	Misclassified = 47	20.43 %				
Priors	Actual Experimental	1	2	3	4	5
0.145	1	19	7	10	1	5
0.19	2	3	23	9	2	3
0.27	3	1	5	52	4	9
0.215	4	1	4	24	14	8
0.17	5	0	0	6	2	18
	Misclassified = 42	18.26 %				

B.2 Confusion Matrices: Pattern Weights of 1 = 0.25, 2 = 0.3, 3 = 1, 4 = 0.3, and 5 = 1

Table B.6: Pattern Weights of 1 = 0.25, 2 = 0.3, 3 = 1, 4 = 0.3, and 5 = 1

Priors	Actual Experimental	1	2	3	4	5
0.2	1	17	1	13	0	11
0.2	2	4	7	15	1	13
0.2	3	2	0	45	1	23
0.2	4	2	2	24	1	22
0.2	5	0	0	6	0	20
	Misclassified = 73	31.73%				
Priors	Actual Experimental	1	2	3	4	5
0.1	1	7	2	30	0	3
0.1	2	1	6	28	0	5
0.6	3	0	0	71	0	0
0.1	4	2	2	41	0	6
0.1	5	0	0	12	0	14
	Misclassified = 54	23.47%				
Priors	Actual Experimental	1	2	3	4	5
0.25	1	23	3	11	0	5
0.25	2	6	15	11	0	8
0.2	3	4	1	48	0	18
0.1	4	2	4	30	0	15
0.2	5	0	0	4	0	22
	Misclassified = 56	24.34%				
Priors	Actual Experimental	1	2	3	4	5
0.1	1	6	0	24	0	12
0.1	2	4	0	24	0	12
0.35	3	0	0	62	0	9
0.1	4	1	0	37	0	13
0.35	5	0	0	3	0	23
	Misclassified = 61	26.52 %				

Table B.7: Pattern Weights of 1 = 0.25, 2 = 0.3, 3 = 1, 4 = 0.3, and 5 = 1

Priors	Actual Experimental	1	2	3	4	5
0.15	1	13	2	22	0	5
0.15	2	4	5	27	1	3
0.4	3	0	0	69	0	2
0.15	4	2	2	36	1	10
0.15	5	0	0	10	0	16
	Misclassified = 47	20.43%				
Priors	Actual Experimental	1	2	3	4	5
0.1	1	8	0	15	0	19
0.1	2	2	2	17	0	19
0.2	3	0	0	48	0	23
0.1	4	0	1	19	0	31
0.5	5	0	0	0	0	26
	Misclassified = 77	33.47%				
Priors	Actual Experimental	1	2	3	4	5
0.3	1	28	4	5	1	4
0.25	2	10	18	2	5	5
0.1	3	9	4	27	12	19
0.25	4	6	4	10	20	11
0.1	5	1	0	2	3	20
	Misclassified = 61	26.52%				
Priors	Actual Experimental	1	2	3	4	5
0.3	1	22	9	5	1	3
0.3	2	7	22	5	2	4
0.1	3	8	11	31	4	17
0.2	4	4	9	15	13	10
0.1	5	1	0	4	2	19
	Misclassified = 60	26.08 %				
Priors	Actual Experimental	1	2	3	4	5
0.35	1	25	0	12	0	5
0.1	2	10	3	20	0	7
0.25	3	3	0	58	0	10
0.1	4	3	2	32	0	14
0.2	5	1	0	3	0	22
	Misclassified = 46	20.00%				

Table B.8: Pattern Weights of 1 = 0.25, 2 = 0.3, 3 = 1, 4 = 0.3, and 5 = 1

Priors	Actual Experimental	1	2	3	4	5
0.15	1	14	2	19	0	7
0.15	2	2	7	19	1	11
0.25	3	0	0	57	0	14
0.2	4	1	1	34	2	13
0.25	5	0	0	4	0	22
	Misclassified = 58	25.21 %				
Priors	Actual Experimental	1	2	3	4	5
0.15	1	16	3	16	0	7
0.15	2	3	9	20	2	6
0.25	3	0	0	60	0	11
0.25	4	1	2	34	5	9
0.2	5	0	0	4	0	22
	Misclassified = 49	21.30 %				
Priors	Actual Experimental	1	2	3	4	5
0.17	1	16	4	17	0	5
0.17	2	3	11	18	1	7
0.23	3	1	0	57	0	13
0.23	4	1	1	35	4	10
0.2	5	0	0	4	0	22
	Misclassified = 50	21.73 %				
Priors	Actual Experimental	1	2	3	4	5
0.15	1	14	4	18	0	6
0.17	2	3	8	17	3	9
0.25	3	0	0	60	1	10
0.25	4	1	0	29	11	10
0.18	5	0	0	7	1	18
	Misclassified = 54	23.47 %				
Priors	Actual Experimental	1	2	3	4	5
0.16	1	19	2	16	0	5
0.17	2	5	10	20	2	3
0.26	3	1	0	62	1	7
0.24	4	1	1	32	3	14
0.18	5	0	0	7	0	19
	Misclassified =43	18.69%				

Table B.9: Pattern Weights of 1 = 0.25, 2 = 0.3, 3 = 1, 4 = 0.3, and 5 = 1

Priors	Actual Experimental	1	2	3	4	5
0.15	1	11	6	19	0	6
0.17	2	4	10	18	1	7
0.27	3	1	0	63	0	7
0.23	4	1	1	32	7	10
0.18	5	0	0	6	0	20
	Misclassified = 49	21.30 %				
Priors	Actual Experimental	1	2	3	4	5
0.16	1	14	1	23	0	4
0.17	2	3	10	19	0	8
0.27	3	1	0	59	2	9
0.22	4	1	2	30	3	15
0.18	5	0	0	7	0	19
	Misclassified = 55	23.91 %				
Priors	Actual Experimental	1	2	3	4	5
0.16	1	17	3	16	0	6
0.16	2	5	4	20	2	9
0.27	3	1	0	62	0	8
0.23	4	1	1	31	9	9
0.18	5	0	0	5	0	21
	Misclassified = 49	21.30%				
Priors	Actual Experimental	1	2	3	4	5
0.15	1	17	3	17	0	5
0.18	2	3	9	24	0	4
0.27	3	1	0	62	1	7
0.23	4	1	3	34	0	13
0.17	5	0	0	8	0	18
	Misclassified = 46	20.00%				
Priors	Actual Experimental	1	2	3	4	5
0.15	1	15	4	18	0	5
0.19	2	2	10	22	0	6
0.27	3	1	0	63	0	7
0.22	4	1	3	34	3	10
0.17	5	0	0	5	0	21
	Misclassified = 46	20.00%				

Table B.10: Pattern Weights of 1 = 0.25, 2 = 0.3, 3 = 1, 4 = 0.3, and 5 = 1

Priors	Actual Experimental	1	2	3	4	5
0.15	1	13	5	18	0	6
0.2	2	3	10	22	1	4
0.27	3	0	0	65	0	6
0.21	4	1	3	30	5	12
0.17	5	0	0	4	0	22
	Misclassified = 43	18.69%				
Priors	Actual Experimental	1	2	3	4	5
0.15	1	12	5	17	0	8
0.19	2	3	11	18	2	6
0.26	3	1	0	61	1	8
0.23	4	1	5	30	5	10
0.17	5	0	0	7	0	19
	Misclassified = 55	23.91%				
Priors	Actual Experimental	1	2	3	4	5
0.145	1	11	5	20	0	6
0.19	2	2	12	22	0	4
0.27	3	0	0	62	0	9
0.215	4	2	3	33	3	10
0.17	5	0	0	8	0	18
	Misclassified = 52	22.60%				

B.3 Confusion Matrices: Pattern Weights of 1 = 0.25, 2 = 0.3, 3 = 0.75, 4 = 0.3 and 5 = 0.75

Table B.11: Pattern Weights of 1 = 0.25, 2 = 0.3, 3 = 0.75, 4 = 0.3, and 5 = 0.75

Priors	Actual Experimental	1	2	3	4	5
0.2	1	20	6	8	1	7
0.2	2	4	18	7	2	9
0.2	3	6	2	44	1	18
0.2	4	2	3	25	12	9
0.2	5	1	0	4	0	21
	Misclassified = 61	26.52%				
Priors	Actual Experimental	1	2	3	4	5
0.1	1	12	2	23	0	5
0.1	2	1	8	25	0	6
0.6	3	0	0	70	0	1
0.1	4	1	2	41	0	7
0.1	5	0	0	15	0	11
	Misclassified = 53	23.04%				

Table B.12: Pattern Weights of 1 = 0.25, 2 = 0.3, 3 = 1, 4 = 0.3, and 5 = 1

Priors	Actual Experimental	1	2	3	4	5
0.25	1	25	2	9	0	6
0.25	2	5	19	10	0	6
0.2	3	5	4	46	0	16
0.1	4	2	6	28	0	15
0.2	5	0	0	6	0	20
	Misclassified = 56	24.34%				
Priors	Actual Experimental	1	2	3	4	5
0.1	1	6	0	27	0	9
0.1	2	2	4	25	0	9
0.35	3	0	0	52	0	19
0.1	4	1	1	29	0	20
0.35	5	0	0	4	0	22
	Misclassified = 70	30.43%				
Priors	Actual Experimental	1	2	3	4	5
0.15	1	11	2	20	0	9
0.15	2	3	8	23	2	4
0.4	3	0	0	69	0	2
0.15	4	1	1	35	3	11
0.15	5	0	0	7	0	19
	Misclassified = 46	20.00%				
Priors	Actual Experimental	1	2	3	4	5
0.1	1	8	1	156	0	17
0.1	2	3	5	13	0	19
0.2	3	0	0	39	0	32
0.1	4	2	2	12	0	35
0.5	5	0	0	0	0	26
	Misclassified = 88	38.26%				
Priors	Actual Experimental	1	2	3	4	5
0.3	1	32	0	5	1	4
0.25	2	10	20	1	7	2
0.1	3	10	8	25	14	14
0.25	4	5	5	7	28	6
0.1	5	1	0	1	4	20
	Misclassified = 55	23.91%				

Table B.13: Pattern Weights of 1 = 0.25, 2 = 0.3, 3 = 0.75, 4 = 0.3, and 5 = 0.75

Priors	Actual Experimental	1	2	3	4	5
0.3	1	32	2	3	1	4
0.3	2	7	25	1	4	3
0.1	3	8	10	27	10	16
0.2	4	5	11	10	19	6
0.1	5	1	1	2	2	20
	Misclassified = 59	25.65%				
Priors	Actual Experimental	1	2	3	4	5
0.35	1	27	0	12	0	3
0.1	2	11	3	20	0	6
0.25	3	4	0	50	0	17
0.1	4	4	1	32	1	13
0.2	5	1	0	4	0	21
	Misclassified = 52	22.60%				
Priors	Actual Experimental	1	2	3	4	5
0.15	1	17	3	16	0	6
0.15	2	3	5	21	2	9
0.25	3	2	0	47	0	22
0.2	4	2	1	29	1	18
0.25	5	0	0	6	0	20
	Misclassified = 66	28.69%				
Priors	Actual Experimental	1	2	3	4	5
0.15	1	17	4	14	0	7
0.15	2	5	9	15	4	7
0.25	3	2	0	56	2	11
0.25	4	1	2	28	11	9
0.2	5	0	0	4	1	21
	Misclassified = 52	22.60%				
Priors	Actual Experimental	1	2	3	4	5
0.17	1	19	4	13	0	6
0.17	2	4	12	16	3	5
0.23	3	2	0	50	1	18
0.23	4	2	1	31	8	9
0.2	5	0	0	5	0	21
	Misclassified = 55	23.91%				

Table B.14: Pattern Weights of 1 = 0.25, 2 = 0.3, 3 = 0.75, 4 = 0.3, and 5 = 0.75

Priors	Actual Experimental	1	2	3	4	5
0.15	1	6	4	15	1	6
0.17	2	3	13	14	3	7
0.25	3	1	0	59	1	10
0.25	4	1	1	29	11	9
0.18	5	0	0	5	0	21
	Misclassified = 59	25.65%				
Priors	Actual Experimental	1	2	3	4	5
0.16	1	17	4	14	0	7
0.17	2	3	13	18	3	3
0.26	3	1	0	59	1	10
0.24	4	1	1	31	11	7
0.18	5	0	0	4	0	22
	Misclassified = 44	19.13%				
Priors	Actual Experimental	1	2	3	4	5
0.15	1	13	4	18	0	7
0.17	2	3	12	17	3	5
0.27	3	0	1	62	1	7
0.23	4	1	1	28	13	8
0.18	5	0	0	4	0	22
	Misclassified = 46	20.00%				
Priors	Actual Experimental	1	2	3	4	5
0.16	1	14	5	19	0	4
0.17	2	2	13	21	0	4
0.27	3	2	0	59	0	10
0.22	4	1	3	34	2	11
0.18	5	0	0	5	0	21
	Misclassified = 48	20.86%				
Priors	Actual Experimental	1	2	3	4	5
0.16	1	20	2	14	1	5
0.16	2	5	9	19	2	5
0.27	3	2	0	60	2	7
0.23	4	1	0	32	9	9
0.18	5	0	0	7	0	19
	Misclassified = 44	19.13%				

Table B.15: Pattern Weights of 1 = 0.25, 2 = 0.3, 3 = 0.75, 4 = 0.3, and 5 = 0.75

Priors	Actual Experimental	1	2	3	4	5
0.15	1	19	2	16	0	5
0.18	2	3	11	14	3	9
0.27	3	2	0	60	1	8
0.23	4	1	1	31	9	9
0.17	5	0	0	7	1	18
	Misclassified = 52	22.60%				
Priors	Actual Experimental	1	2	3	4	5
0.15	1	17	3	16	1	5
0.19	2	5	13	15	3	4
0.27	3	0	1	63	2	5
0.22	4	1	4	30	8	8
0.17	5	0	0	5	0	21
	Misclassified = 44	19.13%				
Priors	Actual Experimental	1	2	3	4	5
0.15	1	14	8	15	0	5
0.2	2	2	15	18	1	4
0.27	3	1	1	63	0	6
0.21	4	1	6	34	2	8
0.17	5	0	0	5	0	21
	Misclassified = 44	19.13%				
Priors	Actual Experimental	1	2	3	4	5
0.15	1	14	6	17	0	5
0.19	2	3	14	16	3	4
0.26	3	0	1	62	1	7
0.23	4	1	3	28	10	9
0.17	5	0	0	7	1	18
	Misclassified = 47	20.43%				
Priors	Actual Experimental	1	2	3	4	5
0.145	1	13	4	21	0	4
0.19	2	2	11	20	2	5
0.27	3	0	1	61	2	7
0.215	4	1	2	31	10	7
0.17	5	0	0	6	0	20
	Misclassified = 48	20.86%				

B.4 Confusion Matrix: Pattern Weight of 1 = 0.25, 2 = 0.45, 3 = 0.75, 4 = 0.45, and 5 = 0.75

Table B.16: Pattern Weights of 1 = 0.25, 2 = 0.45, 3 = 0.75, 4 = 0.45, and 5 = 0.75

Priors	Actual Experimental	1	2	3	4	5
0.2	1	18	6	11	1	6
0.2	2	2	25	7	2	4
0.2	3	3	2	46	6	14
0.2	4	0	4	20	18	9
0.2	5	0	0	2	2	22
	Misclassified = 47	20.43%				
Priors	Actual Experimental	1	2	3	4	5
0.1	1	4	4	32	0	2
0.1	2	2	8	28	0	2
0.6	3	0	0	71	0	0
0.1	4	1	3	39	2	6
0.1	5	0	0	18	0	8
	Misclassified = 58	23.91%				
Priors	Actual Experimental	1	2	3	4	5
0.25	1	22	9	6	0	5
0.25	2	5	23	7	0	5
0.2	3	4	11	35	0	21
0.1	4	1	14	23	0	13
0.2	5	0	1	4	0	21
	Misclassified = 61	26.53%				
Priors	Actual Experimental	1	2	3	4	5
0.1	1	5	1	28	0	8
0.1	2	2	6	21	0	11
0.35	3	0	0	53	0	18
0.1	4	1	2	28	0	20
0.35	5	0	0	4	0	22
	Misclassified = 72	31.30%				

Table B.17: Pattern Weights of 1 = 0.25, 2 = 0.3, 3 = 1, 4 = 0.3, and 5 = 1

Priors	Actual Experimental	1	2	3	4	5
0.15	1	11	5	22	0	4
0.15	2	2	13	21	1	3
0.40	3	1	0	69	0	1
0.15	4	1	4	33	7	6
0.15	5	0	0	11	0	15
	Misclassified = 48	20.86%				
Priors	Actual Experimental	1	2	3	4	5
0.1	1	10	4	14	0	14
0.1	2	1	8	15	0	16
0.2	3	3	0	33	0	35
0.1	4	0	3	19	0	29
0.5	5	0	0	1	0	25
	Misclassified = 86	37.39%				
Priors	Actual Experimental	1	2	3	4	5
0.3	1	22	11	4	1	4
0.25	2	5	24	1	8	2
0.1	3	7	13	20	20	11
0.25	4	3	5	5	34	4
0.1	5	1	0	2	6	17
	Misclassified = 48	20.86%				
Priors	Actual Experimental	1	2	3	4	5
0.3	1	24	11	3	1	3
0.3	2	3	33	0	4	0
0.1	3	5	22	21	12	11
0.2	4	4	12	6	26	3
0.1	5	1	1	1	6	17
	Misclassified = 46	20.00%				
Priors	Actual Experimental	1	2	3	4	5
0.35	1	27	0	10	0	5
0.1	2	10	6	16	1	7
0.25	3	5	0	48	1	17
0.1	4	2	2	31	1	15
0.2	5	0	0	6	0	20
	Misclassified = 55	23.91%				

Table B.18: Pattern Weights of 1 = 0.25, 2 = 0.45, 3 = 0.75, 4 = 0.45, and 5 = 0.75

Priors	Actual Experimental	1	2	3	4	5
0.15	1	16	4	15	1	6
0.15	2	4	13	15	2	6
0.25	3	2	1	48	2	18
0.2	4	1	4	26	10	10
0.25	5	0	0	5	0	21
	Misclassified = 60	26.08%				
Priors	Actual Experimental	1	2	3	4	5
0.15	1	15	6	13	1	7
0.15	2	2	14	11	7	6
0.25	3	2	2	49	6	12
0.25	4	1	2	21	20	7
0.2	5	0	0	3	2	21
	Misclassified = 54	23.47%				
Priors	Actual Experimental	1	2	3	4	5
0.17	1	17	7	13	1	4
0.17	2	2	15	11	6	6
0.23	3	1	4	43	6	17
0.23	4	2	0	19	22	8
0.2	5	0	0	7	2	17
	Misclassified = 57	24.78%				
Priors	Actual Experimental	1	2	3	4	5
0.15	1	17	7	11	1	6
0.17	2	2	13	11	9	5
0.25	3	1	3	52	8	7
0.25	4	1	1	15	26	8
0.18	5	0	1	4	2	19
	Misclassified = 47	20.43%				
Priors	Actual Experimental	1	2	3	4	5
0.16	1	17	6	13	1	5
0.17	2	2	14	17	3	4
0.26	3	3	3	49	5	11
0.24	4	1	2	26	16	6
0.18	5	0	0	7	2	17
	Misclassified = 50	21.73%				

Table B.19: Pattern Weights of 1 = 0.25, 2 = 0.45, 3 = 0.75, 4 = 0.45, and 5 = 0.75

Priors	Actual Experimental	1	2	3	4	5
0.15	1	12	6	16	2	6
0.17	2	2	17	12	4	5
0.27	3	1	2	52	5	11
0.23	4	1	2	22	18	8
0.18	5	0	0	6	2	18
	Misclassified = 54	23.47%				
Priors	Actual Experimental	1	2	3	4	5
0.16	1	14	7	16	1	4
0.17	2	2	14	18	3	3
0.27	3	2	1	56	1	11
0.22	4	1	2	29	12	7
0.18	5	0	0	8	1	17
	Misclassified = 51	22.17%				
Priors	Actual Experimental	1	2	3	4	5
0.16	1	11	9	14	1	7
0.16	2	3	16	12	4	5
0.27	3	0	1	59	3	8
0.23	4	1	1	28	14	7
0.18	5	0	0	5	2	19
	Misclassified = 46	20.00%				
Priors	Actual Experimental	1	2	3	4	5
0.15	1	14	10	12	1	5
0.18	2	2	16	15	3	4
0.27	3	0	3	53	4	11
0.23	4	1	2	27	15	6
0.17	5	0	0	7	2	17
	Misclassified = 46	20.00%				
Priors	Actual Experimental	1	2	3	4	5
0.15	1	11	12	11	1	7
0.19	2	2	19	14	4	1
0.27	3	0	3	59	5	4
0.22	4	1	2	26	15	7
0.17	5	0	0	6	3	17
	Misclassified = 37	16.08%				

Table B.20: Pattern Weights of 1 = 0.25, 2 = 0.45, 3 = 0.75, 4 = 0.45, and 5 = 0.75

Priors	Actual Experimental	1	2	3	4	5
0.15	1	12	11	10	1	8
0.2	2	1	19	13	4	3
0.27	3	2	3	56	2	8
0.21	4	1	4	27	12	7
0.17	5	0	0	7	1	18
	Misclassified = 48	20.86%				
Priors	Actual Experimental	1	2	3	4	5
0.15	1	10	13	9	1	9
0.19	2	2	17	14	2	5
0.26	3	0	3	57	4	7
0.23	4	1	3	24	17	6
0.17	5	0	0	5	1	20
	Misclassified = 42	18.26%				
Priors	Actual Experimental	1	2	3	4	5
0.145	1	13	10	12	1	6
0.19	2	2	16	14	4	4
0.27	3	0	2	55	5	9
0.215	4	1	3	26	13	8
0.17	5	0	0	7	1	18
	Misclassified = 47	20.43%				

B.5 Confusion Matrices: Pattern Weights of 1 = 0.25, 2 = 0.45, 3 = 0.5, 4 = 0.45, and 5 = 0.5

Table B.21: Pattern Weights of 1 = 0.25, 2 = 0.45, 3 = 0.5, 4 = 0.45, and 5 = 0.5

Priors	Actual Experimental	1	2	3	4	5
0.2	1	21	9	6	1	5
0.2	2	1	28	1	7	3
0.2	3	3	9	30	13	16
0.2	4	1	6	12	22	10
0.2	5	0	0	3	3	20
	Misclassified = 51	22.17%				
Priors	Actual Experimental	1	2	3	4	5
0.1	1	9	6	23	0	4
0.1	2	2	14	23	0	1
0.6	3	0	0	71	0	0
0.1	4	1	6	40	2	2
0.1	5	0	0	20	0	6
	Misclassified = 55	23.91%				
Priors	Actual Experimental	1	2	3	4	5
0.25	1	21	11	4	1	5
0.25	2	4	30	2	0	4
0.2	3	5	14	32	1	19
0.1	4	3	14	19	4	11
0.2	5	1	2	3	0	20
	Misclassified = 61	26.53%				
Priors	Actual Experimental	1	2	3	4	5
0.1	1	13	4	16	0	9
0.1	2	2	8	21	0	9
0.35	3	1	1	46	0	23
0.1	4	1	3	34	0	13
0.35	5	0	0	5	0	21
	Misclassified = 67	29.13%				

Table B.22: Pattern Weights of 1 = 0.25, 2 = 0.45, 3 = 0.5, 4 = 0.45, and 5 = 0.5

Priors	Actual Experimental	1	2	3	4	5
0.15	1	12	7	20	0	3
0.15	2	2	17	18	2	1
0.40	3	1	1	66	1	2
0.15	4	1	2	31	15	2
0.15	5	0	0	14	1	11
	Misclassified = 46	20.00%				
Priors	Actual Experimental	1	2	3	4	5
0.1	1	18	2	7	0	15
0.1	2	3	10	7	0	20
0.2	3	2	0	26	0	43
0.1	4	3	3	11	1	33
0.5	5	0	0	0	0	26
	Misclassified = 90	39.13%				
Priors	Actual Experimental	1	2	3	4	5
0.3	1	26	9	2	3	2
0.25	2	7	24	1	8	0
0.1	3	8	17	11	30	5
0.25	4	4	11	3	34	1
0.1	5	2	2	2	9	11
	Misclassified = 49	21.30%				
Priors	Actual Experimental	1	2	3	4	5
0.3	1	25	10	2	2	3
0.3	2	4	31	0	5	0
0.1	3	11	20	13	17	10
0.2	4	3	13	4	30	1
0.1	5	3	3	1	6	13
	Misclassified = 56	24.34%				
Priors	Actual Experimental	1	2	3	4	5
0.35	1	32	0	7	0	3
0.1	2	14	8	12	3	3
0.25	3	5	2	51	1	12
0.1	4	4	5	24	8	10
0.2	5	1	0	5	0	20
	Misclassified = 48	20.86%				

Table B.23: Pattern Weights of 1 = 0.25, 2 = 0.45, 3 = 0.5, 4 = 0.45, and 5 = 0.5

Priors	Actual Experimental	1	2	3	4	5
0.15	1	17	7	10	1	7
0.15	2	2	18	8	6	6
0.25	3	2	5	41	8	15
0.2	4	1	3	14	25	8
0.25	5	0	0	1	2	23
	Misclassified = 52	22.60%				
Priors	Actual Experimental	1	2	3	4	5
0.15	1	18	9	9	1	5
0.15	2	4	19	3	10	4
0.25	3	2	4	38	15	12
0.25	4	1	3	10	30	7
0.2	5	0	0	4	3	19
	Misclassified = 51	22.17%				
Priors	Actual Experimental	1	2	3	4	5
0.17	1	18	9	8	1	6
0.17	2	2	24	4	6	4
0.23	4	3	4	41	11	12
0.23	4	1	5	14	25	6
0.2	5	0	0	2	4	20
	Misclassified = 48	20.86%				
Priors	Actual Experimental	1	2	3	4	5
0.15	1	15	13	8	1	5
0.17	2	3	19	6	10	2
0.25	3	3	6	39	14	9
0.25	4	1	3	14	31	2
0.18	5	0	0	3	3	20
	Misclassified = 45	19.56%				
Priors	Actual Experimental	1	2	3	4	5
0.16	1	19	7	9	1	6
0.17	2	2	20	7	8	3
0.26	3	4	7	36	14	10
0.24	4	1	4	14	28	4
0.18	5	0	0	5	3	18
	Misclassified = 51	22.17%				

Priors	Actual Experimental	1	2	3	4	5
0.15	1	17	9	9	1	6
0.17	2	2	21	8	8	1
0.27	3	3	5	43	13	7
0.23	4	1	3	15	27	5
0.18	5	0	0	5	2	19
	Misclassified = 44	19.13%				

Priors	Actual Experimental	1	2	3	4	5
0.16	1	17	11	8	1	5
0.17	2	3	22	8	5	2
0.27	3	3	6	41	13	8
0.22	4	1	4	16	25	5
0.18	5	0	0	4	3	19
	Misclassified = 41	17.82%				

Priors	Actual Experimental	1	2	3	4	5
0.16	1	19	6	14	1	5
0.16	2	4	20	7	6	3
0.27	3	2	6	43	12	8
0.23	4	1	2	16	27	5
0.18	5	0	0	4	4	18
	Misclassified = 43	18.69%				

Priors	Actual Experimental	1	2	3	4	5
0.15	1	16	10	11	1	4
0.18	2	2	20	6	10	2
0.27	3	1	6	43	11	10
0.23	4	0	4	15	28	4
0.17	5	0	0	7	2	17
	Misclassified = 50	21.72%				

Priors	Actual Experimental	1	2	3	4	5
0.15	1	16	13	7	1	5
0.19	2	2	22	10	3	3
0.27	3	2	7	45	8	9
0.22	4	1	2	20	25	3
0.17	5	0	0	6	2	18
	Misclassified = 39	16.95%				

Priors	Actual Experimental	1	2	3	4	5
0.15	1	14	15	8	1	4
0.2	2	2	22	7	7	2
0.27	3	0	10	43	8	10
0.21	4	1	6	15	22	7
0.17	5	0	0	6	3	17
	Misclassified = 45	19.56%				
Priors	Actual Experimental	1	2	3	4	5
0.15	1	9	12	14	1	6
0.19	2	1	27	4	7	1
0.26	3	2	6	47	9	7
0.23	4	1	3	14	29	4
0.17	5	0	1	5	3	17
	Misclassified = 47	20.43%				
Priors	Actual Experimental	1	2	3	4	5
0.145	1	14	14	8	1	5
0.19	2	2	25	7	4	2
0.27	3	1	8	46	7	9
0.215	4	1	5	17	24	4
0.17	5	0	0	5	2	19
	Misclassified = 41	17.82%				

B.6 Confusion Matrices: Pattern Weights of 1 = 0.3, 2 = 0.45, 3 = 0.25, 4 = 0.45, and 5 = 0.25

Table B.24: Pattern Weights of 1 = 0.25, 2 = 0.45, 3 = 0.25, 4 = 0.45, and 5 = 0.25

Priors	Actual Experimental	1	2	3	4	5
0.2	1	26	8	1	4	3
0.2	2	5	27	0	8	0
0.2	3	6	13	18	23	11
0.2	4	3	5	4	35	4
0.2	5	2	2	2	4	16
	Misclassified = 47	20.43%				
Priors	Actual Experimental	1	2	3	4	5
0.1	1	17	5	19	0	1
0.1	2	2	16	20	2	0
0.6	3	0	1	68	2	0
0.1	4	2	2	32	13	2
0.1	5	0	0	20	2	4
	Misclassified = 46	20.00%				
Priors	Actual Experimental	1	2	3	4	5
0.25	1	28	8	2	1	3
0.25	2	4	32	0	3	1
0.2	3	10	22	18	6	15
0.1	4	4	20	8	13	6
0.2	5	2	3	4	1	16
	Misclassified = 68	29.56%				
Priors	Actual Experimental	1	2	3	4	5
0.1	1	19	7	8	1	7
0.1	2	3	18	9	4	6
0.35	3	4	4	44	3	16
0.1	4	2	3	22	14	10
0.35	5	0	0	6	0	20
	Misclassified = 57	24.78%				

Table B.25: Pattern Weights of 1 = 0.3, 2 = 0.45, 3 = 0.25, 4 = 0.45, and 5 = 0.25

Priors	Actual Experimental	1	2	3	4	5
0.15	1	20	9	11	1	1
0.15	2	6	21	6	7	0
0.40	3	3	5	51	12	0
0.15	4	1	6	17	25	2
0.15	5	0	1	13	4	8
	Misclassified = 44	19.13%				
Priors	Actual Experimental	1	2	3	4	5
0.1	1	22	7	1	1	11
0.1	2	7	18	2	4	9
0.2	3	7	8	16	7	33
0.1	4	3	7	5	19	17
0.5	5	1	0	0	2	23
	Misclassified = 77	33.47%				
Priors	Actual Experimental	1	2	3	4	5
0.3	1	34	5	0	3	0
0.25	2	6	27	0	7	0
0.1	3	9	20	1	37	4
0.25	4	5	5	0	41	0
0.1	5	5	4	0	15	2
	Misclassified = 42	18.26%				
Priors	Actual Experimental	1	2	3	4	5
0.3	1	31	8	0	2	1
0.3	2	6	29	0	5	0
0.1	3	12	24	6	27	2
0.2	4	3	13	2	33	0
0.1	5	3	4	1	12	6
	Misclassified = 46	20.00%				
Priors	Actual Experimental	1	2	3	4	5
0.35	1	37	1	1	1	2
0.1	2	18	9	5	4	4
0.25	3	16	3	32	5	15
0.1	4	8	3	13	21	6
0.2	5	6	0	4	3	13
	Misclassified = 64	27.82%				

Table B.26: Pattern Weights of 1 = 0.3, 2 = 0.45, 3 = 0.25, 4 = 0.45, and 5 = 0.25

Priors	Actual Experimental	1	2	3	4	5
0.15	1	20	11	5	1	5
0.15	2	6	22	0	8	4
0.25	3	3	12	22	20	14
0.2	4	2	4	6	34	5
0.25	5	1	0	1	5	19
	Misclassified = 48	20.86%				
Priors	Actual Experimental	1	2	3	4	5
0.15	1	23	7	4	5	3
0.15	2	5	20	1	13	1
0.25	3	4	6	19	32	10
0.25	4	1	2	4	41	3
0.2	5	1	0	3	10	12
	Misclassified = 47	20.43%				
Priors	Actual Experimental	1	2	3	4	5
0.17	1	21	11	3	4	3
0.17	2	6	24	1	9	0
0.23	3	6	10	19	24	12
0.23	4	2	5	4	38	2
0.2	5	1	0	3	8	14
	Misclassified = 48	20.86%				
Priors	Actual Experimental	1	2	3	4	5
0.15	1	22	10	4	4	2
0.17	2	5	26	0	9	0
0.25	3	4	11	22	28	6
0.25	4	3	4	5	38	1
0.18	5	1	0	4	9	12
	Misclassified = 42	18.26%				
Priors	Actual Experimental	1	2	3	4	5
0.16	1	24	9	3	3	3
0.17	2	5	23	1	10	1
0.26	3	6	11	20	27	7
0.24	4	2	3	6	38	2
0.18	5	1	0	4	9	12
	Misclassified = 43	18.69%				

Table B.27: Pattern Weights of 1 = 0.3, 2 = 0.45, 3 = 0.25, 4 = 0.45, and 5 = 0.25

Priors	Actual Experimental	1	2	3	4	5
0.15	1	19	12	4	5	2
0.17	2	4	25	2	9	0
0.27	3	5	12	18	27	9
0.23	4	3	4	6	36	2
0.18	5	0	1	3	9	13
	Misclassified = 45	19.56%				
Priors	Actual Experimental	1	2	3	4	5
0.16	1	22	10	5	2	3
0.17	2	6	24	1	9	0
0.27	3	3	13	25	21	9
0.22	4	3	5	6	35	2
0.18	5	1	0	3	7	15
	Misclassified = 43	18.69%				
Priors	Actual Experimental	1	2	3	4	5
0.16	1	22	10	4	2	4
0.16	2	7	21	2	10	0
0.27	3	5	10	21	26	9
0.23	4	3	4	6	36	2
0.18	5	1	0	3	9	13
	Misclassified = 45	19.56%				
Priors	Actual Experimental	1	2	3	4	5
0.15	1	21	10	4	5	2
0.18	2	5	26	0	9	0
0.27	3	3	11	25	26	6
0.23	4	2	3	8	37	1
0.17	5	1	0	3	9	13
	Misclassified = 38	16.52%				
Priors	Actual Experimental	1	2	3	4	5
0.15	1	20	12	4	3	3
0.19	2	3	27	0	9	1
0.27	3	3	16	25	22	5
0.22	4	3	6	5	34	3
0.17	5	1	0	5	9	11
	Misclassified = 43	18.69%				

Table B.28: Pattern Weights of 1 = 0.3, 2 = 0.45, 3 = 0.25, 4 = 0.45, and 5 = 0.25

Priors	Actual Experimental	1	2	3	4	5
0.15	1	21	11	5	2	3
0.2	2	5	26	1	8	0
0.27	3	3	11	30	23	4
0.21	4	2	5	10	33	1
0.17	5	2	2	5	7	10
	Misclassified = 41	17.82%				
Priors	Actual Experimental	1	2	3	4	5
0.15	1	18	14	4	4	2
0.19	2	3	29	1	7	0
0.26	3	2	13	24	28	4
0.23	4	1	6	5	39	0
0.17	5	1	1	3	9	12
	Misclassified = 35	15.21%				
Priors	Actual Experimental	1	2	3	4	5
0.145	1	18	16	3	3	2
0.19	2	2	28	1	9	0
0.27	3	4	15	23	23	6
0.215	4	4	5	5	35	2
0.17	5	1	1	6	8	10
	Misclassified = 44	19.13%				

B.7 Confusion Matrices: Pattern Weights of 1 = 0.3, 2 = 0.45, 3 = 0.1, 4 = 0.45, and 5 = 0.1

Table B.29: Pattern Weights of 1 = 0.3, 2 = 0.45, 3 = 0.1, 4 = 0.45, and 5 = 0.1

Priors	Actual Experimental	1	2	3	4	5
0.2	1	30	6	0	6	0
0.2	2	6	28	0	6	0
0.2	3	10	21	0	37	3
0.2	4	4	6	0	41	0
0.2	5	3	0	1	18	4
	Misclassified = 39	16.95%				
Priors	Actual Experimental	1	2	3	4	5
0.1	1	23	7	11	1	0
0.1	2	7	24	4	5	0
0.6	3	5	9	47	10	0
0.1	4	1	4	15	31	0
0.1	5	2	0	14	9	1
	Misclassified = 43	18.69%				
Priors	Actual Experimental	1	2	3	4	5
0.25	1	33	6	0	2	1
0.25	2	3	36	0	1	0
0.2	3	22	23	6	16	4
0.1	4	6	20	2	23	0
0.2	5	8	4	1	6	7
	Misclassified = 65	28.26%				
Priors	Actual Experimental	1	2	3	4	5
0.1	1	23	10	4	1	4
0.1	2	6	25	0	8	1
0.35	3	5	14	22	18	12
0.1	4	1	9	4	30	5
0.35	5	1	1	3	2	19
	Misclassified = 52	22.60%				

Table B.30: Pattern Weights of 1 = 0.3, 2 = 0.45, 3 = 0.1, 4 = 0.45, and 5 = 0.1

Priors	Actual Experimental	1	2	3	4	5
0.15	1	28	7	2	5	0
0.15	2	5	27	1	7	0
0.40	3	12	11	20	28	0
0.15	4	5	7	4	35	0
0.15	5	4	0	8	11	3
	Misclassified = 50	21.73%				
Priors	Actual Experimental	1	2	3	4	5
0.1	1	25	9	0	2	6
0.1	2	6	23	0	6	5
0.2	3	6	15	6	18	26
0.1	4	3	8	1	27	12
0.5	5	0	2	0	3	21
	Misclassified = 64	27.82%				
Priors	Actual Experimental	1	2	3	4	5
0.3	1	29	8	0	5	0
0.25	2	7	27	0	6	0
0.1	3	8	21	0	42	0
0.25	4	4	5	0	42	0
0.1	5	6	2	0	18	0
	Misclassified = 36	15.65%				
Priors	Actual Experimental	1	2	3	4	5
0.3	1	29	9	0	4	0
0.3	2	5	31	0	4	0
0.1	3	14	24	0	33	0
0.2	4	5	10	0	36	0
0.1	5	4	6	0	16	0
	Misclassified = 47	20.43%				
Priors	Actual Experimental	1	2	3	4	5
0.35	1	38	2	1	1	0
0.1	2	13	22	0	5	0
0.25	3	21	15	11	22	2
0.1	4	10	4	2	35	0
0.2	5	9	1	0	10	6
	Misclassified = 54	23.47%				

Table B.31: Pattern Weights of 1 = 0.3, 2 = 0.45, 3 = 0.1, 4 = 0.45, and 5 = 0.1

Priors	Actual Experimental	1	2	3	4	5
0.15	1	22	9	1	8	2
0.15	2	5	25	1	8	1
0.25	3	7	14	4	40	6
0.2	4	3	4	1	43	0
0.25	5	2	0	0	17	7
	Misclassified = 42	18.26%				
Priors	Actual Experimental	1	2	3	4	5
0.15	1	22	10	0	10	0
0.15	2	7	25	0	8	0
0.25	3	4	17	6	40	4
0.25	4	1	4	1	45	0
0.2	5	3	1	1	15	6
	Misclassified = 40	17.39%				
Priors	Actual Experimental	1	2	3	4	5
0.17	1	31	4	0	6	1
0.17	2	5	29	0	6	0
0.23	3	8	15	0	47	1
0.23	4	5	2	0	44	0
0.2	5	6	2	1	14	3
	Misclassified = 38	16.52%				
Priors	Actual Experimental	1	2	3	4	5
0.15	1	25	7	0	10	0
0.17	2	5	27	0	8	0
0.25	3	8	16	1	45	1
0.25	4	0	4	0	47	0
0.18	5	5	1	1	17	2
	Misclassified = 38	16.52%				
Priors	Actual Experimental	1	2	3	4	5
0.16	1	24	7	0	11	0
0.17	2	5	28	0	7	0
0.26	3	6	17	0	48	0
0.24	4	1	3	0	47	0
0.18	5	4	1	1	17	3
	Misclassified = 34	14.78%				

Table B.32: Pattern Weights of 1 = 0.3, 2 = 0.45, 3 = 0.1, 4 = 0.45, and 5 = 0.1

Priors	Actual Experimental	1	2	3	4	5
0.15	1	23	7	0	12	0
0.17	2	7	25	0	8	0
0.27	3	4	14	2	50	1
0.23	4	1	3	0	47	0
0.18	5	1	4	2	18	1
	Misclassified = 35	15.21%				
Priors	Actual Experimental	1	2	3	4	5
0.16	1	27	7	1	6	1
0.17	2	4	30	0	6	0
0.27	3	6	18	3	44	0
0.22	4	4	4	0	43	0
0.18	5	5	2	1	15	3
	Misclassified = 36	15.65%				
Priors	Actual Experimental	1	2	3	4	5
0.16	1	30	5	0	6	1
0.16	2	6	27	0	7	0
0.27	3	7	16	1	46	1
0.23	4	3	3	0	45	0
0.18	5	3	0	1	17	5
	Misclassified = 32	13.91%				
Priors	Actual Experimental	1	2	3	4	5
0.15	1	24	9	0	8	1
0.18	2	5	29	0	6	0
0.27	3	7	18	1	44	1
0.23	4	1	3	0	47	0
0.17	5	5	1	1	16	3
	Misclassified = 34	14.78%				
Priors	Actual Experimental	1	2	3	4	5
0.15	1	24	8	2	8	0
0.19	2	4	30	0	6	0
0.27	3	4	20	6	41	0
0.22	4	1	5	0	45	0
0.17	5	5	2	4	15	0
	Misclassified = 37	16.08%				

Table B.33: Pattern Weights of 1 = 0.3, 2 = 0.45, 3 = 0.1, 4 = 0.45, and 5 = 0.1

Priors	Actual Experimental	1	2	3	4	5
0.15	1	19	13	1	9	0
0.2	2	4	29	1	6	0
0.27	3	4	21	8	38	0
0.21	4	0	6	1	44	0
0.17	5	3	3	2	18	0
	Misclassified = 34	14.78%				
Priors	Actual Experimental	1	2	3	4	5
0.15	1	21	10	0	11	0
0.19	2	2	30	0	8	0
0.26	3	6	14	2	48	1
0.23	4	1	5	1	44	0
0.17	5	3	3	1	18	1
	Misclassified = 39	16.95%				
Priors	Actual Experimental	1	2	3	4	5
0.145	1	22	7	0	13	0
0.19	2	7	26	0	7	0
0.27	3	4	16	1	49	1
0.215	4	1	4	0	46	0
0.17	5	2	0	0	17	1
	Misclassified = 38	16.52%				

C. Research Codes

C.1 R® Code for Random Selection of Fingerprint Images

```
library(tools)
library(base)
library(utils)

inDir <- "F:/Latents" #where the latents prints are located
outDir <- "F:/Latents for research" #where the latents will be written out to

setwd("F:/Latents") #need to set the working directory for the code to pull the files, if the
working directory is not set it will not pull the list of files
latent_list<- list.files(pattern= ".bmp", recursive = TRUE)#find and lists fingerprints with
the pattern bmp for all the folders in the directory
latent_list_basename<-basename(latent_list) #removes the folder name
latent_list_basename_ext <- file_path_sans_ext(latent_list_basename)#removes the bmp extension

x<-data.frame(latent_list_basename_ext)#forces the list into a data frame
type_latent = substr(x$latent_list_basename_ext, 9,9 )#finds and extracts the character string
within each file and reports it
x = cbind(x,type_latent)#combines the data frame of the list and the corresponding character
string

latent_fingers= subset(x,type_latent=="F")#finds only the fingerprints that have an F within
the name


set.seed(42)#a random number generator state for random number generation
random_latents = data.frame(sample(latent_fingers$latent_list_basename_ext, 500))#random
sampling of 500 latent fingerprints from the list of just fingerprints or f in the name
colnames(random_latents) <- "Rlatents" #see the column name for the random_latents
random_latents_extent = paste0(random_latents$Rlatents, ".bmp")#adds the bmp extension


latentDir<- substr(random_latents$Rlatents, 1,7) #gives the first seven characters from the
list of random latents
outDir <- "F:/Latents for research" #output directory
inDir <- "F:/Latents/" #input directory

##k<-1

for(i in 1:length(random_latents_extent)){
  inFile<- paste0(inDir,as.character(latentDir[i]), "/", random_latents_extent[i])
  outFile <- paste0(outDir, random_latents_extent[i])

  file.copy(from=inFile, to=outDir, overwrite=TRUE, copy.mode = TRUE, copy.date = TRUE )
#copies the files needed

} #for loop to locate the fingerprints and move them the to output directory

write.table(random_latents, file="F:/Latents for research/random sampling with r.txt", sep="")
```

C.2 AFIX Tracker® Data Extraction R® Code

```
library(RODBC)

# channel ==========================================

channel <- odbcConnectAccess("F:/DATA/Tracker")

# Get data Match Results=======
Match_Result <- sqlQuery(channel, paste ("select [Match ID], [Job Number], [Control ID],
                                   [Match Score], [Status], [Fingers Matched], [Percent 
Matched]
                                         from [Match Result]"))
cols <- c("Match_ID", "Job_Number", "Control_ID", "Match_Score", "Status", "Fingers_Matched", 
"Percent_Matched")
colnames(Match_Result) <- cols


Match_Result <-as.data.frame(apply(Match_Result,2,function(x)gsub('\\s+', '',x)))

Finger_Position <- log10(as.numeric(as.character(Match_Result$Fingers_Matched)))/log10(2)

Match_Result <- cbind(Match_Result, Finger_Position)
# Get data Search Jobs ========
Search_Jobs <- sqlQuery(channel, paste ("select [Job Number], Verification, [Control ID],
                                   Description, [Search Type]
                                   from [Search Jobs]"))
cols <- c("Job_Number", "Verification", "Control_ID", "Description", "Search_Type")
colnames(Search_Jobs) <- cols


Search_Jobs <-as.data.frame(apply(Search_Jobs,2,function(x)gsub('\\s+', '',x)))

# Get data Latents=============
Latents <- sqlQuery(channel, paste ("select [Control ID], [Latent Case ID], [Latent Case 
Number],
                              Officer,  Verification, Comments
                             from Latents"))

cols <- c("Control_ID", "Latent_Case_ID", "Latent_Case_Number", "Officer", 
           "Verification", "Comments")
colnames(Latents) <- cols

Latents <-as.data.frame(apply(Latents,2,function(x)gsub('\\s+', '',x)))

write.csv(Latents, "F:/DATA/latents.csv")

# Get data Latent Cases========
Latent_Cases <- sqlQuery(channel, paste ("select [Control ID], [Case Number], [City]
                                   from [Latent Cases]"))
cols <- c("Control_ID", "Case_Number", "Cases_Number")
colnames(Latent_Cases) <- cols

Latent_Cases <-as.data.frame(apply(Latent_Cases,2,function(x)gsub('\\s+', '',x)))

write.csv(Latent_Cases, "F:/DATA/latent cases.csv")

#city gives the repeated name and finger number but different fingerprint
# Get data Fingers=============
Fingers <- sqlQuery(channel, paste("select [Control ID], [Print Type], [Print Number],
                              Classification, [Minutiae Count], [Finger ID]
                              from Fingers"))
cols <- c("Control_ID", "Print Type", "Print_Number", "Classification",  "Minutiae_Count", 
          "Finger_ID")
colnames(Fingers) <- cols

Fingers <-as.data.frame(apply(Fingers,2,function(x)gsub('\\s+', '',x)))

Fingers_latents<-which(Fingers[,2] == "LL1")

fingers_latents_subset<-Fingers[Fingers_latents,]

write.csv(fingers_latents_subset, "F:/DATA/fingers.csv")
```

```
# Get data Biographical=============
Biographical <- sqlQuery(channel, paste("select [Control ID], [Last Name], [Classification 1],
[Tenprint Status 1]
                          from Biographical"))
cols <- c("Control_ID", "Last_Name",  "Classification_1", "Tenprint_Status_1")

colnames(Biographical) <- cols

Biographical <-as.data.frame(apply(Biographical,2,function(x)gsub('\\s+', '',x)))
#Create Results File ===========================
Output <- c()
# Pull verified data from Match Result Table========

verified.Match_Result<-(Match_Result)

# Pull data from Search Jobs Table========
SJ.Verification <- c()
SJ.Search_Type <-c()
SJ.Control_ID <-c()
SJ.Description <-c()
SJ.Job_Number <-c()

k <- 1
for (i in 1:length(verified.Match_Result$Match_ID)){
  for (j in 1:length(Search_Jobs$Job_Number)){
    if (as.character(verified.Match_Result$Job_Number[i]) ==
as.character(Search_Jobs$Job_Number[j])){
      SJ.Job_Number[k] <- as.character(Search_Jobs$Job_Number[j])
      SJ.Verification[k] <- as.character(Search_Jobs$Verification[j])
      SJ.Search_Type[k] <- as.character(Search_Jobs$Search_Type[j])
      SJ.Control_ID[k] <- as.character(Search_Jobs$Control_ID[j])
      SJ.Description[k] <- as.character(Search_Jobs$Description[j])
      k<-k+1
    }
  }
}

Output<- cbind(verified.Match_Result,SJ.Job_Number, SJ.Verification, SJ.Search_Type,
SJ.Control_ID, SJ.Description)
Output <- subset(Output, SJ.Search_Type=="Latent")
match_results_and_search<-cbind(verified.Match_Result,SJ.Job_Number, SJ.Verification,
SJ.Search_Type, SJ.Control_ID, SJ.Description)



# Pull data from Latents Table=======
L.Control_ID <- c()
L.Latent_Case_ID <- c()
L.Latent_Case_Number <- c()
L.Officer <- c()
L.Verification <- c()
L.Comments <- c()

k <- 1
for (i in 1:length(Output$Match_ID)){
  for (j in 1:length(Latents$Control_ID)){
    if (as.character(Output$SJ.Control_ID[i]) == as.character(Latents$Control_ID[j])){
      L.Control_ID[k] <- as.character(Latents$Control_ID[j])
      L.Latent_Case_ID[k] <- as.character(Latents$Latent_Case_ID[j])
      L.Latent_Case_Number[k] <- as.character(Latents$Latent_Case_Number[j])
      L.Officer[k] <- as.character(Latents$Officer[j])
      L.Verification[k] <- as.character(Latents$Verification[j])
      L.Comments[k] <- as.character(Latents$Comments[j])
      k<-k+1
    }
  }
}

latent_pulled_data<-cbind(L.Control_ID, L.Latent_Case_ID, L.Latent_Case_Number,
                          L.Officer,  L.Verification,
                          L.Comments)

Output <- cbind(Output, L.Control_ID, L.Latent_Case_ID, L.Latent_Case_Number,
                 L.Officer,  L.Verification,
                L.Comments)
```

```
# Pull data from Latent Cases Table=======

LC.Control_ID <- c()
LC.Case_Number <- c()
LC.Cases_Number<-c() #repeated case number but different fingerprint

k <- 1
for (i in 1:length(Output$L.Latent_Case_ID)){
  for (j in 1:length(Latent_Cases$Case_Number)){
    if (as.character(Output$L.Latent_Case_ID[i]) == as.character(Latent_Cases$Control_ID[j])){
      LC.Control_ID[k] <- as.character(Latent_Cases$Control_ID[j])
      LC.Case_Number[k] <- as.character(Latent_Cases$Case_Number[j])
      LC.Cases_Number[k] <- as.character(Latent_Cases$Cases_Number[j])
      k<-k+1
    }
    }
}

Output<- cbind(Output, LC.Control_ID, LC.Case_Number, LC.Cases_Number)



# Pull data from Biographical Table=======

B.Control_ID <- c()
B.Last_Name <- c()
B.Classification_1 <- c()
B.Tenprint_Status_1 <- c()

k <- 1
for (i in 1:length(Output$Control_ID)){
  for (j in 1:length(Biographical$Control_ID)){
    if (as.character(Output$Control_ID[i]) == as.character(Biographical$Control_ID[j])){
      B.Control_ID[k] <- as.character(Biographical$Control_ID[j])
      B.Last_Name[k] <- as.character(Biographical$Last_Name[j])
      B.Classification_1[k] <- as.character(Biographical$Classification_1[j])
      B.Tenprint_Status_1[k] <- as.character(Biographical$Tenprint_Status_1[j])
      k<-k+1
    }
  }
}

Output <- cbind(Output, B.Control_ID, B.Last_Name,B.Classification_1,
                B.Tenprint_Status_1)



# Pull data from Fingers Table=======

bioFingers <- subset(Fingers)

F.Control_ID <- c()
F.Print_Number <- c()
F.Classification <- c()
F.Minutiae_Count <- c()
F.Finger_ID <- c()


Output.Control_ID.Finger_Position <- paste0(as.character(Output$B.Control_ID), "_",
as.character(Output$Finger_Position))
bioFingers.Control_ID.Print_Number <- paste0(as.character(bioFingers$Control_ID), "_",
as.character(bioFingers$Print_Number))

k <- 1
for (i in 1:length(Output.Control_ID.Finger_Position)){
  for (j in 1:length(bioFingers.Control_ID.Print_Number)){
    if (as.character(Output.Control_ID.Finger_Position[i]) ==
as.character(bioFingers.Control_ID.Print_Number[j])){
      F.Control_ID[k] <- as.character(bioFingers$Control_ID[j])
      F.Print_Number[k] <- as.character(bioFingers$Print_Number[j])
      F.Classification[k] <- as.character(bioFingers$Classification[j])
      F.Minutiae_Count[k] <- as.character(bioFingers$Minutiae_Count[j])
      F.Finger_ID[k] <- as.character(bioFingers$Finger_ID[j])
      k<-k+1
    }
  }
}
Output2 <- cbind( F.Control_ID, F.Print_Number, F.Classification, F.Minutiae_Count,
F.Finger_ID)

write.csv(Output,"F:/DATA/May 27-2017.csv", row.names = FALSE)



# http://rprogramming.net/connect-to-database-in-r/
## Query the database and put the results into the data frame "dataframe"
#dataframe <- sqlQuery(channel, "
#                       SELECT SCHOOL,
#                       STUDENT_NAME
#                       FROM
#                       SCHEMA.DATATABLE
#                       WHERE
#                       SCHOOL_YEAR='2011-12'")

# addding rank to the afix output========
library(BioPhysConnectoR)#needed to sort the afix output based on the job number

afix_output<-read.csv("E:/AFIX output/May 27-2017.csv") #the output from afix
afix_output<-as.matrix(afix_output)


trim.leading <- function (x)
  sub("^\\s+", "", x) #removes the blank spaces in the names

afix_output<-mat.sort(afix_output, 2) #sorting the matrix based on the second column of job
number

job = afix_output[,2] #specifying the the variable job to be the job number column
job2 = trim.leading(job)#removing the blank spaces in the job column
job_numbers_unique<-unique(job2) #finding the unique identities in the job number

match<-afix_output[,5] #specifying the variable match to be the match score column


#i<-2
rank<-vector("numeric") #creating a blank vector to place the ranks into

for(i in 1:length(job_numbers_unique)){ #finding the ranks for each job number
  row_numbers<-which((job2 ==job_numbers_unique[i])) #finds the rows in the job number that
are the same as the unique

  total_search_return<-length(row_numbers) #the number of row returned in the finding of the
same rows
  row_value<-rbind(1:total_search_return) #creates a row based on the length of the number of
rows that are the same

  rank<-append(rank, row_value) #appends the row numbers from row value to those created in
the previous job numbers
}

rank<-as.matrix(rank) #the final output of ranks from all of the job numbers and places them
into a matrix form
#it takes the rows and places them into a column

cols<-c("Rank")

rank<-`colnames<-`(rank, cols)

updated_output<-cbind(afix_output, rank) #binds the rank column to the end of the afix input
data

need_to_split<-updated_output[,19]

need_to_split<-strsplit(need_to_split, "-")
split<-data.frame(need_to_split)
split<-t(split)
cols2<-c("Area", "Extraction")

split<-`colnames<-`(split, cols2)

updated_output<-cbind(updated_output[,(1:18)], split, updated_output[,(20:27)])
```

```
finger<-as.numeric(updated_output[,8])
finger<-finger+1

updated_output<-cbind(updated_output[,(1:7)], finger, updated_output[,(9:28)])

#k<-1
for(k in 1:length(updated_output[,23])){
  if(is.na(updated_output[k,23])){updated_output[k,23]=1}
}

changed<-gsub("1219147_F01_BP_CP1", "1219247_F01_BP_CP1", updated_output[,13])
changed<-as.matrix(changed)
changed<-gsub("1473933_F06_CA_CD2", "1473993_F06_CA_CD2", changed)
changed<- as.matrix(changed)
changed<-gsub("1581969_F01_BP_CP2", "1581696_F01_BP_CP2", changed)
changed<- as.matrix(changed)
changed<-gsub("1737457_F01_CA_BG1", "1737547_F01_CA_BG1", changed)
changed<- as.matrix(changed)
changed<-gsub("199145_F03_NI_PA1", "1991415_F03_NI_PA1", changed)
changed<- as.matrix(changed)
changed<-gsub("1030930_F06_NI_PA1", "1030930_F08_NI_PA1", changed)
changed<- as.matrix(changed)
changed<-gsub("1768313_F01_BP_CP2", "1768313_F02_BP_CP2", changed)
changed<-as.matrix(changed)

updated_output<-cbind(updated_output[,(1:12)], changed, updated_output[,(14:28)])

changed<-gsub("1219147_F01_BP_CP1", "1219247_F01_BP_CP1", updated_output[,16])
changed<-as.matrix(changed)
changed<-gsub("1473933_F06_CA_CD2", "1473993_F06_CA_CD2", changed)
changed<- as.matrix(changed)
changed<-gsub("1581969_F01_BP_CP2", "1581696_F01_BP_CP2", changed)
changed<- as.matrix(changed)
changed<-gsub("1737457_F01_CA_BG1", "1737547_F01_CA_BG1", changed)
changed<- as.matrix(changed)
changed<-gsub("199145_F03_NI_PA1", "1991415_F03_NI_PA1", changed)
changed<- as.matrix(changed)
changed<-gsub("1030930_F06_NI_PA1", "1030930_F08_NI_PA1", changed)
changed<- as.matrix(changed)
changed<-gsub("1768313_F01_BP_CP2", "1768313_F02_BP_CP2", changed)
changed<-as.matrix(changed)


updated_output<-cbind(updated_output[,(1:15)], changed, updated_output[,(17:28)])

changed<-gsub("1219147_F01_BP_CP1", "1219247_F01_BP_CP1", updated_output[,22])
changed<-as.matrix(changed)
changed<-gsub("1473933_F06_CA_CD2", "1473993_F06_CA_CD2", changed)
changed<- as.matrix(changed)
changed<-gsub("1581969_F01_BP_CP2", "1581696_F01_BP_CP2", changed)
changed<- as.matrix(changed)
changed<-gsub("1737457_F01_CA_BG1", "1737547_F01_CA_BG1", changed)
changed<- as.matrix(changed)
changed<-gsub("199145_F03_NI_PA1", "1991415_F03_NI_PA1", changed)
changed<- as.matrix(changed)
changed<-gsub("1030930_F06_NI_PA1", "1030930_F08_NI_PA1", changed)
changed<- as.matrix(changed)
changed<-gsub("1768313_F01_BP_CP2", "1768313_F02_BP_CP2", changed)
changed<-as.matrix(changed)

updated_output<-cbind(updated_output[,(1:21)], changed, updated_output[,(23:28)])

write.csv(updated_output, "E:/AFIX output/updated afix output with rank.csv", row.names =
FALSE) #writes out the update afix data with the ranks
#without the row numbers


# pulling the matches from afix output part 1==========
afix_output<-read.csv("E:/AFIX output/updated afix output with rank.csv")
afix_output<-as.matrix(afix_output)


trim.leading <- function (x)
  sub("^\\s+", "", x) #removes the blank spaces in the names at the begining
```

```
sample_name<-afix_output[,13]
sample_name<-substr(sample_name, 1,7)
sample_name_unique<-unique(sample_name)

finger_location<-as.numeric(afix_output[,8])

method<-afix_output[,17]

area<-afix_output[,19]
area<-trim.leading(area)

extraction<-afix_output[,20]
extraction<-trim.leading(extraction)

job = afix_output[,2]
job_numbers_unique<-unique(job)

job = trim.leading(job)

job_numbers_unique = trim.leading(job_numbers_unique)

search_job<-afix_output[,2]
latent_case_number<-afix_output[,15]

combo_job_latent<-cbind(search_job, latent_case_number)
combo_job<-!duplicated(combo_job_latent)
combo_job<-combo_job_latent[combo_job,]


blank_matrix<-matrix( , ,ncol=15)

# pulling the matches from afix output part 2==================================================
#i<-228

 for(i in 1:length(job_numbers_unique)){
  job_numbers2<-which((job==job_numbers_unique[i]))
  narrow_jobs<-afix_output[job_numbers2,]

  top_rank<-narrow_jobs[1,28] #top rank for the job number  number
  top_match_score<-narrow_jobs[1,4] #top match score for the job number

  name<-narrow_jobs[,13]
  name<-substr(name, 1,7)
  name_narrow<-name[1]
  #
  method_narrow<-narrow_jobs[1,17]
  area_narrow<-narrow_jobs[1,19]
  extraction_narrow<-narrow_jobs[1,20]
  repeated<-narrow_jobs[1,23]

  finger_in_name<-narrow_jobs[1,13]
  finger_in_name<-substr(finger_in_name, 10, 11)
  if(finger_in_name != "10"){finger_in_name<-substr(finger_in_name, 2, 2)}
  if(finger_in_name == "10"){finger_in_name<-substr(finger_in_name, 1, 2)}
  finger_in_name<-as.character(finger_in_name)


  status<-which((narrow_jobs[,5] == "I"))
  status_identified<-as.matrix(narrow_jobs[status,])
  if(nrow((status_identified))== 28 & ncol(status_identified) == 1){status_identified<-
t(status_identified)}


  if(length(status_identified) == "0"){
    output=cbind(name_narrow, finger_in_name[1], repeated,  job_numbers_unique[i],
method_narrow, area_narrow,
                 extraction_narrow, top_rank, top_match_score,0,0,0,0,0,0)

    cols<-c("Name","Finger","Repeated Sample", "Job Number","Method","Area", "Extraction",
"Top Rank", "Top Match Score", "Match 1-Rank",
            "Match 1-Score", "Match 2-Rank", "Match 2-Score","Match 3-Rank", "Match 3-Score" )

    pulled_data<-`colnames<-`(output, cols)
  }
  if(length(status_identified) != "0"  & nrow(status_identified) == 1){
    fingers<-which(status_identified[1,8]==status_identified[,8])
    fingers_identified<-as.matrix(status_identified[fingers,])
    if(nrow(fingers_identified) == 28 & ncol(fingers_identified) == 1){fingers_identified<-
t(fingers_identified)}


    more_than_one_case_same_finger<-which(fingers_identified[1,23] == fingers_identified[,23])
    more_than_one_case_same_finger_identified<-
fingers_identified[more_than_one_case_same_finger,]
    more_than_one_case_same_finger_identified<-
as.matrix(more_than_one_case_same_finger_identified)
    if(nrow(more_than_one_case_same_finger_identified) == 28 &
ncol(more_than_one_case_same_finger_identified) == 1){
      more_than_one_case_same_finger_identified<-t(more_than_one_case_same_finger_identified)}

    output<-cbind(name_narrow, finger_in_name[1],
more_than_one_case_same_finger_identified[1,23],
                  job_numbers_unique[i], more_than_one_case_same_finger_identified[1,17],
#method
                  more_than_one_case_same_finger_identified[1,19],
more_than_one_case_same_finger_identified[1,20],
                  top_rank, top_match_score,
as.numeric(more_than_one_case_same_finger_identified[1,28]), #match 1 rank
                  more_than_one_case_same_finger_identified[1,4],0, 0 #match score 1
                  ,0,0) ##name, finger, job number, method, area, extration, top rank, top
match score, match 1 rank
                        ##match 1 score, match 2 rank, match 2 score, match 3 rank, match 3
score
    cols<-c("Name","Finger","Repeated Sample", "Job Number","Method","Area", "Extraction",
"Top Rank", "Top Match Score", "Match 1-Rank",
            "Match 1-Score", "Match 2-Rank", "Match 2-Score","Match 3-Rank", "Match 3-Score" )
    pulled_data<-`colnames<-`(output, cols)
  }

  if(length(status_identified) != "0"  & nrow(status_identified) == 2){
    fingers<-which(status_identified[1,8]==status_identified[,8])
    fingers_identified<-status_identified[fingers,]

    more_than_one_case_same_finger<-which(fingers_identified[1,23] == fingers_identified[,23])
    more_than_one_case_same_finger_identified<-
fingers_identified[more_than_one_case_same_finger,]

    output<-cbind(name_narrow,
finger_in_name[1],more_than_one_case_same_finger_identified[1,23],
                  job_numbers_unique[i], more_than_one_case_same_finger_identified[1,17],
#method
                  more_than_one_case_same_finger_identified[1,19],
more_than_one_case_same_finger_identified[1,20],#area and extraction
                  top_rank, top_match_score, more_than_one_case_same_finger_identified[1,28],
#top rank, top match, match 1 rank

more_than_one_case_same_finger_identified[1,4],more_than_one_case_same_finger_identified[2,28],

                  more_than_one_case_same_finger_identified[2,4] #match score 1
                  ,0,0) ##name, finger, job number, method, area, extration, top rank, top
match score, match 1 rank
    ##match 1 score, match 2 rank, match 2 score, match 3 rank, match 3 score
    cols<-c("Name","Finger","Repeated Sample", "Job Number","Method","Area", "Extraction",
"Top Rank", "Top Match Score", "Match 1-Rank",
            "Match 1-Score", "Match 2-Rank", "Match 2-Score","Match 3-Rank", "Match 3-Score" )

    pulled_data<-`colnames<-`(output, cols)
  }

  if(length(status_identified) != "0" & nrow(status_identified) == 3){
    fingers<-which(status_identified[1,8]==status_identified[,8])
    fingers_identified<-status_identified[fingers,]

    more_than_one_case_same_finger<-which(fingers_identified[1,23] == fingers_identified[,23])
    more_than_one_case_same_finger_identified<-
fingers_identified[more_than_one_case_same_finger,]

    output<-cbind(name_narrow,
finger_in_name[1],more_than_one_case_same_finger_identified[1,23],
                  job_numbers_unique[i], more_than_one_case_same_finger_identified[1,17],
#method
                  more_than_one_case_same_finger_identified[1,19],
more_than_one_case_same_finger_identified[1,20],#area and extraction
                  top_rank, top_match_score, more_than_one_case_same_finger_identified[1,28],
#top rank, top match, match 1 rank

more_than_one_case_same_finger_identified[1,4],more_than_one_case_same_finger_identified[2,28],

                  more_than_one_case_same_finger_identified[2,4],
more_than_one_case_same_finger_identified[3,28],
                  more_than_one_case_same_finger_identified[3,4] ) ##name, finger, job number,
method, area, extration, top rank, top match score, match 1 rank
    ##match 1 score, match 2 rank, match 2 score, match 3 rank, match 3 score
    cols<-c("Name","Finger","Repeated Sample", "Job Number","Method","Area", "Extraction",
"Top Rank", "Top Match Score", "Match 1-Rank",
            "Match 1-Score", "Match 2-Rank", "Match 2-Score","Match 3-Rank", "Match 3-Score" )
    pulled_data<-`colnames<-`(output, cols)
  }

  blank_matrix<-rbind(blank_matrix, pulled_data)
  }



blank_row<-nrow(blank_matrix)

blank_matrix<-blank_matrix[2:blank_row,]


#z<-1
t<-matrix(,,ncol = 16)

for (z in 1:length(blank_matrix[,1])) {
  search_jobs<-which(as.numeric(combo_job[,1]) == blank_matrix[z,4])
  search_jobs<-as.numeric(search_jobs)
  binding_latents<-append(blank_matrix[search_jobs,], combo_job[z,1])

  binding_latents<-as.matrix(binding_latents)
  if(nrow(binding_latents)== 16 & ncol(binding_latents) == 1){binding_latents<-
t(binding_latents)}


  t<-rbind(t, binding_latents)
}

t_nrow<-nrow(t)

write.csv(t[(2:t_nrow),], "E:/AFIX output/research results from afix.csv", row.names = FALSE)

# combination of auto and manual extraction  =======================================
updated_research_results<-read.csv("E:/AFIX output/research results from afix.csv")

trim.leading <- function (x)
  sub("^\\s+", "", x) #removes the blank spaces in the names

name_match<-updated_research_results[,1]
finger_match<-as.character(updated_research_results[,2])
repeats_match<-updated_research_results[,3]
method_match<-updated_research_results[,5]
area_match<-updated_research_results[,6]

#i<-233
blank_matrix2<-matrix( , ,ncol=23)
for(i in 1:length(name_match)){
  name_match_search<-which((name_match[i] == name_match))
  pulled_name_match<-as.matrix(updated_research_results[name_match_search,])
  if(nrow(pulled_name_match) == 16 & ncol(pulled_name_match) ==1 ){
    pulled_name_match<-t(pulled_name_match)}

  method_match_search<-which(method_match[i]==pulled_name_match[,5])
  pulled_method_search<-as.matrix(pulled_name_match[method_match_search,])
  if(nrow(pulled_method_search) == 16 & ncol(pulled_method_search) ==1){
    pulled_method_search<-t(pulled_method_search)}

  area_match_search<-which((area_match[i] == pulled_method_search[,6]))
  pulled_area_search<-as.matrix(pulled_method_search[area_match_search,])
  if(nrow(pulled_area_search) == 16 & ncol(pulled_area_search) ==1){
    pulled_area_search<-t(pulled_area_search)}

  t<-trim.leading(pulled_area_search[,2])
  finger_match_search<-which((finger_match[i] == t))
  pulled_finger_search<-as.matrix(pulled_area_search[finger_match_search,])
  if(nrow(pulled_finger_search) == 16 & ncol(pulled_finger_search) ==1){
    pulled_finger_search<-t(pulled_finger_search)}

  repeats_match_search<-which(repeats_match[i] == pulled_finger_search[,3])
  pulled_repeats_search<-as.matrix(pulled_finger_search[repeats_match_search,])
  if(nrow(pulled_repeats_search) == 16 & ncol(pulled_repeats_search) ==1){
    pulled_repeats_search<-t(pulled_repeats_search)}



  #if there is 1 row and is auto
  if(nrow(pulled_repeats_search) == 1 & pulled_repeats_search[1,7] == "AUTO"){ #if there is
not two rows, in the case of 1 row
    extraction_combo<-cbind(name_match[i], pulled_repeats_search[1,2],
pulled_repeats_search[1,3],
                            as.character(method_match[i]), as.character(area_match[i]),
pulled_repeats_search[1,16],
                            pulled_repeats_search[1,8], pulled_repeats_search[1,9],
pulled_repeats_search[1,10],
                            pulled_repeats_search[1,11],
pulled_repeats_search[1,12],pulled_repeats_search[1,13],
                            pulled_repeats_search[1,14], pulled_repeats_search[1,15], 0, 0, 0,
0, 0, 0, 0, 0, 0 )

    cols<-c("Name", "Finger", "Repeated Sample", "Method","Area","Auto Search Number","Auto-
Top Rank", "Auto-Top Match Score", "Auto-Match 1-Rank",
            "Auto-Match 1-Score", "Auto-Match 2-Rank", "Auto-Match 2-Score","Auto-Match 3-
Rank", "Auto-Match 3-Score",
            "Manual Search Number","Manual-Top Rank", "Manual-Top Match Score", "Manual-Match
1-Rank",
            "Manual-Match 1-Score", "Manual-Match 2-Rank", "Manual-Match 2-Score","Manual-
Match 3-Rank", "Manual-Match 3-Score")

    extraction_combo_names<-`colnames<-`(extraction_combo, cols)
  }

  ####if there is 1 row and is manual
  if (nrow(pulled_repeats_search) == 1 & pulled_repeats_search[1,7] == "MANUAL"){
    extraction_combo<-cbind(name_match[i],pulled_repeats_search[1,2],
pulled_repeats_search[1,3],
                            as.character(method_match[i]), as.character(area_match[i]),
                            0, 0, 0, 0,0, 0, 0, 0, 0, pulled_repeats_search[1,16],
                            pulled_repeats_search[1,8], pulled_repeats_search[1,9],
pulled_repeats_search[1,10],
                            pulled_repeats_search[1,11],
pulled_repeats_search[1,12],pulled_repeats_search[1,13],
                            pulled_repeats_search[1,14], pulled_repeats_search[1,15])

    cols<-c("Name", "Finger", "Repeated Sample", "Method","Area","Auto Search Number","Auto-
Top Rank", "Auto-Top Match Score", "Auto-Match 1-Rank",
            "Auto-Match 1-Score", "Auto-Match 2-Rank", "Auto-Match 2-Score","Auto-Match 3-
Rank", "Auto-Match 3-Score",
            "Manual Search Number","Manual-Top Rank", "Manual-Top Match Score", "Manual-Match
1-Rank",
            "Manual-Match 1-Score", "Manual-Match 2-Rank", "Manual-Match 2-Score","Manual-
Match 3-Rank", "Manual-Match 3-Score")

    extraction_combo_names<-`colnames<-`(extraction_combo, cols)
  }

  #if there is 2 rows and the first is auto and the second is manual
  if(nrow(pulled_repeats_search) == 2){
    if(pulled_repeats_search[1,7] == "AUTO" & pulled_repeats_search[2,7]== "MANUAL"){
      extraction_combo<-cbind(name_match[i],pulled_repeats_search[1,2],
pulled_repeats_search[1,3],
                              as.character(method_match[i]), as.character(area_match[i]),
pulled_repeats_search[1,16],
                              pulled_repeats_search[1,8], pulled_repeats_search[1,9],
pulled_repeats_search[1,10],
                              pulled_repeats_search[1,11],
pulled_repeats_search[1,12],pulled_repeats_search[1,13],
                              pulled_repeats_search[1,14], pulled_repeats_search[1,15],
pulled_repeats_search[2,16],
                              pulled_repeats_search[2,8], pulled_repeats_search[2,9],
pulled_repeats_search[2,10],
                              pulled_repeats_search[2,11],
pulled_repeats_search[2,12],pulled_repeats_search[2,13],
                              pulled_repeats_search[2,14], pulled_repeats_search[2,15])

      cols<-c("Name", "Finger", "Repeated Sample", "Method","Area","Auto Search Number","Auto-
Top Rank", "Auto-Top Match Score", "Auto-Match 1-Rank",
              "Auto-Match 1-Score", "Auto-Match 2-Rank", "Auto-Match 2-Score","Auto-Match 3-
Rank", "Auto-Match 3-Score",
              "Manual Search Number","Manual-Top Rank", "Manual-Top Match Score", "Manual-
Match 1-Rank",
              "Manual-Match 1-Score", "Manual-Match 2-Rank", "Manual-Match 2-Score","Manual-
Match 3-Rank", "Manual-Match 3-Score")

      extraction_combo_names<-`colnames<-`(extraction_combo, cols)
    }
  }
  ##if there is 2 rows and the first is manual and the second is auto
  if(nrow(pulled_repeats_search) == 2){
    if (pulled_repeats_search[1,7] == "MANUAL" & pulled_repeats_search[2,7] == "AUTO"){
      extraction_combo<-cbind(name_match[i],pulled_repeats_search[1,2],
pulled_repeats_search[1,3],
                              as.character(method_match[i]), as.character(area_match[i]),
pulled_repeats_search[2,16],
                              pulled_repeats_search[2,8], pulled_repeats_search[2,9],
pulled_repeats_search[2,10],
                              pulled_repeats_search[2,11],
pulled_repeats_search[2,12],pulled_repeats_search[2,13],
                              pulled_repeats_search[2,14], pulled_repeats_search[2,15],
                              pulled_repeats_search[1,16], pulled_repeats_search[1,8],
                              pulled_repeats_search[1,9], pulled_repeats_search[1,10],
                              pulled_repeats_search[1,11],
pulled_repeats_search[1,12],pulled_repeats_search[1,13],
                              pulled_repeats_search[1,14], pulled_repeats_search[1,15])

      cols<-c("Name", "Finger", "Repeated Sample", "Method","Area","Auto Search Number","Auto-
Top Rank", "Auto-Top Match Score", "Auto-Match 1-Rank",
              "Auto-Match 1-Score", "Auto-Match 2-Rank", "Auto-Match 2-Score","Auto-Match 3-
Rank", "Auto-Match 3-Score",
              "Manual Search Number","Manual-Top Rank", "Manual-Top Match Score", "Manual-
Match 1-Rank",
              "Manual-Match 1-Score", "Manual-Match 2-Rank", "Manual-Match 2-Score","Manual-
Match 3-Rank", "Manual-Match 3-Score")

      extraction_combo_names<-`colnames<-`(extraction_combo, cols)
    }
  }

  #if there is two rows and are both manual
  if(nrow(pulled_repeats_search) == 2){
    if (pulled_repeats_search[1,7] == "MANUAL" & pulled_repeats_search[2,7] == "MANUAL"){
      extraction_combo<-cbind(name_match[i],pulled_repeats_search[1,2],
pulled_repeats_search[1,3],
                              as.character(method_match[i]), as.character(area_match[i]),
                              0, 0, 0, 0,0, 0, 0, 0, 0,
                              pulled_repeats_search[1,16], pulled_repeats_search[1,8],
                              pulled_repeats_search[1,9], pulled_repeats_search[1,10],
                              pulled_repeats_search[1,11],
pulled_repeats_search[1,12],pulled_repeats_search[1,13],
                              pulled_repeats_search[1,14], pulled_repeats_search[1,15])

      cols<-c("Name", "Finger", "Repeated Sample", "Method","Area","Auto Search Number","Auto-
Top Rank", "Auto-Top Match Score", "Auto-Match 1-Rank",
              "Auto-Match 1-Score", "Auto-Match 2-Rank", "Auto-Match 2-Score","Auto-Match 3-
Rank", "Auto-Match 3-Score",
              "Manual Search Number","Manual-Top Rank", "Manual-Top Match Score", "Manual-
Match 1-Rank",
              "Manual-Match 1-Score", "Manual-Match 2-Rank", "Manual-Match 2-Score","Manual-
Match 3-Rank", "Manual-Match 3-Score")

      extraction_combo_names<-`colnames<-`(extraction_combo, cols)
    }
  }

  #if there is two rows and are both auto
  if(nrow(pulled_repeats_search) == 2){
    if (pulled_repeats_search[1,7] == "AUTO" & pulled_repeats_search[2,7] == "AUTO"){
      extraction_combo<-cbind(name_match[i],pulled_repeats_search[1,2], 
pulled_repeats_search[1,3],
                              as.character(method_match[i]), as.character(area_match[i]), 
pulled_repeats_search[1,16],
                              pulled_repeats_search[1,8],
                              pulled_repeats_search[1,9], pulled_repeats_search[1,10],
                              pulled_repeats_search[1,11], 
pulled_repeats_search[1,12],pulled_repeats_search[1,13],
                              pulled_repeats_search[1,14], pulled_repeats_search[1,15],
                              0, 0, 0, 0,0, 0, 0, 0, 0)

      cols<-c("Name", "Finger", "Repeated Sample", "Method","Area","Auto Search Number","Auto-
Top Rank", "Auto-Top Match Score", "Auto-Match 1-Rank",
              "Auto-Match 1-Score", "Auto-Match 2-Rank", "Auto-Match 2-Score","Auto-Match 3-
Rank", "Auto-Match 3-Score",
              "Manual Search Number","Manual-Top Rank", "Manual-Top Match Score", "Manual-
Match 1-Rank",
              "Manual-Match 1-Score", "Manual-Match 2-Rank", "Manual-Match 2-Score","Manual-
Match 3-Rank", "Manual-Match 3-Score")

      extraction_combo_names<-`colnames<-`(extraction_combo, cols)

    }
  }

  if(nrow(pulled_repeats_search)== 3){#1 auto and 2 manual
    if(pulled_repeats_search[1,7] == "AUTO" & pulled_repeats_search[2,7]== "MANUAL" & 
pulled_repeats_search[3,7] == "MANUAL"){
      extraction_combo<-cbind(name_match[i],pulled_repeats_search[1,2], 
pulled_repeats_search[1,3],
                              as.character(method_match[i]), as.character(area_match[i]), 
pulled_repeats_search[1,16],
                              pulled_repeats_search[1,8], pulled_repeats_search[1,9], 
pulled_repeats_search[1,10],
                              pulled_repeats_search[1,11], 
pulled_repeats_search[1,12],pulled_repeats_search[1,13],
                              pulled_repeats_search[1,14], pulled_repeats_search[1,15], 
pulled_repeats_search[3,16],
                              pulled_repeats_search[3,8], pulled_repeats_search[3,9], 
pulled_repeats_search[3,10],
                              pulled_repeats_search[3,11], 
pulled_repeats_search[3,12],pulled_repeats_search[3,13],
                              pulled_repeats_search[3,14], pulled_repeats_search[3,15])

      cols<-c("Name", "Finger", "Repeated Sample", "Method","Area","Auto Search Number","Auto-
Top Rank", "Auto-Top Match Score", "Auto-Match 1-Rank",
              "Auto-Match 1-Score", "Auto-Match 2-Rank", "Auto-Match 2-Score","Auto-Match 3-
Rank", "Auto-Match 3-Score",
              "Manual Search Number","Manual-Top Rank", "Manual-Top Match Score", "Manual-
Match 1-Rank",
              "Manual-Match 1-Score", "Manual-Match 2-Rank", "Manual-Match 2-Score","Manual-
Match 3-Rank", "Manual-Match 3-Score")

      extraction_combo_names<-`colnames<-`(extraction_combo, cols)
    }
  }

  if(nrow(pulled_repeats_search)== 3){#2 auto and 1 manual
    if(pulled_repeats_search[1,7] == "AUTO" & pulled_repeats_search[2,7]== "AUTO" & 
pulled_repeats_search[3,7] == "MANUAL"){
      extraction_combo<-cbind(name_match[i],pulled_repeats_search[1,2], 
pulled_repeats_search[1,3],
                              as.character(method_match[i]), as.character(area_match[i]), 
pulled_repeats_search[1,16],
                              pulled_repeats_search[1,8], pulled_repeats_search[1,9], 
pulled_repeats_search[1,10],
                              pulled_repeats_search[1,11], 
pulled_repeats_search[1,12],pulled_repeats_search[1,13],
                              pulled_repeats_search[1,14], pulled_repeats_search[1,15], 
pulled_repeats_search[3,16],
                              pulled_repeats_search[3,8], pulled_repeats_search[3,9],
pulled_repeats_search[3,10],
                              pulled_repeats_search[3,11],
pulled_repeats_search[3,12],pulled_repeats_search[3,13],
                              pulled_repeats_search[3,14], pulled_repeats_search[3,15])

      cols<-c("Name", "Finger", "Repeated Sample", "Method","Area","Auto Search Number","Auto-
Top Rank", "Auto-Top Match Score", "Auto-Match 1-Rank",
              "Auto-Match 1-Score", "Auto-Match 2-Rank", "Auto-Match 2-Score","Auto-Match 3-
Rank", "Auto-Match 3-Score",
              "Manual Search Number","Manual-Top Rank", "Manual-Top Match Score", "Manual-
Match 1-Rank",
              "Manual-Match 1-Score", "Manual-Match 2-Rank", "Manual-Match 2-Score","Manual-
Match 3-Rank", "Manual-Match 3-Score")

      extraction_combo_names<-`colnames<-`(extraction_combo, cols)
    }
  }

  if(nrow(pulled_repeats_search)== 3){#2 auto and 1 manual
    if(pulled_repeats_search[1,7] == "MANUAL" & pulled_repeats_search[2,7]== "AUTO" &
pulled_repeats_search[3,7] == "MANUAL"){
      extraction_combo<-cbind(name_match[i],pulled_repeats_search[1,2],
pulled_repeats_search[1,3],
                              as.character(method_match[i]), as.character(area_match[i]),
pulled_repeats_search[2,16],
                              pulled_repeats_search[2,8], pulled_repeats_search[2,9],
pulled_repeats_search[1,10],
                              pulled_repeats_search[2,11],
pulled_repeats_search[2,12],pulled_repeats_search[2,13],
                              pulled_repeats_search[2,14], pulled_repeats_search[2,15],
pulled_repeats_search[3,16],
                              pulled_repeats_search[3,8], pulled_repeats_search[3,9],
pulled_repeats_search[3,10],
                              pulled_repeats_search[3,11],
pulled_repeats_search[3,12],pulled_repeats_search[3,13],
                              pulled_repeats_search[3,14], pulled_repeats_search[3,15])

      cols<-c("Name", "Finger", "Repeated Sample", "Method","Area","Auto Search Number","Auto-
Top Rank", "Auto-Top Match Score", "Auto-Match 1-Rank",
              "Auto-Match 1-Score", "Auto-Match 2-Rank", "Auto-Match 2-Score","Auto-Match 3-
Rank", "Auto-Match 3-Score",
              "Manual Search Number","Manual-Top Rank", "Manual-Top Match Score", "Manual-
Match 1-Rank",
              "Manual-Match 1-Score", "Manual-Match 2-Rank", "Manual-Match 2-Score","Manual-
Match 3-Rank", "Manual-Match 3-Score")

      extraction_combo_names<-`colnames<-`(extraction_combo, cols)
    }
  }



  #if there is 3 rows and the first is auto and the second is manual
  if(nrow(pulled_repeats_search)== 4){#3 auto and 2 manual
    if(pulled_repeats_search[1,7] == "AUTO" & pulled_repeats_search[2,7]== "AUTO" &
pulled_repeats_search[3,7] == "AUTO" &
       pulled_repeats_search[4,7] == "MANUAL"){
      extraction_combo<-cbind(name_match[i],pulled_repeats_search[1,2],
pulled_repeats_search[1,3],
                              as.character(method_match[i]), as.character(area_match[i]),
pulled_repeats_search[1,16],
                              pulled_repeats_search[1,8], pulled_repeats_search[1,9],
pulled_repeats_search[1,10],
                              pulled_repeats_search[1,11],
pulled_repeats_search[1,12],pulled_repeats_search[1,13],
                              pulled_repeats_search[1,14], pulled_repeats_search[1,15],
pulled_repeats_search[4,16],
                              pulled_repeats_search[4,8], pulled_repeats_search[4,9],
pulled_repeats_search[4,10],
                              pulled_repeats_search[4,11],
pulled_repeats_search[4,12],pulled_repeats_search[4,13],
                              pulled_repeats_search[4,14], pulled_repeats_search[4,15])

      cols<-c("Name", "Finger", "Repeated Sample", "Method","Area","Auto Search Number","Auto-
Top Rank", "Auto-Top Match Score", "Auto-Match 1-Rank",
              "Auto-Match 1-Score", "Auto-Match 2-Rank", "Auto-Match 2-Score","Auto-Match 3-
```

```
Rank", "Auto-Match 3-Score",
              "Manual Search Number","Manual-Top Rank", "Manual-Top Match Score", "Manual-
Match 1-Rank",
              "Manual-Match 1-Score", "Manual-Match 2-Rank", "Manual-Match 2-Score","Manual-
Match 3-Rank", "Manual-Match 3-Score")

      extraction_combo_names<-`colnames<-`(extraction_combo, cols)
    }
  }


  if(nrow(pulled_repeats_search) == 4){#2 auto and 2 manual
    if(pulled_repeats_search[1,7] == "AUTO" & pulled_repeats_search[2,7]== "AUTO" &
pulled_repeats_search[3,7] == "MANUAL" &
       pulled_repeats_search[4,7] == "MANUAL"){
      extraction_combo<-cbind(name_match[i],pulled_repeats_search[1,2],
pulled_repeats_search[1,3],
                              as.character(method_match[i]), as.character(area_match[i]),
pulled_repeats_search[1,16],
                              pulled_repeats_search[1,8], pulled_repeats_search[1,9],
pulled_repeats_search[1,10],
                              pulled_repeats_search[1,11],
pulled_repeats_search[1,12],pulled_repeats_search[1,13],
                              pulled_repeats_search[1,14], pulled_repeats_search[1,15],
pulled_repeats_search[4,16],
                              pulled_repeats_search[4,8], pulled_repeats_search[4,9],
pulled_repeats_search[4,10],
                              pulled_repeats_search[4,11],
pulled_repeats_search[4,12],pulled_repeats_search[4,13],
                              pulled_repeats_search[4,14], pulled_repeats_search[4,15])

      cols<-c("Name", "Finger", "Repeated Sample", "Method","Area","Auto Search Number","Auto-
Top Rank", "Auto-Top Match Score", "Auto-Match 1-Rank",
              "Auto-Match 1-Score", "Auto-Match 2-Rank", "Auto-Match 2-Score","Auto-Match 3-
Rank", "Auto-Match 3-Score",
              "Manual Search Number","Manual-Top Rank", "Manual-Top Match Score", "Manual-
Match 1-Rank",
              "Manual-Match 1-Score", "Manual-Match 2-Rank", "Manual-Match 2-Score","Manual-
Match 3-Rank", "Manual-Match 3-Score")

      extraction_combo_names<-`colnames<-`(extraction_combo, cols)
    }
  }

  if(nrow(pulled_repeats_search) == 4){#1 auto, 3 manual
    if(pulled_repeats_search[1,7] == "AUTO" & pulled_repeats_search[2,7]== "MANUAL" &
pulled_repeats_search[3,7] == "MANUAL" &
       pulled_repeats_search[4,7] == "MANUAL"){
      extraction_combo<-cbind(name_match[i],pulled_repeats_search[1,2],
pulled_repeats_search[1,3],
                              as.character(method_match[i]), as.character(area_match[i]),
pulled_repeats_search[1,16],
                              pulled_repeats_search[1,8], pulled_repeats_search[1,9],
pulled_repeats_search[1,10],
                              pulled_repeats_search[1,11],
pulled_repeats_search[1,12],pulled_repeats_search[1,13],
                              pulled_repeats_search[1,14], pulled_repeats_search[1,15],
pulled_repeats_search[4,16],
                              pulled_repeats_search[4,8], pulled_repeats_search[4,9],
pulled_repeats_search[4,10],
                              pulled_repeats_search[4,11],
pulled_repeats_search[4,12],pulled_repeats_search[4,13],
                              pulled_repeats_search[4,14], pulled_repeats_search[4,15])

      cols<-c("Name", "Finger", "Repeated Sample", "Method","Area","Auto Search Number","Auto-
Top Rank", "Auto-Top Match Score", "Auto-Match 1-Rank",
              "Auto-Match 1-Score", "Auto-Match 2-Rank", "Auto-Match 2-Score","Auto-Match 3-
Rank", "Auto-Match 3-Score",
              "Manual Search Number","Manual-Top Rank", "Manual-Top Match Score", "Manual-
Match 1-Rank",
              "Manual-Match 1-Score", "Manual-Match 2-Rank", "Manual-Match 2-Score","Manual-
Match 3-Rank", "Manual-Match 3-Score")

      extraction_combo_names<-`colnames<-`(extraction_combo, cols)
    }
  }
  if(nrow(pulled_repeats_search) == 4){#1 auto, 1 manual, 1 auto , 1 manual
    if(pulled_repeats_search[1,7] == "AUTO" & pulled_repeats_search[2,7]== "MANUAL" &
pulled_repeats_search[3,7] == "AUTO" &
       pulled_repeats_search[4,7] == "MANUAL"){
      extraction_combo<-cbind(name_match[i],pulled_repeats_search[1,2],
pulled_repeats_search[1,3],
                              as.character(method_match[i]), as.character(area_match[i]),
pulled_repeats_search[1,16],
                              pulled_repeats_search[1,8], pulled_repeats_search[1,9],
pulled_repeats_search[1,10],
                              pulled_repeats_search[1,11],
pulled_repeats_search[1,12],pulled_repeats_search[1,13],
                              pulled_repeats_search[1,14], pulled_repeats_search[1,15],
pulled_repeats_search[4,16],
                              pulled_repeats_search[4,8], pulled_repeats_search[4,9],
pulled_repeats_search[4,10],
                              pulled_repeats_search[4,11],
pulled_repeats_search[4,12],pulled_repeats_search[4,13],
                              pulled_repeats_search[4,14], pulled_repeats_search[4,15])

      cols<-c("Name", "Finger", "Repeated Sample", "Method","Area","Auto Search Number","Auto-
Top Rank", "Auto-Top Match Score", "Auto-Match 1-Rank",
              "Auto-Match 1-Score", "Auto-Match 2-Rank", "Auto-Match 2-Score","Auto-Match 3-
Rank", "Auto-Match 3-Score",
              "Manual Search Number","Manual-Top Rank", "Manual-Top Match Score", "Manual-
Match 1-Rank",
              "Manual-Match 1-Score", "Manual-Match 2-Rank", "Manual-Match 2-Score","Manual-
Match 3-Rank", "Manual-Match 3-Score")

      extraction_combo_names<-`colnames<-`(extraction_combo, cols)
    }
  }

  if(nrow(pulled_repeats_search) == 5){#1 auto, 1 manual, 1 auto , 1 manual
    if(pulled_repeats_search[1,7] == "AUTO" & pulled_repeats_search[2,7]== "MANUAL" &
pulled_repeats_search[3,7] == "MANUAL" &
       pulled_repeats_search[4,7] == "MANUAL" & pulled_repeats_search[5,7] == "MANUAL"){
      extraction_combo<-cbind(name_match[i],pulled_repeats_search[1,2],
pulled_repeats_search[1,3],
                              as.character(method_match[i]), as.character(area_match[i]),
pulled_repeats_search[1,16],
                              pulled_repeats_search[1,8], pulled_repeats_search[1,9],
pulled_repeats_search[1,10],
                              pulled_repeats_search[1,11],
pulled_repeats_search[1,12],pulled_repeats_search[1,13],
                              pulled_repeats_search[1,14], pulled_repeats_search[1,15],
pulled_repeats_search[4,16],
                              pulled_repeats_search[4,8], pulled_repeats_search[4,9],
pulled_repeats_search[4,10],
                              pulled_repeats_search[4,11],
pulled_repeats_search[4,12],pulled_repeats_search[4,13],
                              pulled_repeats_search[4,14], pulled_repeats_search[4,15])

      cols<-c("Name", "Finger", "Repeated Sample", "Method","Area","Auto Search Number","Auto-
Top Rank", "Auto-Top Match Score", "Auto-Match 1-Rank",
              "Auto-Match 1-Score", "Auto-Match 2-Rank", "Auto-Match 2-Score","Auto-Match 3-
Rank", "Auto-Match 3-Score",
              "Manual Search Number","Manual-Top Rank", "Manual-Top Match Score", "Manual-
Match 1-Rank",
              "Manual-Match 1-Score", "Manual-Match 2-Rank", "Manual-Match 2-Score","Manual-
Match 3-Rank", "Manual-Match 3-Score")

      extraction_combo_names<-`colnames<-`(extraction_combo, cols)
    }
  }


  blank_matrix2<-rbind(blank_matrix2, extraction_combo_names)
}


blank_row<-nrow(blank_matrix2)

write.csv(blank_matrix2[(2:blank_row),], "E:/AFIX output/research results from afix-manual and
auto combined.csv", row.names = FALSE)
```

```
# pulling the information from the manual and auto extract combo ======
combination_extraction<-read.csv("E:/AFIX output/research results from afix-manual and auto
combined.csv")

research_data<-read.csv("E:/AFIX output/research results-without afix.csv")

trim.leading <- function (x)
  sub("^\\s+", "", x) #removes the blank spaces in the names

combo_name<-combination_extraction[,1]
combo_name<-trim.leading(combo_name)

combo_finger<-combination_extraction[,2]

combo_repeated<-combination_extraction[,3]

combo_method<-combination_extraction[,4]
combo_method<-trim.leading(combo_method)

combo_area<-combination_extraction[,5]
combo_area<-trim.leading(combo_area)

rr_name<-research_data[,2]
rr_name<-trim.leading(rr_name)

rr_finger<-research_data[,3]

rr_repeat<-research_data[,4]

rr_method<-research_data[,7]
rr_method<-trim.leading(rr_method)

rr_area<-research_data[,11]
rr_area<-trim.leading(rr_area)

#i<-10

for(i in 1:length(combo_name)){
  same_name<-which((combo_name[i] == rr_name), arr.ind = TRUE)
  same_name_data<-as.matrix(research_data[same_name,])
  if(nrow(same_name_data) == 38 & ncol(same_name_data) ==1){
    same_name_data<-t(same_name_data)}

  same_method<-grep(combo_method[i], same_name_data[,7])
  same_method_data<-as.matrix(same_name_data[same_method,])
  if(nrow(same_method_data) == 38 & ncol(same_method_data) ==1){
    same_method_data<-t(same_method_data)}
  x<-row.names(same_method_data)
  same_method_data<-cbind(x, same_method_data)

  same_repeat<-which(combo_repeated[i] == same_method_data[,5])
  same_repeat_data<-as.matrix(same_method_data[same_repeat,])
  if(nrow(same_repeat_data) == 38 & ncol(same_repeat_data) ==1){
    same_repeat_data<-t(same_repeat_data)}


  same_finger<-which(combo_finger[i] == trim.leading(same_repeat_data[,4]))
  same_finger_data<-as.matrix(same_repeat_data[same_finger,])
  if(nrow(same_finger_data) == 38 & ncol(same_finger_data) ==1){
    same_finger_data<-t(same_finger_data)}

  same_area<-which(combo_area[i]== same_finger_data[,12])
  same_area_data<-as.matrix(same_finger_data[same_area,])
  if(nrow(same_area_data) == 38 & ncol(same_area_data) ==1){
    same_area_data<-t(same_area_data)}

  row_number<-same_area_data[1]

    research_data[row_number, 13] = combination_extraction[i,6]#need to add minutiae count
    research_data[row_number, 14] = combination_extraction[i,7]
    research_data[row_number, 15] =combination_extraction[i,8]
    research_data[row_number, 16] =combination_extraction[i,9]
    research_data[row_number, 17] =combination_extraction[i,10]
    research_data[row_number, 18] = combination_extraction[i,11]
    research_data[row_number, 19] = combination_extraction[i,12]
    research_data[row_number, 20] = combination_extraction[i,13]
    research_data[row_number, 21] = combination_extraction[i,14]

    research_data[row_number, 23] = combination_extraction[i,15]#need to add minutiae count
    research_data[row_number, 24] = combination_extraction[i,16]
    research_data[row_number, 25] = combination_extraction[i,17]
    research_data[row_number, 26] = combination_extraction[i,18]
    research_data[row_number, 27] = combination_extraction[i,19]
    research_data[row_number, 28] = combination_extraction[i,20]
    research_data[row_number, 29] = combination_extraction[i,21]
    research_data[row_number, 30] = combination_extraction[i,22]
    research_data[row_number, 31] = combination_extraction[i,23]
}

for(t in 1:length(research_data[,13])){
  if(is.na(research_data[t,13])){research_data[t,13] = as.numeric("0")}
  if(research_data[t,13] == "0"){research_data[t,13] = as.numeric("0")}
}

for(k in 1:length(research_data[,23])){
  if(is.na(research_data[k,23])){research_data[k,23] = as.numeric("0")}
  if(research_data[k,23] == "0"){research_data[k,23] = as.numeric("0")}
}

write.csv(research_data, "E:/AFIX output/research-data-combined with afix.csv")

# combination of latents and minutiae count==============================
latents<-read.csv("E:/AFIX output/latents.csv")

split<-as.character(latents[,7])
need_to_split<-strsplit(split, "-")
split<-data.frame(need_to_split)
split<-t(split)
cols2<-c("Area", "Extraction")

split<-`colnames<-`(split, cols2)
latents<-cbind(latents[,(1:6)], split)

changed<-gsub("1219147_F01_BP_CP1", "1219247_F01_BP_CP1", latents[,4])
changed<-as.matrix(changed)
changed<-gsub("1473933_F06_CA_CD2", "1473993_F06_CA_CD2", changed)
changed<- as.matrix(changed)
changed<-gsub("1581969_F01_BP_CP2", "1581696_F01_BP_CP2", changed)
changed<- as.matrix(changed)
changed<-gsub("1737457_F01_CA_BG1", "1737547_F01_CA_BG1", changed)
changed<- as.matrix(changed)
changed<-gsub("199145_F03_NI_PA1", "1991415_F03_NI_PA1", changed)
changed<- as.matrix(changed)
changed<-gsub("1030930_F06_NI_PA1", "1030930_F08_NI_PA1", changed)
changed<- as.matrix(changed)
changed<-gsub("1768313_F01_BP_CP2", "1768313_F02_BP_CP2", changed)
changed<-as.matrix(changed)

latents<-cbind(latents[,(1:3)], changed, latents[,(5:8)])


Latent_Cases<-read.csv("F:/AFIX output/latent cases.csv")
changed<-gsub("1219147_F01_BP_CP1", "1219247_F01_BP_CP1", Latent_Cases[,3])
changed<-as.matrix(changed)
changed<-gsub("1473933_F06_CA_CD2", "1473993_F06_CA_CD2", changed)
changed<- as.matrix(changed)
changed<-gsub("1581969_F01_BP_CP2", "1581696_F01_BP_CP2", changed)
changed<- as.matrix(changed)
changed<-gsub("1737457_F01_CA_BG1", "1737547_F01_CA_BG1", changed)
changed<- as.matrix(changed)
changed<-gsub("199145_F03_NI_PA1", "1991415_F03_NI_PA1", changed)
changed<- as.matrix(changed)
changed<-gsub("1030930_F06_NI_PA1", "1030930_F08_NI_PA1", changed)
changed<- as.matrix(changed)
changed<-gsub("1768313_F01_BP_CP2", "1768313_F02_BP_CP2", changed)
changed<-as.matrix(changed)
Latent_Cases<-cbind(Latent_Cases[,(1:2)], changed, Latent_Cases[,4])


fingers<-read.csv("E:/AFIX output/fingers.csv")
```

```
control_fingers<-fingers[,2]
control_latents<-latents[,2]


control_latents_id<-latents[,3]
control_latent_cases_id<-Latent_Cases[,2]

control_latent_cases<-Latent_Cases[,2]
fingers_minutiae<-fingers[,6]



t<-matrix(,,ncol = 9)

#z<-41
for(z in 1:length(control_latent_cases_id)){ #want to match the latent latent case id number
to the control number in latent cases
  matching_latent_files<-which(control_latent_cases_id[z] == control_latents_id)

  binding_latents<-cbind(latents[matching_latent_files,], Latent_Cases[z,4])

  for(k in 1:length(binding_latents[,9])){
    if(is.na(binding_latents[k,9])){binding_latents[k,9] = "1"}
  }


  binding_latents<-as.matrix(binding_latents)

  t<-rbind(t, binding_latents)

}

t_nrow<-nrow(t)
t<-t[2:t_nrow,]

#i<-1
minutiae_addition<-matrix(,,ncol = 10)
for(i in 1:length(control_fingers)){

  row_numbers<-which((control_fingers[i]==as.numeric(t[,2])))


  minutiae_update<-append(t[row_numbers,], fingers_minutiae[i])
  minutiae_update<-as.matrix(minutiae_update)
  minutiae_update<-t(minutiae_update)

  cols<-c("X", "Control ID", "Latent Case Number", "Latent Case Number", "Method",
          "Verification", "Area", "Extraction", "Repeated Cases", "Minutiae Count")

  minutiae_update<-`colnames<-`(minutiae_update, cols)

  minutiae_addition<-rbind(minutiae_addition, minutiae_update)
}

x<-nrow(minutiae_addition)
y<-ncol(minutiae_addition)

write.csv(minutiae_addition[(2:x), (2:y)], "E:/AFIX output/update latents with minutiae
counts.csv", row.names = FALSE)

# latents combination of manual and auto minutiae counts================
latents_minutiae<-read.csv("E:/AFIX output/update latents with minutiae counts.csv")
latents_minutiae<-as.matrix(latents_minutiae)

name_match<-latents_minutiae[,3]
#name_match<-substr(name_match, 1,7)

method_match<-latents_minutiae[,4]

area_match<-latents_minutiae[,6]

repeat_match<-latents_minutiae[,8]

#i<-1
blank_matrix2<-matrix( , ,ncol=6)
```

```
for(i in 1:length(name_match)){
  name_match_search<-which((name_match[i] == name_match))
  pulled_name_match<-as.matrix(latents_minutiae[name_match_search,])
  if(nrow(pulled_name_match) == 9 & ncol(pulled_name_match) ==1 ){
    pulled_name_match<-t(pulled_name_match)}

  method_match_search<-which(method_match[i]==pulled_name_match[,4])
  pulled_method_search<-as.matrix(pulled_name_match[method_match_search,])
  if(nrow(pulled_method_search) == 9 & ncol(pulled_method_search) ==1){
    pulled_method_search<-t(pulled_method_search)}

  repeated_match_search<-which(repeat_match[i]==pulled_method_search[,8])
  pulled_repeated_search<-as.matrix(pulled_method_search[repeated_match_search,])
  if(nrow(pulled_repeated_search) == 9 & ncol(pulled_repeated_search) ==1){
    pulled_repeated_search<-t(pulled_repeated_search)}

  area_match_search<-which((area_match[i] == pulled_repeated_search[,6]))
  pulled_area_search<-as.matrix(pulled_repeated_search[area_match_search,])
  if(nrow(pulled_area_search) == 9 & ncol(pulled_area_search) ==1){
    pulled_area_search<-t(pulled_area_search)}



  if(nrow(pulled_area_search) != 2 & pulled_area_search[1,7] == "AUTO"){
    extraction_combo<-cbind(name_match[i], as.character(method_match[i]),
as.character(area_match[i]),
                            pulled_area_search[1,8], pulled_area_search[1,9],0)

    cols<-c("Name", "Method","Area", "Repeated Cases", "Auto-Minutiae Count", "Manual-Minutiae
Count")

    extraction_combo_names<-`colnames<-`(extraction_combo, cols)
  }

  if (nrow(pulled_area_search) != 2 & pulled_area_search[1,7] == "MANUAL"){
    extraction_combo<-cbind(name_match[i], as.character(method_match[i]),
as.character(area_match[i]),
                            pulled_area_search[1,8], 0, pulled_area_search[1,9])

    cols<-c("Name", "Method","Area", "Repeated Cases", "Auto-Minutiae Count", "Manual-Minutiae
Count")

    extraction_combo_names<-`colnames<-`(extraction_combo, cols)
  }

  if(nrow(pulled_area_search) == 2){
    if(pulled_method_search[1,7] == "AUTO" & pulled_area_search[2,7]== "MANUAL"){
      extraction_combo<-cbind(name_match[i], as.character(method_match[i]),
as.character(area_match[i]),
                              pulled_area_search[1,8],
pulled_area_search[1,9],pulled_area_search[2,9])

      cols<-c("Name", "Method","Area", "Repeated Cases", "Auto-Minutiae Count", "Manual-
Minutiae Count")

      extraction_combo_names<-`colnames<-`(extraction_combo, cols)
    }
  }

  if(nrow(pulled_area_search) == 2){
    if (pulled_area_search[1,7] == "MANUAL" & pulled_area_search[2,7] == "AUTO"){
      extraction_combo<-cbind(name_match[i], as.character(method_match[i]),
as.character(area_match[i]),

pulled_area_search[1,8],pulled_area_search[2,9],pulled_area_search[1,9])

      cols<-c("Name", "Method","Area", "Repeated Cases", "Auto-Minutiae Count", "Manual-
Minutiae Count")

      extraction_combo_names<-`colnames<-`(extraction_combo, cols)
    }
  }

  if(nrow(pulled_area_search) == 2){
    if (pulled_area_search[1,7] == "MANUAL" & pulled_area_search[2,7] == "MANUAL"){
      extraction_combo1<-cbind(name_match[i], as.character(method_match[i]),
as.character(area_match[i]),
                               pulled_area_search[1,8], 0, pulled_area_search[1,9])

      cols<-c("Name", "Method","Area", "Repeated Cases", "Auto-Minutiae Count", "Manual-
Minutiae Count")

      extraction_combo_names<-`colnames<-`(extraction_combo1, cols)

    }
  }

  blank_matrix2<-rbind(blank_matrix2, extraction_combo_names)
}


test_nrow<-nrow(blank_matrix2)

write.csv(blank_matrix2[2:test_nrow,], "E:/AFIX output/latents combination of auto and manual
extraction.csv", row.names = FALSE)

# adding minutiae counts for afix ========================================
combination_extraction<-read.csv("E:/AFIX output/latents combination of auto and manual
extraction.csv")
research_data<-read.csv("E:/AFIX output/research-data-combined with afix.csv")

trim.leading <- function (x)
  sub("^\\s+", "", x) #removes the blank spaces in the names

combo_name<-combination_extraction[,1]
combo_name<-trim.leading(combo_name)

need_to_split<-strsplit(combo_name, "_") #split the string name up into the individual areas
split<-data.frame(need_to_split)
split<-t(split)
cols2<-c("Name", "Finger", "Development", "Substrate")

split<-`colnames<-`(split, cols2)

finger<-split[,2] #need to remove the F on the front of the finger number for matching
finger<-as.matrix(finger)
finger_in_name<-substr(finger, 2, 3)
finger_in_name<-as.matrix(finger_in_name)

finger_new<-vector("numeric") #new finger numbers

#t<-483
for(t in 1:length(finger_in_name)){
  if(finger_in_name[t,1] != "10"){finger_update<-substr(finger_in_name[t,1], 2, 2)}
  if(finger_in_name[t,1] == "10"){finger_update<-substr(finger_in_name[t,1], 1, 2)}
  finger_update<-as.character(finger_update)

  finger_new<-rbind(finger_new, finger_update)
}

combo<-cbind(split[,1], finger_new, combination_extraction[, (2:6)]) #combining everything
back together

combo_name_update<-combo[,1]

combo_finger<-combo[,2]

combo_method<-combo[,3]
combo_method<-trim.leading(combo_method)

combo_area<-combo[,4]
combo_area<-trim.leading(combo_area)

combo_repeated<-combo[,5]

rr_name<-research_data[,3]
rr_name<-trim.leading(rr_name)

rr_finger<-research_data[,4]

rr_repeat<-research_data[,5]
```

```
rr_method<-research_data[,8]
rr_method<-trim.leading(rr_method)

rr_area<-research_data[,12]
rr_area<-trim.leading(rr_area)

#i<-1

for(i in 1:length(combo_name_update)){
  same_name<-which((combo_name_update[i] == rr_name), arr.ind = TRUE)
  same_name_data<-as.matrix(research_data[same_name,])
  if(nrow(same_name_data) == 39 & ncol(same_name_data) ==1){
    same_name_data<-t(same_name_data)}

  same_method<-grep(combo_method[i], same_name_data[,8])
  same_method_data<-as.matrix(same_name_data[same_method,])
  if(nrow(same_method_data) == 39 & ncol(same_method_data) ==1){
    same_method_data<-t(same_method_data)}

  same_repeat<-which(combo_repeated[i] == same_method_data[,5])
  same_repeat_data<-as.matrix(same_method_data[same_repeat,])
  if(nrow(same_repeat_data) == 39 & ncol(same_repeat_data) ==1){
    same_repeat_data<-t(same_repeat_data)}


  same_finger<-which(combo_finger[i] == trim.leading(same_repeat_data[,4]))
  same_finger_data<-as.matrix(same_repeat_data[same_finger,])
  if(nrow(same_finger_data) == 39 & ncol(same_finger_data) ==1){
    same_finger_data<-t(same_finger_data)}

  same_area<-which(combo_area[i]== same_finger_data[,12])
  same_area_data<-as.matrix(same_finger_data[same_area,])
  if(nrow(same_area_data) == 39 & ncol(same_area_data) ==1){
    same_area_data<-t(same_area_data)}

  row_number<-same_area_data[1]
  row_number<-as.numeric(row_number)

  research_data[row_number, 13] = combo[i,6]
  research_data[row_number, 23] = combo[i,7]
}


# removing blanks from the spreadsheet===========================================
for(t in 1:length(research_data[,13])){
  if(is.na(research_data[t,13])){research_data[t,13] = as.numeric("0")}
}

for(k in 1:length(research_data[,15])){
  if(is.na(research_data[k,15])){research_data[k,15] = as.numeric("0")}
}

for(k in 1:length(research_data[,16])){
  if(is.na(research_data[k,16])){research_data[k,16] = as.numeric("0")}
}

for(k in 1:length(research_data[,17])){
  if(is.na(research_data[k,17])){research_data[k,17] = as.numeric("0")}
}

for(k in 1:length(research_data[,18])){
  if(is.na(research_data[k,18])){research_data[k,18] = as.numeric("0")}
}

for(k in 1:length(research_data[,(19)])){
  if(is.na(research_data[k,19])){research_data[k,19] = as.numeric("0")}
}
for(k in 1:length(research_data[,20])){
  if(is.na(research_data[k,20])){research_data[k,20] = as.numeric("0")}
}
for(k in 1:length(research_data[,21])){
  if(is.na(research_data[k,21])){research_data[k,21] = as.numeric("0")}
}
for(k in 1:length(research_data[,22])){
  if(is.na(research_data[k,22])){research_data[k,22] =as.numeric("0")}
}
```

```
for(k in 1:length(research_data[,23])){
  if(is.na(research_data[k,23])){research_data[k,23] = as.numeric("0")}
}

for(k in 1:length(research_data[,25])){
  if(is.na(research_data[k,25])){research_data[k,25] = as.numeric("0")}
}
for(k in 1:length(research_data[,26])){
  if(is.na(research_data[k,26])){research_data[k,26] = as.numeric("0")}
}
for(k in 1:length(research_data[,27])){
  if(is.na(research_data[k,27])){research_data[k,27] = as.numeric("0")}
}
for(k in 1:length(research_data[,28])){
  if(is.na(research_data[k,28])){research_data[k,28] = as.numeric("0")}
}
for(k in 1:length(research_data[,29])){
  if(is.na(research_data[k,29])){research_data[k,29] = as.numeric("0")}
}
for(k in 1:length(research_data[,30])){
  if(is.na(research_data[k,30])){research_data[k,30] = as.numeric("0")}
}
for(k in 1:length(research_data[,31])){
  if(is.na(research_data[k,31])){research_data[k,31] = as.numeric("0")}
}
for(k in 1:length(research_data[,32])){
  if(is.na(research_data[k,32])){research_data[k,32] = as.numeric("0")}
}

research_row<-nrow(research_data)
research_col<-ncol(research_data)

write.csv(research_data[, (2:research_col)], "E:/Research Data-AFIX only.csv", row.names =
FALSE)
```

D. AFIX Tracker® Plots

D.1 Minutiae Scatter Plots

D.1.1 Subjective Method

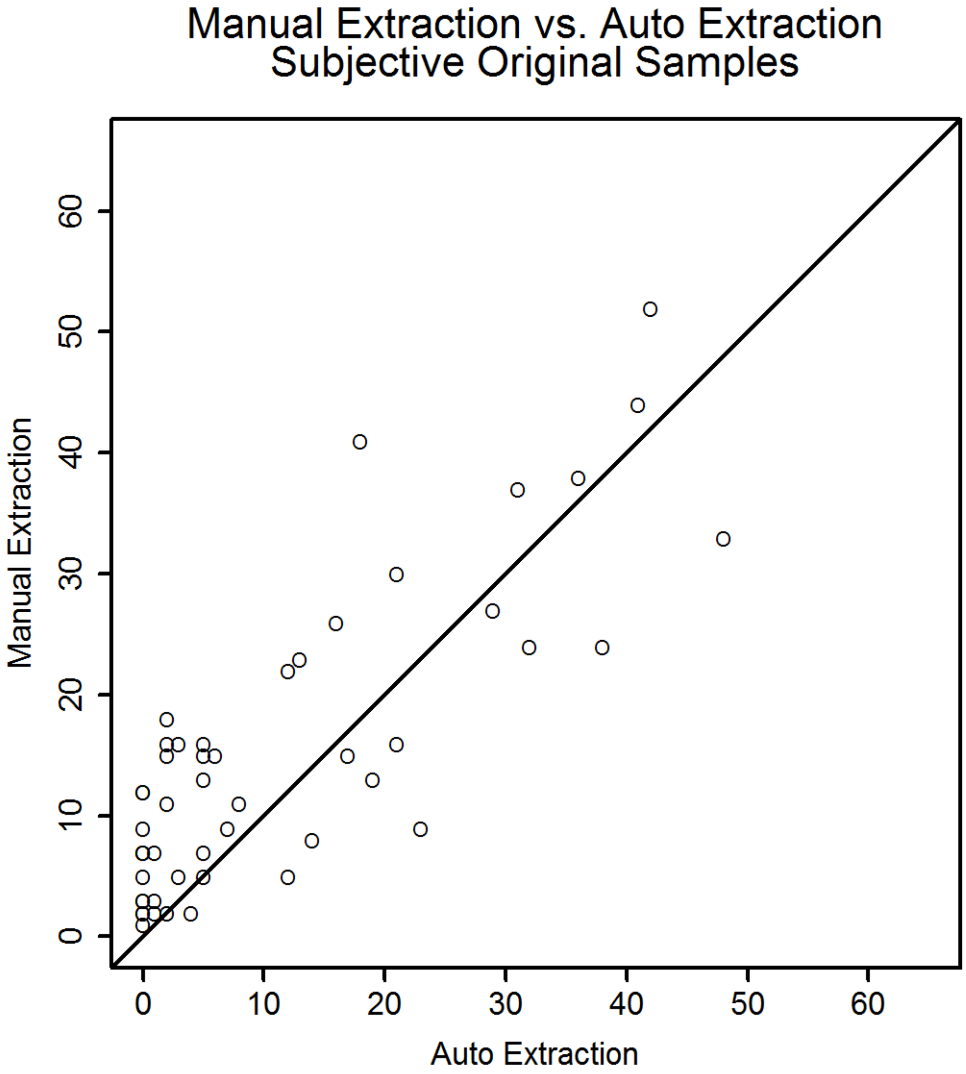


Figure D.1: The scatter plot for the manual minutiae extraction versus the automatic minutiae extraction for subjective original.

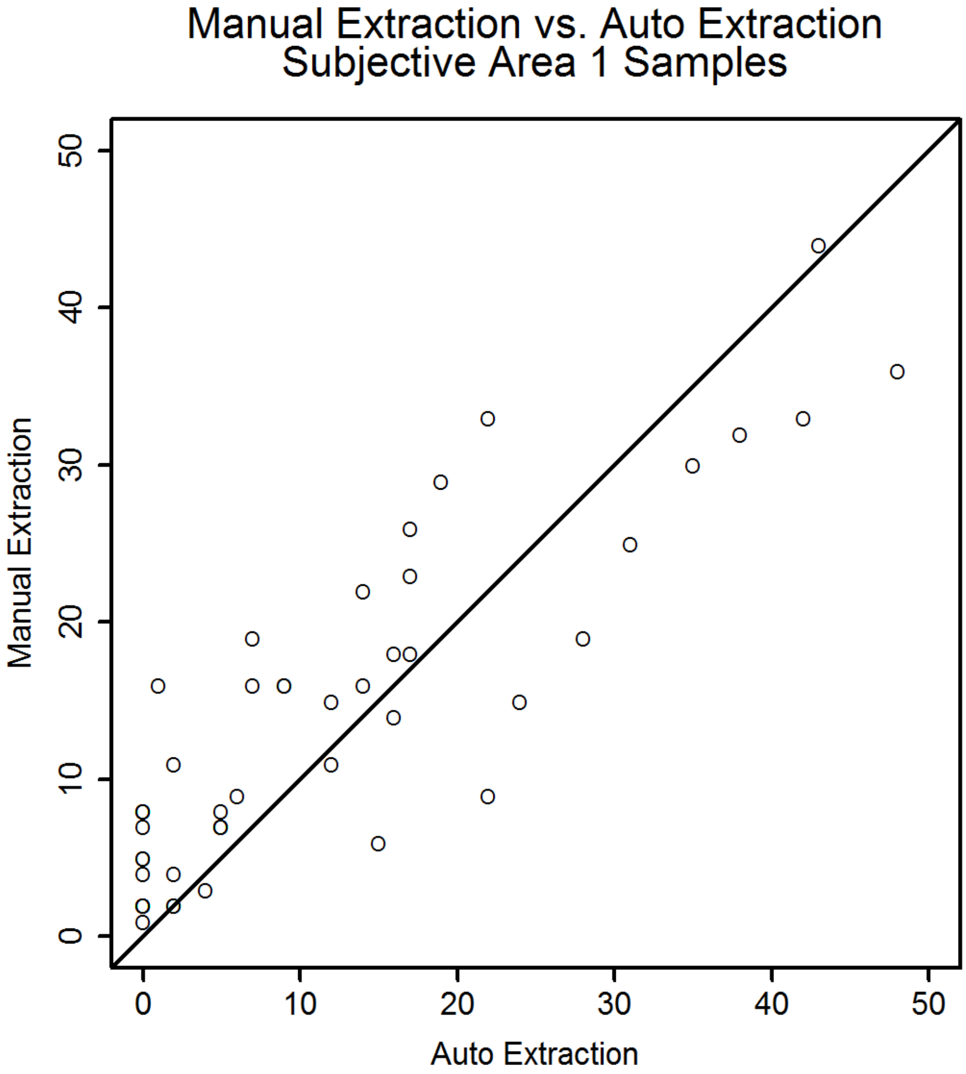


Figure D.2: The scatter plot for the manual minutiae extraction versus the automatic minutiae extraction for subjective Area 1.

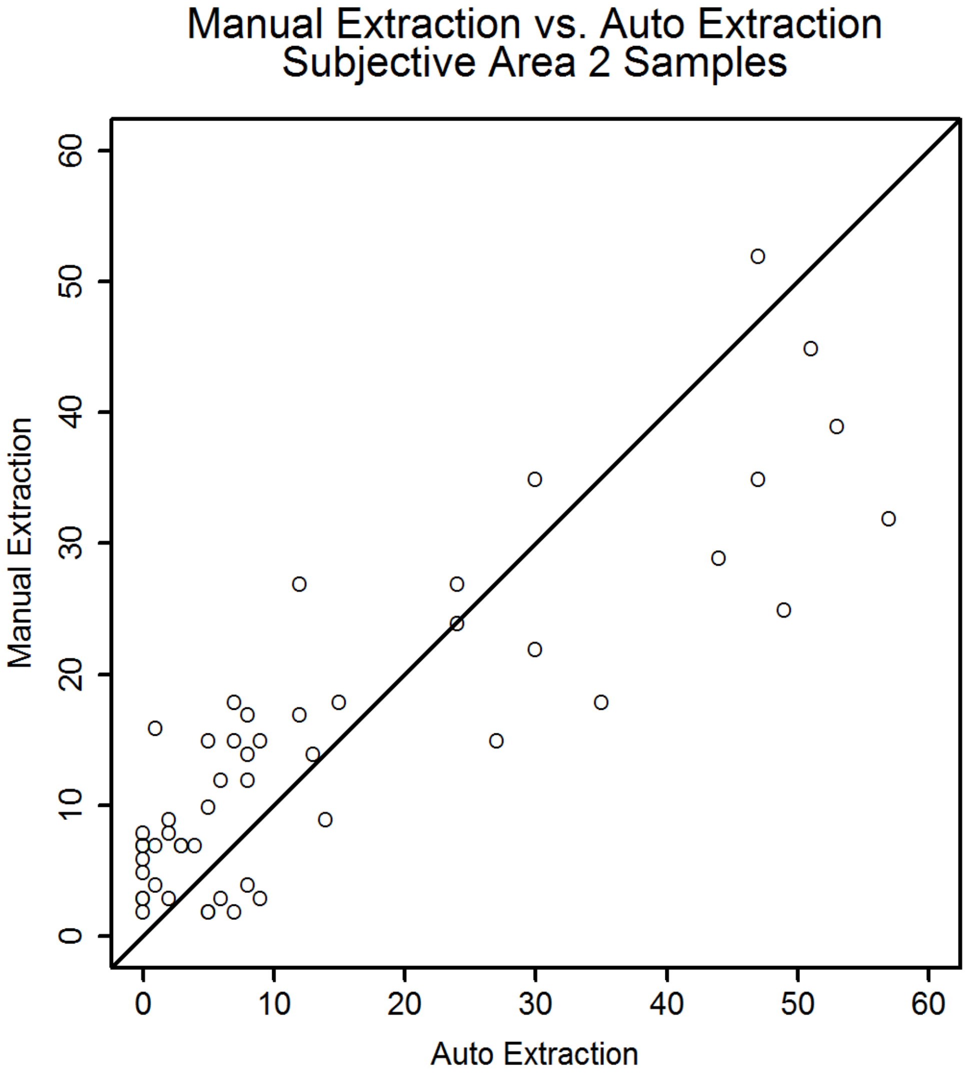


Figure D.3: The scatter plot for the manual minutiae extraction versus the automatic minutiae extraction for subjective Area 2.

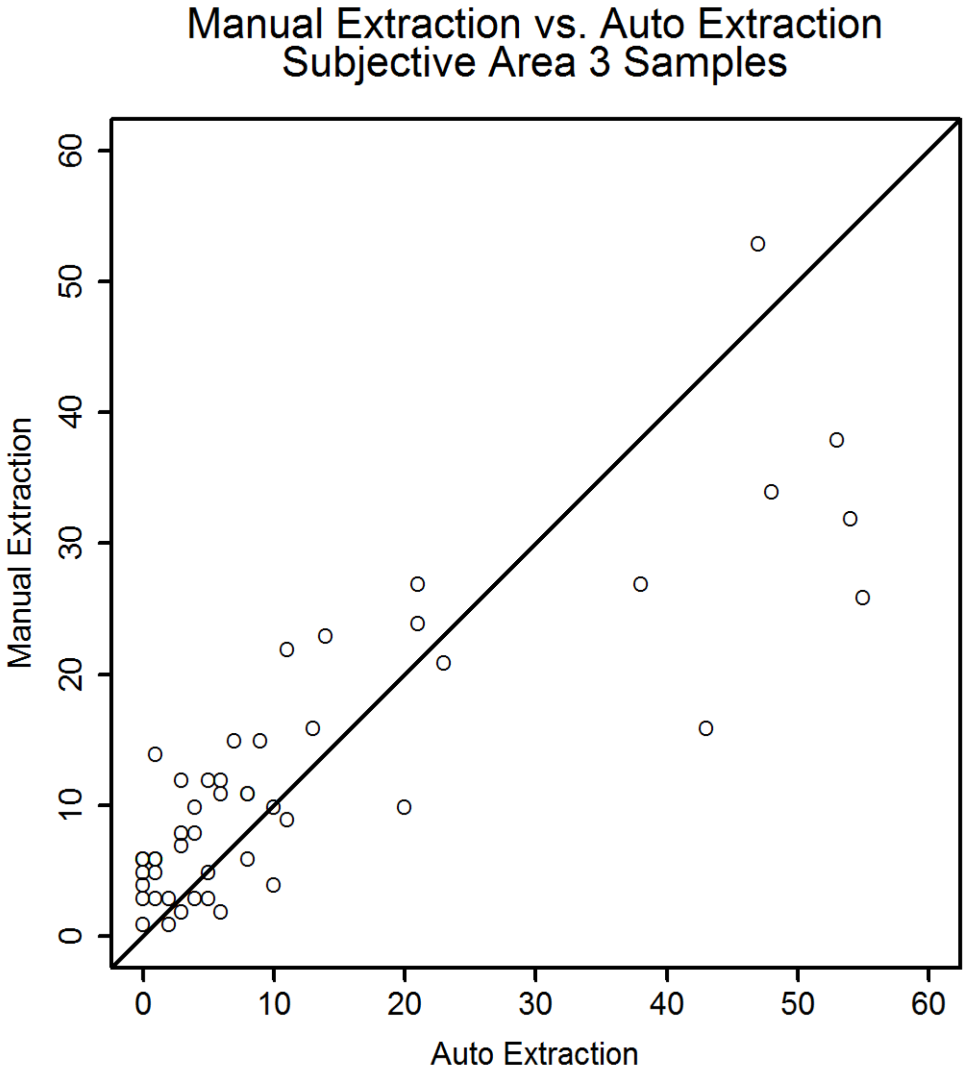


Figure D.4: The scatter plot for the manual minutiae extraction versus the automatic minutiae extraction for subjective Area 3.

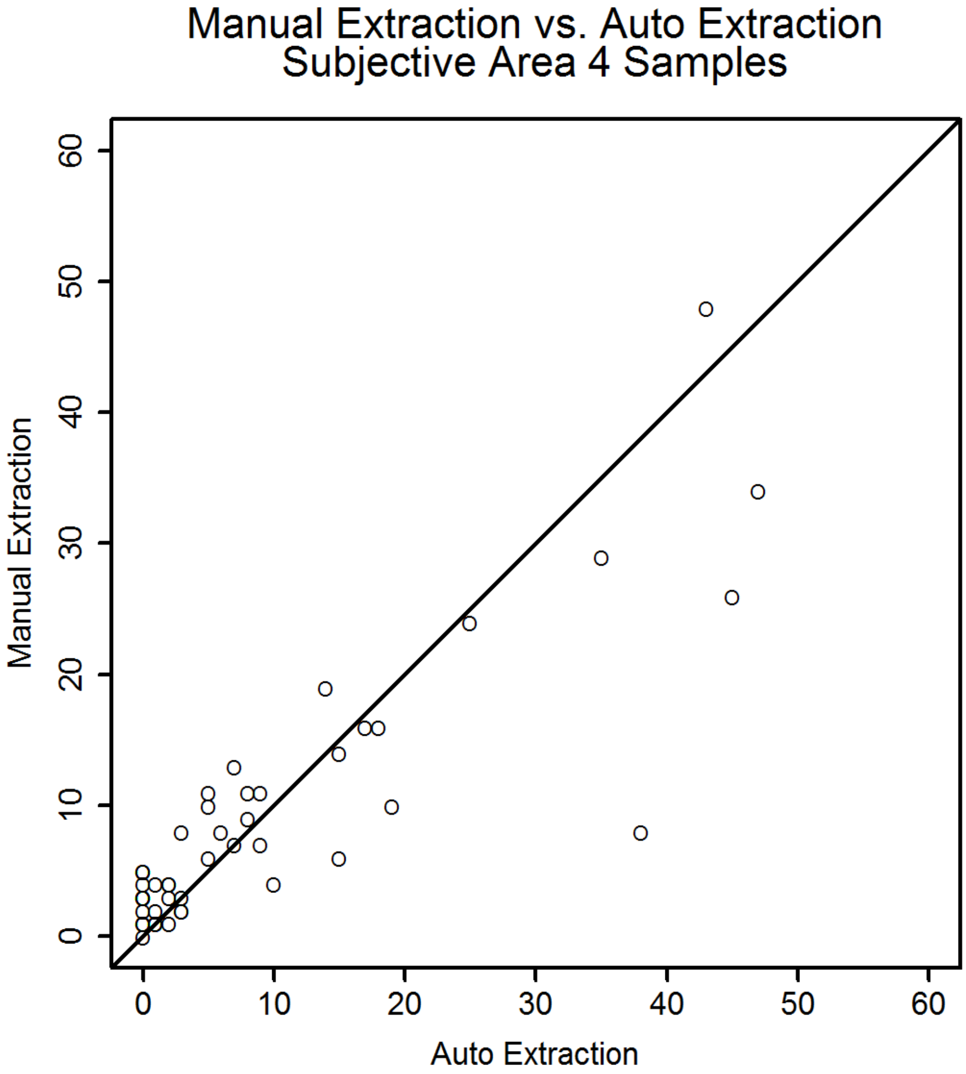


Figure D.5: The scatter plot for the manual minutiae extraction versus the automatic minutiae extraction for subjective Area 4.

D.1.2 Objective Method

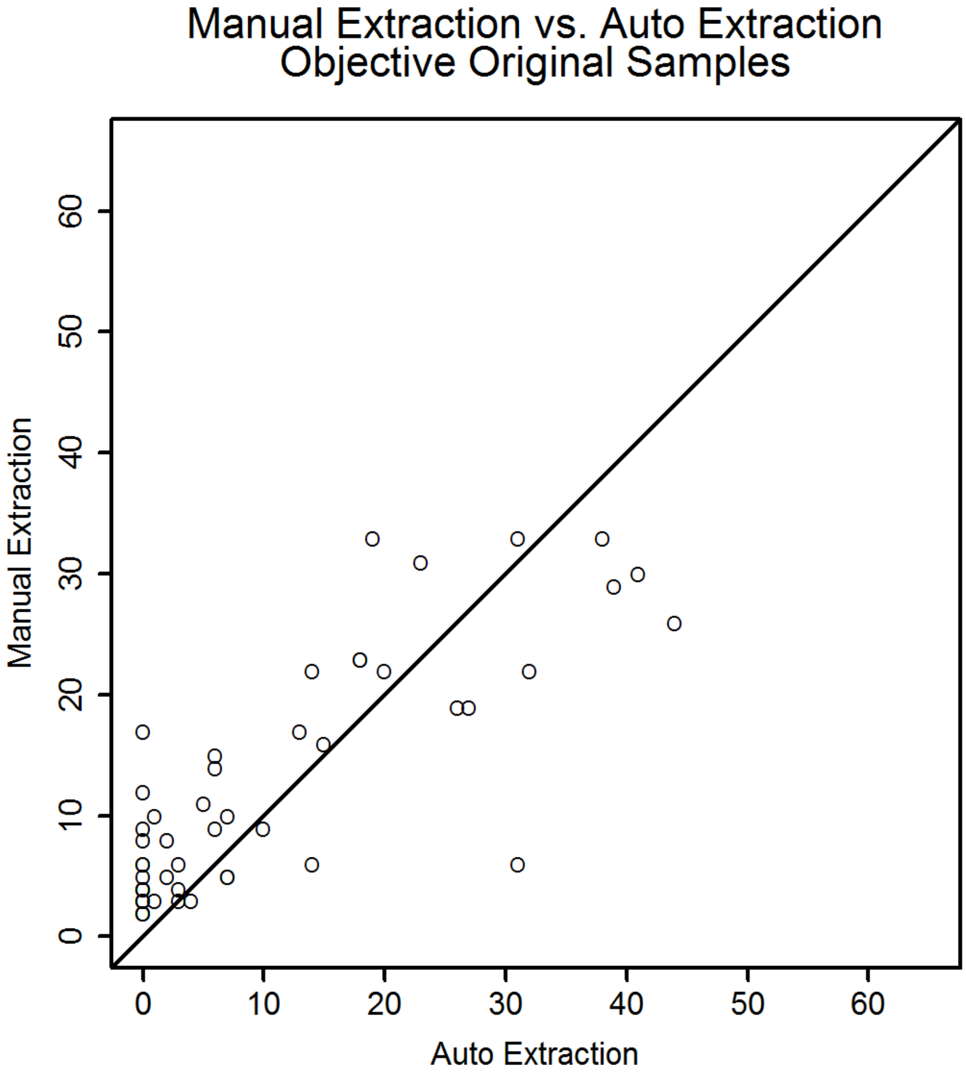


Figure D.6: The scatter plot for the manual minutiae extraction versus the automatic minutiae extraction for objective original samples.

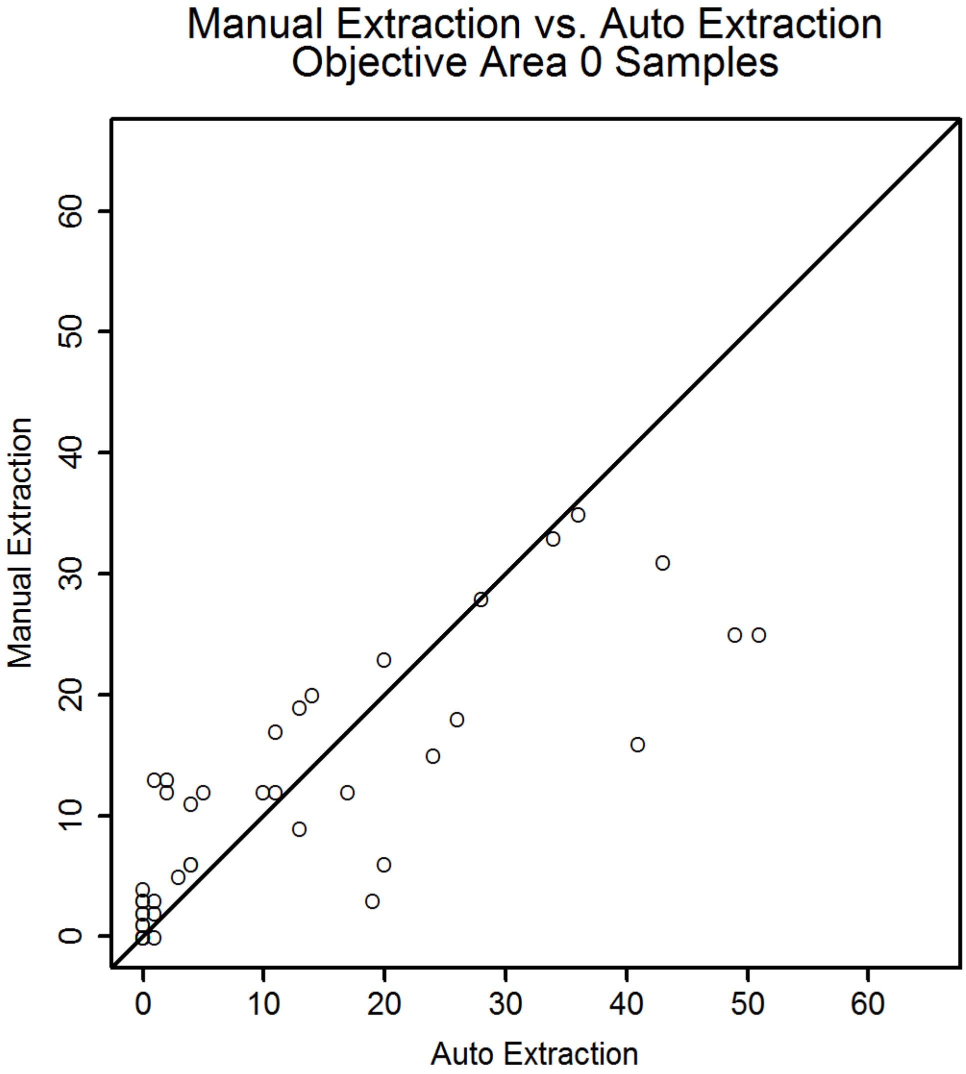


Figure D.7: The scatter plot for the manual minutiae extraction versus the automatic minutiae extraction for objective Area 0.

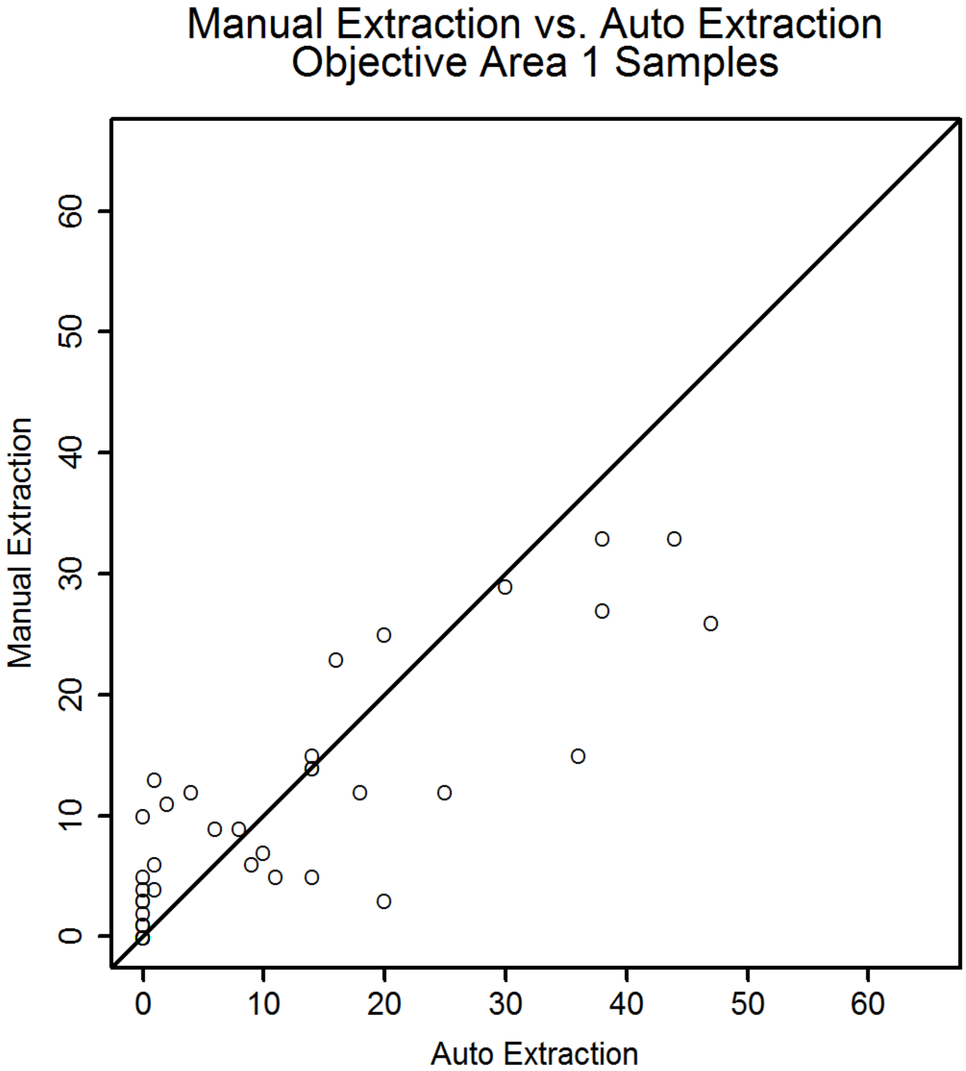


Figure D.8: The scatter plot for the manual minutiae extraction versus the automatic minutiae extraction for objective Area 1.

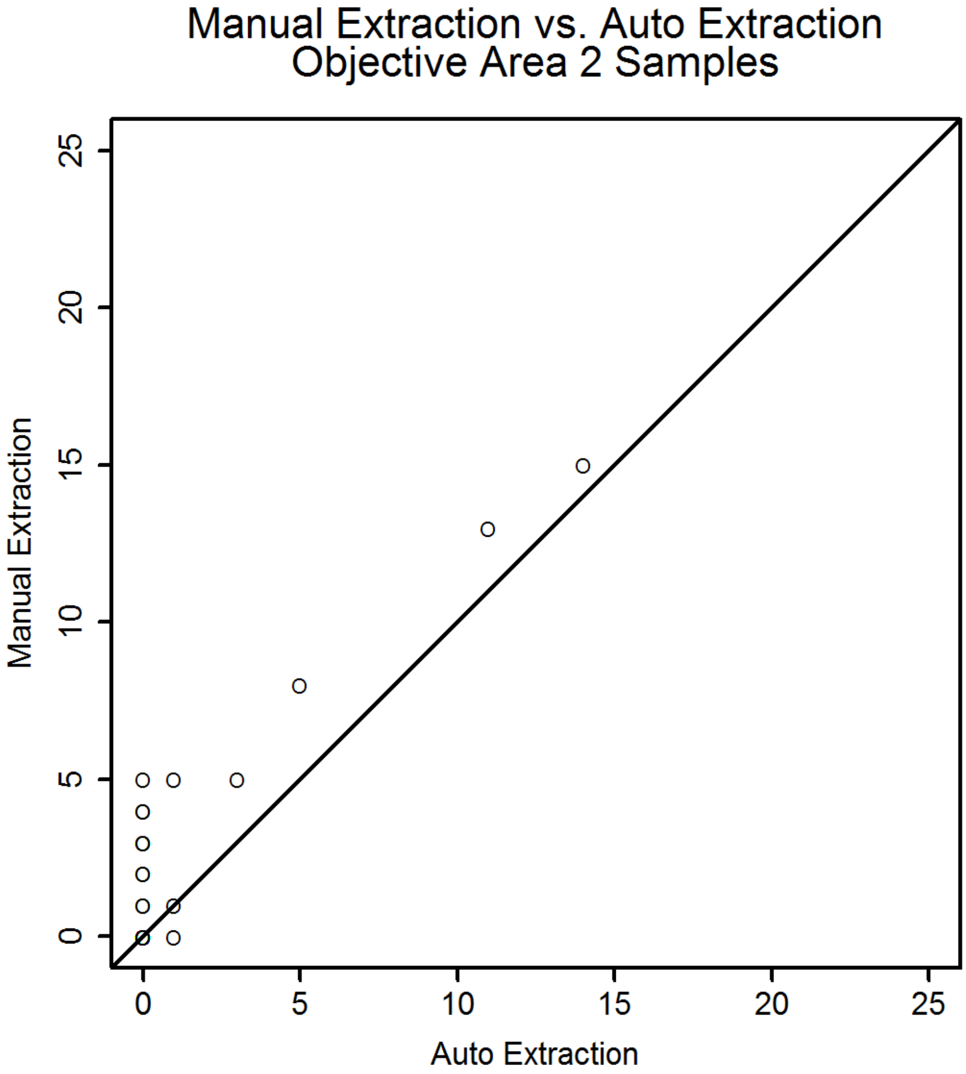


Figure D.9: The scatter plot for the manual minutiae extraction versus the automatic minutiae extraction for objective Area 2.

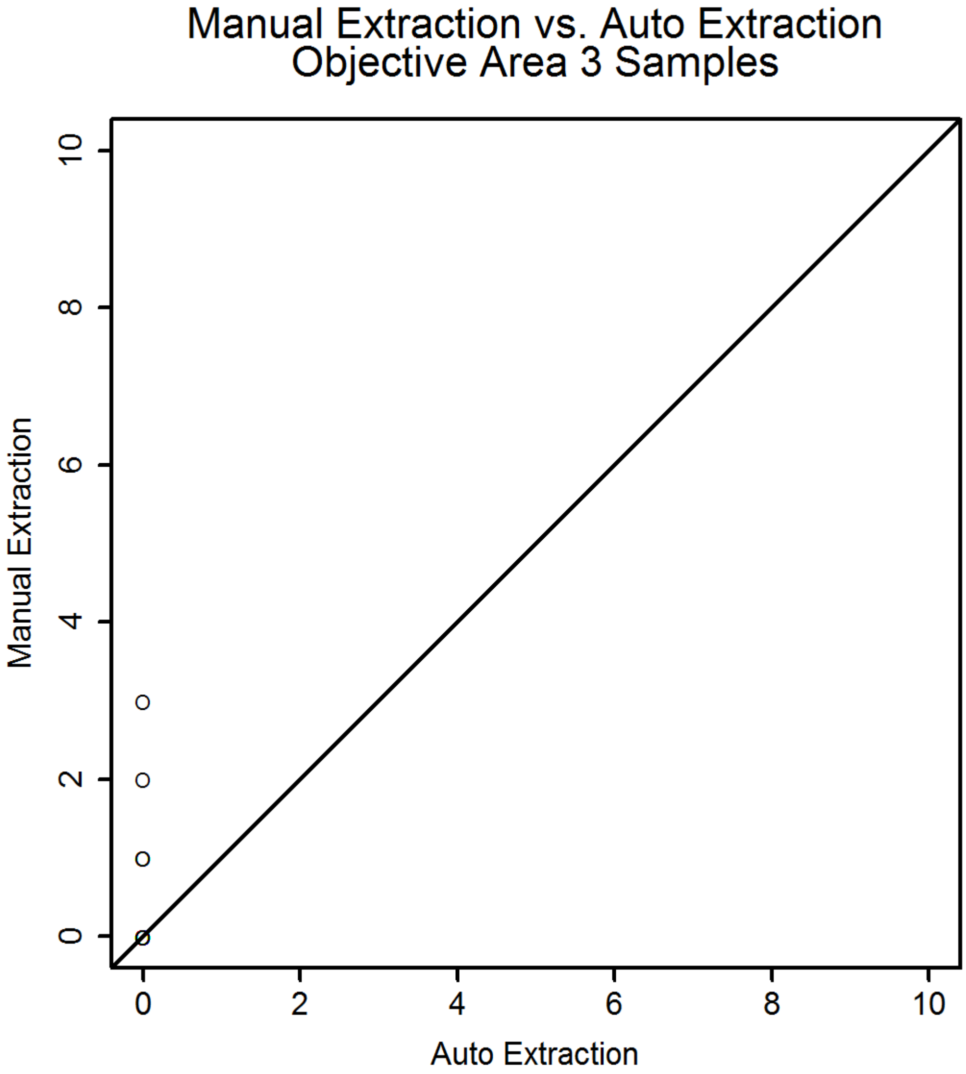


Figure D.10: The scatter plot for the manual minutiae extraction versus the automatic minutiae extraction for objective Area 3.

D.2 Development Methods

D.2.1 Subjective Method

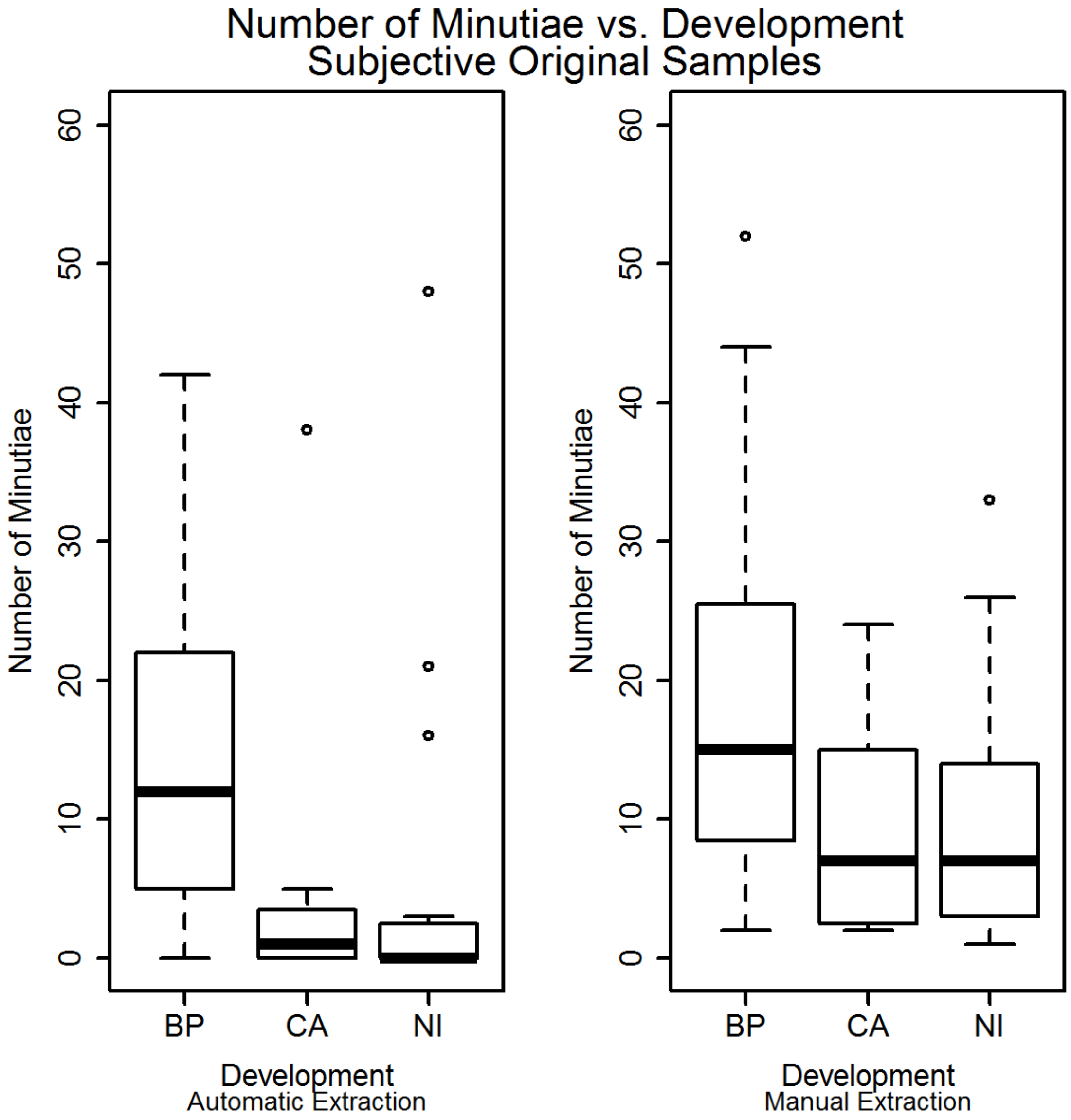


Figure D.11: This figure presents the box plots for the minutiae detected for each development type in the subjective original images. On the left is the automatic extraction box plots while on the right is the manual extraction. The development methods are represented by the two letters. BP represents black powder, CA represents cyanoacrylate and NI represents ninhydrin. The thick black line represents the median in the samples.

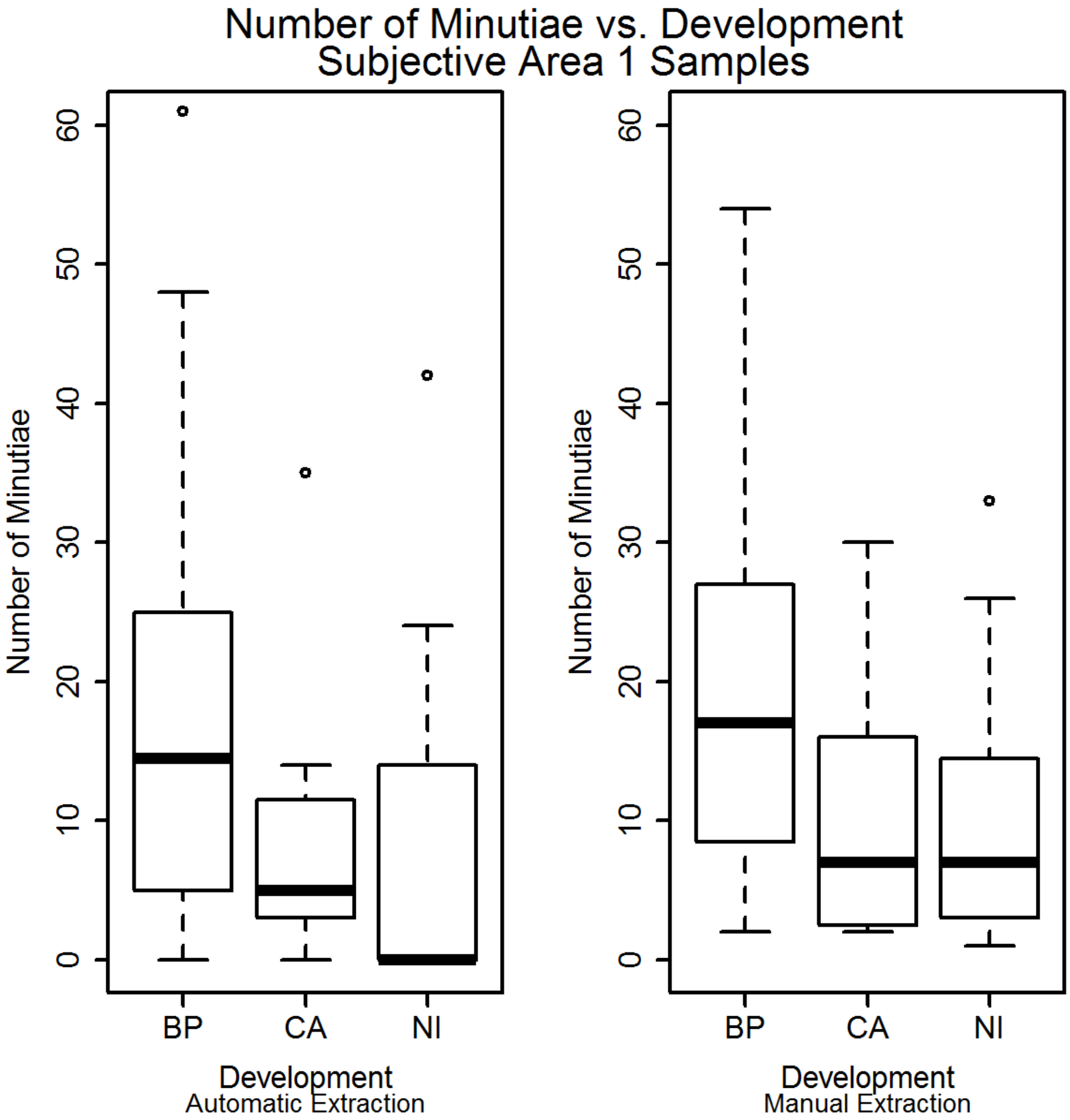


Figure D.12: This figure presents the box plots for the minutiae detected for each development type in subjective Area 1. On the left is the automatic extraction box plots while on the right is the manual extraction. The development methods are represented by the two letters. BP represents black powder, CA represents cyanoacrylate and NI represents ninhydrin. The thick black line represents the median in the samples.

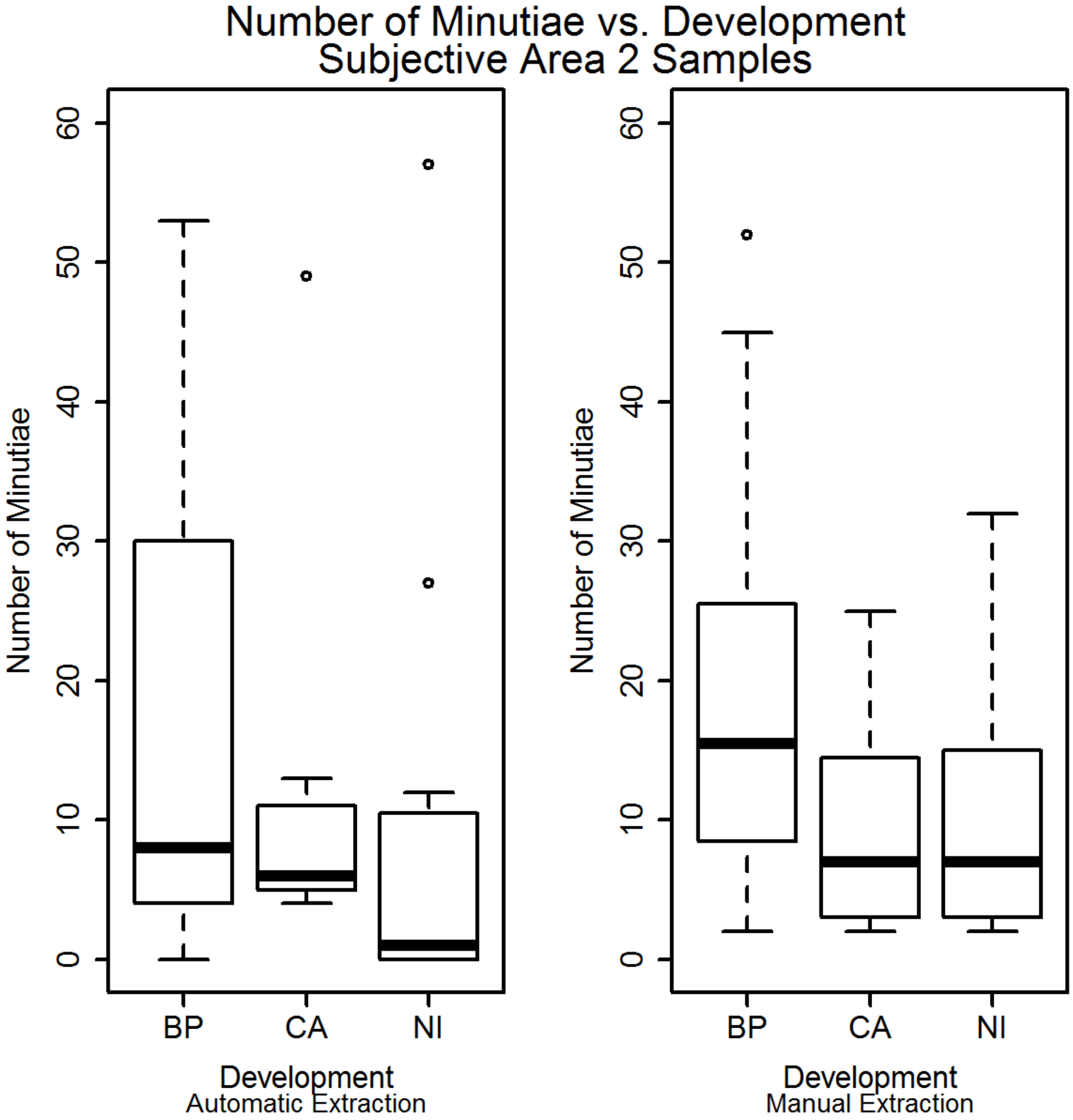


Figure D.13: This figure presents the box plots for the minutiae detected for each development type in subjective Area 2. On the left is the automatic extraction box plots while on the right is the manual extraction. The development methods are represented by the two letters. BP represents black powder, CA represents cyanoacrylate and NI represents ninhydrin. The thick black line represents the median in the samples.

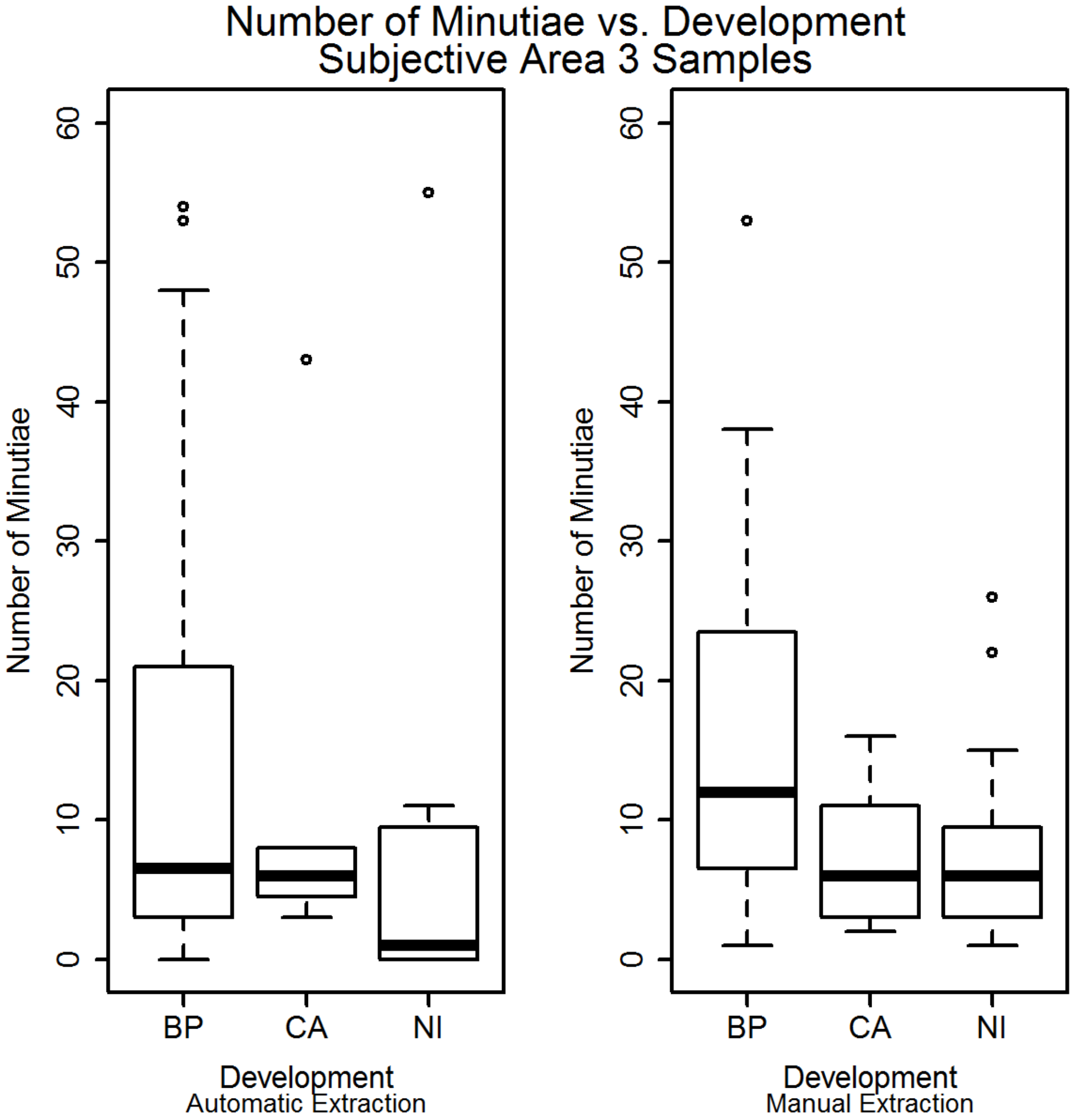


Figure D.14: This figure presents the box plots for the minutiae detected for each development type in subjective Area 3. On the left is the automatic extraction box plots while on the right is the manual extraction. The development methods are represented by the two letters. BP represents black powder, CA represents cyanoacrylate and NI represents ninhydrin. The thick black line represents the median in the samples.

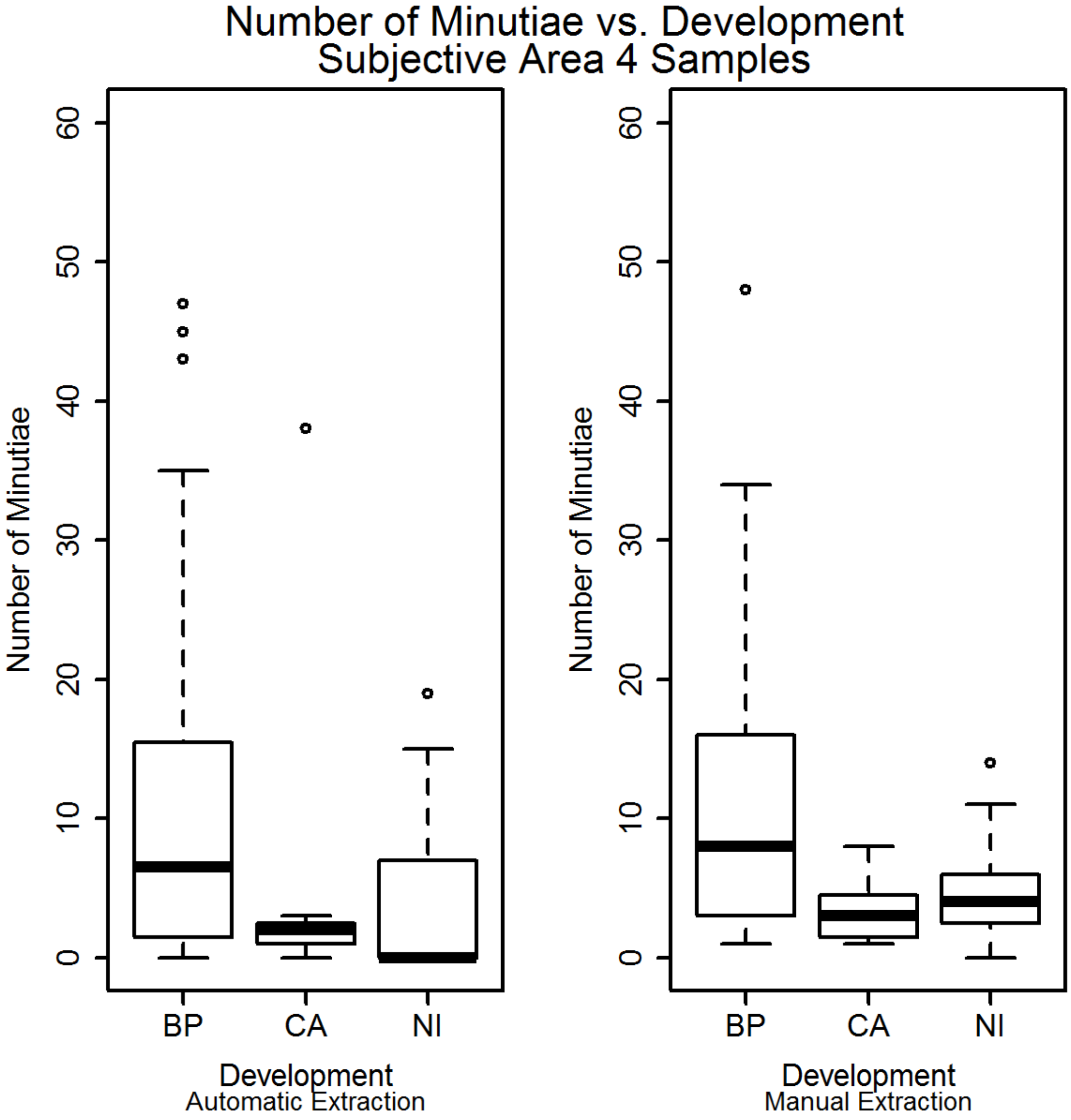


Figure D.15: This figure presents the box plots for the minutiae detected for each development type in objective Area 4. On the left is the automatic extraction box plots while on the right is the manual extraction. The development methods are represented by the two letters. BP represents black powder, CA represents cyanoacrylate and NI represents ninhydrin. The thick black line represents the median in the samples.

D.2.2 Objective Method

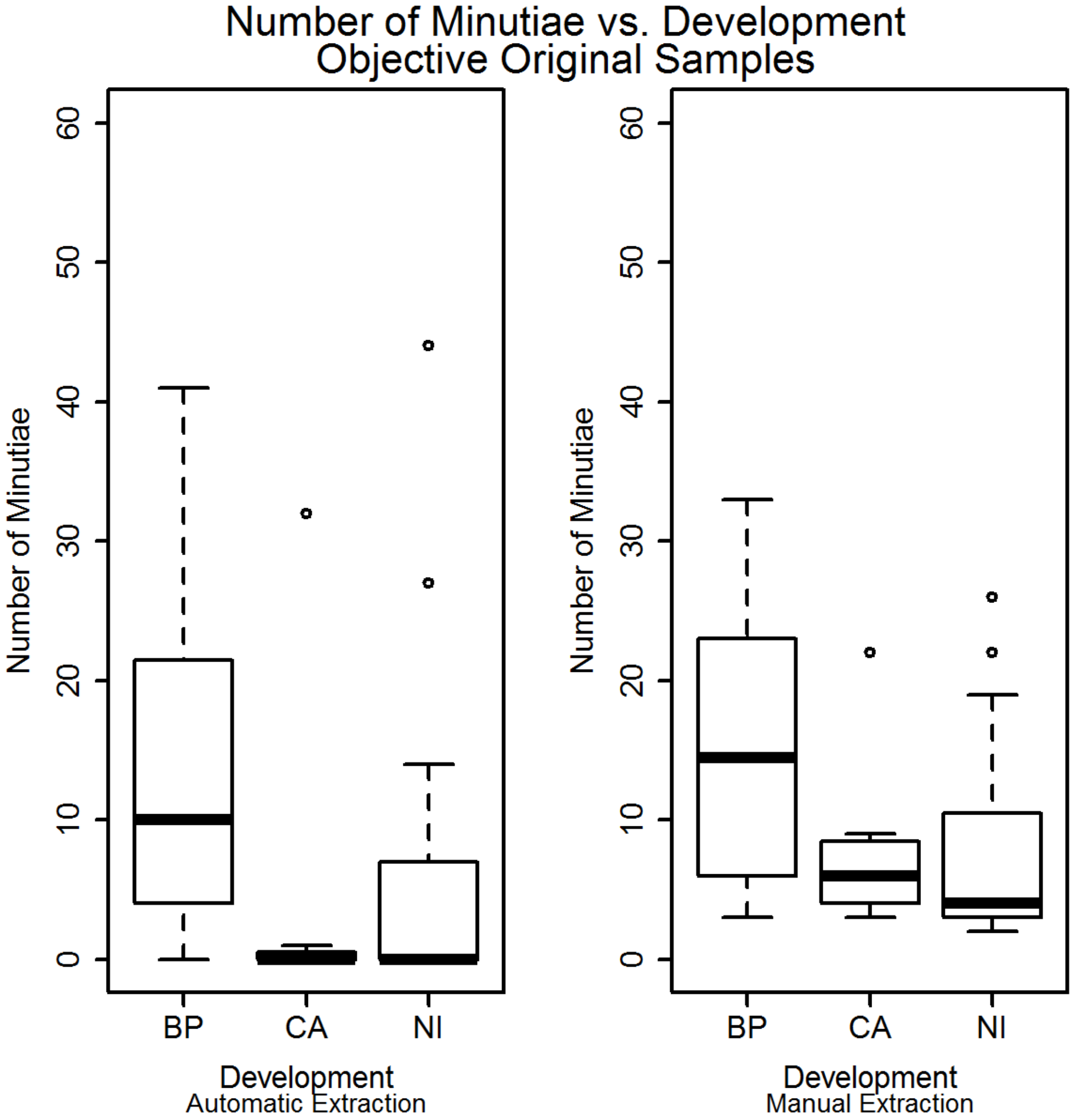


Figure D.16: This figure presents the box plots for the minutiae detected for each development type in the objective original images. On the left is the automatic extraction box plots while on the right is the manual extraction. The development methods are represented by the two letters. BP represents black powder, CA represents cyanoacrylate and NI represents ninhydrin. The thick black line represents the median in the samples.

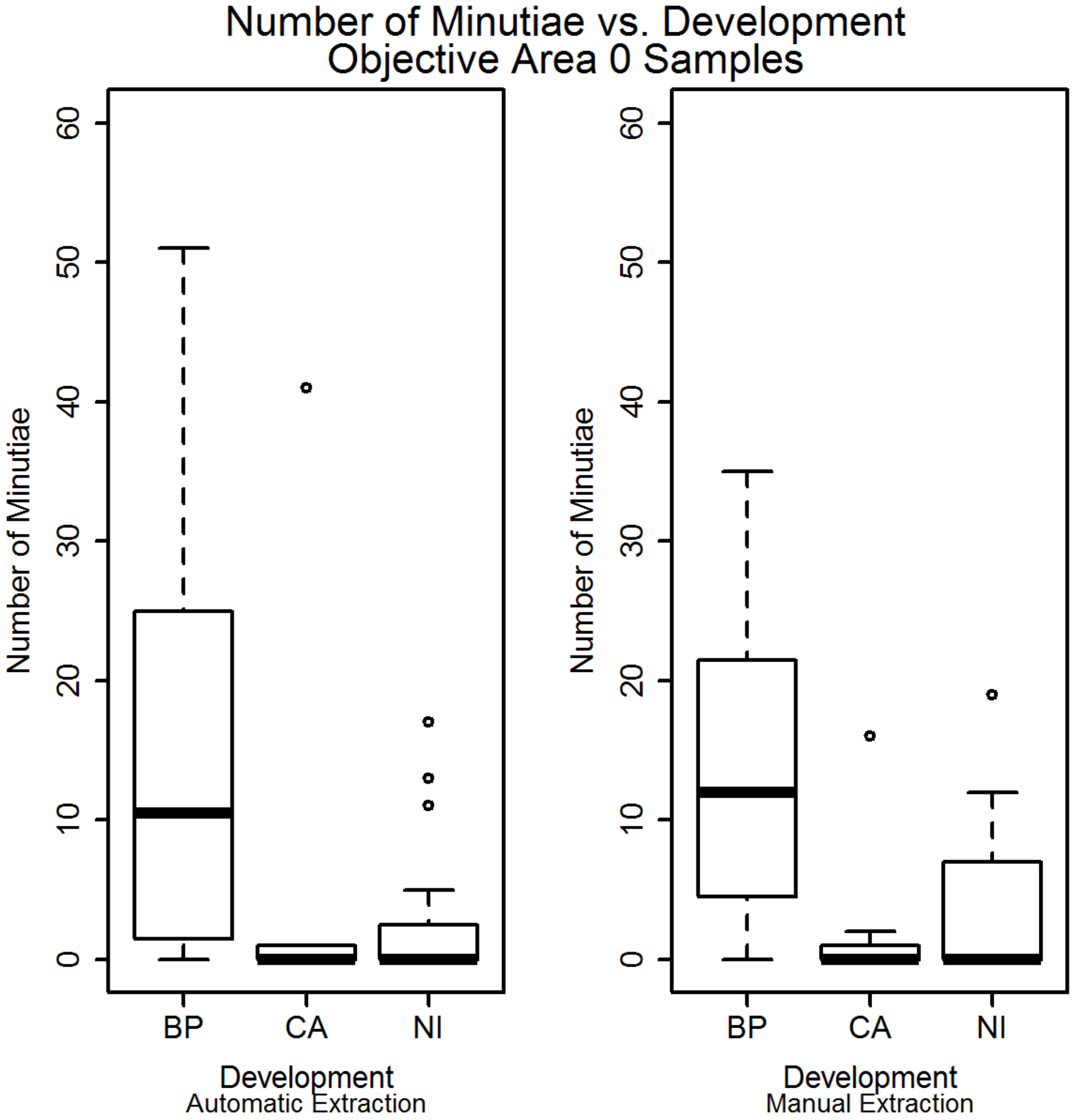


Figure D.17: This figure presents the box plots for the minutiae detected for each development type in objective Area 0. On the left is the automatic extraction box plots while on the right is the manual extraction. The development methods are represented by the two letters. BP represents black powder, CA represents cyanoacrylate and NI represents ninhydrin. The thick black line represents the median in the samples.

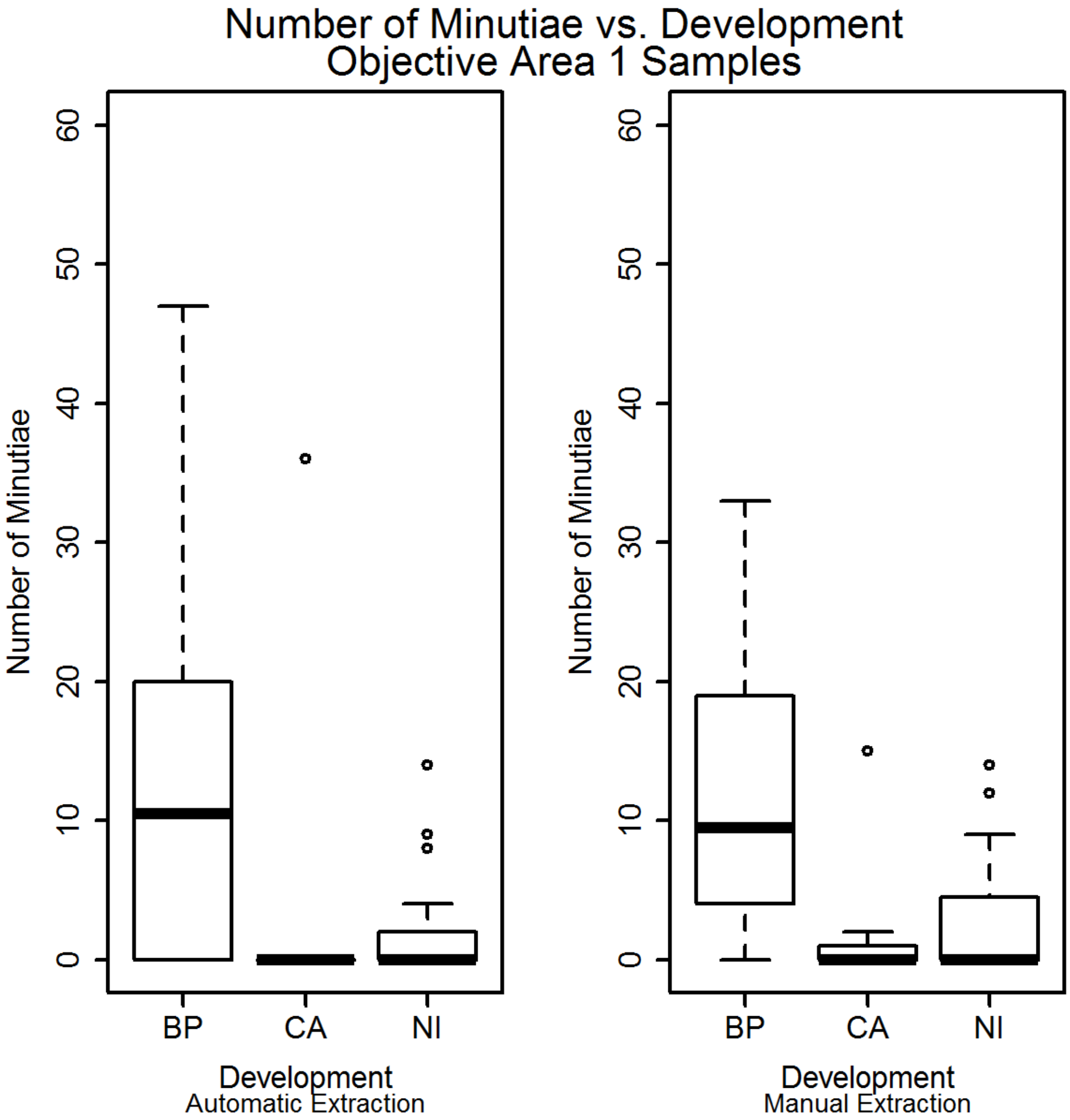


Figure D.18: This figure presents the box plots for the minutiae detected for each development type in objective Area 1. On the left is the automatic extraction box plots while on the right is the manual extraction. The development methods are represented by the two letters. BP represents black powder, CA represents cyanoacrylate and NI represents ninhydrin. The thick black line represents the median in the samples.

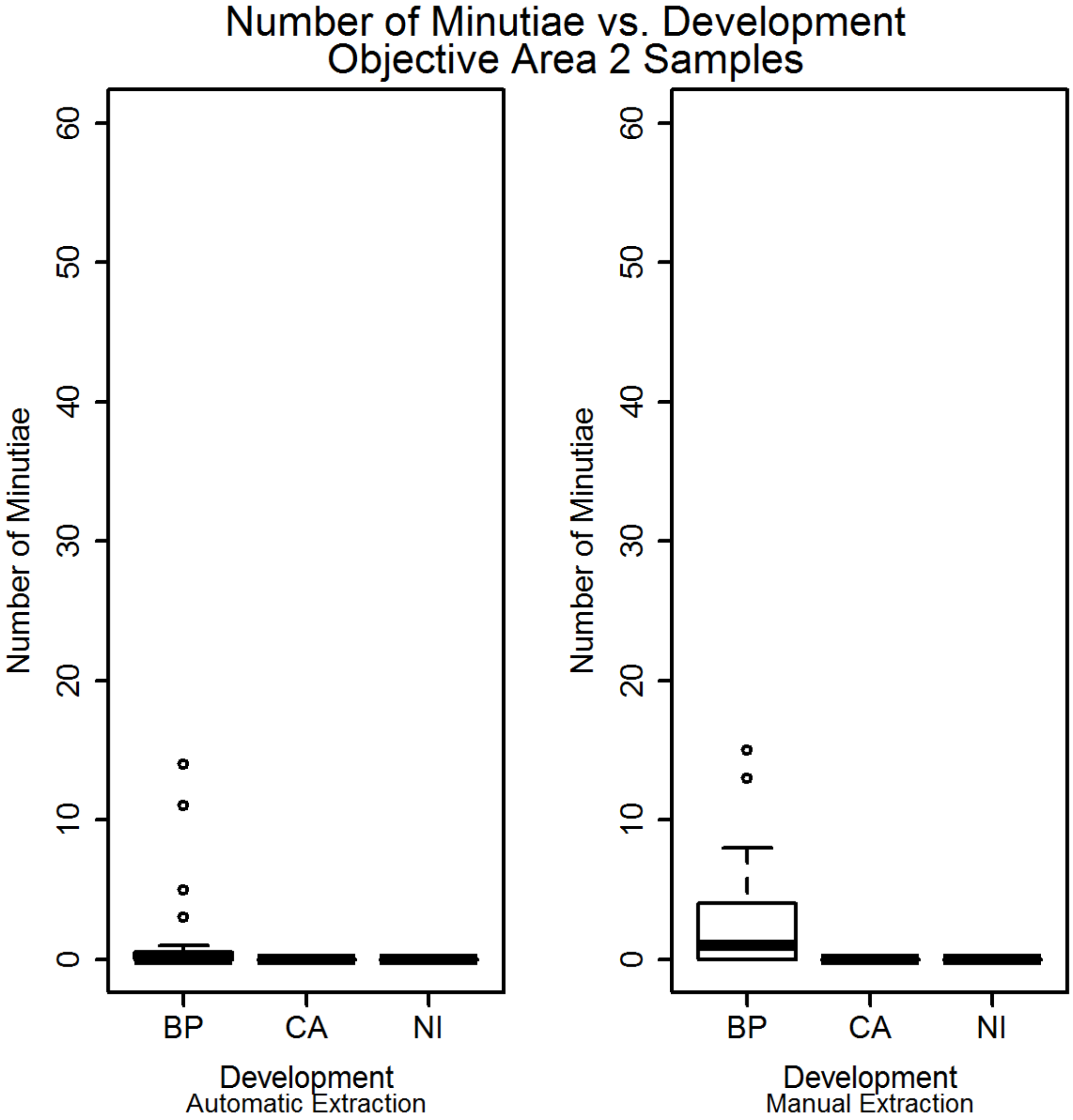


Figure D.19: This figure presents the box plots for the minutiae detected for each development type in objective Area 2. On the left is the automatic extraction box plots while on the right is the manual extraction. The development methods are represented by the two letters. BP represents black powder, CA represents cyanoacrylate and NI represents ninhydrin. The thick black line represents the median in the samples.

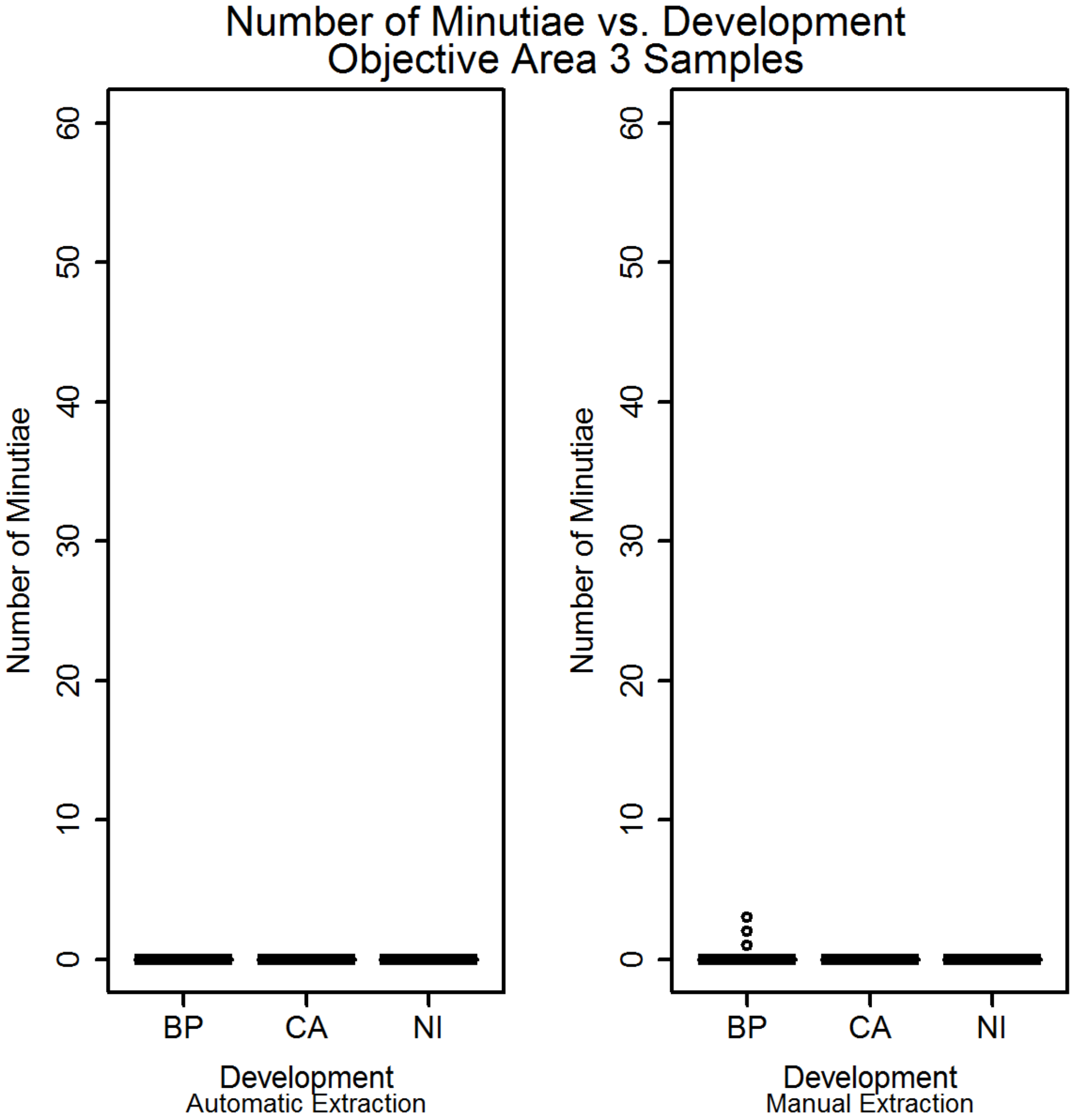


Figure D.20: This figure presents the box plots for the minutiae detected for each development type in objective Area 3. On the left is the automatic extraction box plots while on the right is the manual extraction. The development methods are represented by the two letters. BP represents black powder, CA represents cyanoacrylate and NI represents ninhydrin. The thick black line represents the median in the samples.

D.3 Substrate Types

D.3.1 Subjective Method

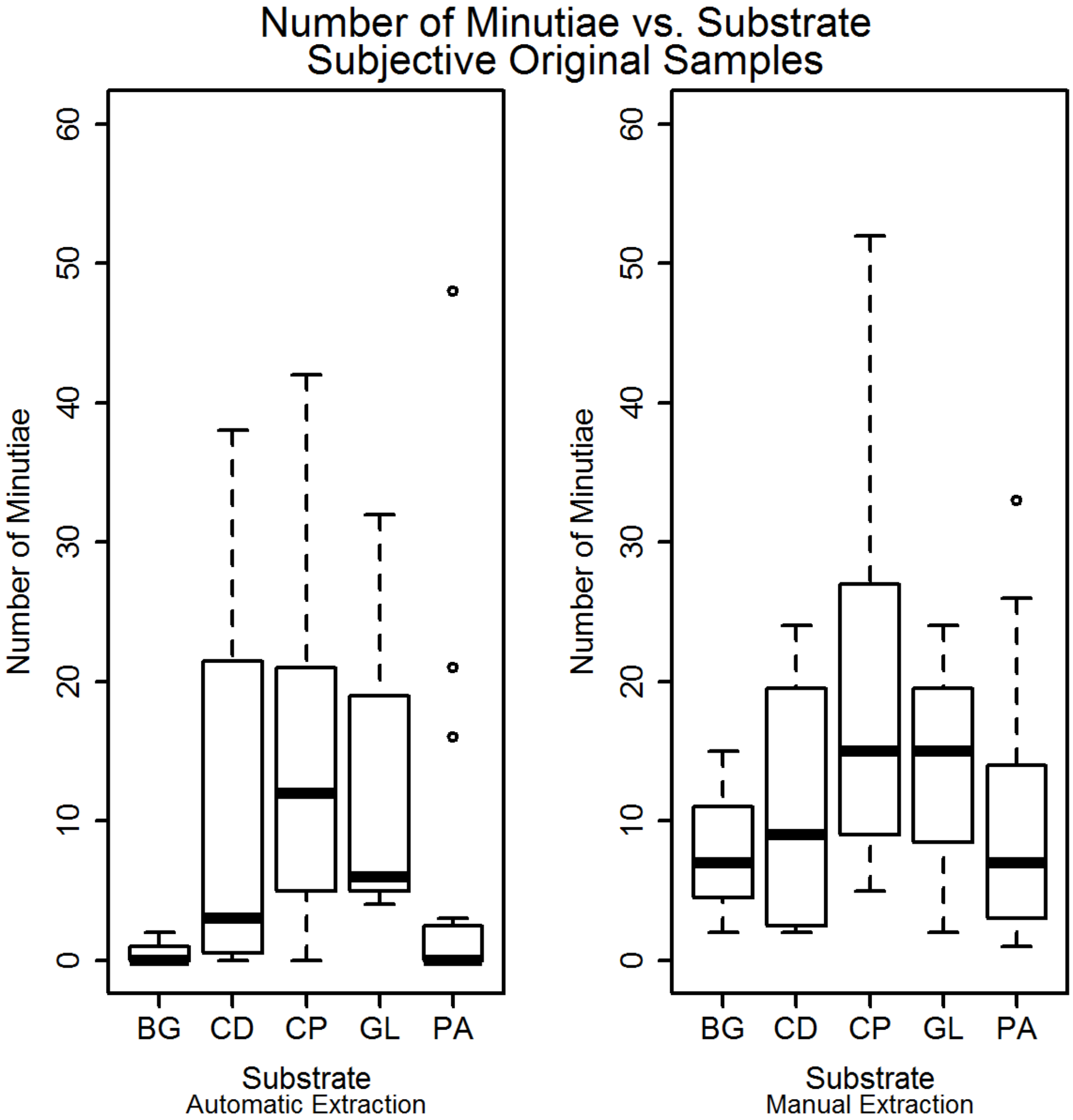


Figure D.21: This figure presents the box plots for the minutiae detected for each substrate type for the subjective original images. On the left is the automatic extraction box plots while on the right is the manual extraction. The substrate types are represented by the two letters. BG represents bag, CD represents a CD, CP represents ceramic plate, GL represents glass and PA represents paper. The thick black line represents the median in the samples.

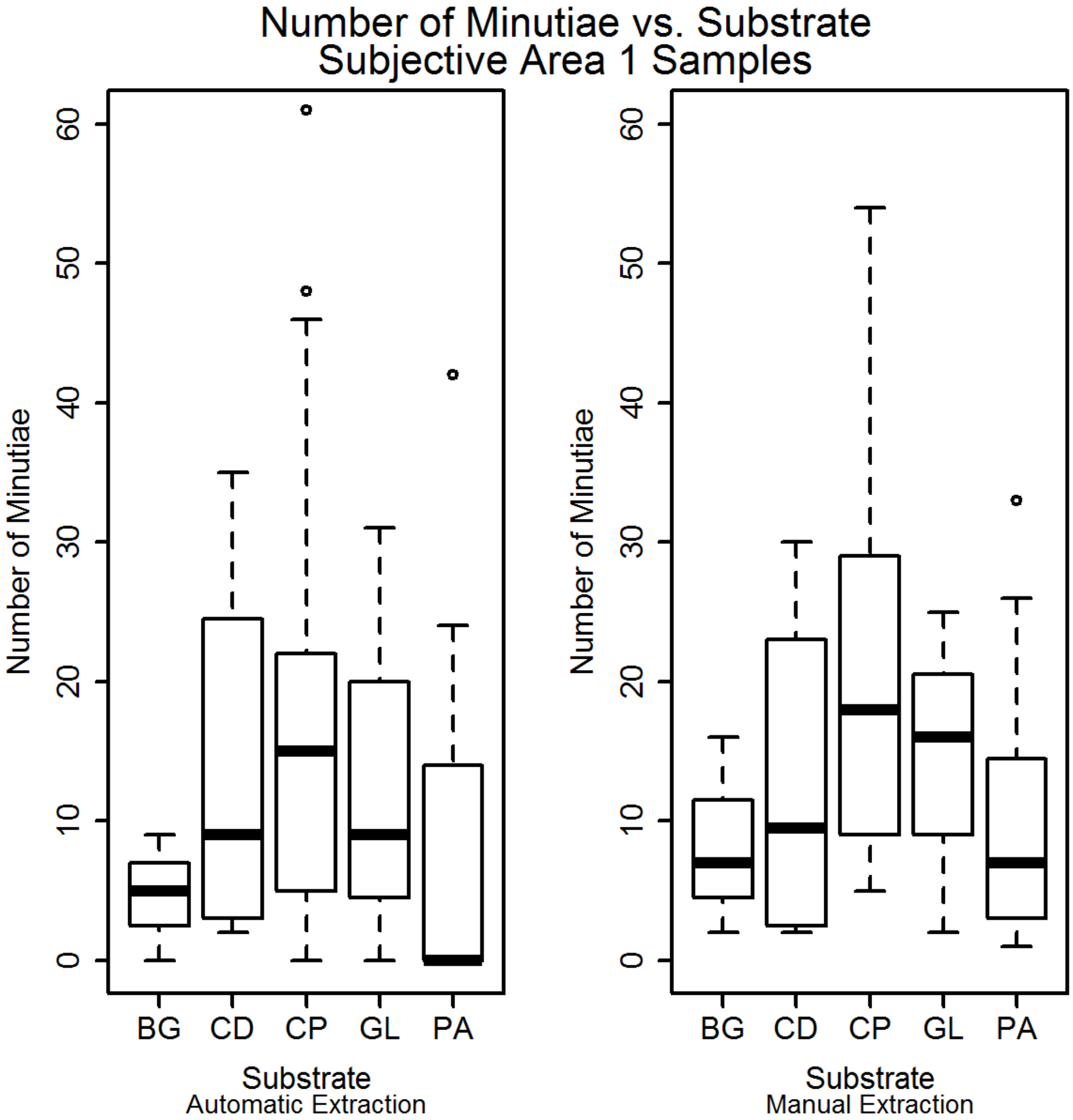


Figure D.22: This figure presents the box plots for the minutiae detected for each substrate type for the subjective Area 1 images. On the left is the automatic extraction box plots while on the right is the manual extraction. The substrate types are represented by the two letters. BG represents bag, CD represents a CD, CP represents ceramic plate, GL represents glass and PA represents paper. The thick black line represents the median in the samples.

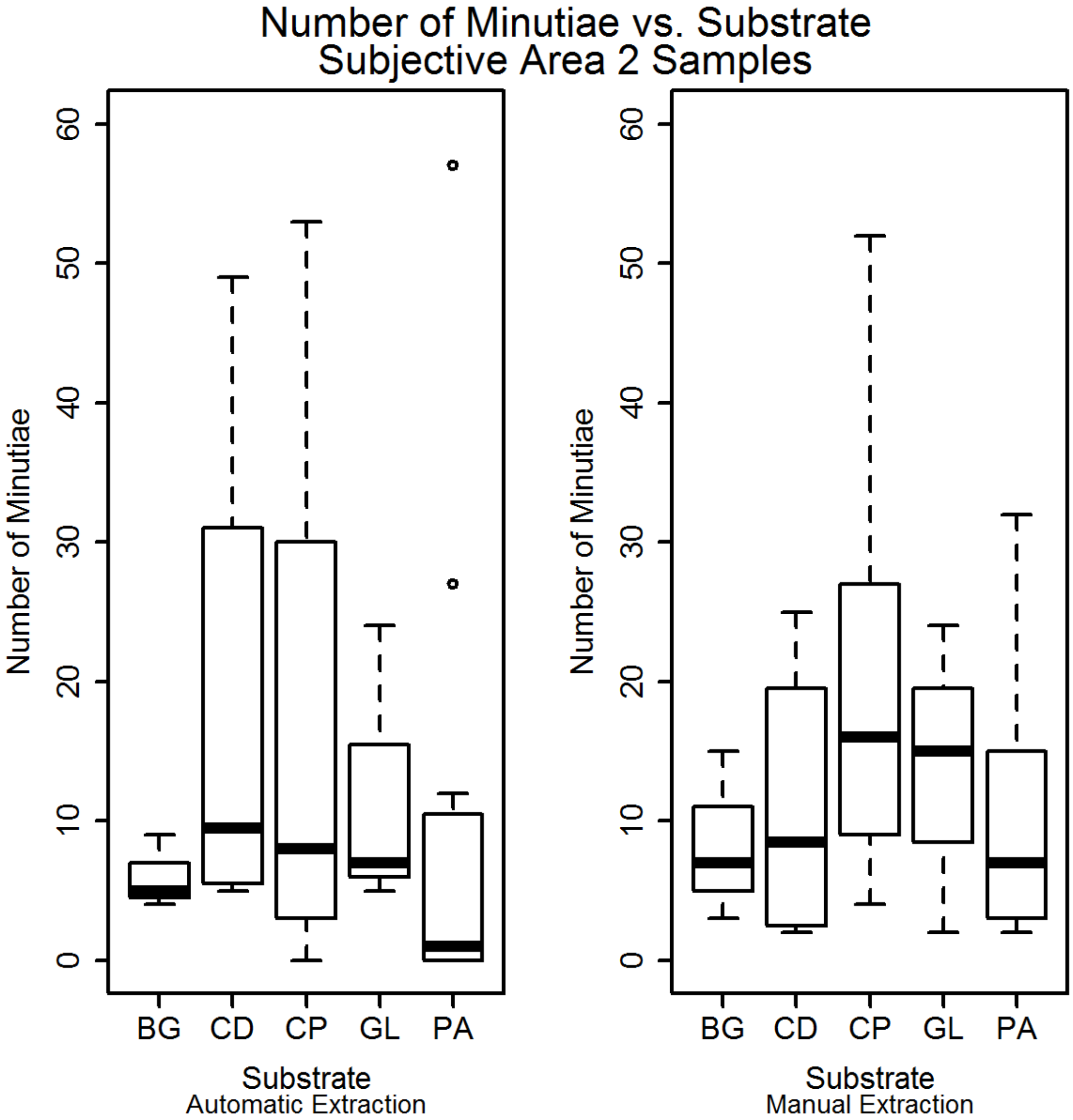


Figure D.23: This figure presents the box plots for the minutiae detected for each substrate type for the subjective Area 2 images. On the left is the automatic extraction box plots while on the right is the manual extraction. The substrate types are represented by the two letters. BG represents bag, CD represents a CD, CP represents ceramic plate, GL represents glass and PA represents paper. The thick black line represents the median in the samples.

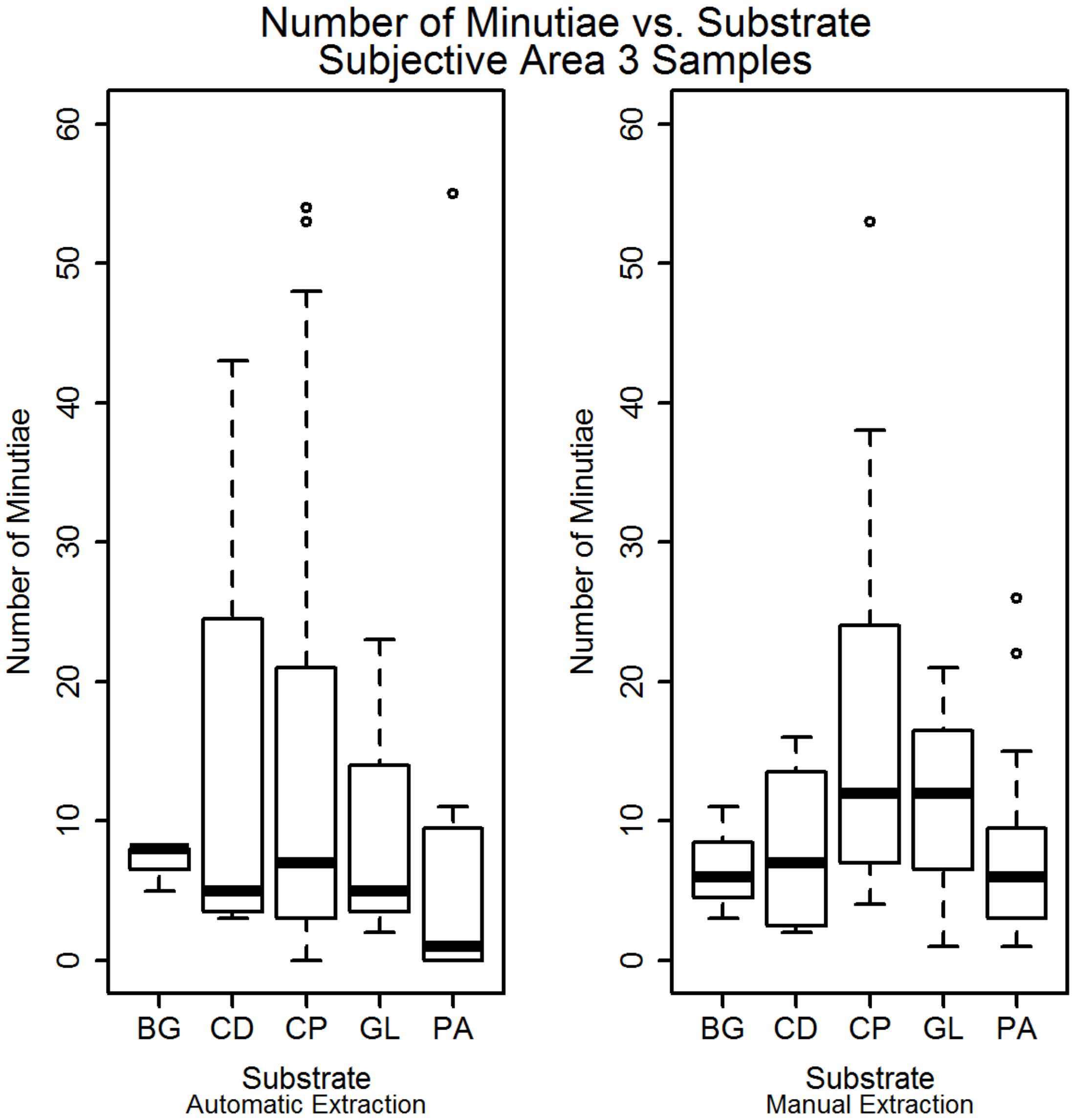


Figure D.24: This figure presents the box plots for the minutiae detected for each substrate type for the subjective Area 3 images. On the left is the automatic extraction box plots while on the right is the manual extraction. The substrate types are represented by the two letters. BG represents bag, CD represents a CD, CP represents ceramic plate, GL represents glass and PA represents paper. The thick black line represents the median in the samples.

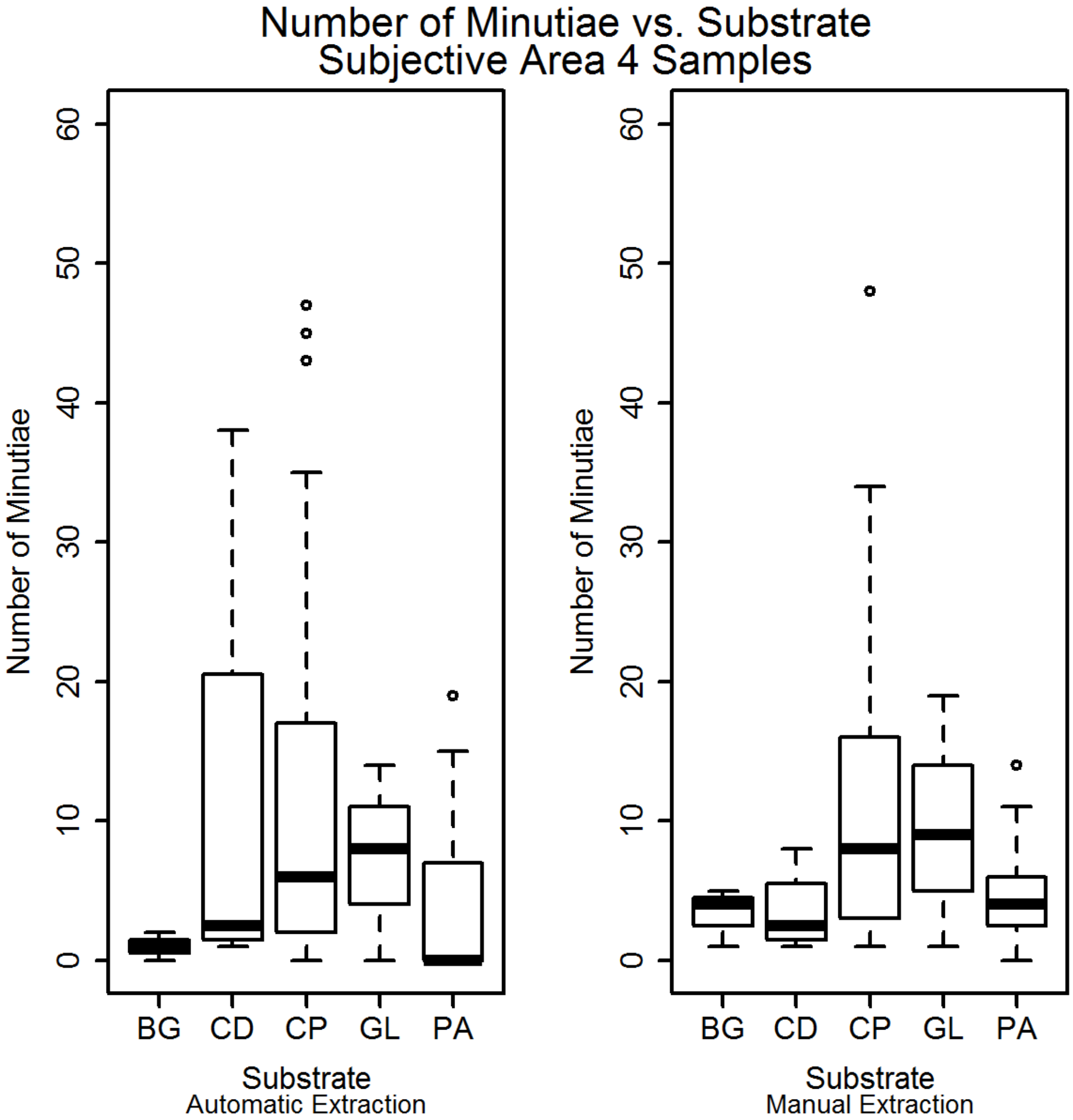


Figure D.25: This figure presents the box plots for the minutiae detected for each substrate type for the subjective Area 4 images. On the left is the automatic extraction box plots while on the right is the manual extraction. The substrate types are represented by the two letters. BG represents bag, CD represents a CD, CP represents ceramic plate, GL represents glass and PA represents paper. The thick black line represents the median in the samples.

D.3.2 Objective Method

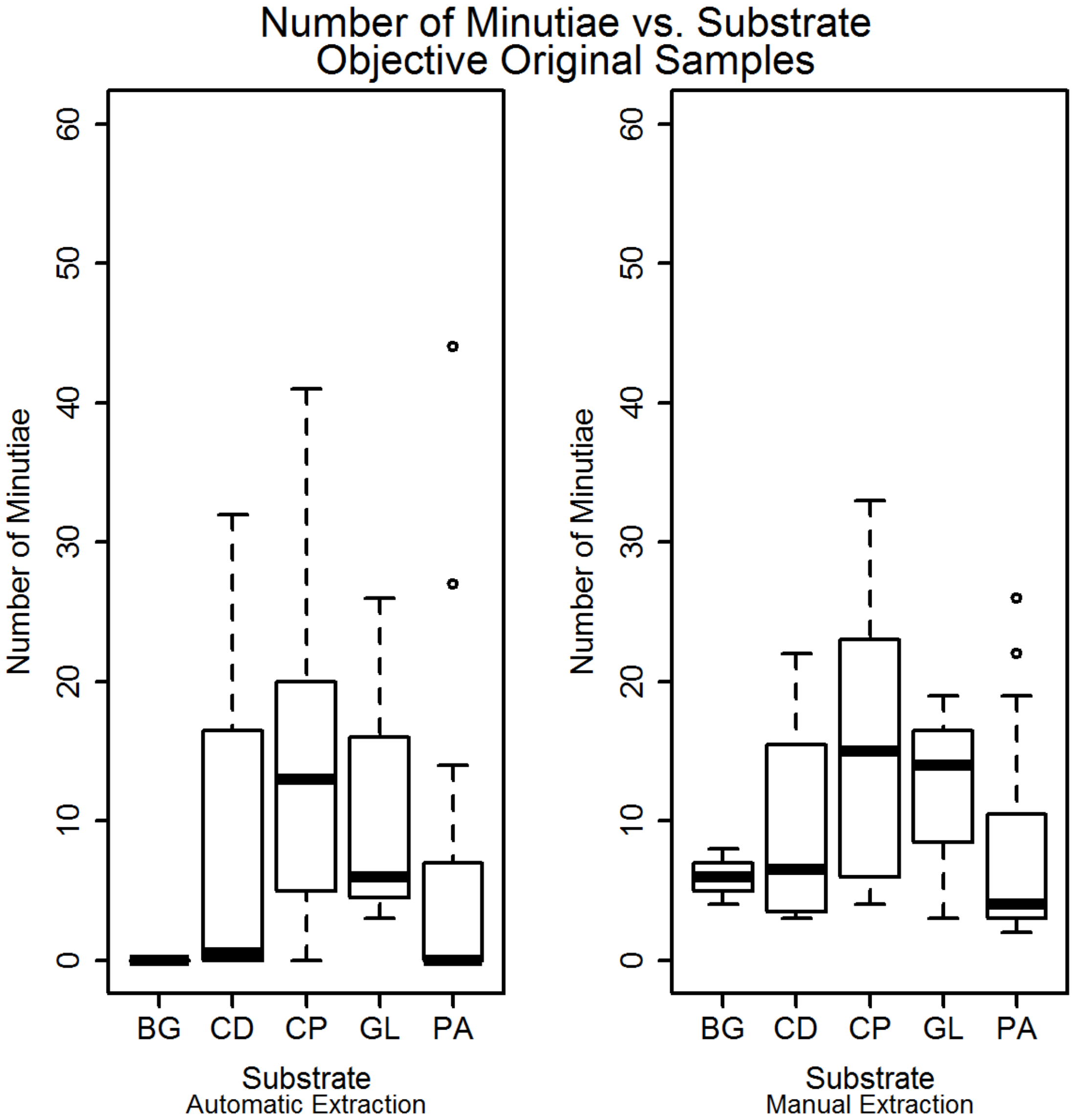


Figure D.26: This figure presents the box plots for the minutiae detected for each substrate type for the objective original images. On the left is the automatic extraction box plots while on the right is the manual extraction. The substrate types are represented by the two letters. BG represents bag, CD represents a CD, CP represents ceramic plate, GL represents glass and PA represents paper. The thick black line represents the median in the samples.

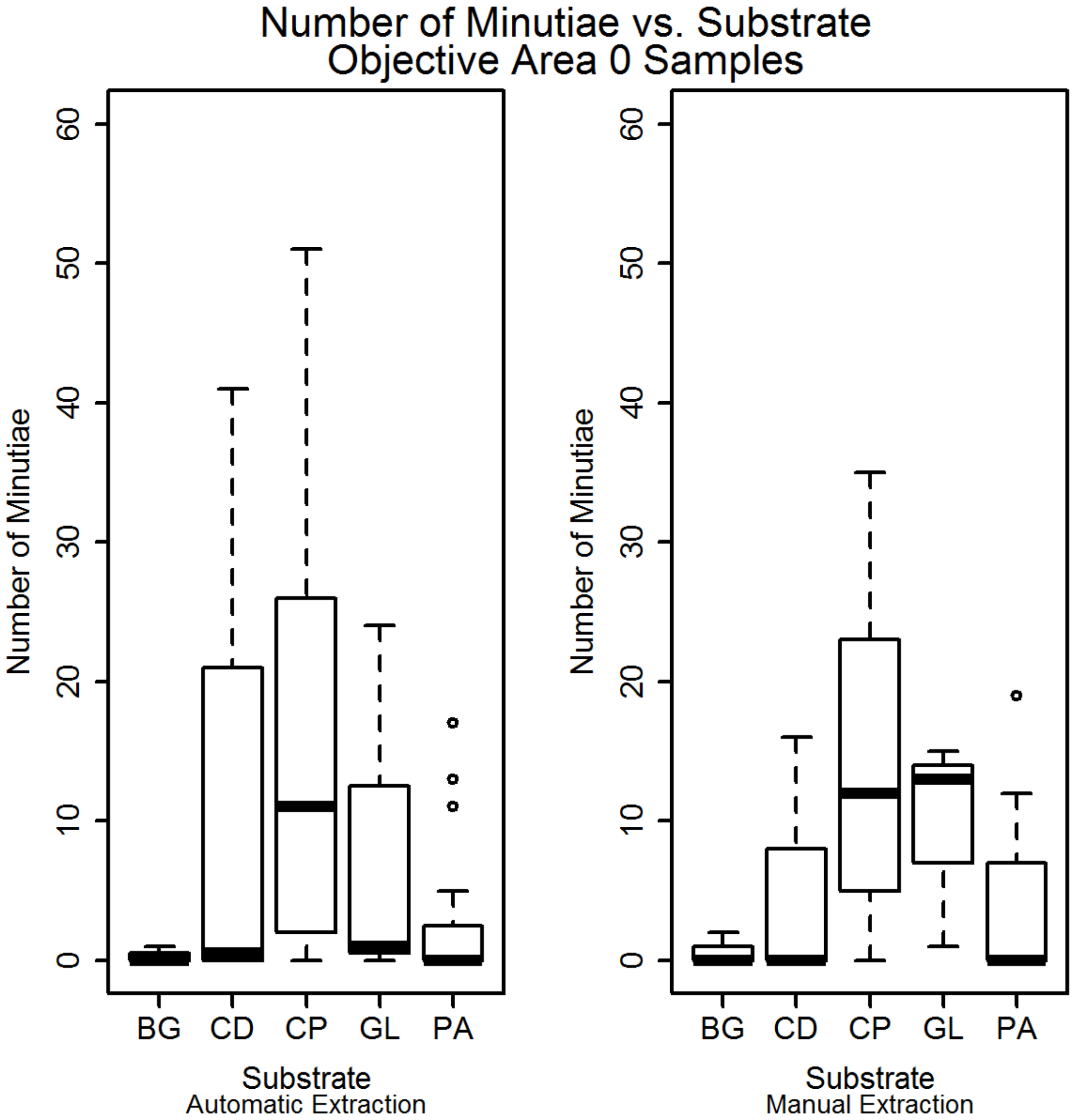


Figure D.27: This figure presents the box plots for the minutiae detected for each substrate type for the objective Area 0 images. On the left is the automatic extraction box plots while on the right is the manual extraction. The substrate types are represented by the two letters. BG represents bag, CD represents a CD, CP represents ceramic plate, GL represents glass and PA represents paper. The thick black line represents the median in the samples.

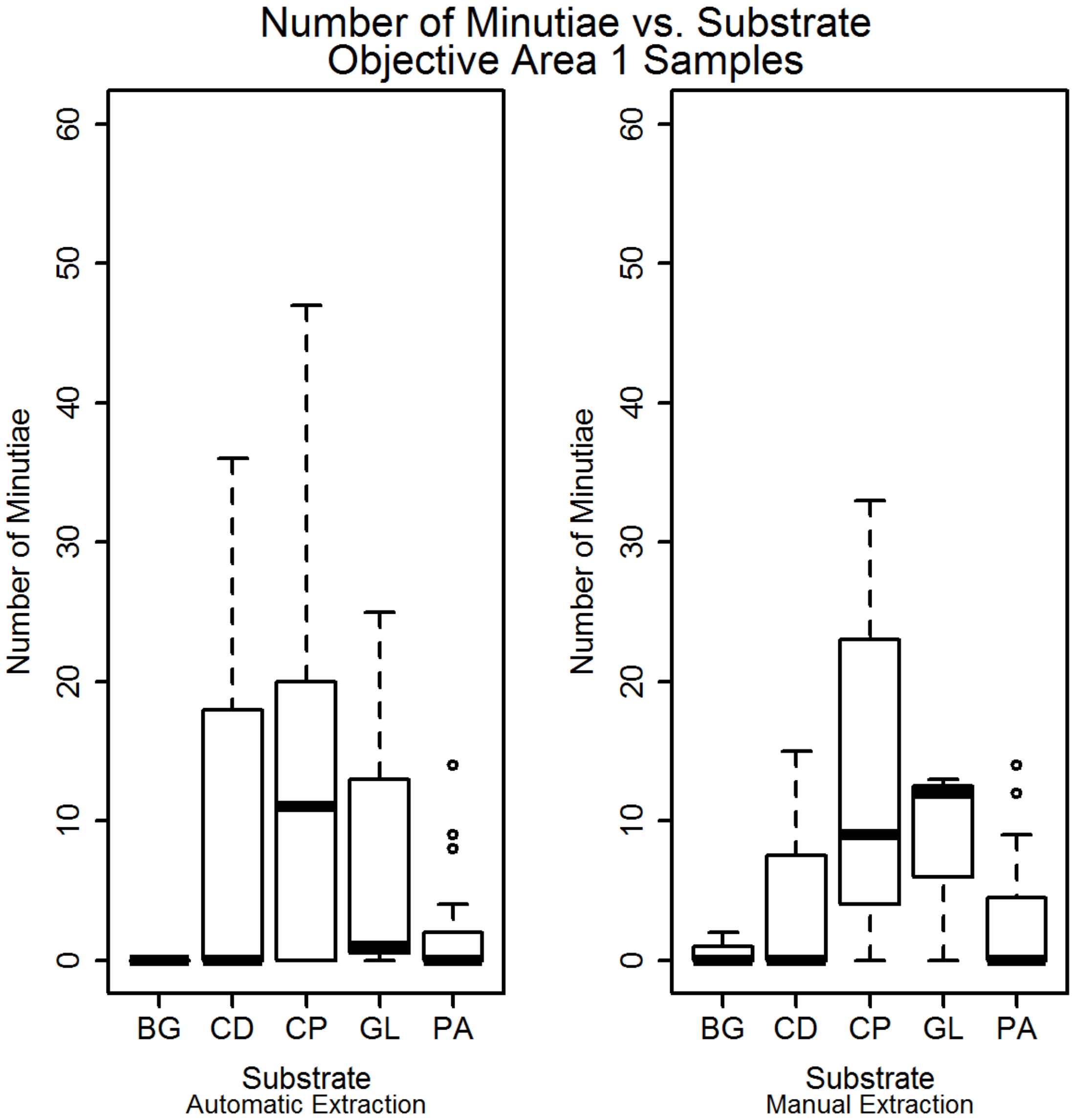


Figure D.28: This figure presents the box plots for the minutiae detected for each substrate type for the objective Area 1 images. On the left is the automatic extraction box plots while on the right is the manual extraction. The substrate types are represented by the two letters. BG represents bag, CD represents a CD, CP represents ceramic plate, GL represents glass and PA represents paper. The thick black line represents the median in the samples.

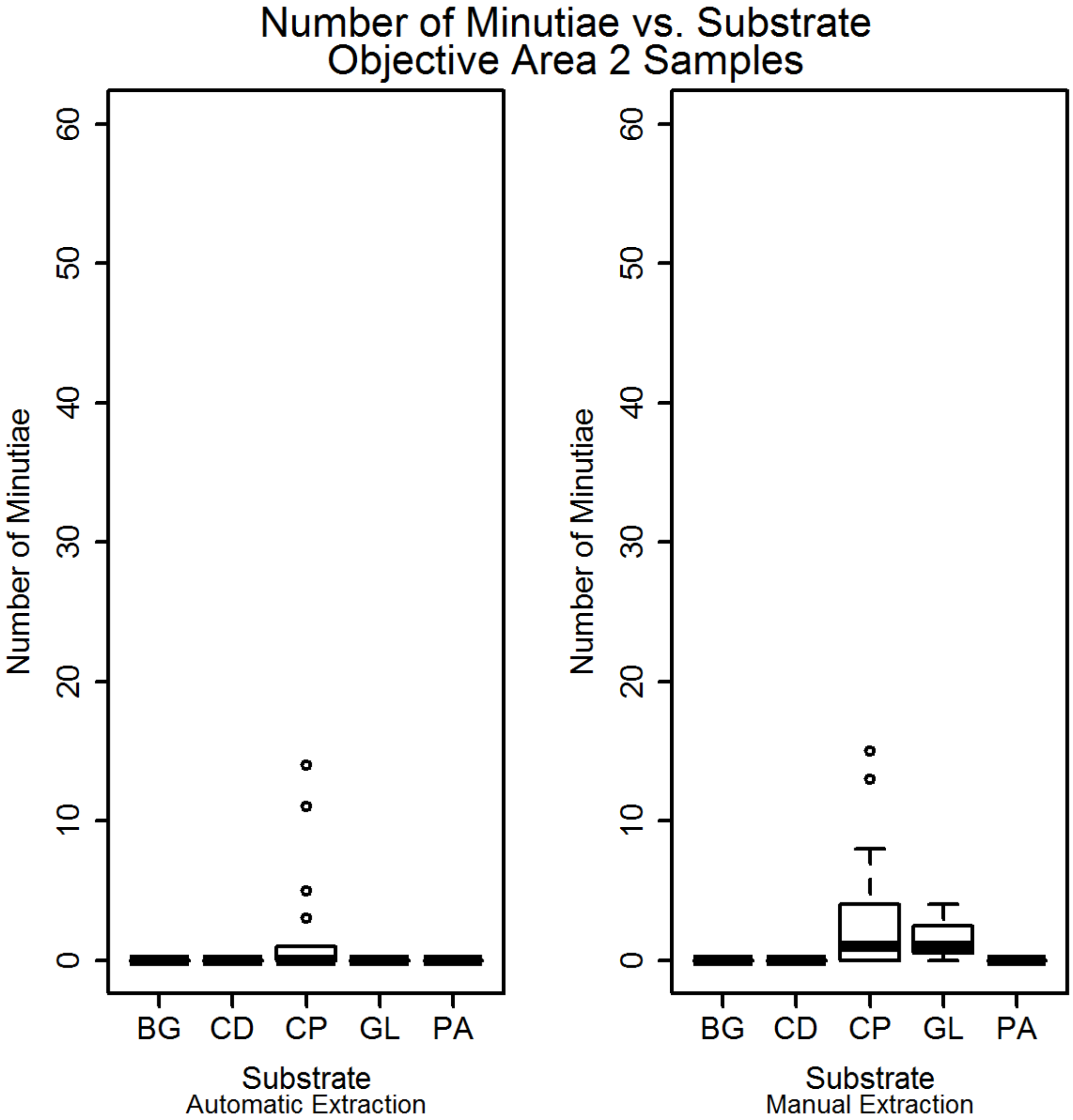


Figure D.29: This figure presents the box plots for the minutiae detected for each substrate type for the objective Area 2 images. On the left is the automatic extraction box plots while on the right is the manual extraction. The substrate types are represented by the two letters. BG represents bag, CD represents a CD, CP represents ceramic plate, GL represents glass and PA represents paper. The thick black line represents the median in the samples.

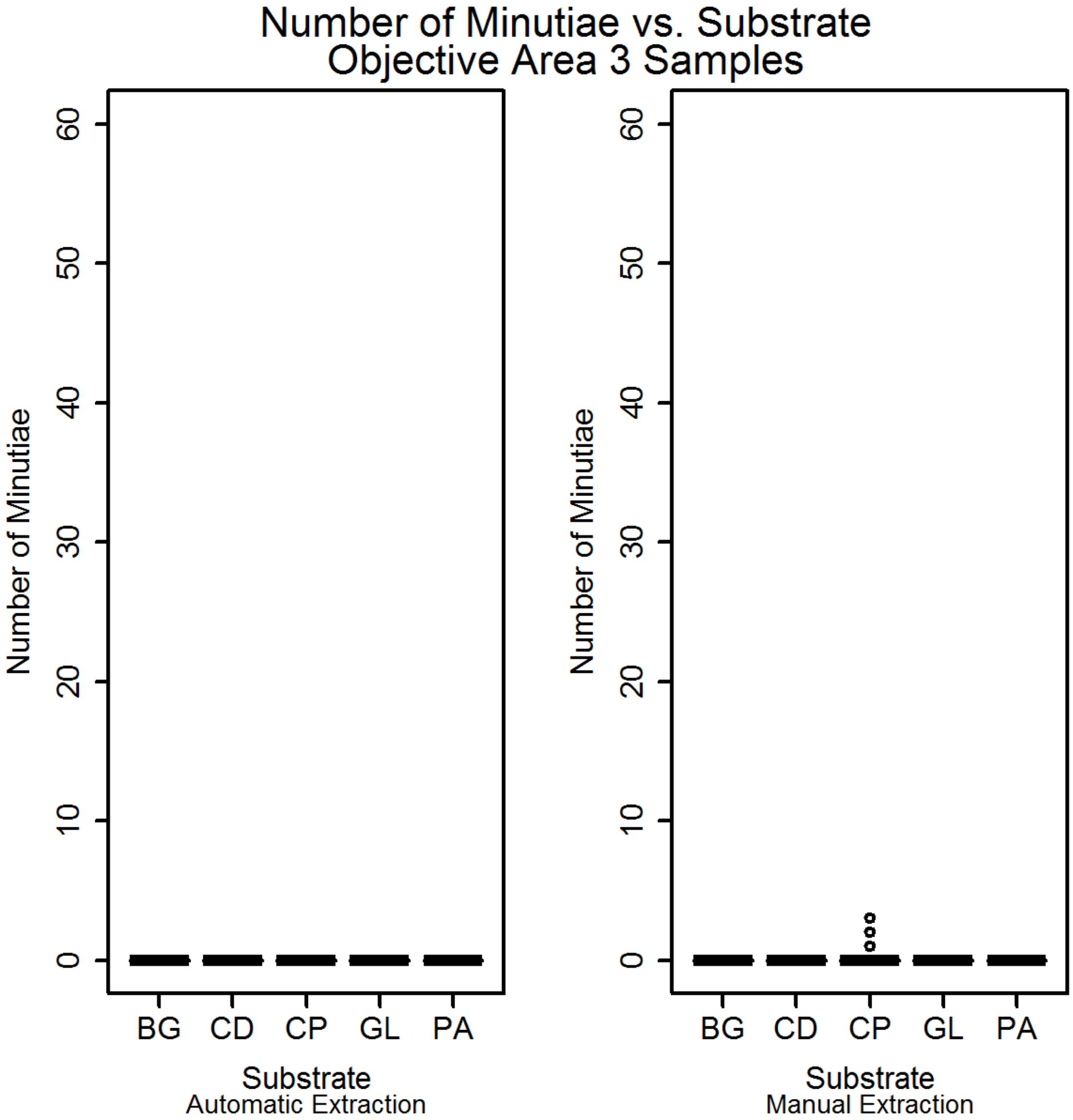


Figure D.30: This figure presents the box plots for the minutiae detected for each substrate type for the objective Area 3 images. On the left is the automatic extraction box plots while on the right is the manual extraction. The substrate types are represented by the two letters. BG represents bag, CD represents a CD, CP represents ceramic plate, GL represents glass and PA represents paper. The thick black line represents the median in the samples.

D.4 Match Score Scatter Plots

D.4.1 Subjective Method

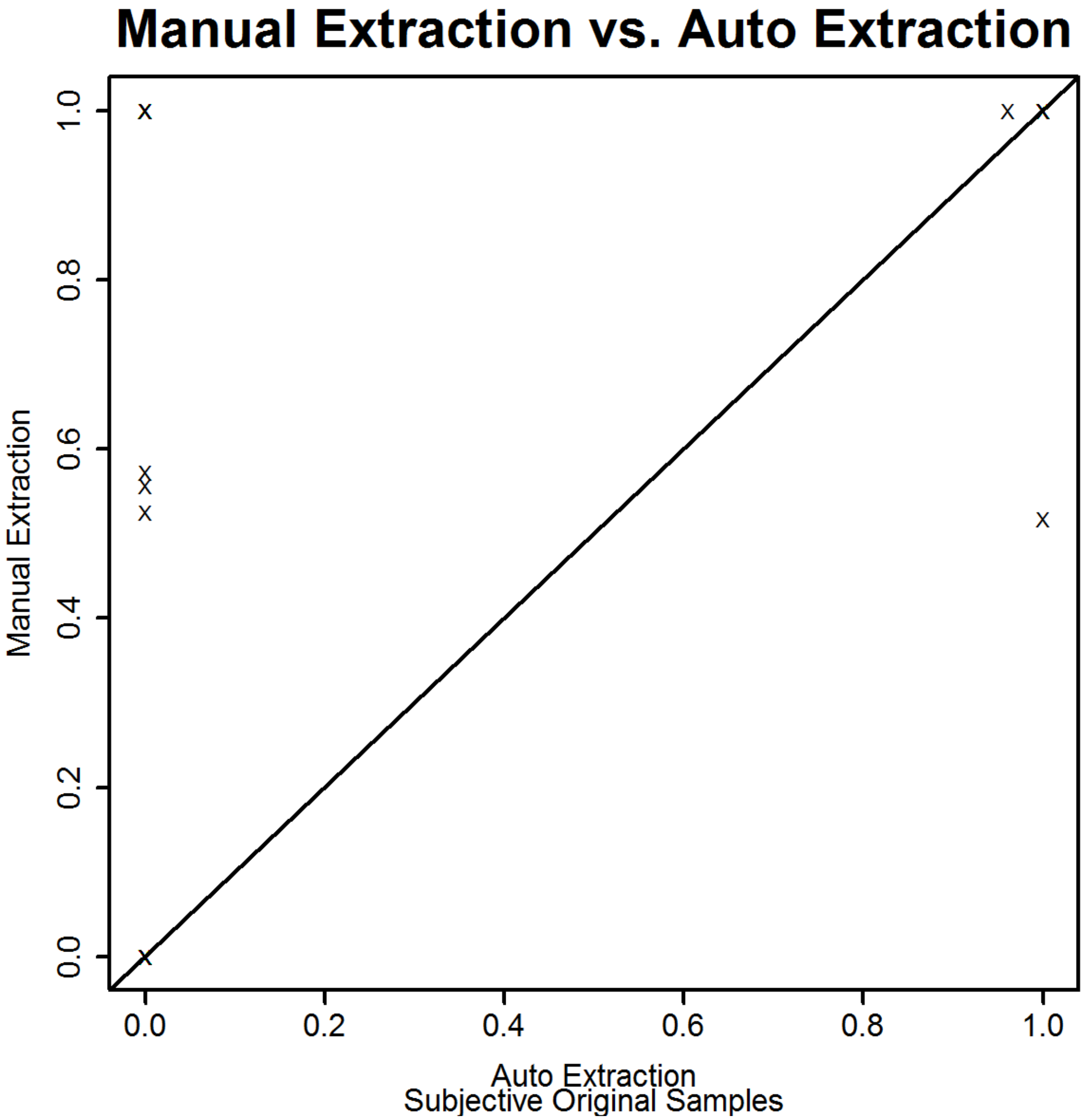


Figure D.31: The scatter plot for the manual minutiae extraction versus the automatic minutiae extraction for subjective original images for the match scores.

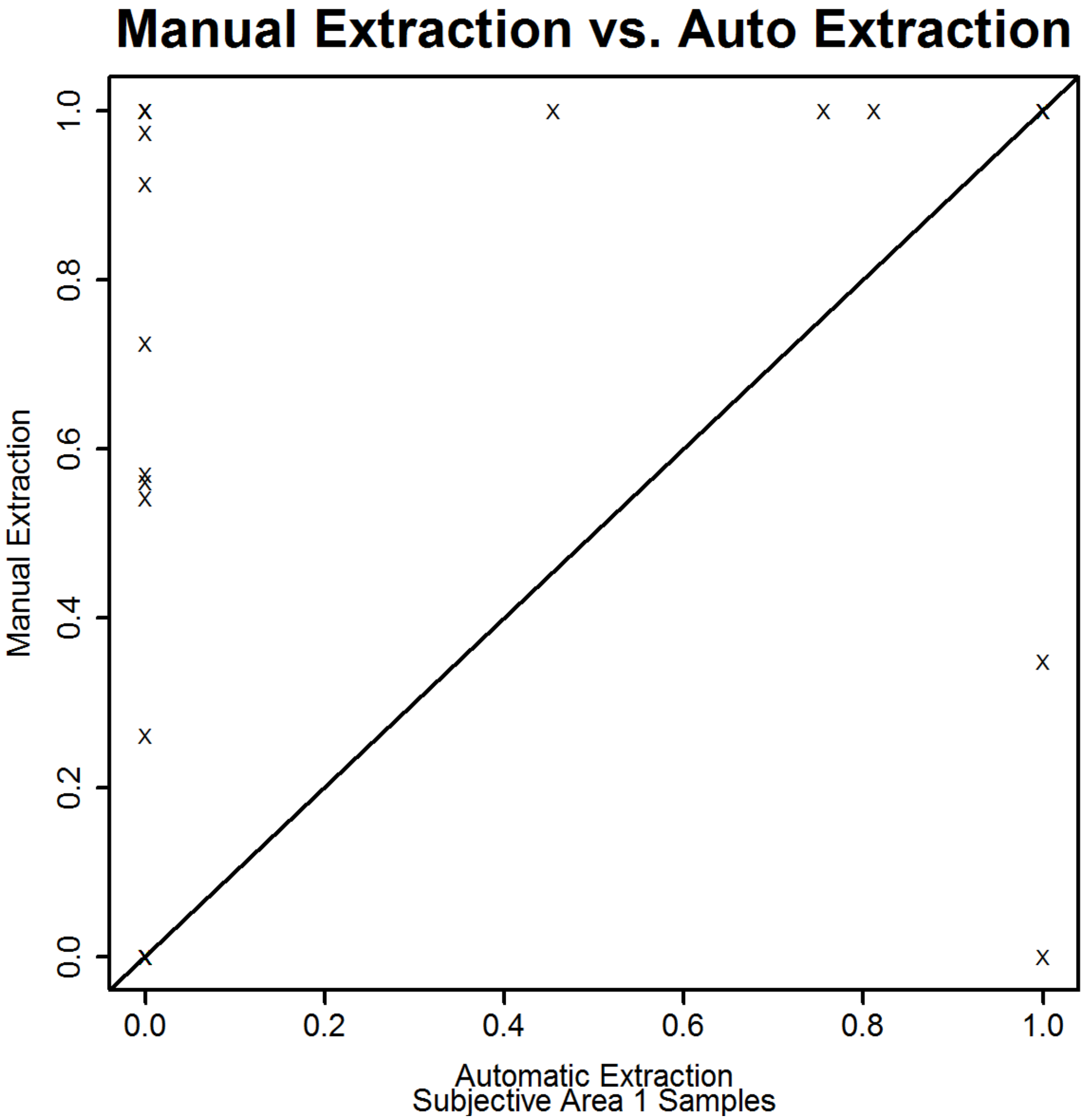


Figure D.32: The scatter plot for the manual minutiae extraction versus the automatic minutiae extraction for subjective Area 1 for the match scores.

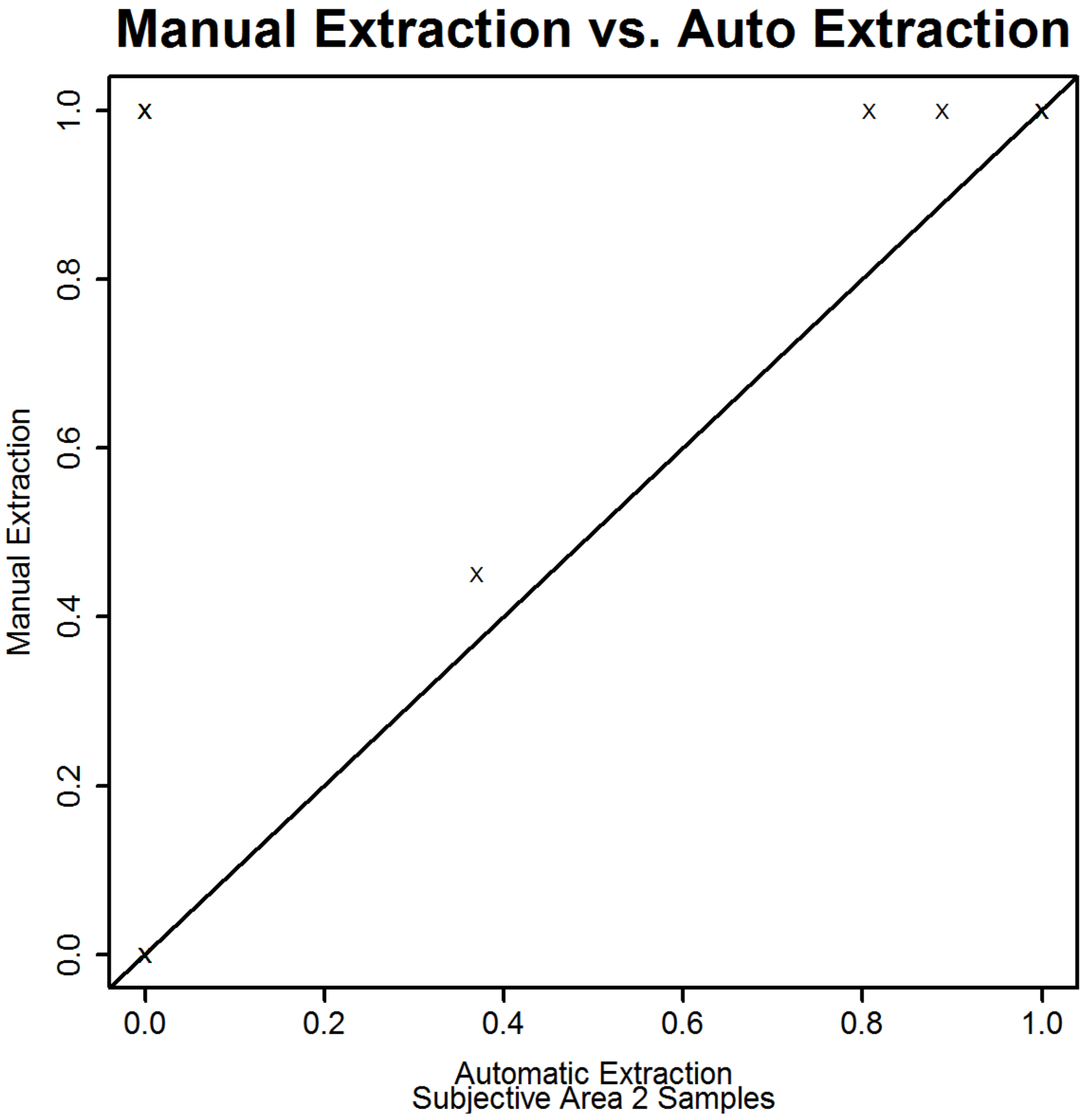


Figure D.33: The scatter plot for the manual minutiae extraction versus the automatic minutiae extraction for subjective Area 2 for the match scores.

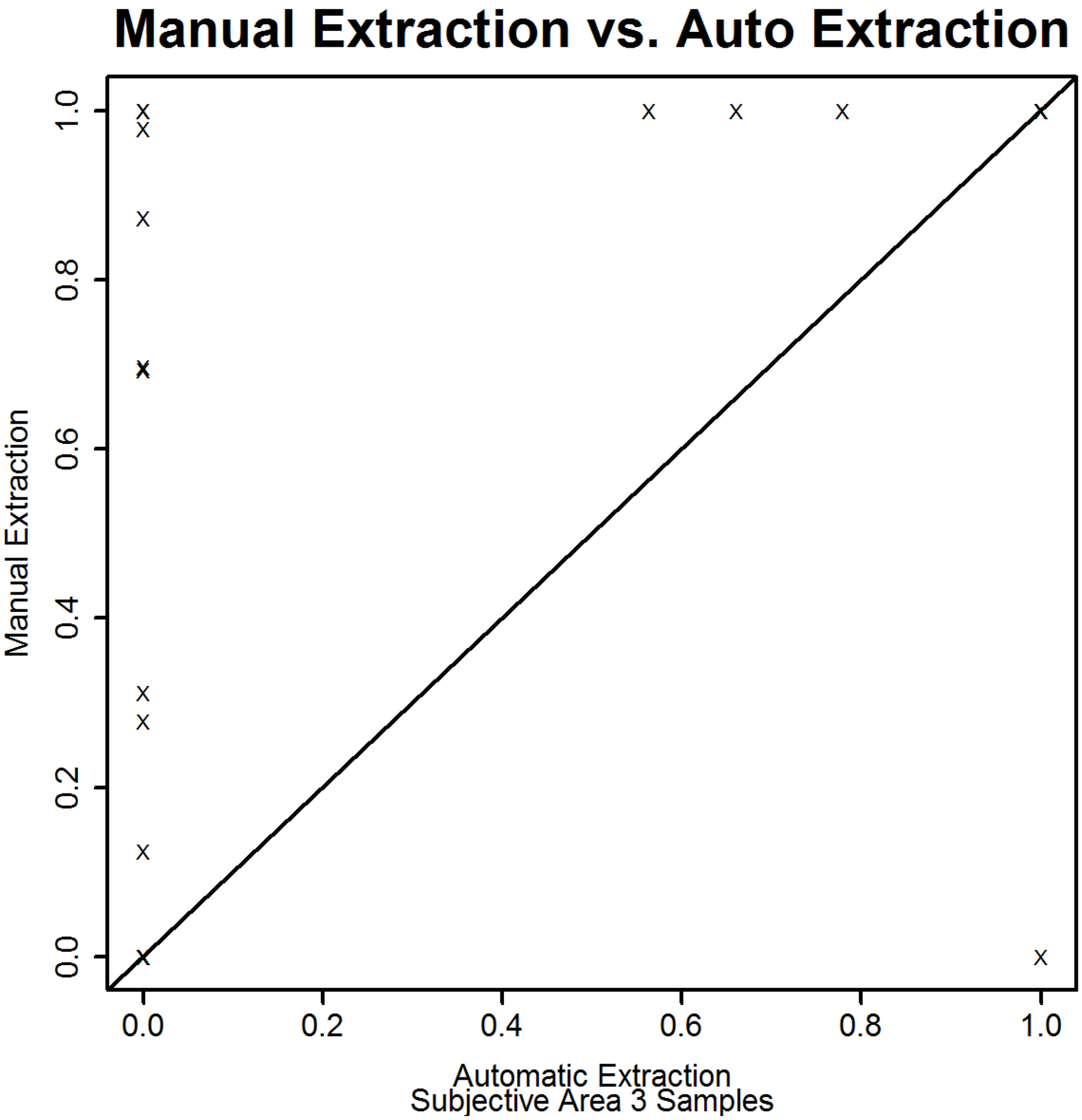


Figure D.34: The scatter plot for the manual minutiae extraction versus the automatic minutiae extraction for subjective Area 3 for the match scores.

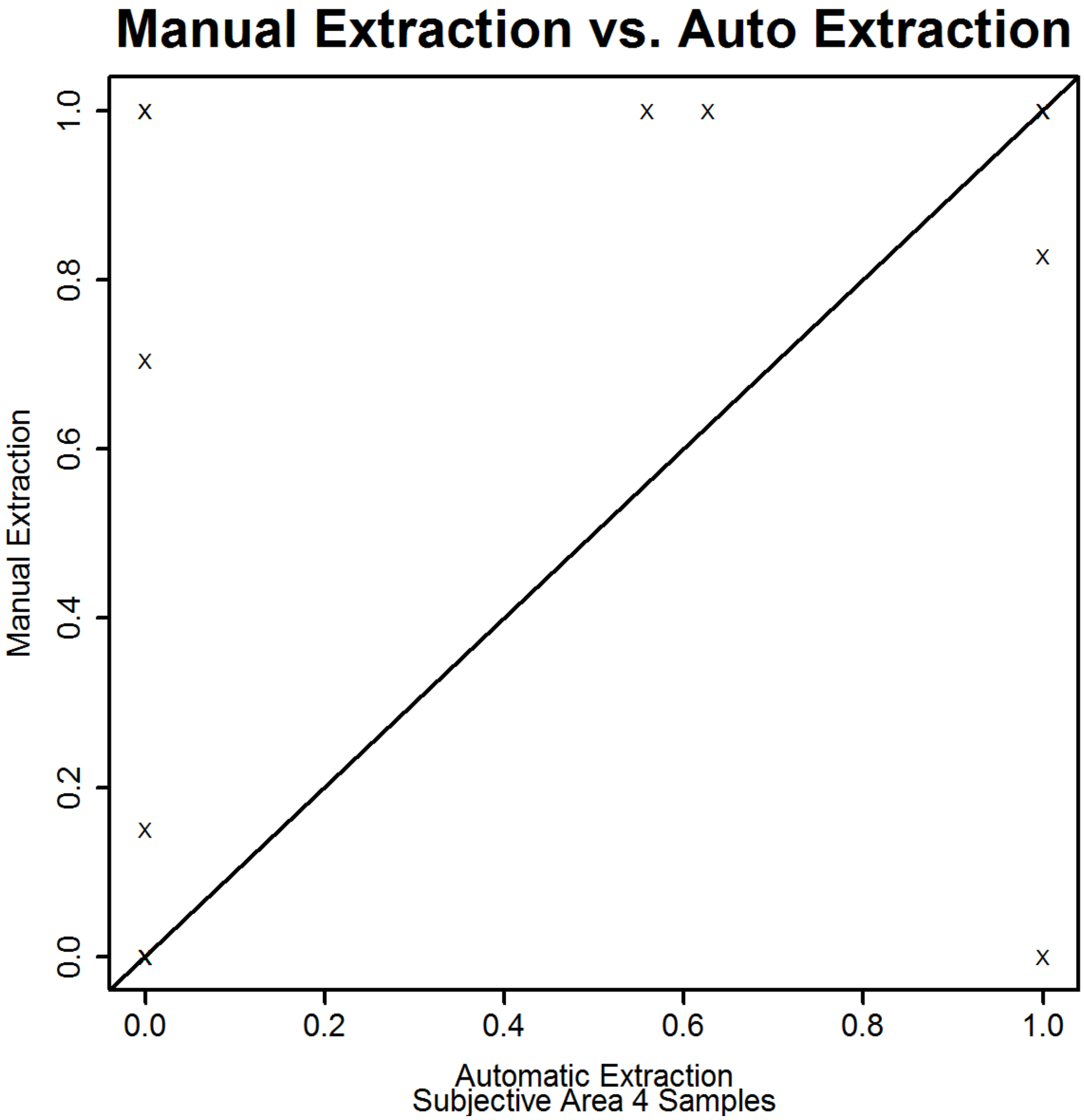


Figure D.35: The scatter plot for the manual minutiae extraction versus the automatic minutiae extraction for subjective Area 4 for the match scores.

D.4.2 Objective Method

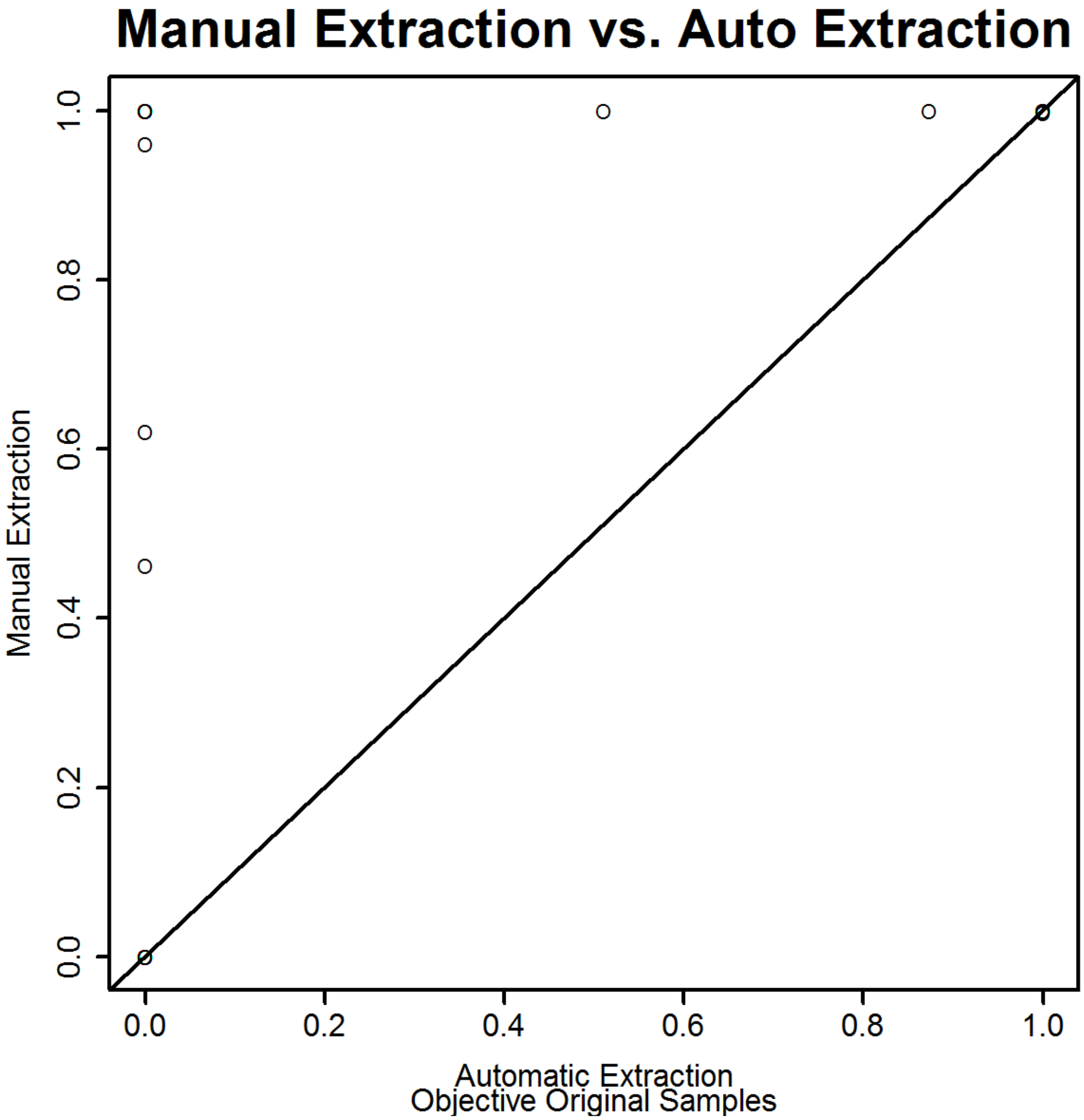


Figure D.36: The scatter plot for the manual minutiae extraction versus the automatic minutiae extraction for objective Area 1 for the match scores.

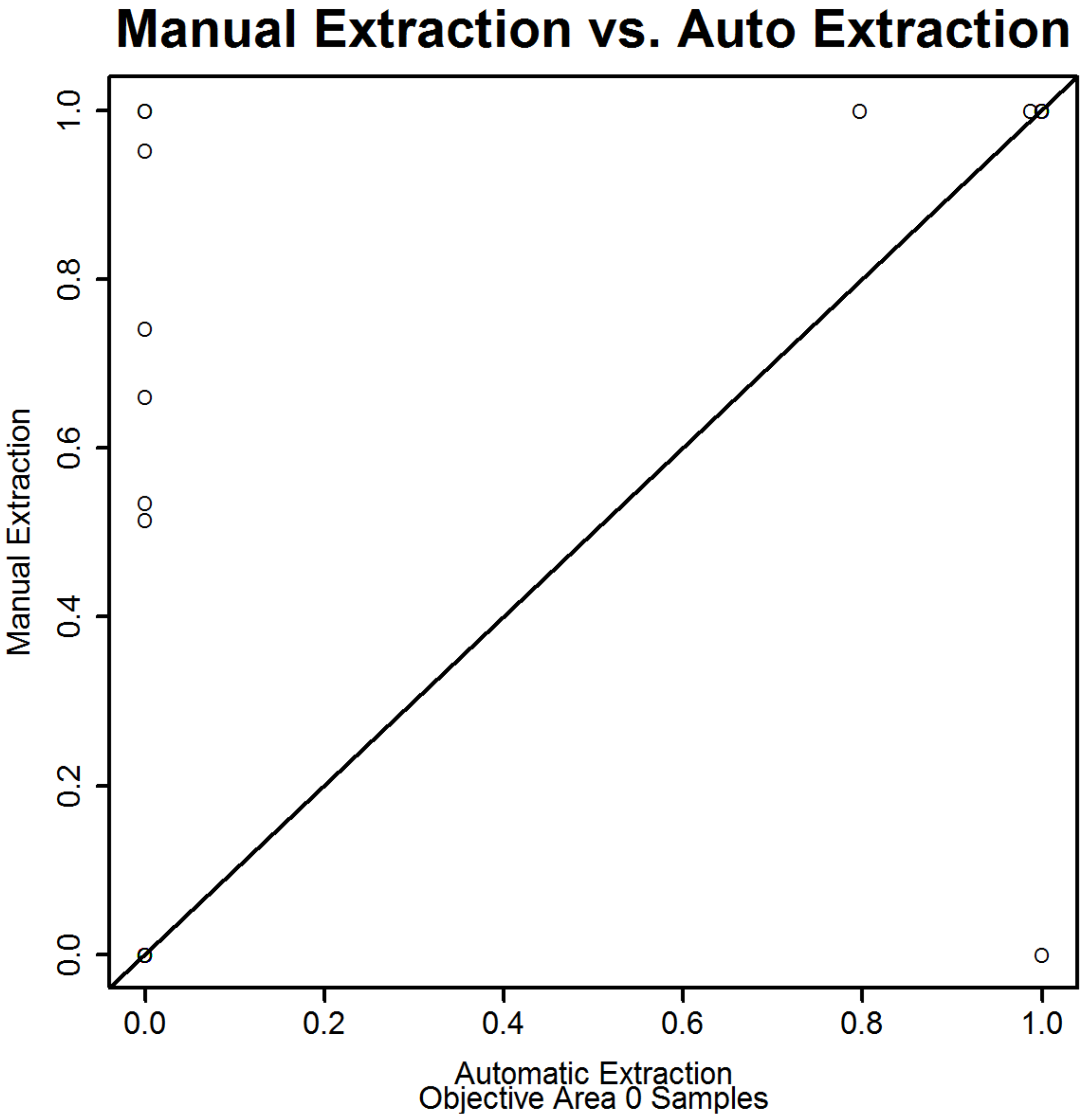


Figure D.37: The scatter plot for the manual minutiae extraction versus the automatic minutiae extraction for objective Area 2 for the match scores.

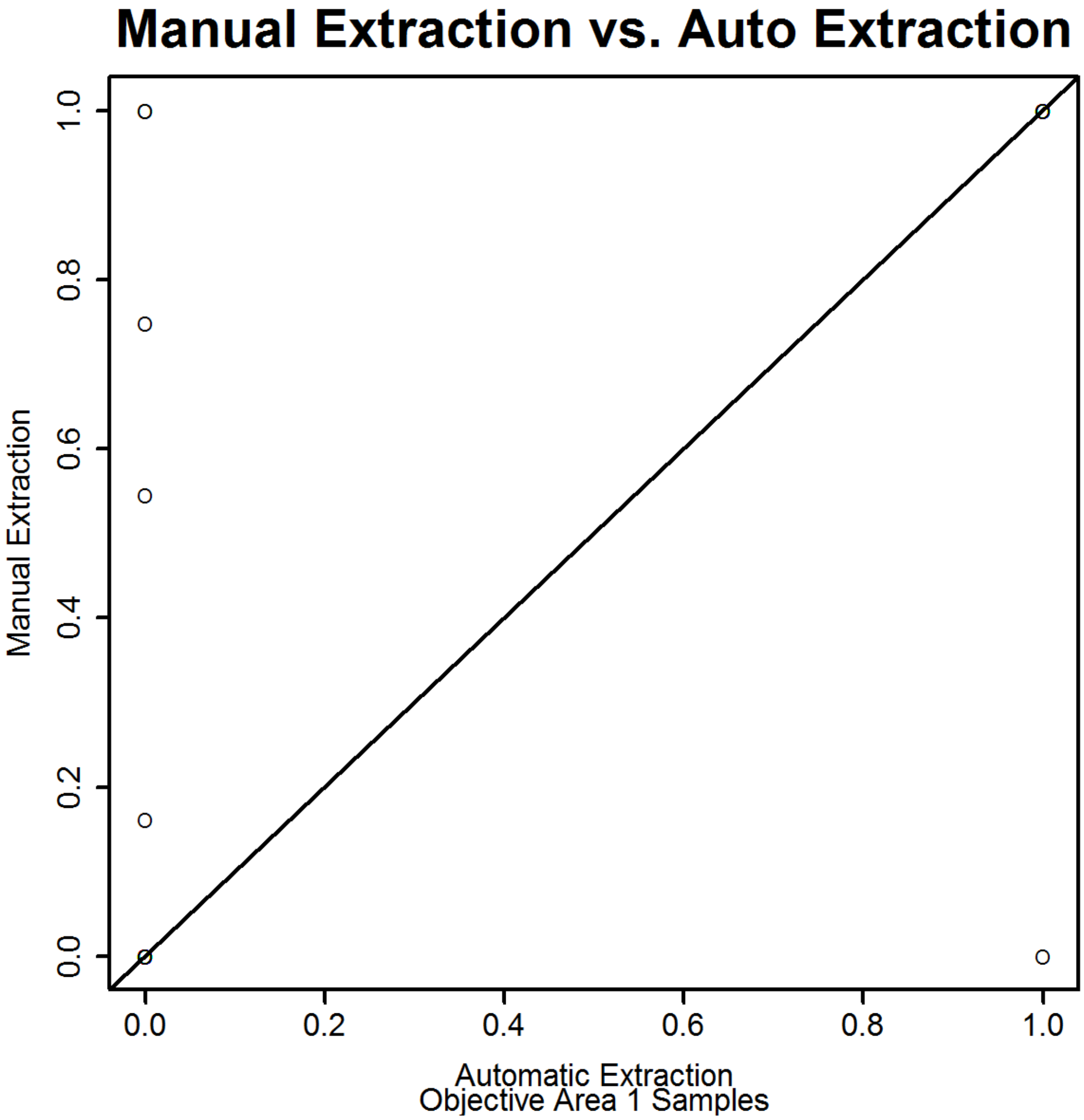


Figure D.38: The scatter plot for the manual minutiae extraction versus the automatic minutiae extraction for objective Area 1 for the match scores.

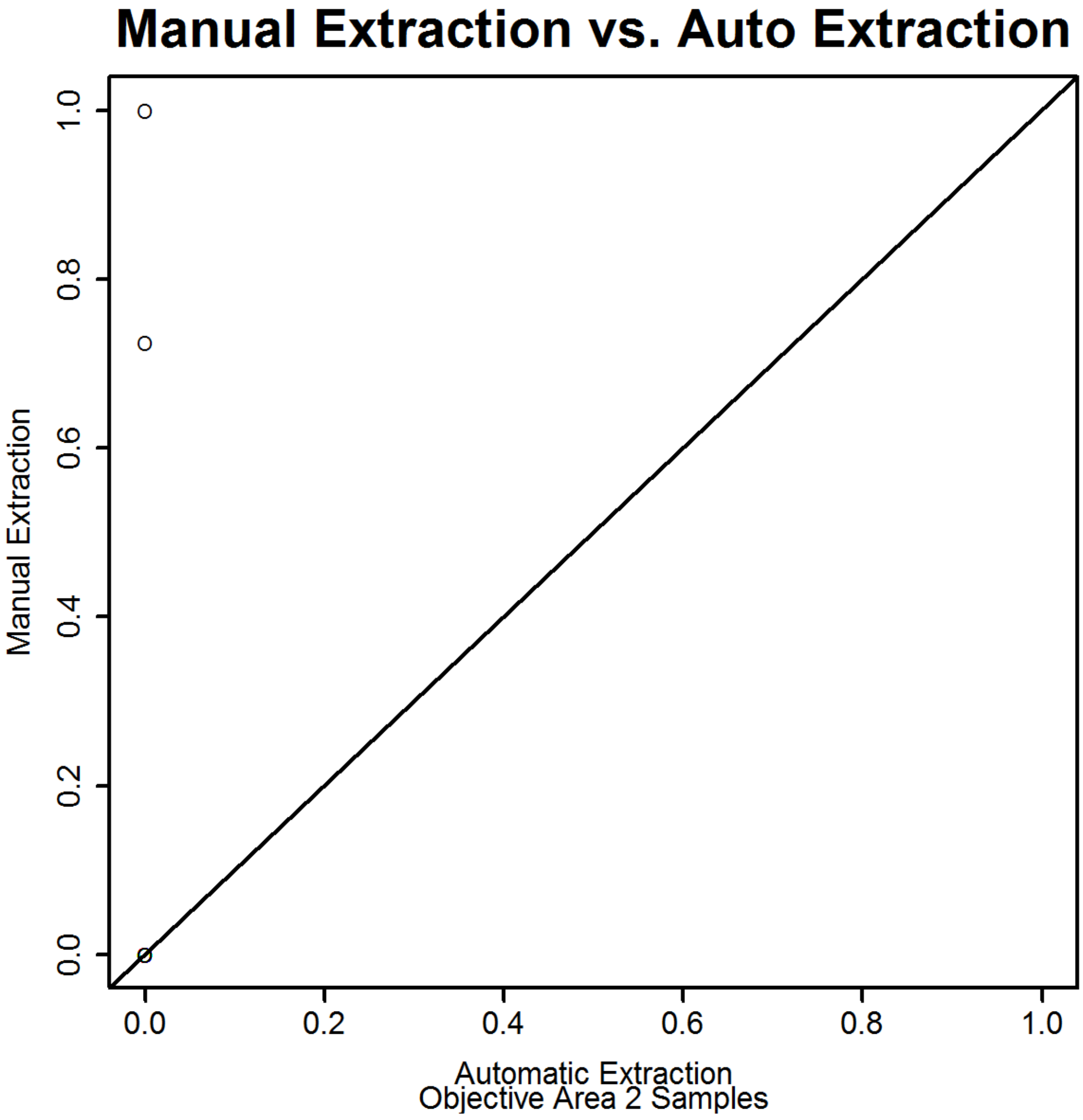


Figure D.39: The scatter plot for the manual minutiae extraction versus the automatic minutiae extraction for objective Area 2 for the match scores.

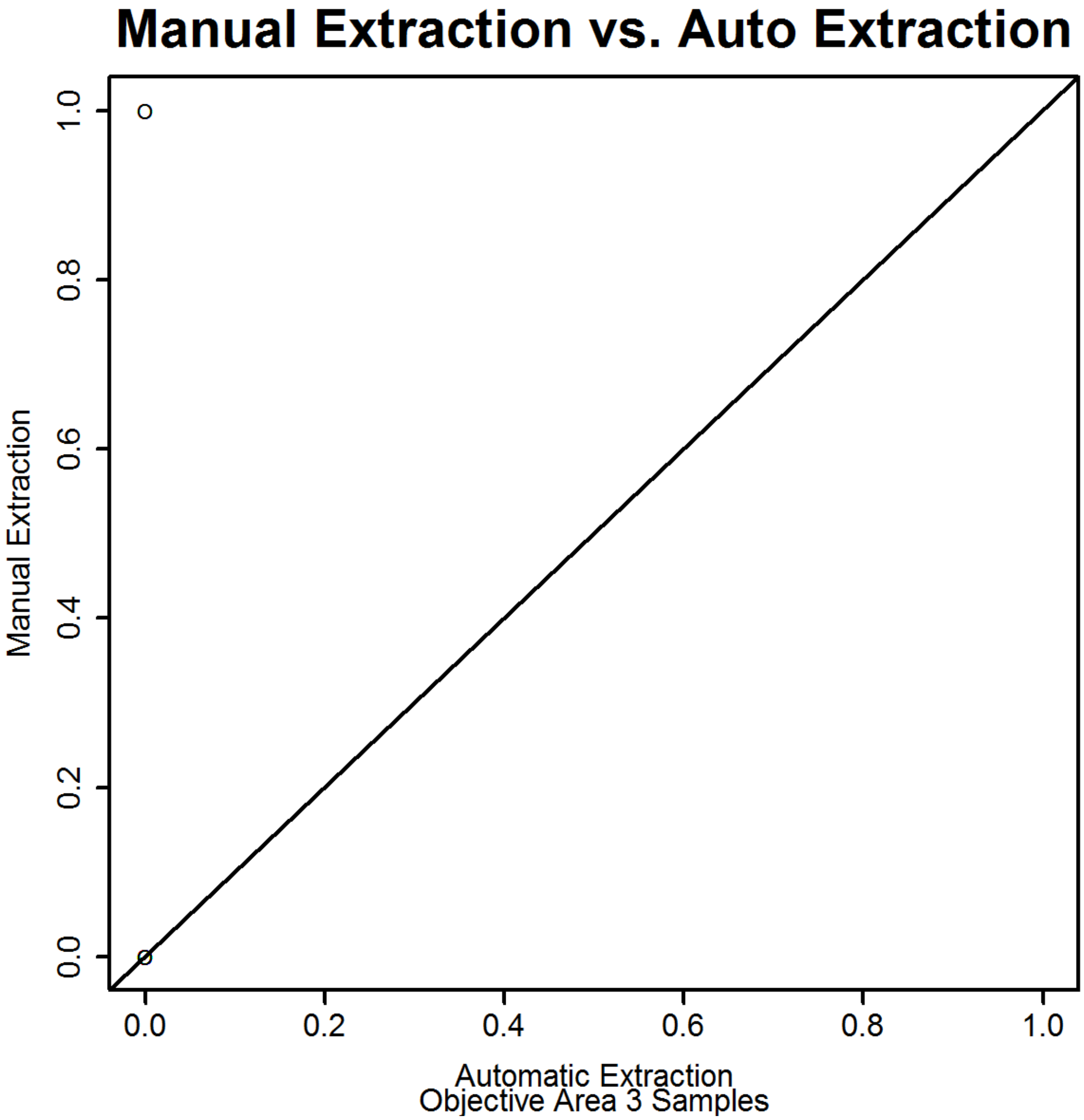


Figure D.40: The scatter plot for the manual minutiae extraction versus the automatic minutiae extraction for objective Area 3 for the match scores.

D.5 Match Score Development Plots

D.5.1 Subjective Method

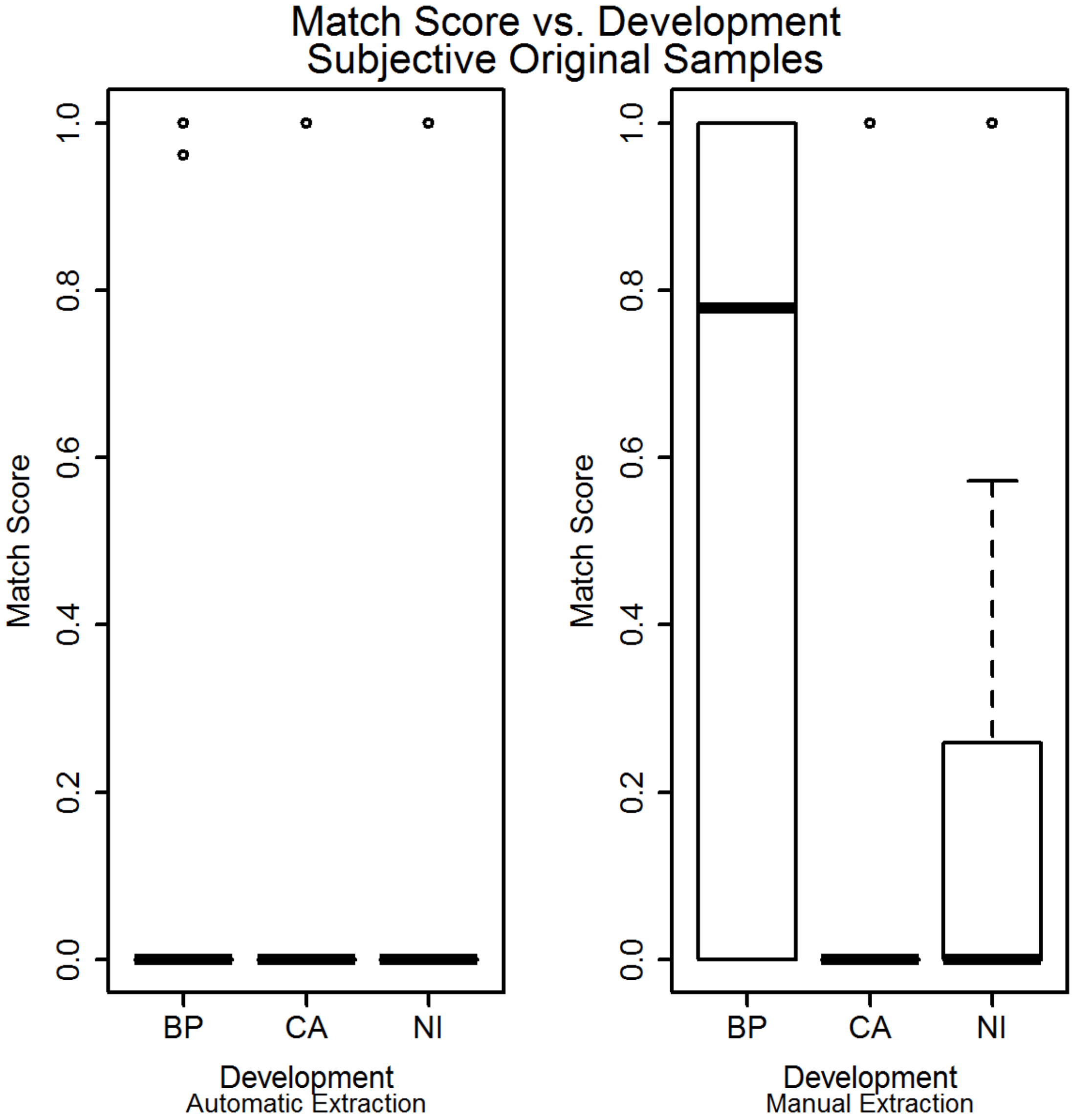


Figure D.41: This figure presents the box plots for the match scores for each development type in the subjective original images. On the left is the automatic extraction box plots while on the right is the manual extraction. The development methods are represented by the two letters. BP represents black powder, CA represents cyanoacrylate and NI represents ninhydrin. The thick black line represents the median in the samples.

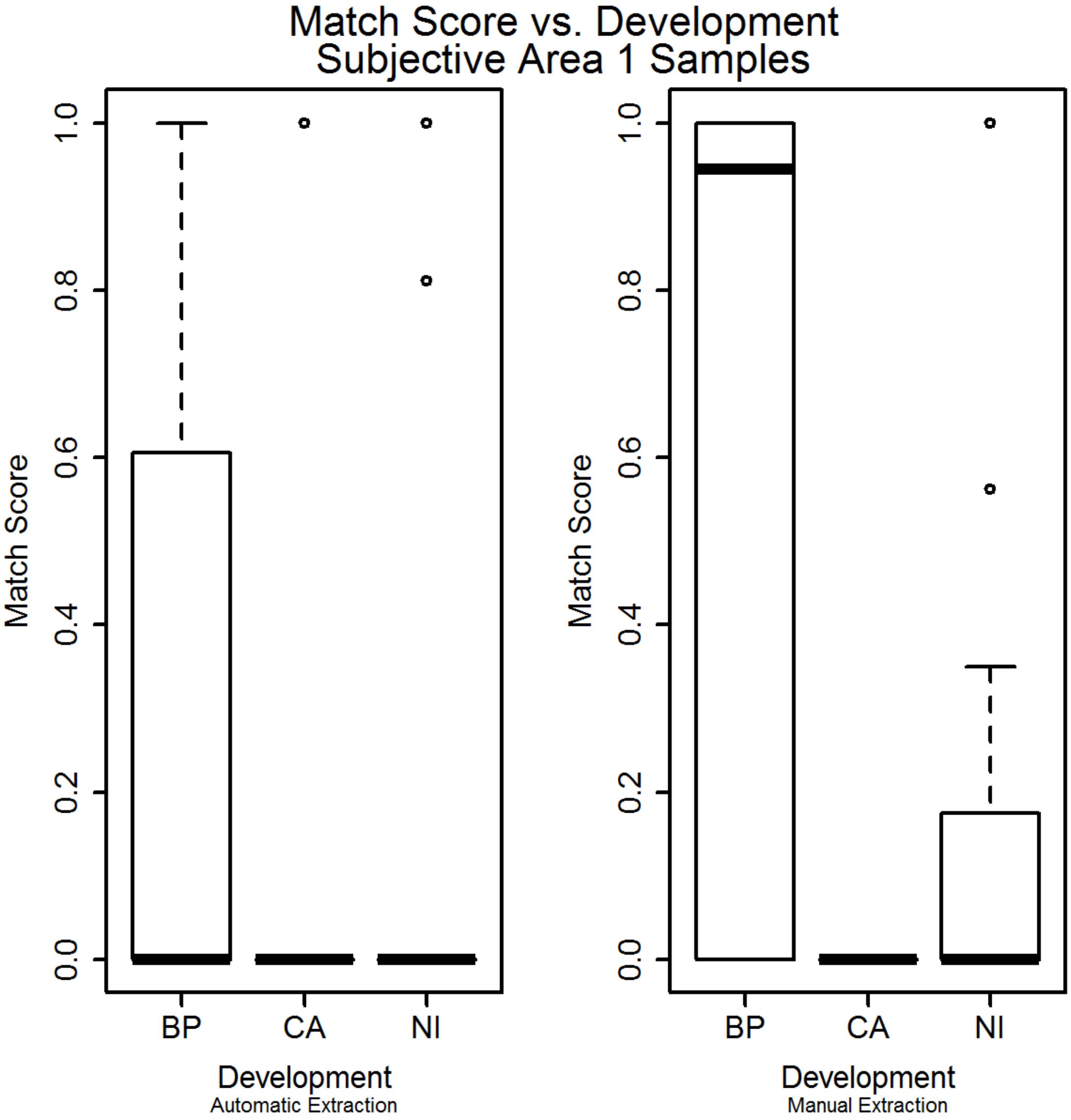


Figure D.42: This figure presents the box plots for the match scores for each development type in the subjective Area 1 images. On the left is the automatic extraction box plots while on the right is the manual extraction. The development methods are represented by the two letters. BP represents black powder, CA represents cyanoacrylate and NI represents ninhydrin. The thick black line represents the median in the samples.

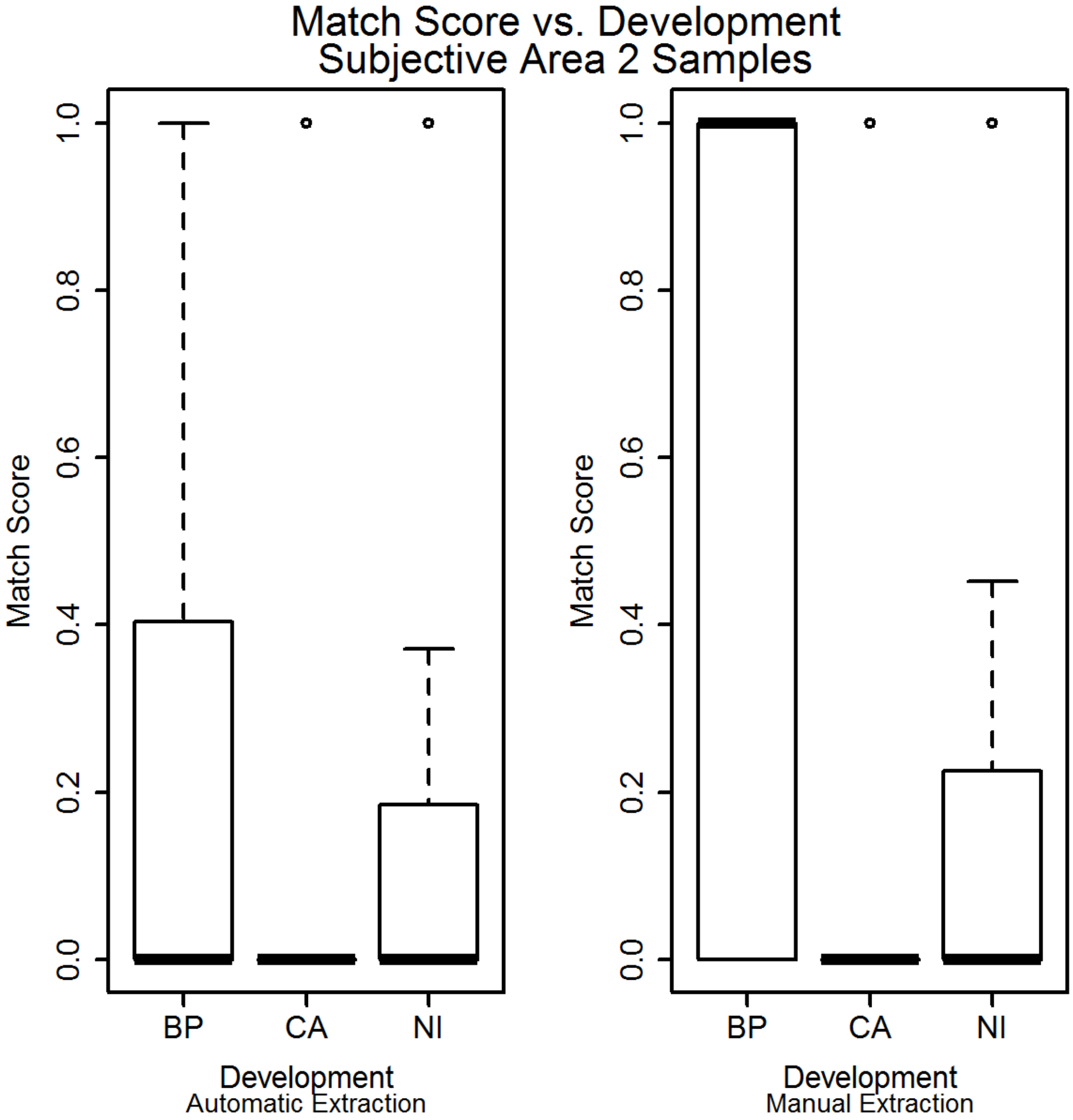


Figure D.43: This figure presents the box plots for the match scores for each development type in the subjective Area 2 images. On the left is the automatic extraction box plots while on the right is the manual extraction. The development methods are represented by the two letters. BP represents black powder, CA represents cyanoacrylate and NI represents ninhydrin. The thick black line represents the median in the samples.

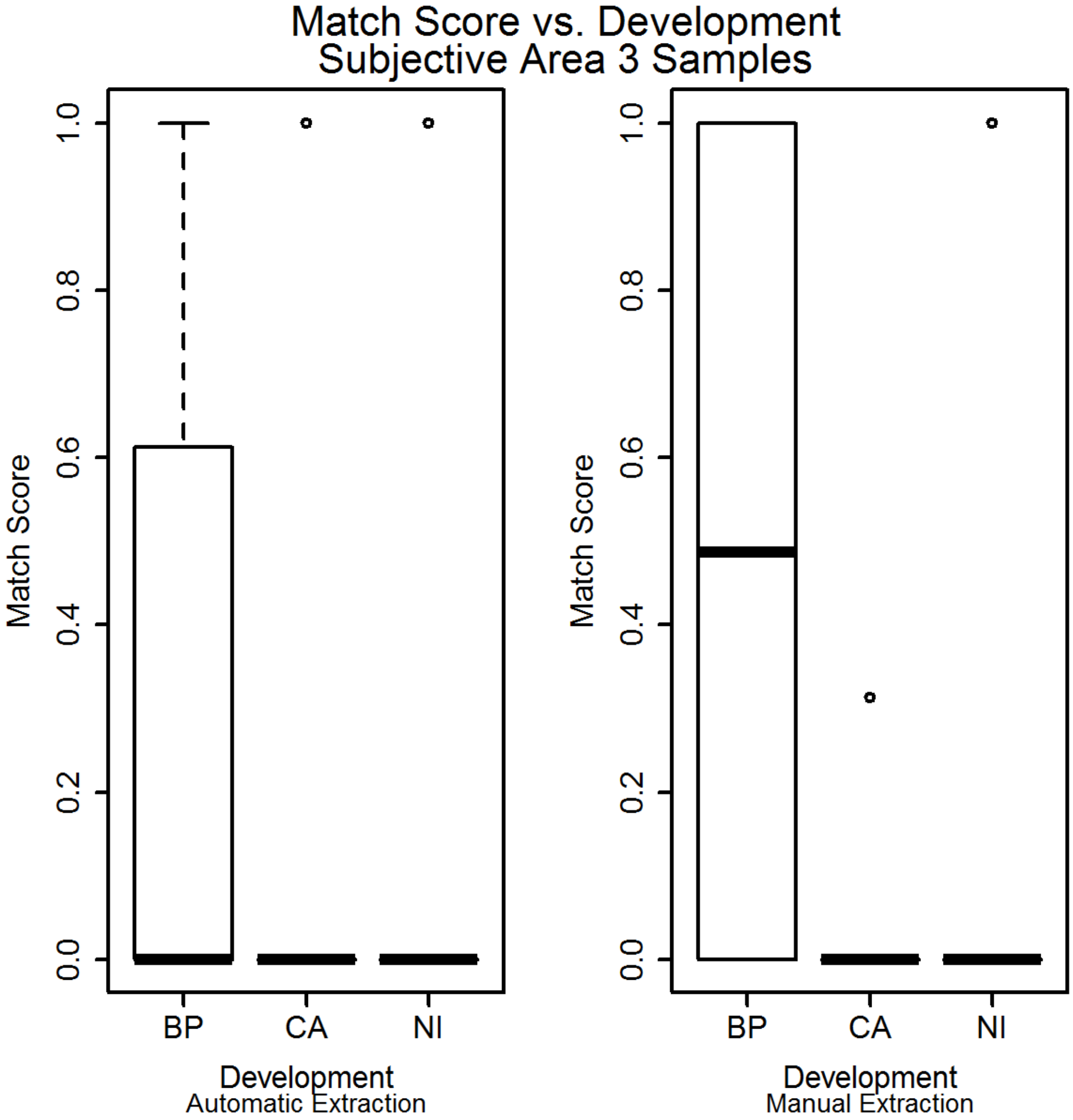


Figure D.44: This figure presents the box plots for the match scores for each development type in the subjective Area 3 images. On the left is the automatic extraction box plots while on the right is the manual extraction. The development methods are represented by the two letters. BP represents black powder, CA represents cyanoacrylate and NI represents ninhydrin. The thick black line represents the median in the samples.

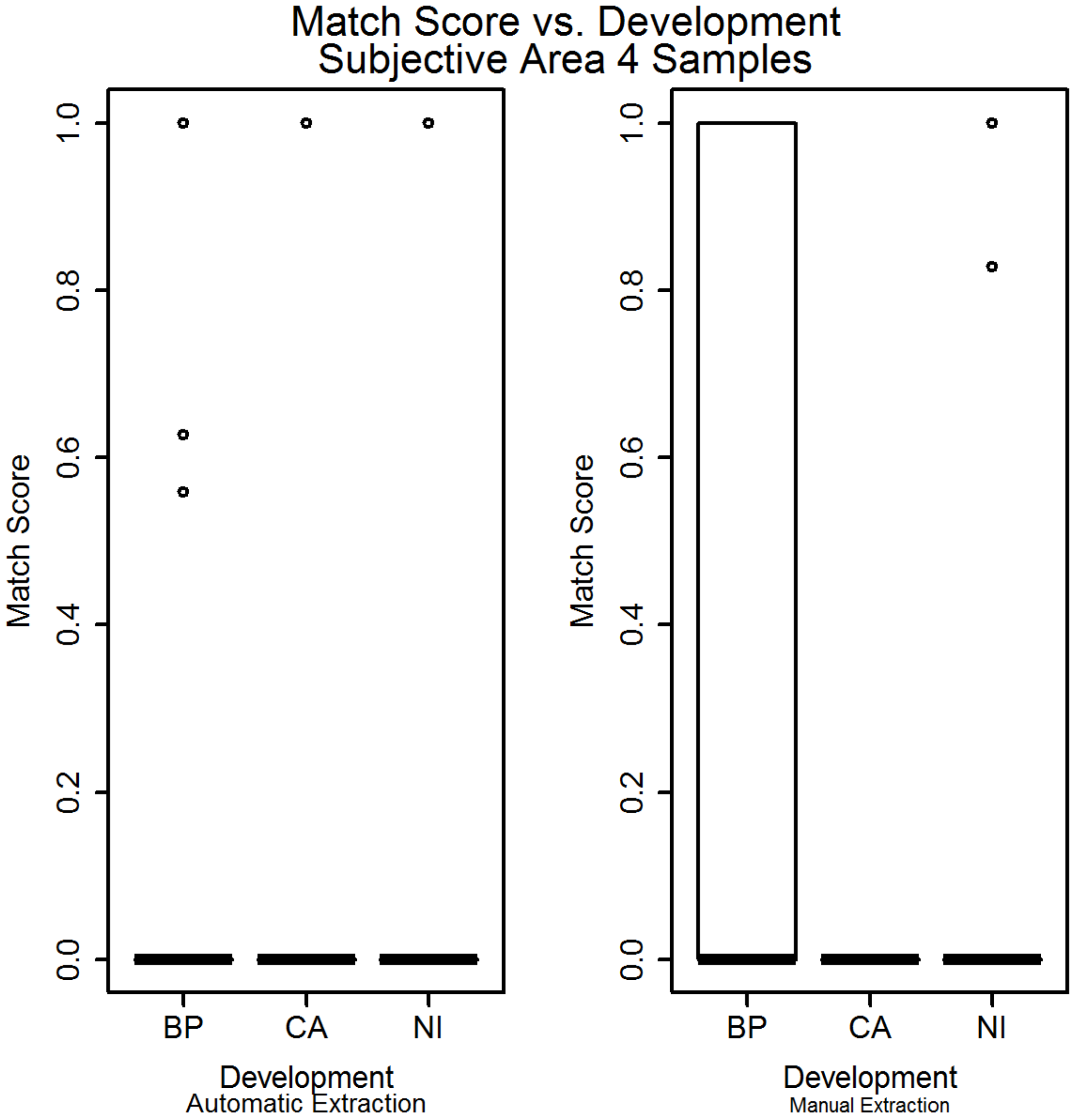


Figure D.45: This figure presents the box plots for the match scores for each development type in the subjective Area 4 images. On the left is the automatic extraction box plots while on the right is the manual extraction. The development methods are represented by the two letters. BP represents black powder, CA represents cyanoacrylate and NI represents ninhydrin. The thick black line represents the median in the samples.

D.5.2 Objective Method

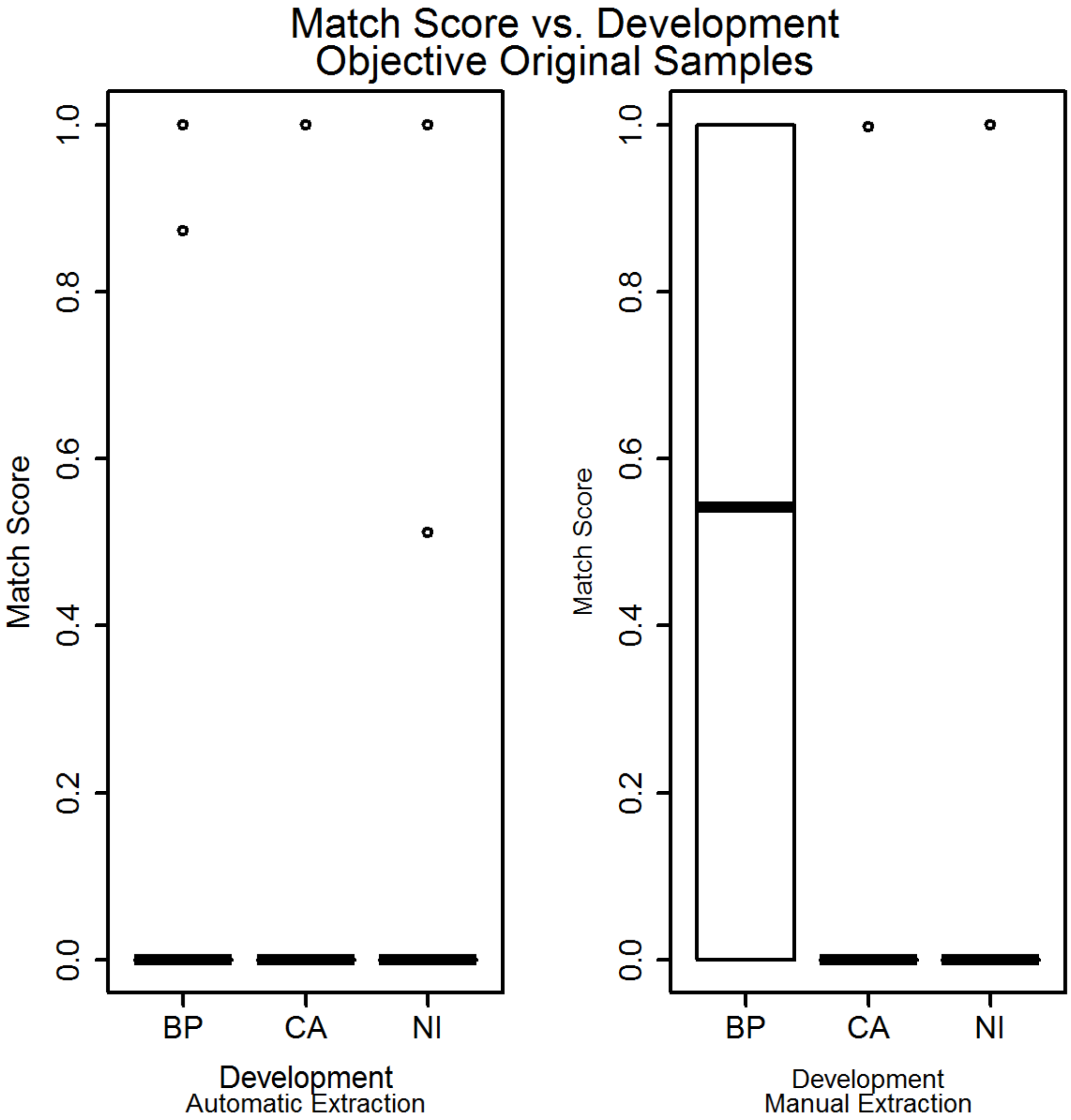


Figure D.46: This figure presents the box plots for the match scores for each development type in the objective original images. On the left is the automatic extraction box plots while on the right is the manual extraction. The development methods are represented by the two letters. BP represents black powder, CA represents cyanoacrylate and NI represents ninhydrin. The thick black line represents the median in the samples.

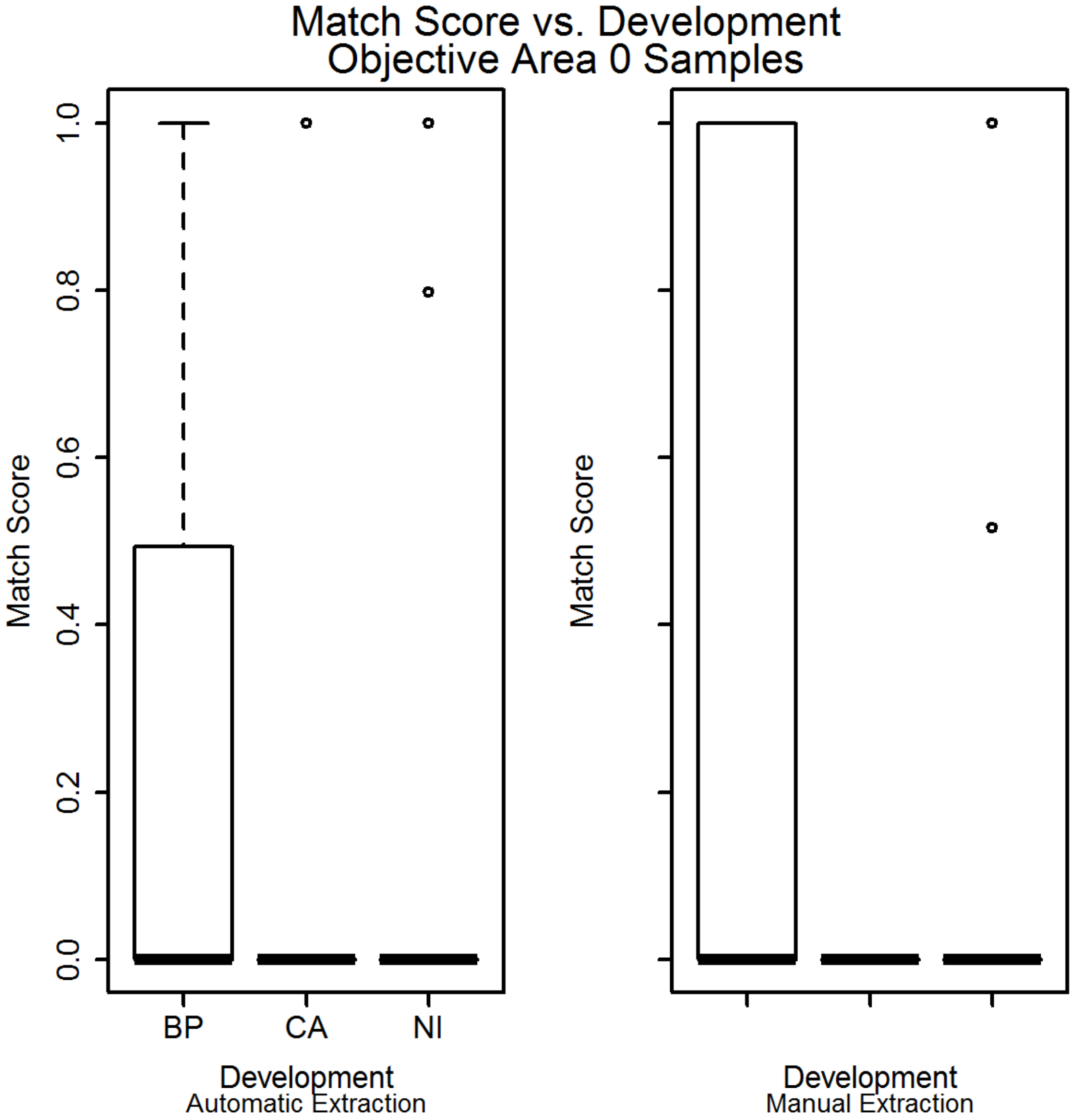


Figure D.47: This figure presents the box plots for the match scores for each development type in the objective Area 0 images. On the left is the automatic extraction box plots while on the right is the manual extraction. The development methods are represented by the two letters. BP represents black powder, CA represents cyanoacrylate and NI represents ninhydrin. The thick black line represents the median in the samples.

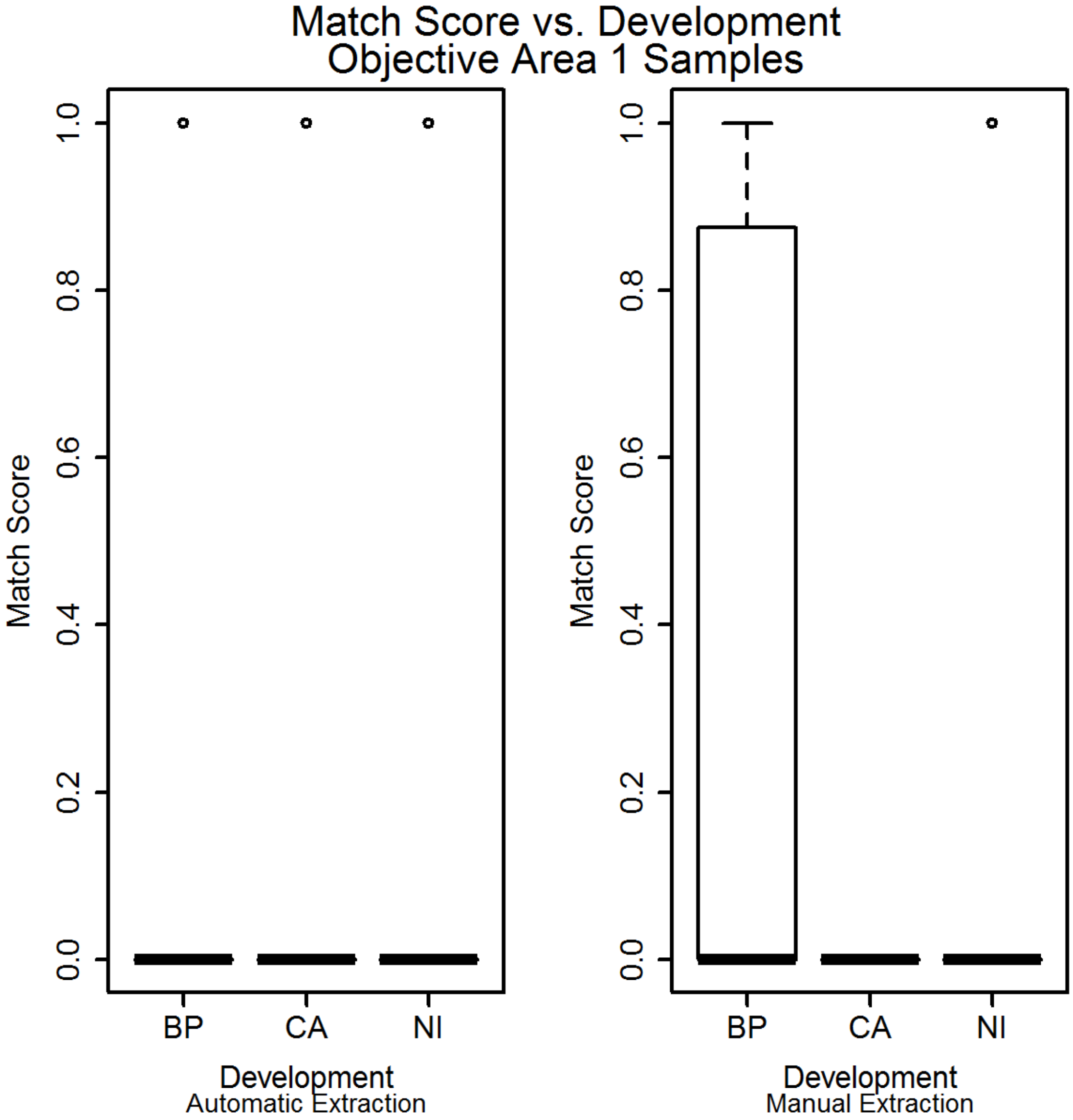


Figure D.48: This figure presents the box plots for the match scores for each development type in the objective Area 1 images. On the left is the automatic extraction box plots while on the right is the manual extraction. The development methods are represented by the two letters. BP represents black powder, CA represents cyanoacrylate and NI represents ninhydrin. The thick black line represents the median in the samples.

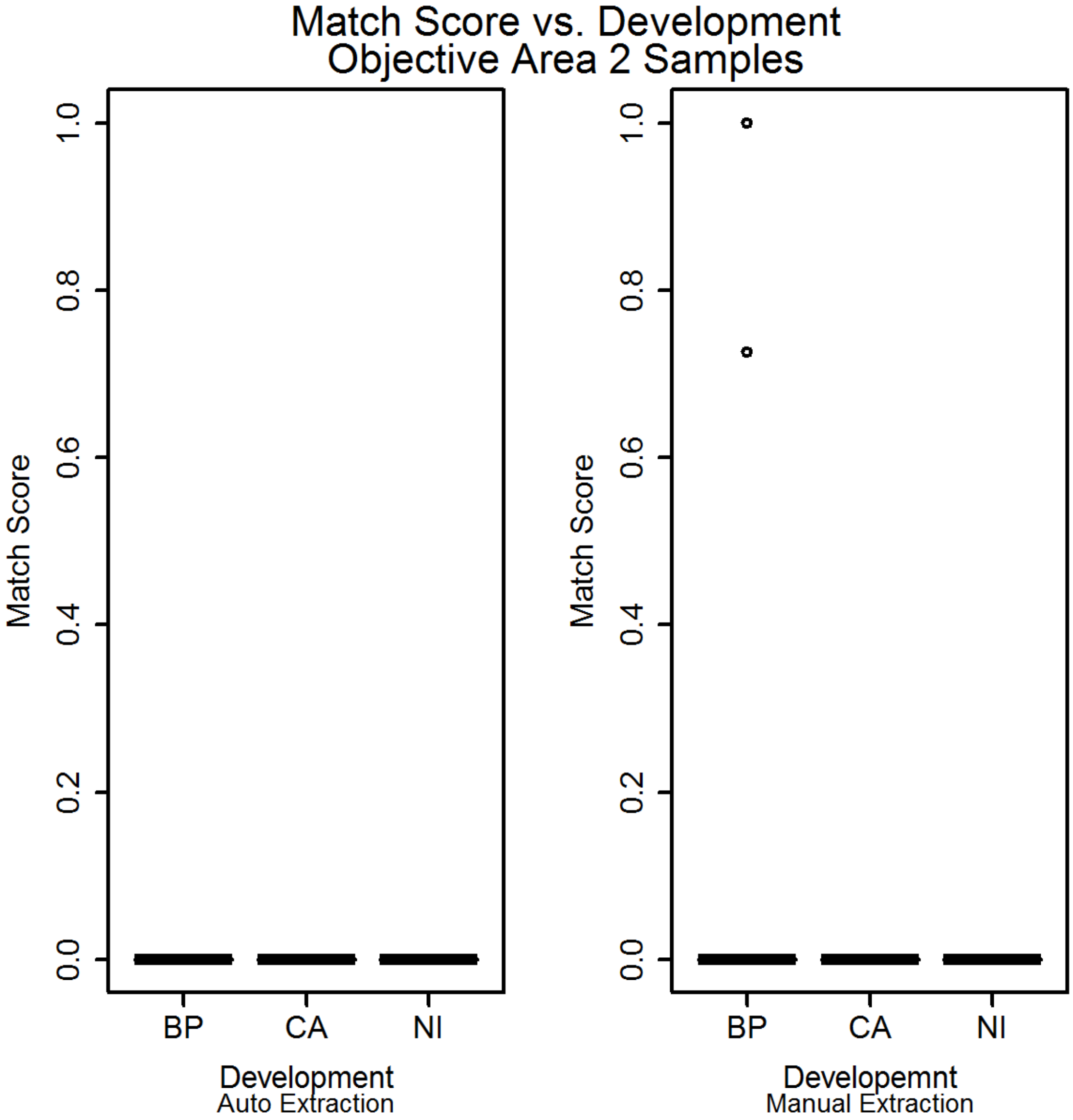


Figure D.49: This figure presents the box plots for the match scores for each development type in the objective Area 2 images. On the left is the automatic extraction box plots while on the right is the manual extraction. The development methods are represented by the two letters. BP represents black powder, CA represents cyanoacrylate and NI represents ninhydrin. The thick black line represents the median in the samples.

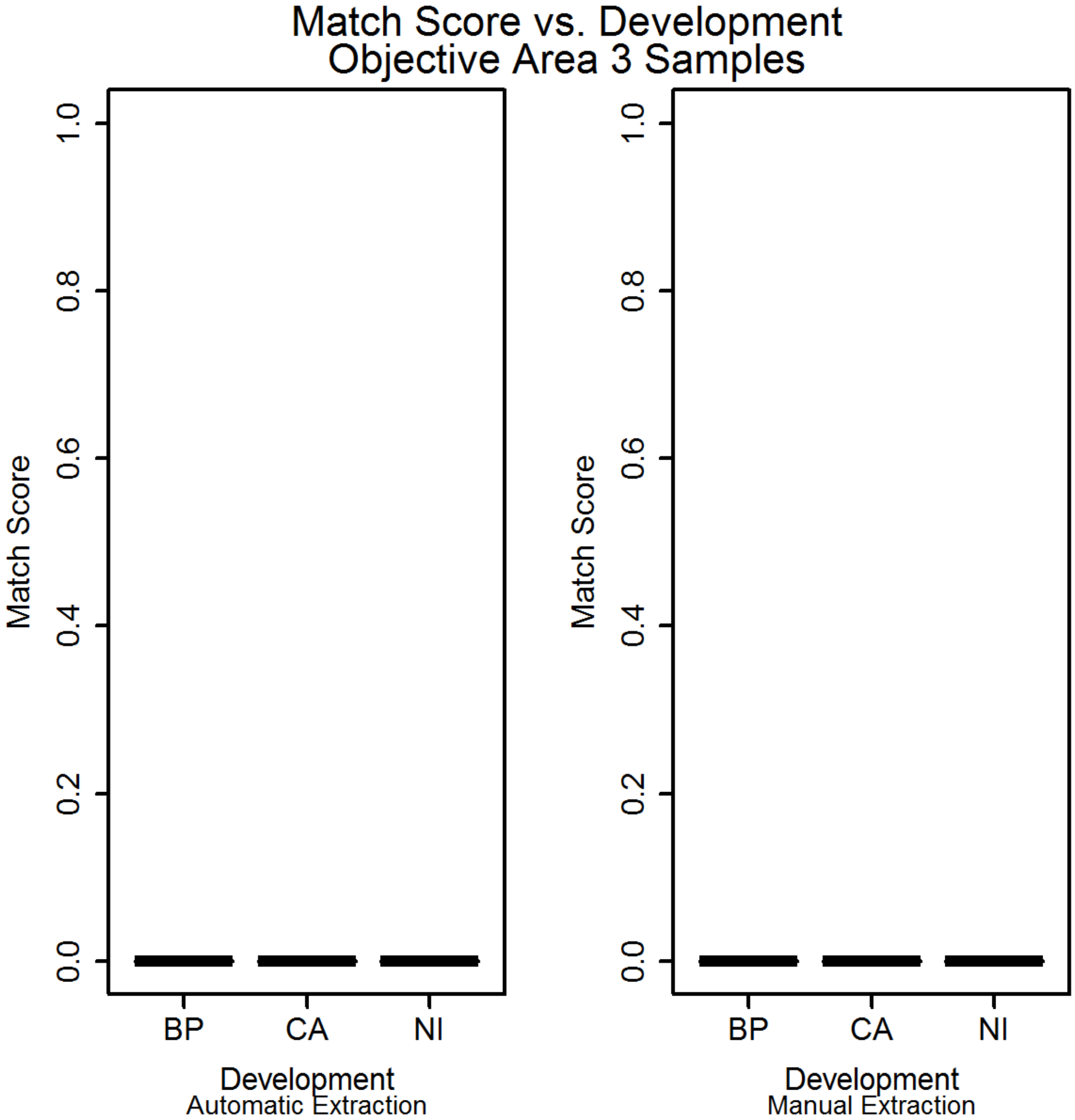


Figure D.50: This figure presents the box plots for the match scores for each development type in the objective Area 3 images. On the left is the automatic extraction box plots while on the right is the manual extraction. The development methods are represented by the two letters. BP represents black powder, CA represents cyanoacrylate and NI represents ninhydrin. The thick black line represents the median in the samples.

D.6 Match Score Substrate Types

D.6.1 Subjective Method

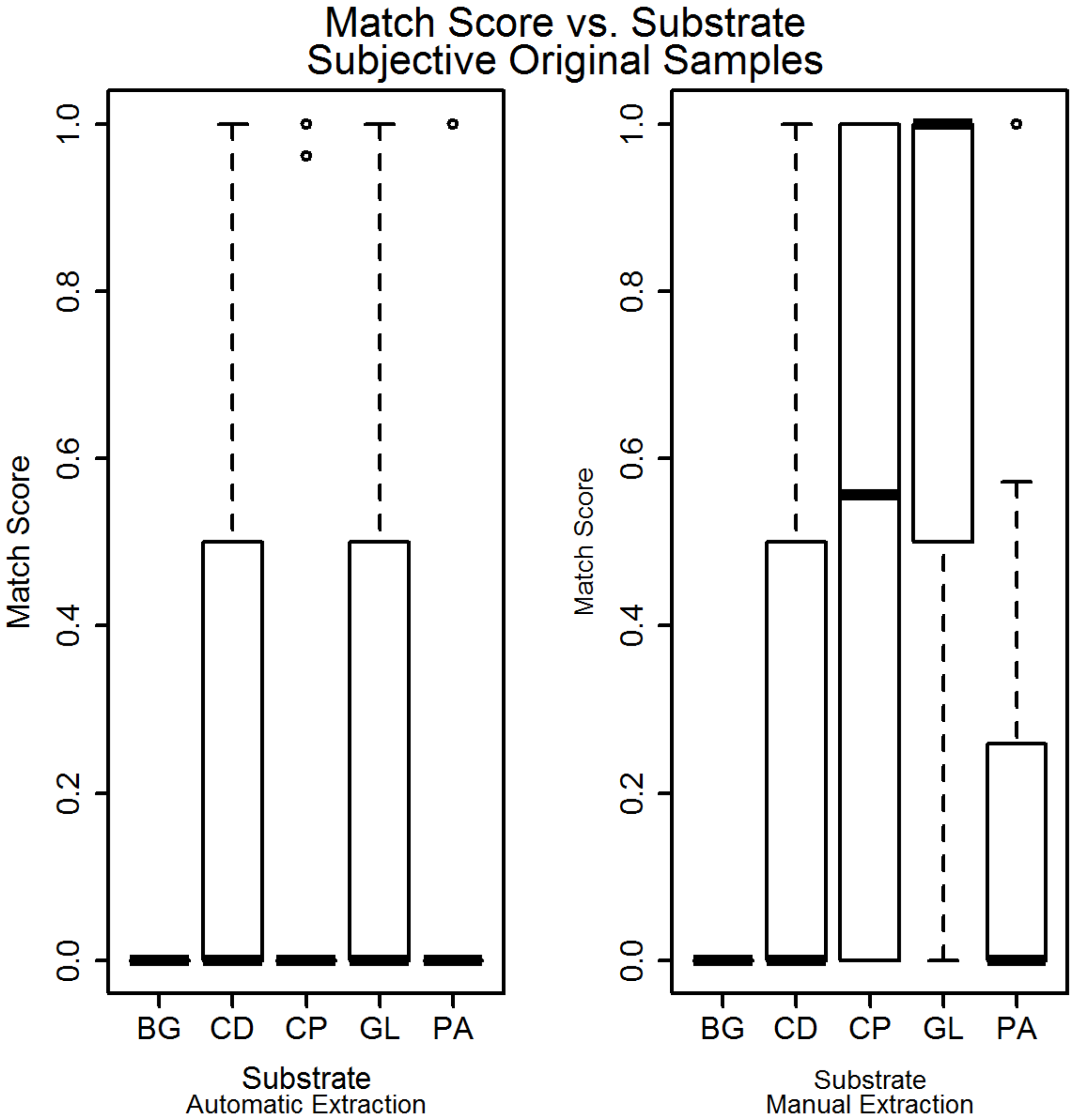


Figure D.51: This figure presents the box plots for the minutiae detected for each substrate type for the subjective original images. On the left is the automatic extraction box plots while on the right is the manual extraction. The substrate types are represented by the two letters. BG represents bag, CD represents a CD, CP represents ceramic plate, GL represents glass and PA represents paper. The thick black line represents the median in the samples.

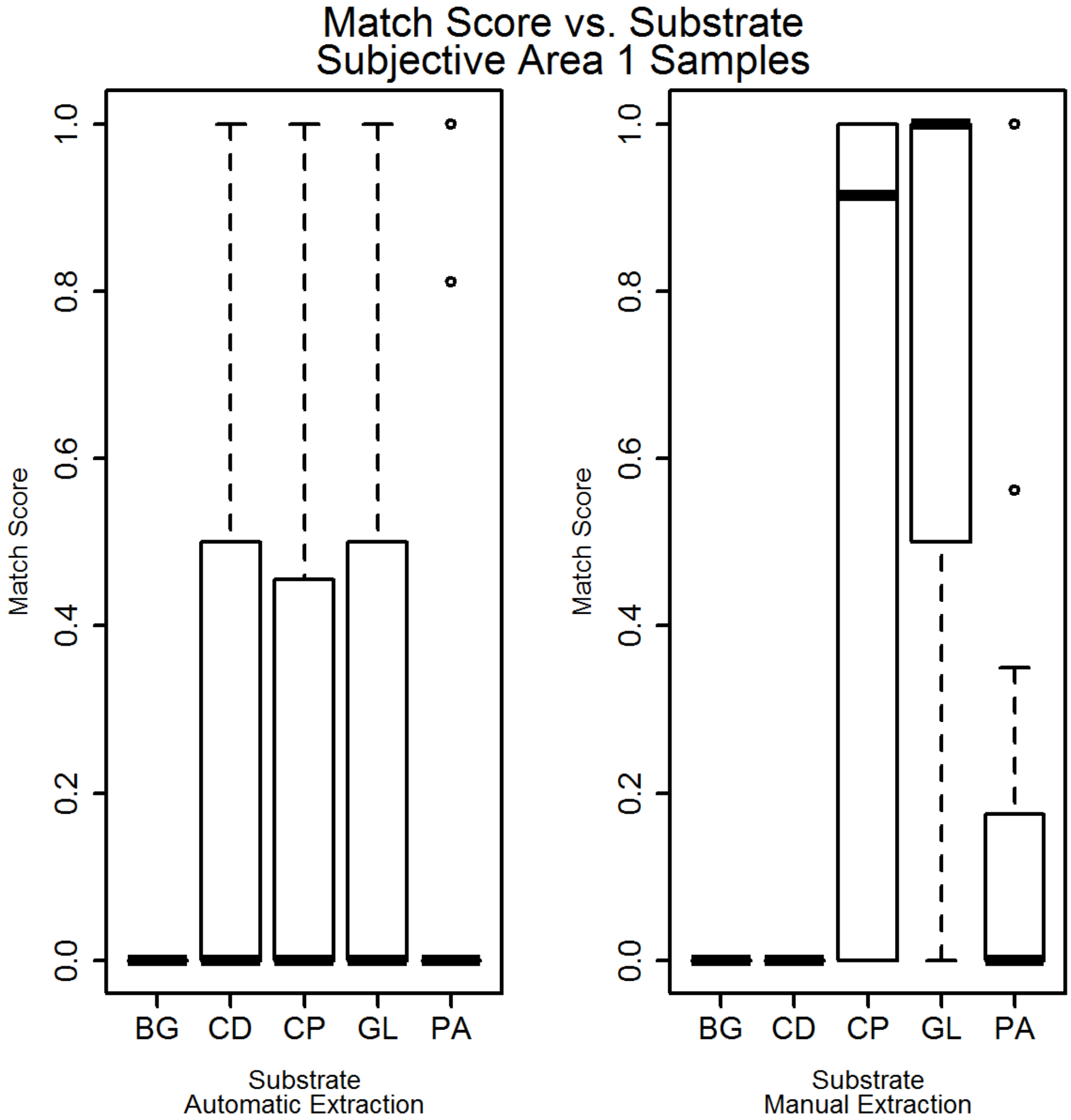


Figure D.52: This figure presents the box plots for the minutiae detected for each substrate type for the subjective Area 1 images. On the left is the automatic extraction box plots while on the right is the manual extraction. The substrate types are represented by the two letters. BG represents bag, CD represents a CD, CP represents ceramic plate, GL represents glass and PA represents paper. The thick black line represents the median in the samples.

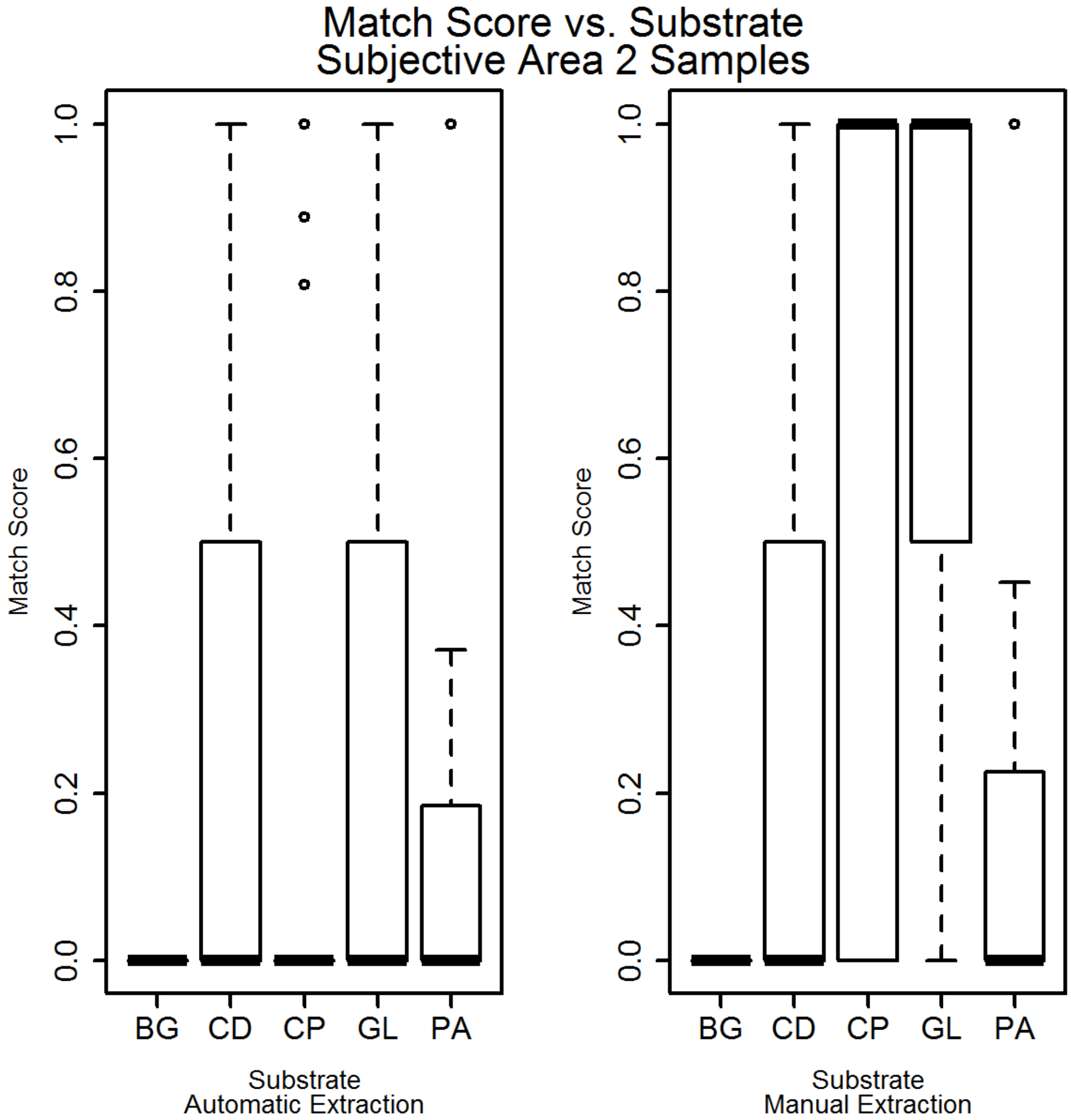


Figure D.53: This figure presents the box plots for the minutiae detected for each substrate type for the subjective Area 2 images. On the left is the automatic extraction box plots while on the right is the manual extraction. The substrate types are represented by the two letters. BG represents bag, CD represents a CD, CP represents ceramic plate, GL represents glass and PA represents paper. The thick black line represents the median in the samples.

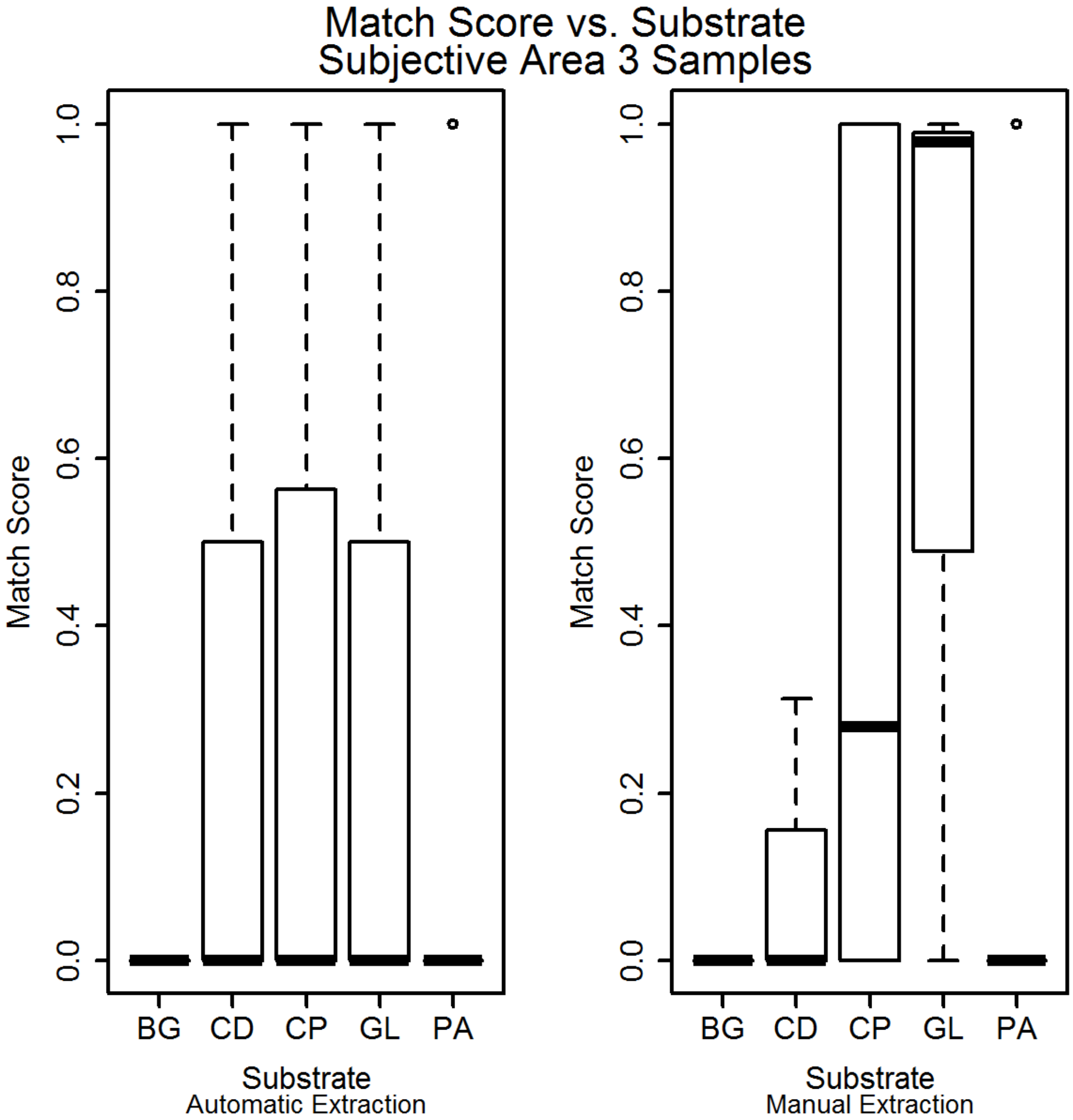


Figure D.54: This figure presents the box plots for the minutiae detected for each substrate type for the subjective Area 3 images. On the left is the automatic extraction box plots while on the right is the manual extraction. The substrate types are represented by the two letters. BG represents bag, CD represents a CD, CP represents ceramic plate, GL represents glass and PA represents paper. The thick black line represents the median in the samples.

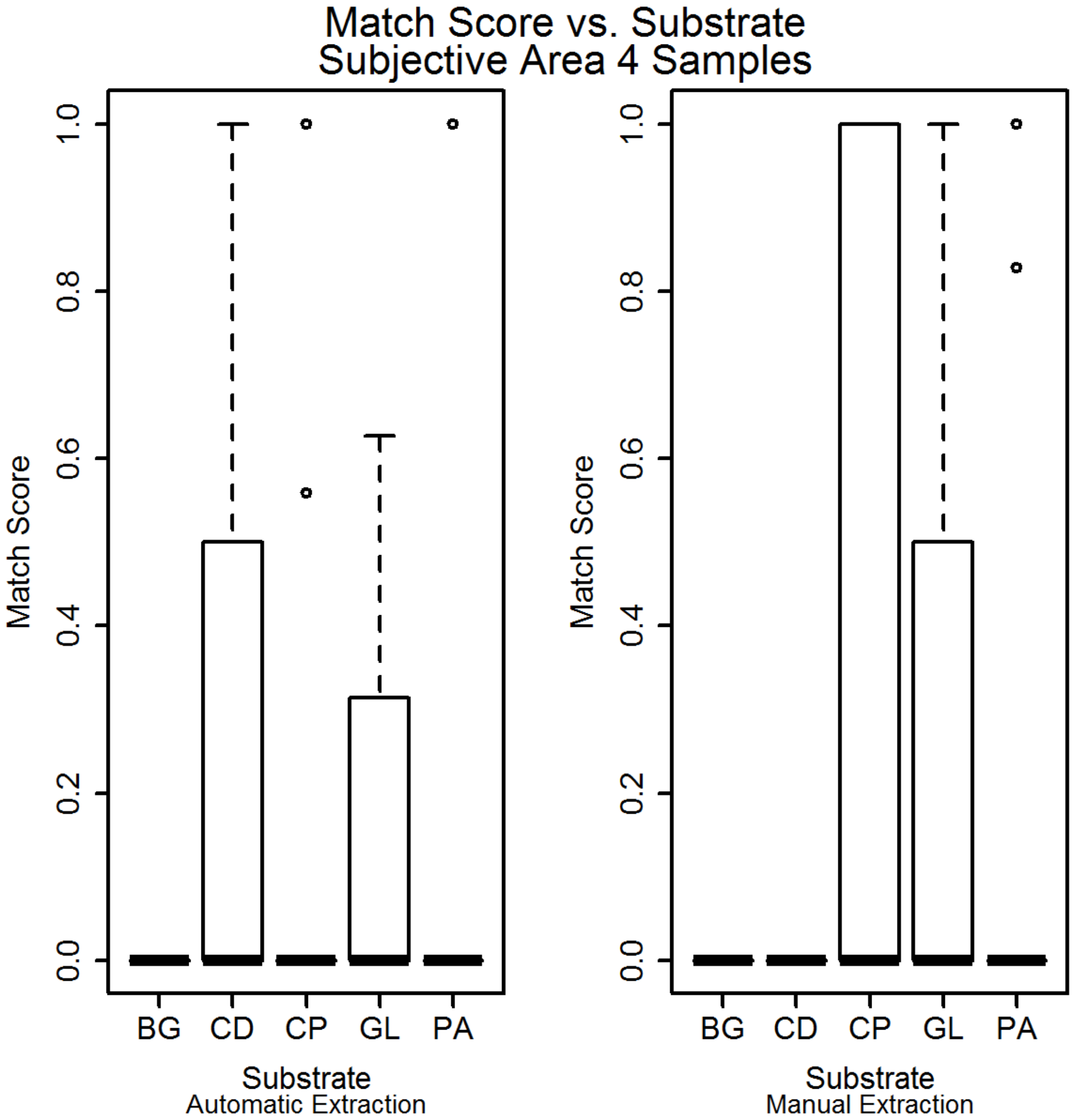


Figure D.55: This figure presents the box plots for the minutiae detected for each substrate type for the subjective Area 4 images. On the left is the automatic extraction box plots while on the right is the manual extraction. The substrate types are represented by the two letters. BG represents bag, CD represents a CD, CP represents ceramic plate, GL represents glass and PA represents paper. The thick black line represents the median in the samples.

D.6.2 Objective Method

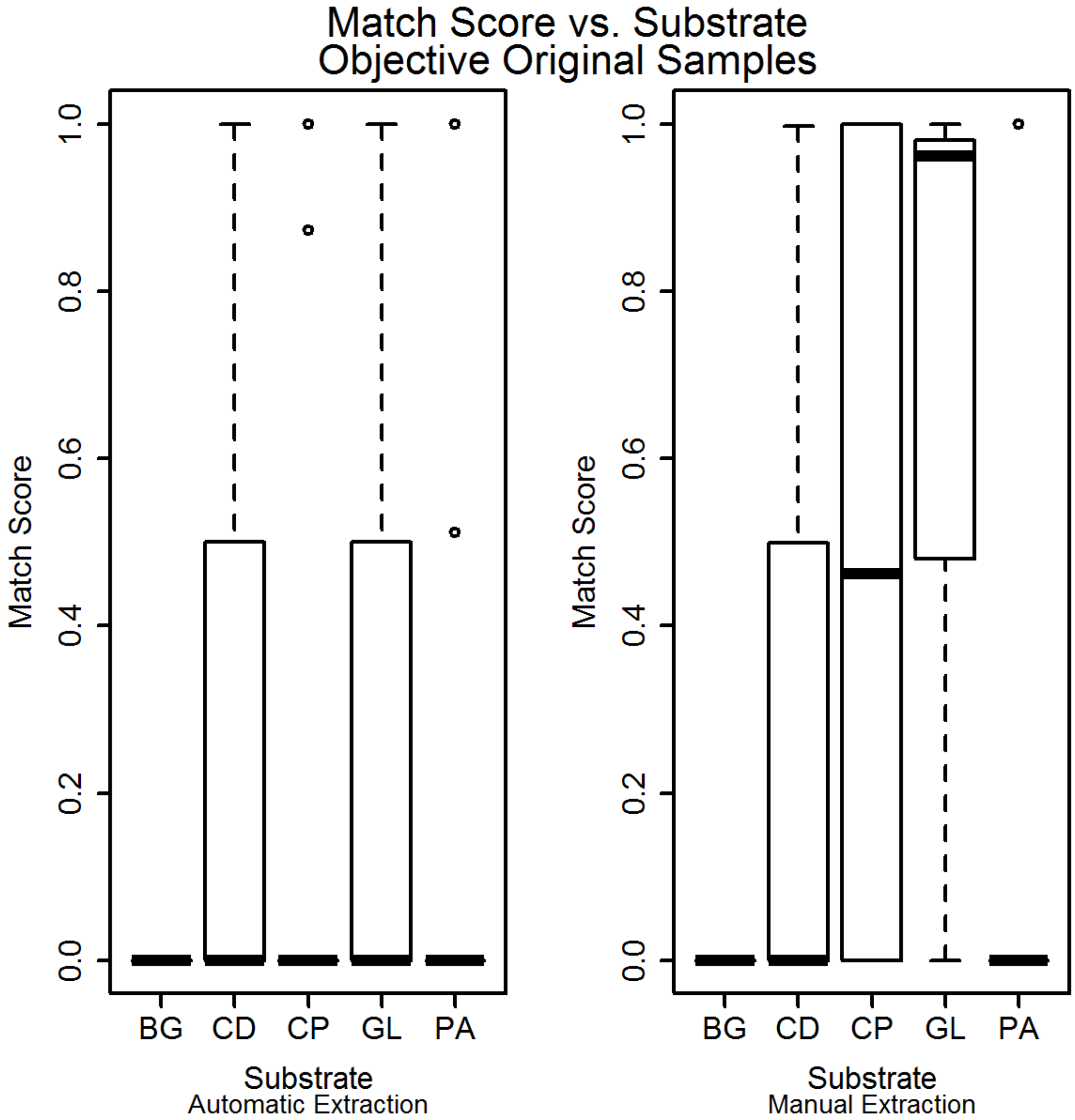


Figure D.56: This figure presents the box plots for the minutiae detected for each substrate type for the objective original images. On the left is the automatic extraction box plots while on the right is the manual extraction. The substrate types are represented by the two letters. BG represents bag, CD represents a CD, CP represents ceramic plate, GL represents glass and PA represents paper. The thick black line represents the median in the samples.

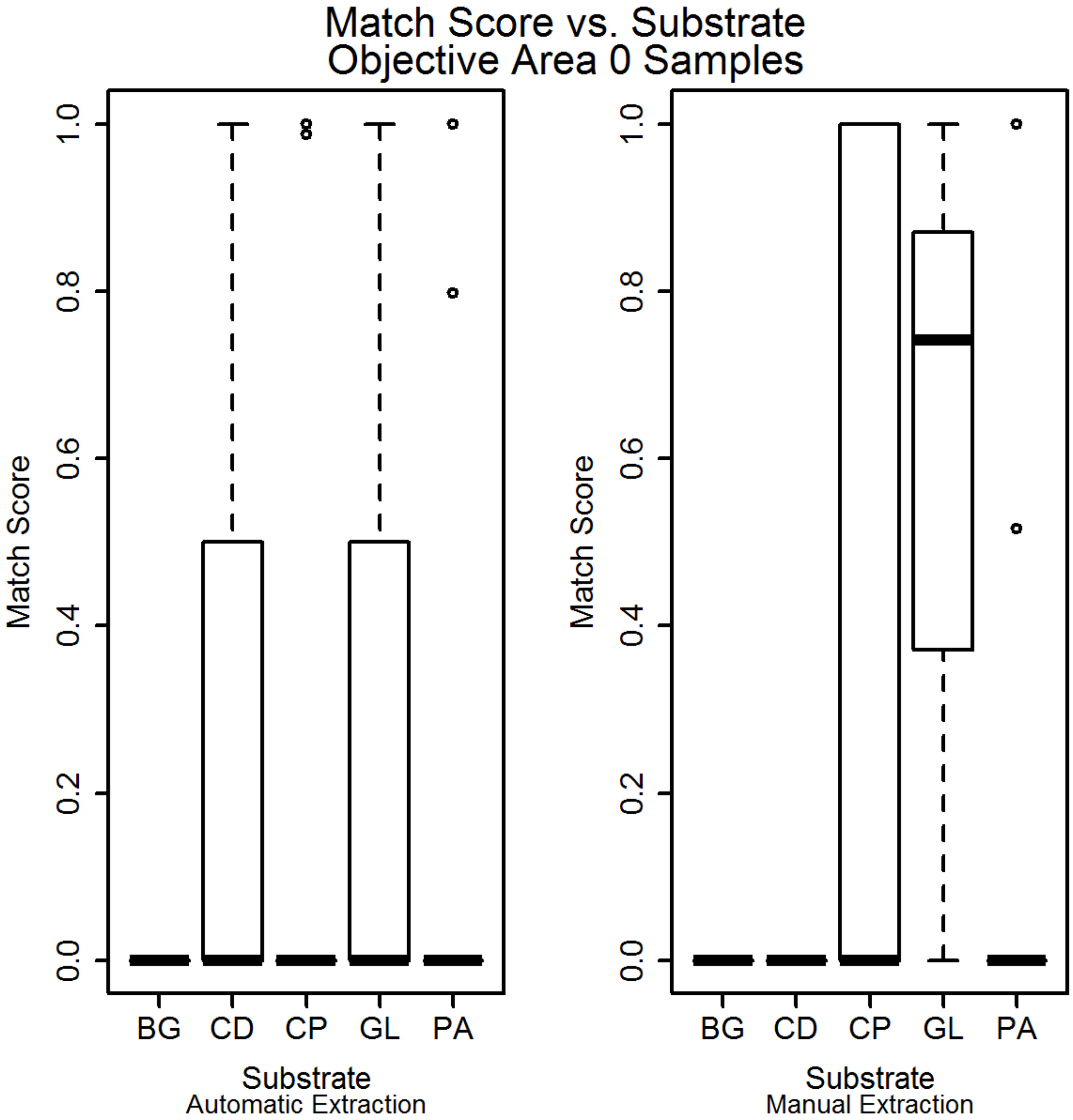


Figure D.57: This figure presents the box plots for the minutiae detected for each substrate type for the objective Area 0 images. On the left is the automatic extraction box plots while on the right is the manual extraction. The substrate types are represented by the two letters. BG represents bag, CD represents a CD, CP represents ceramic plate, GL represents glass and PA represents paper. The thick black line represents the median in the samples.

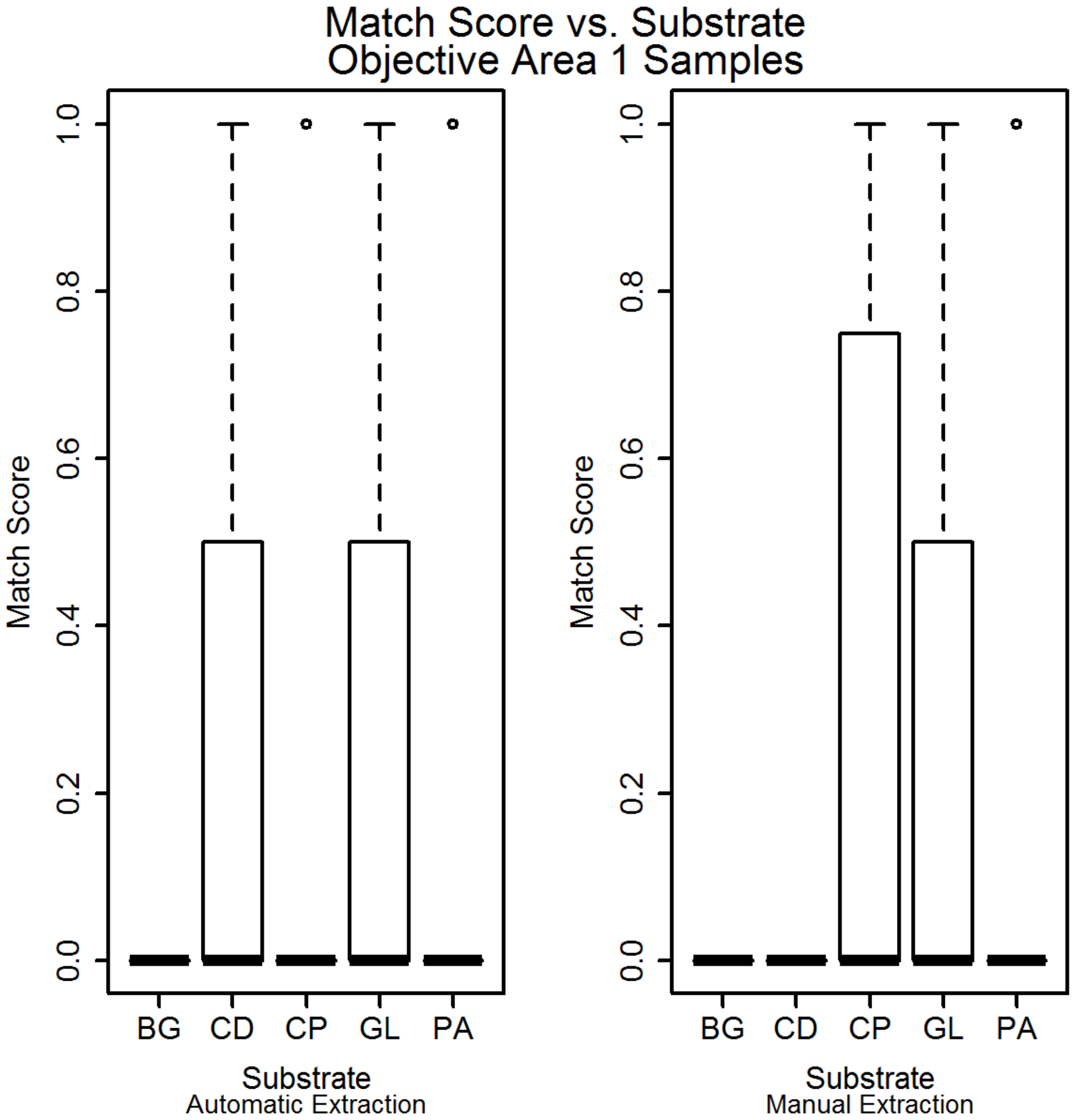


Figure D.58: This figure presents the box plots for the minutiae detected for each substrate type for the objective Area 1 images. On the left is the automatic extraction box plots while on the right is the manual extraction. The substrate types are represented by the two letters. BG represents bag, CD represents a CD, CP represents ceramic plate, GL represents glass and PA represents paper. The thick black line represents the median in the samples.

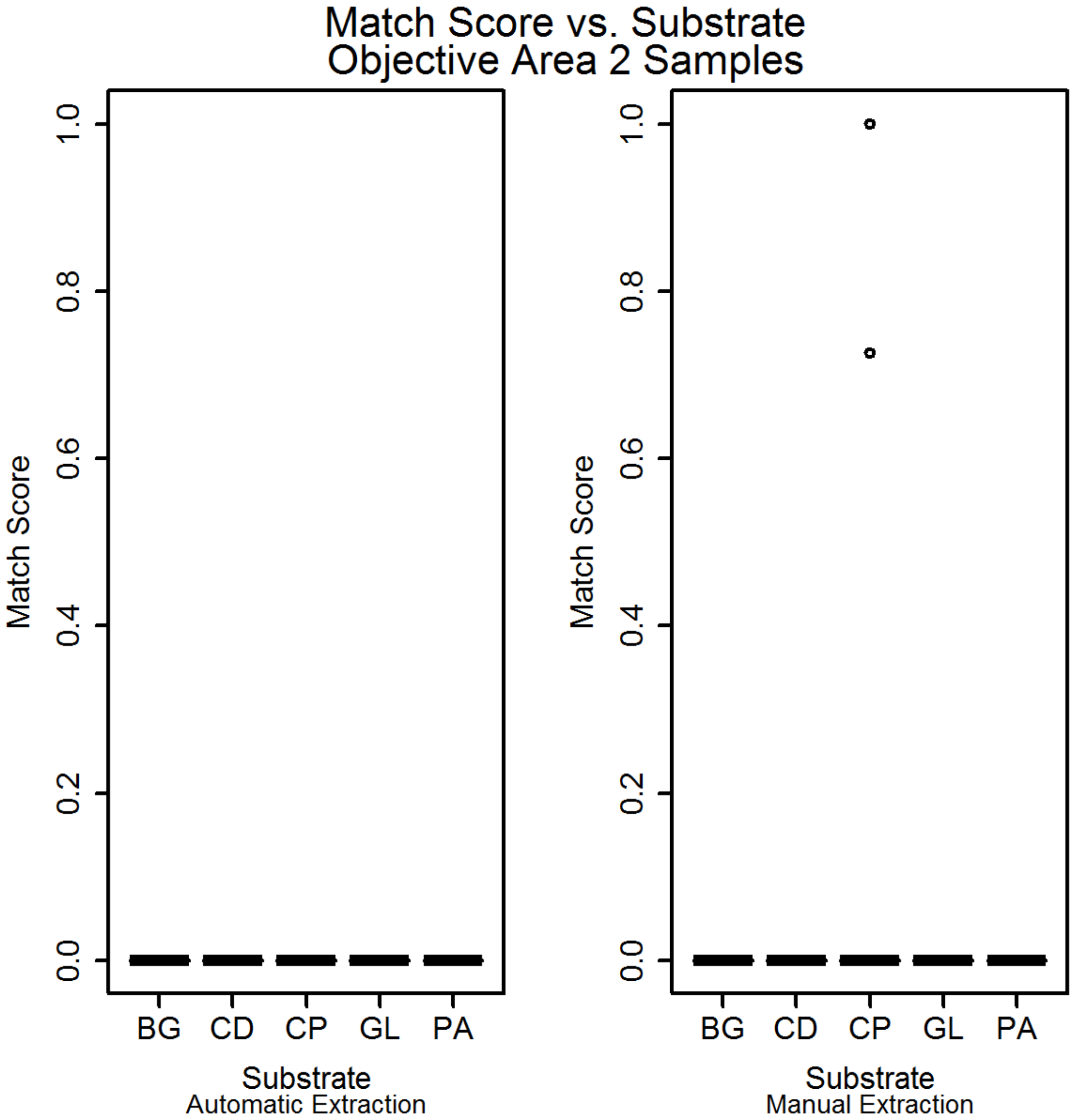


Figure D.59: This figure presents the box plots for the minutiae detected for each substrate type for the objective Area 2 images. On the left is the automatic extraction box plots while on the right is the manual extraction. The substrate types are represented by the two letters. BG represents bag, CD represents a CD, CP represents ceramic plate, GL represents glass and PA represents paper. The thick black line represents the median in the samples.

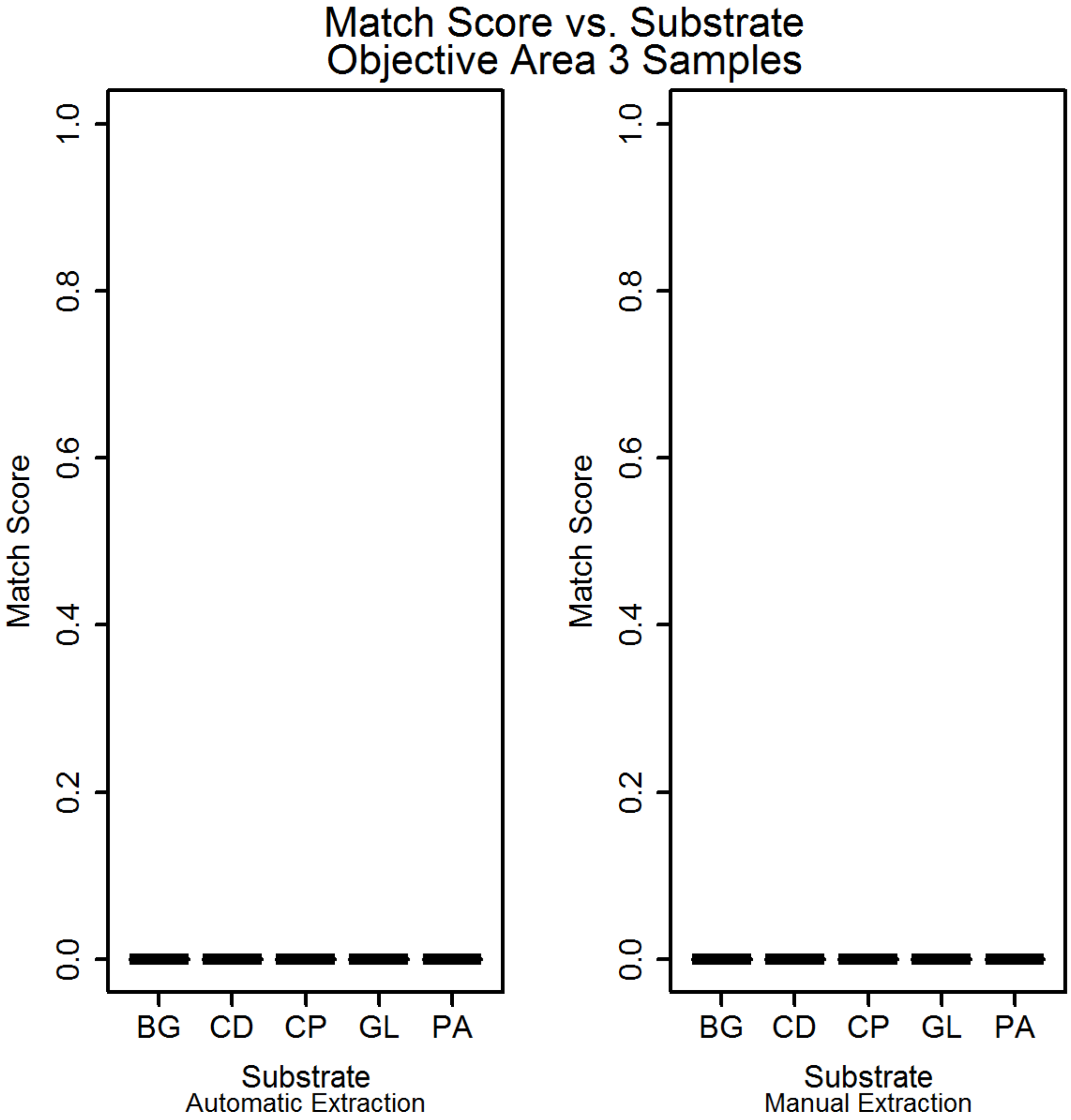


Figure D.60: This figure presents the box plots for the minutiae detected for each substrate type for the objective Area 3 images. On the left is the automatic extraction box plots while on the right is the manual extraction. The substrate types are represented by the two letters. BG represents bag, CD represents a CD, CP represents ceramic plate, GL represents glass and PA represents paper. The thick black line represents the median in the samples.

D.7 ROC Curves-Match Scores

D.7.1 Subjective Method: All Samples

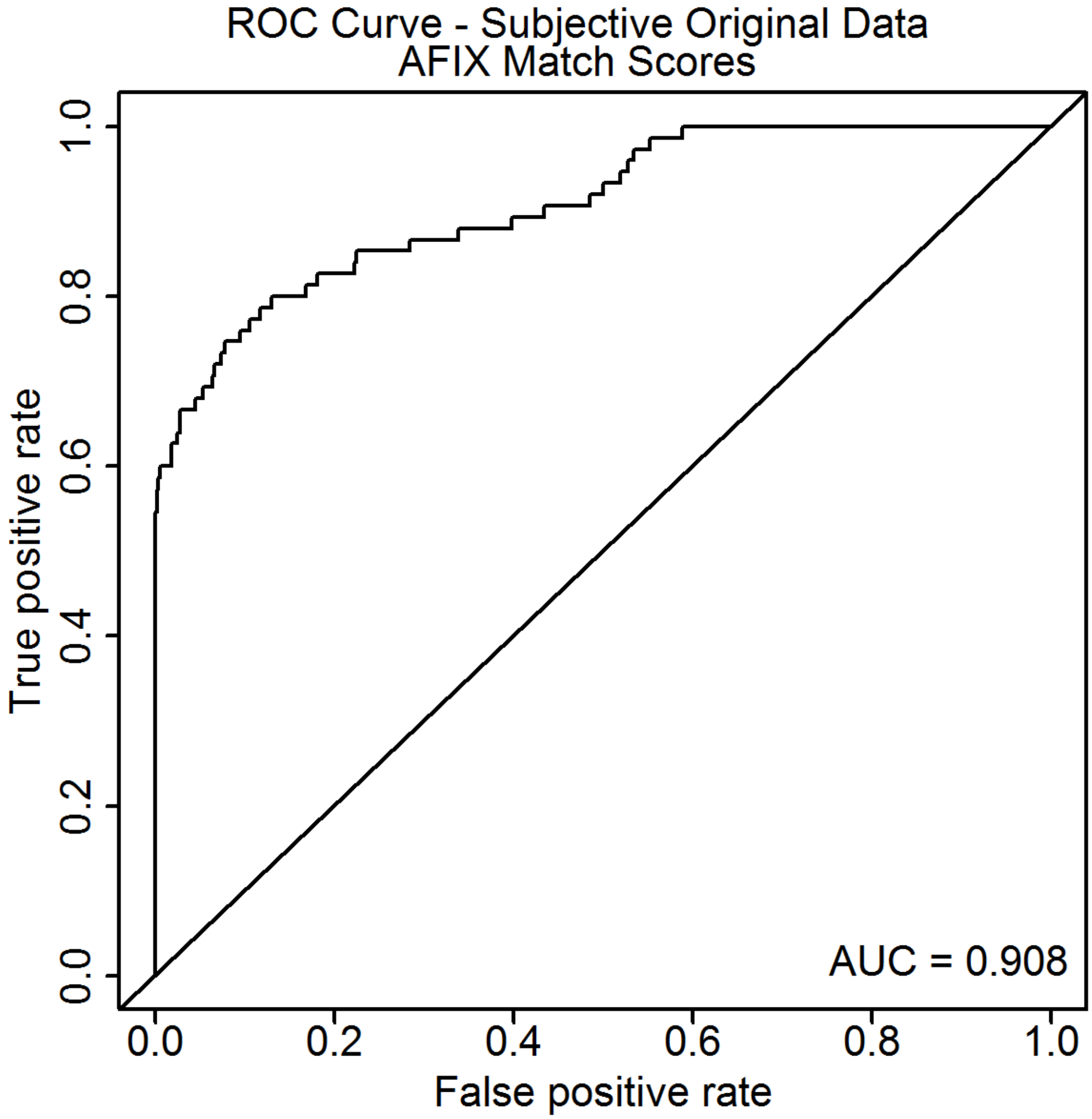


Figure D.61: This figure presents the ROC curve from the subjective original images for the match scores.

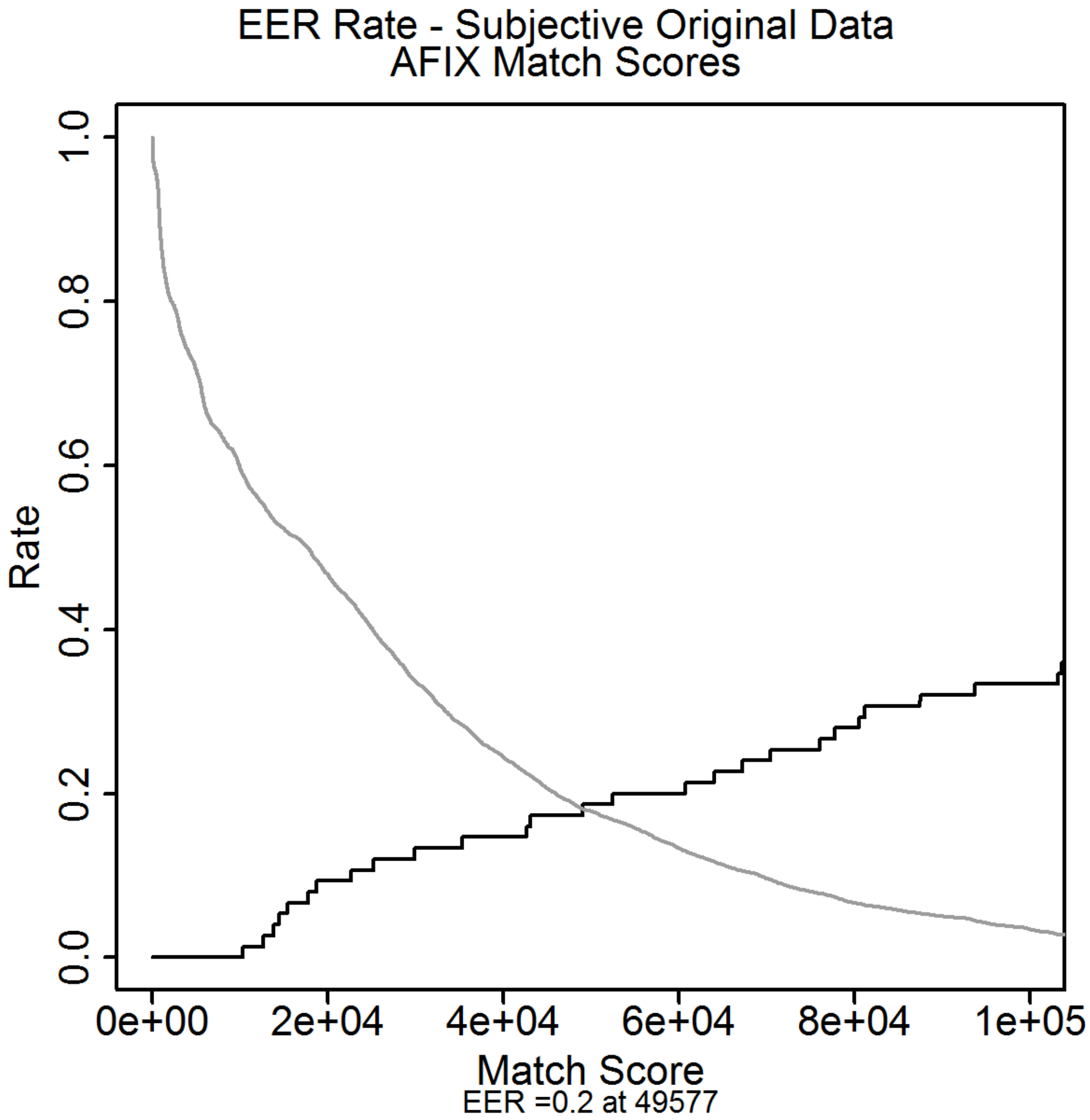


Figure D.62: This figure presents the EER plot from the subjective original images for the match scores.

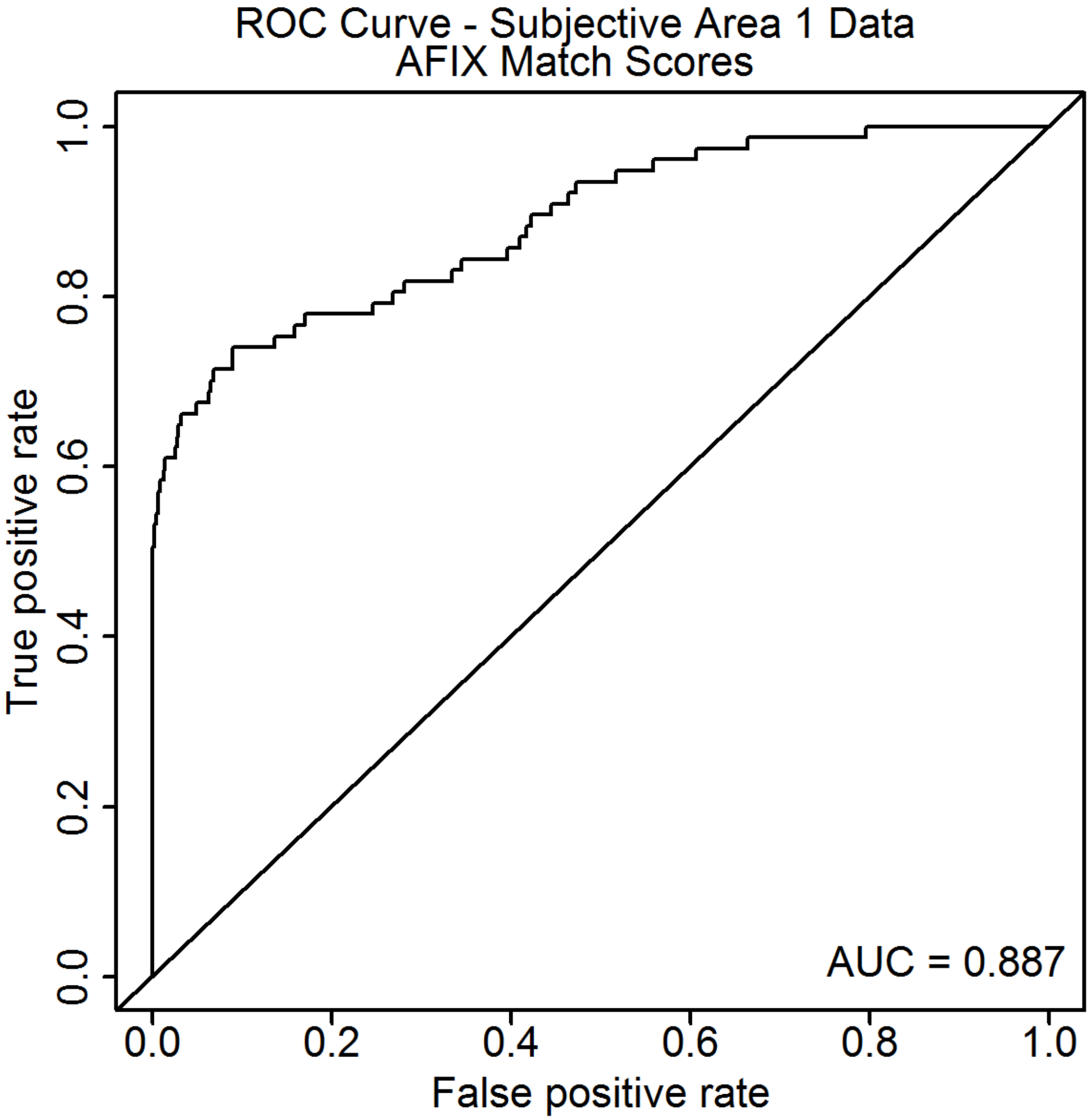


Figure D.63: This figure presents the ROC curve from the subjective Area 1 images for the match scores.

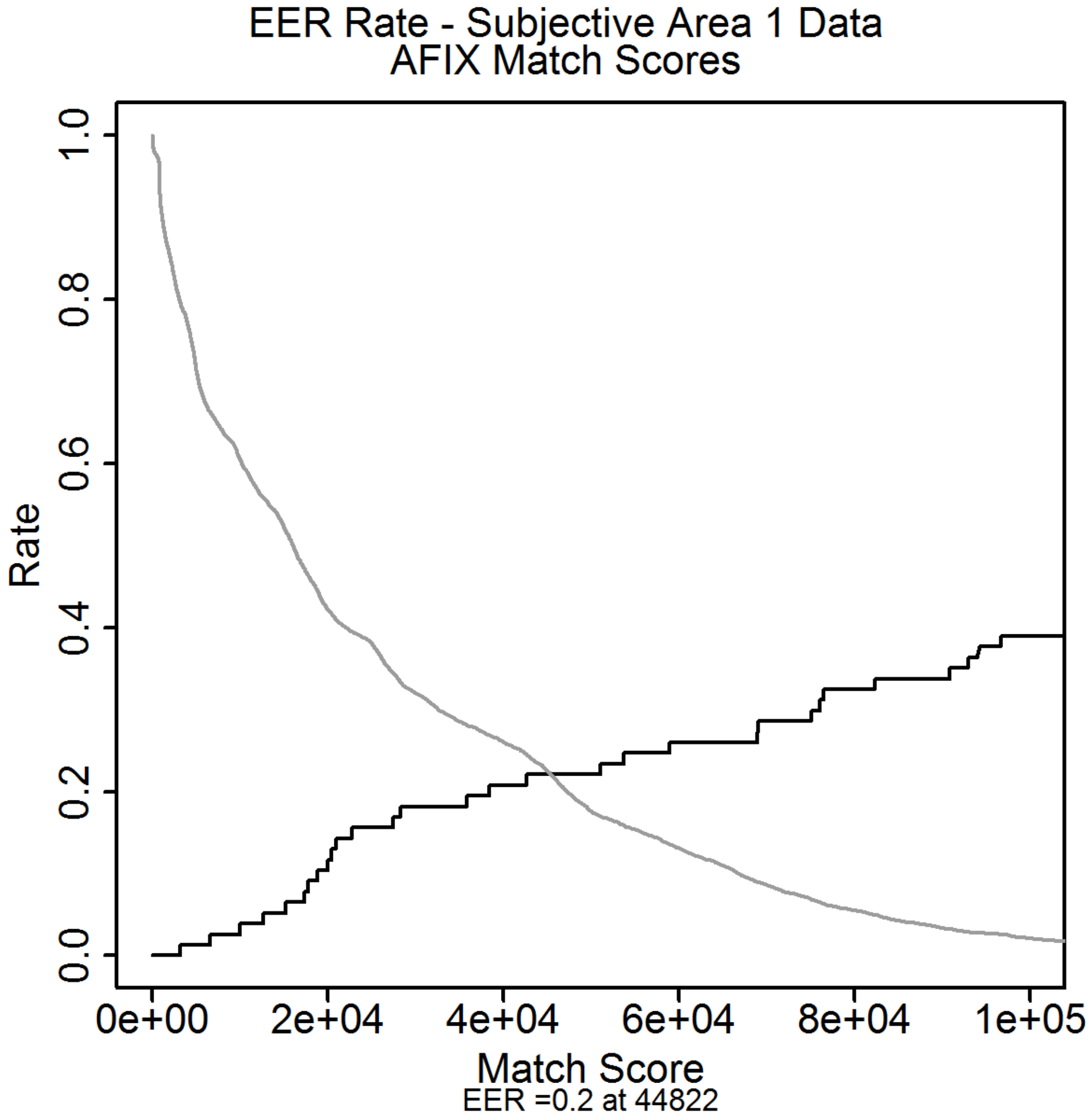


Figure D.64: This figure presents the EER plot from the subjective Area 1 images for the match scores.

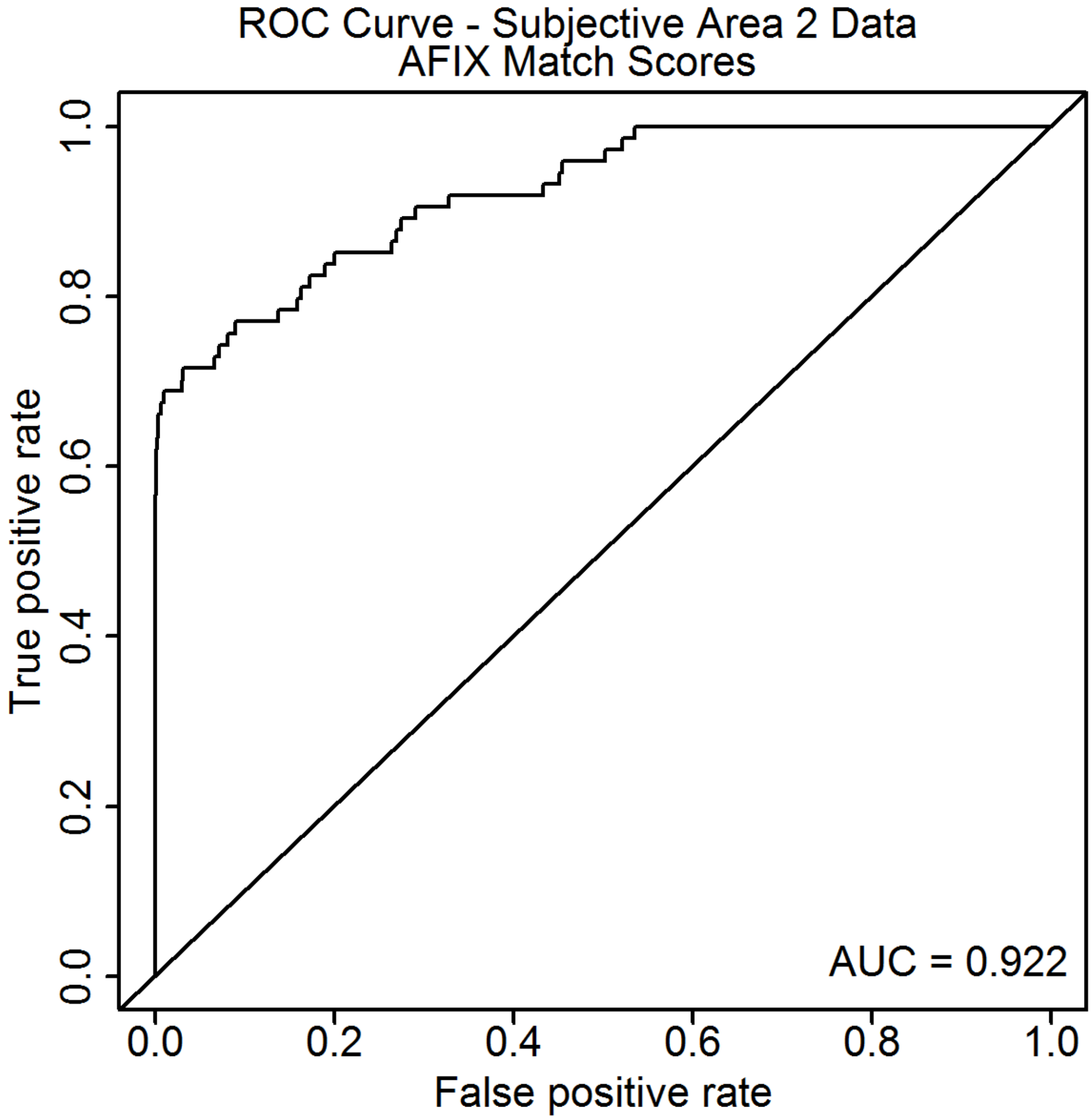


Figure D.65: This figure presents the ROC curve from the subjective Area 2 images for the match scores.

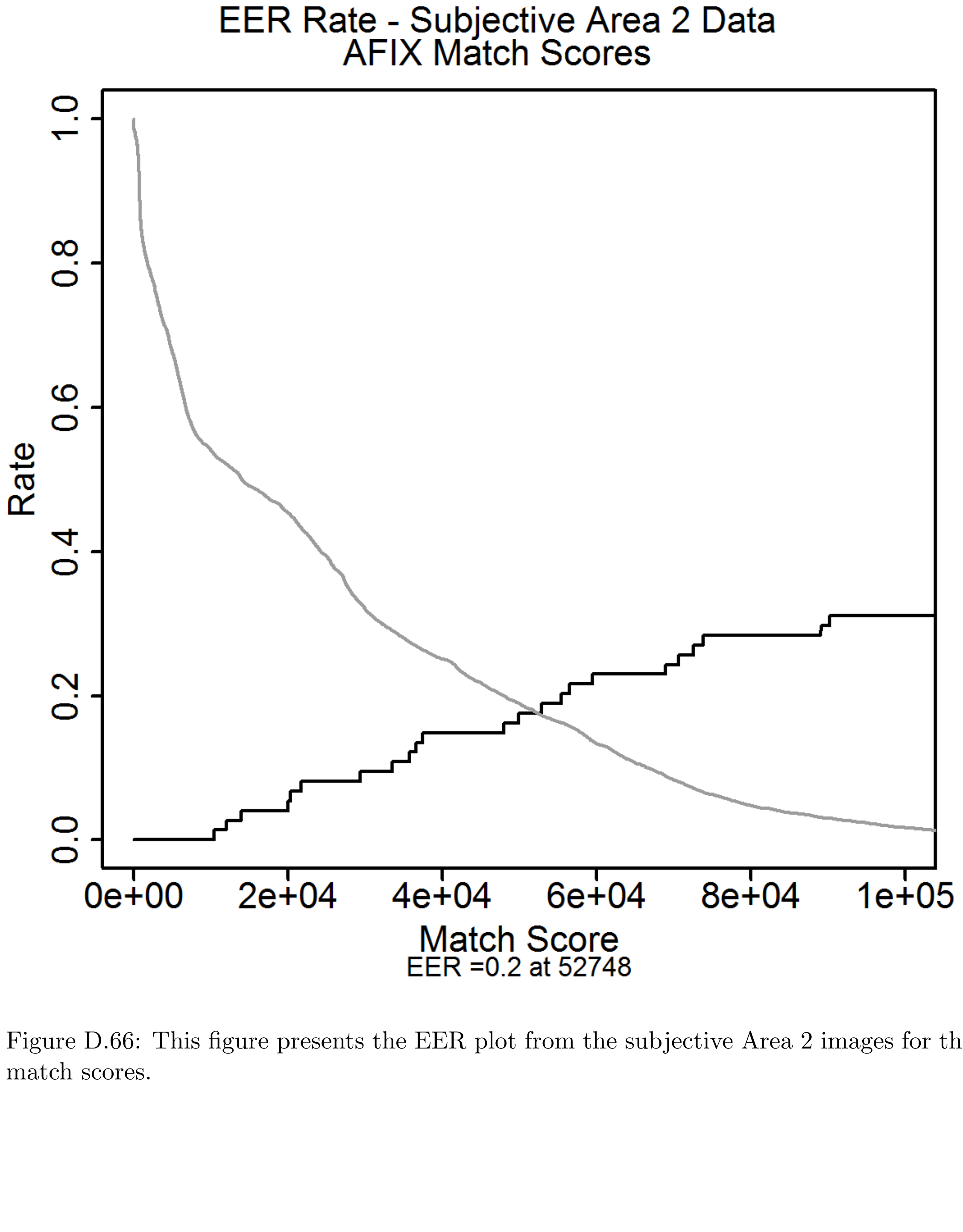


Figure D.66: This figure presents the EER plot from the subjective Area 2 images for the match scores.

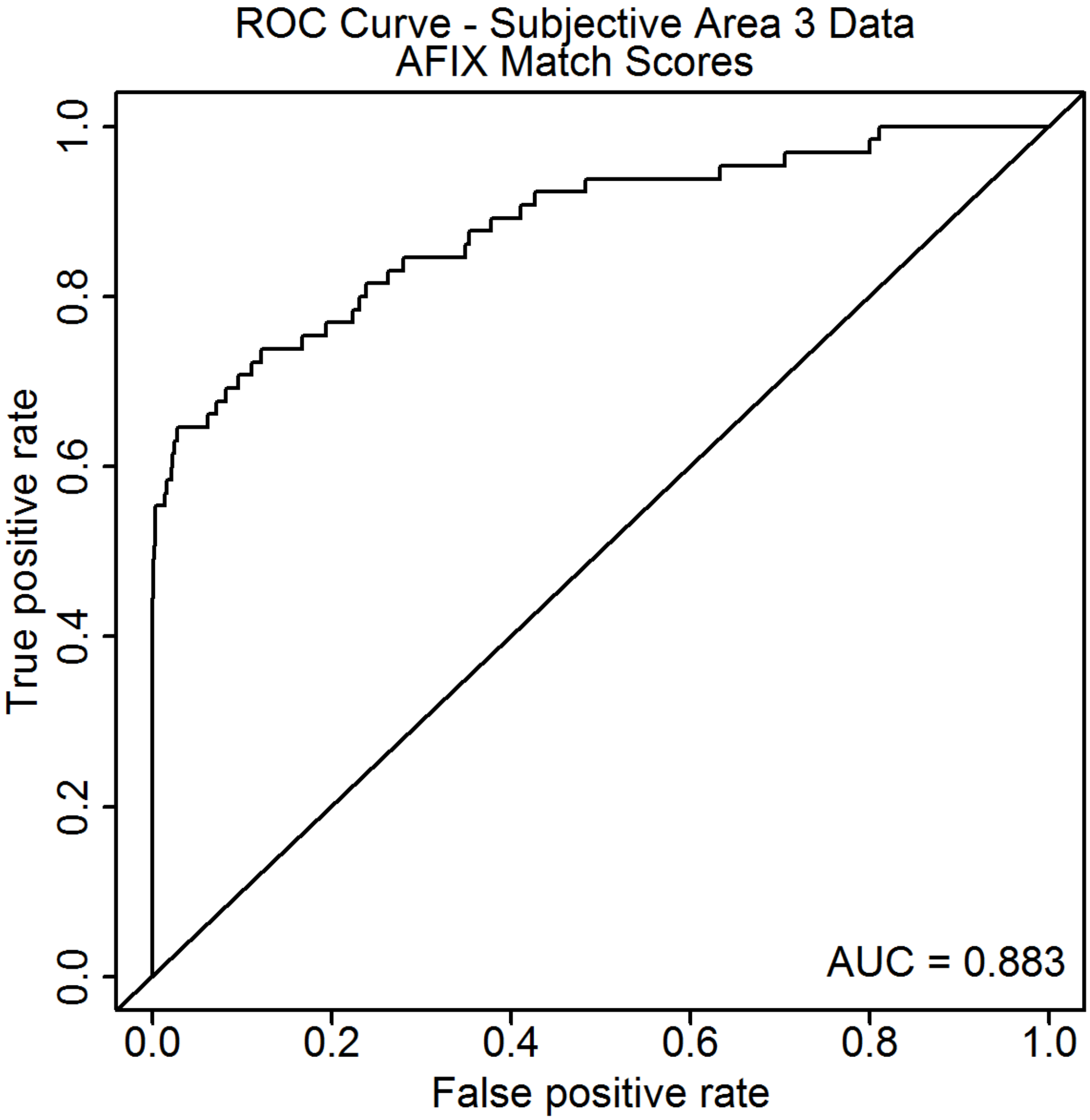


Figure D.67: This figure presents the ROC curve from the subjective Area 3 images for the match scores.

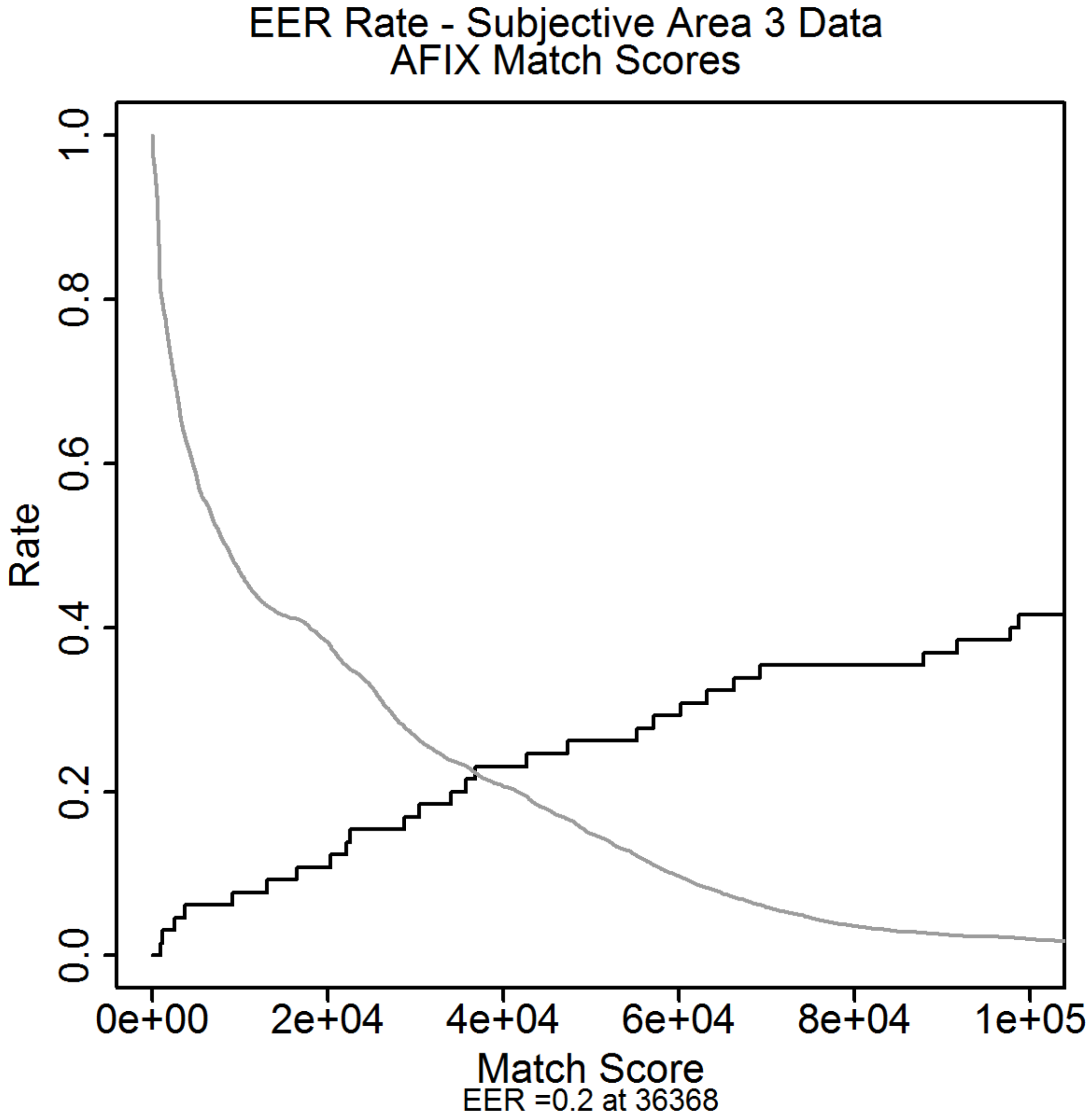


Figure D.68: This figure presents the EER plot from the subjective Area 3 images for the match scores.

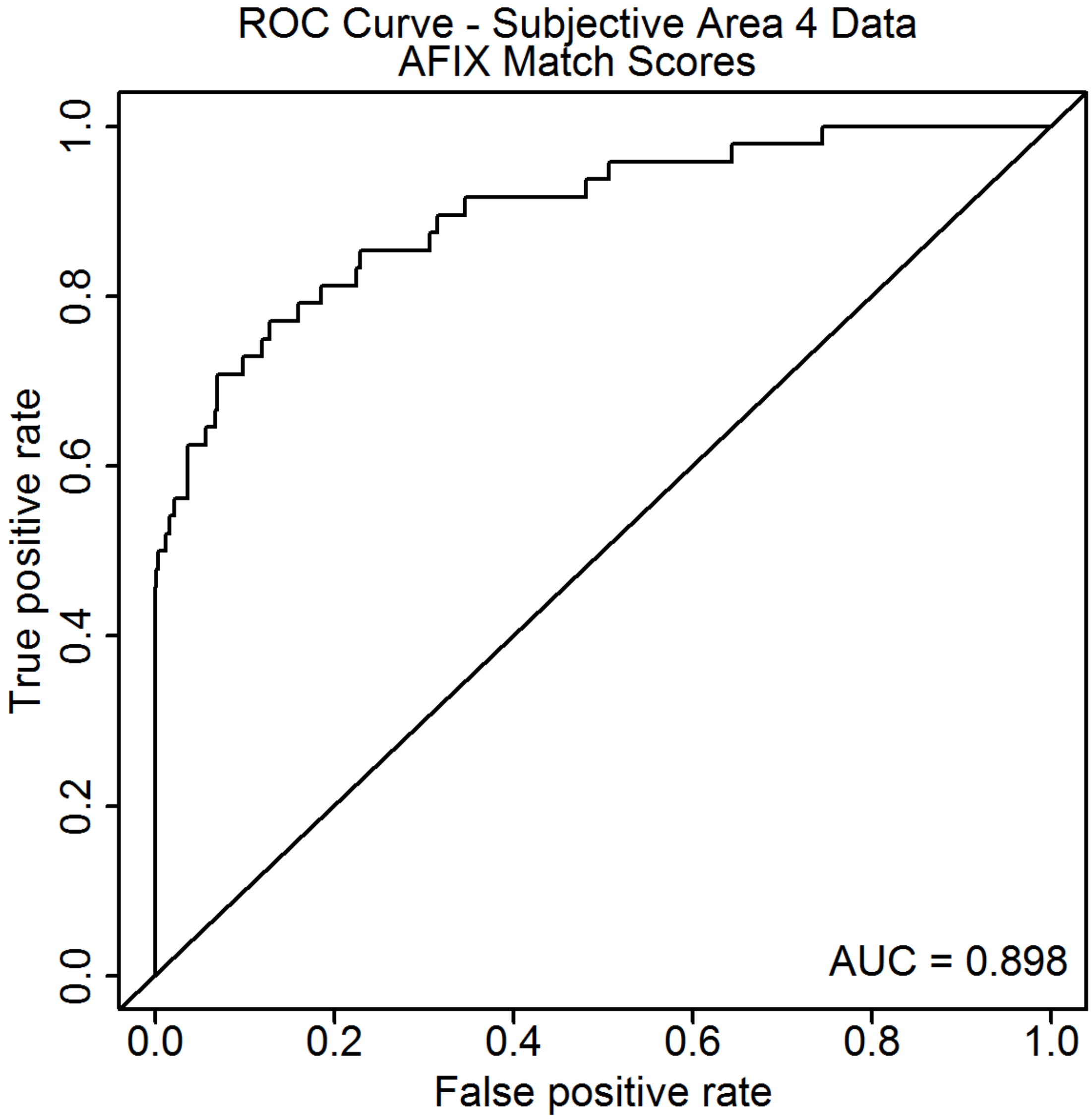


Figure D.69: This figure presents the ROC curve from the subjective Area 4 images for the match scores.

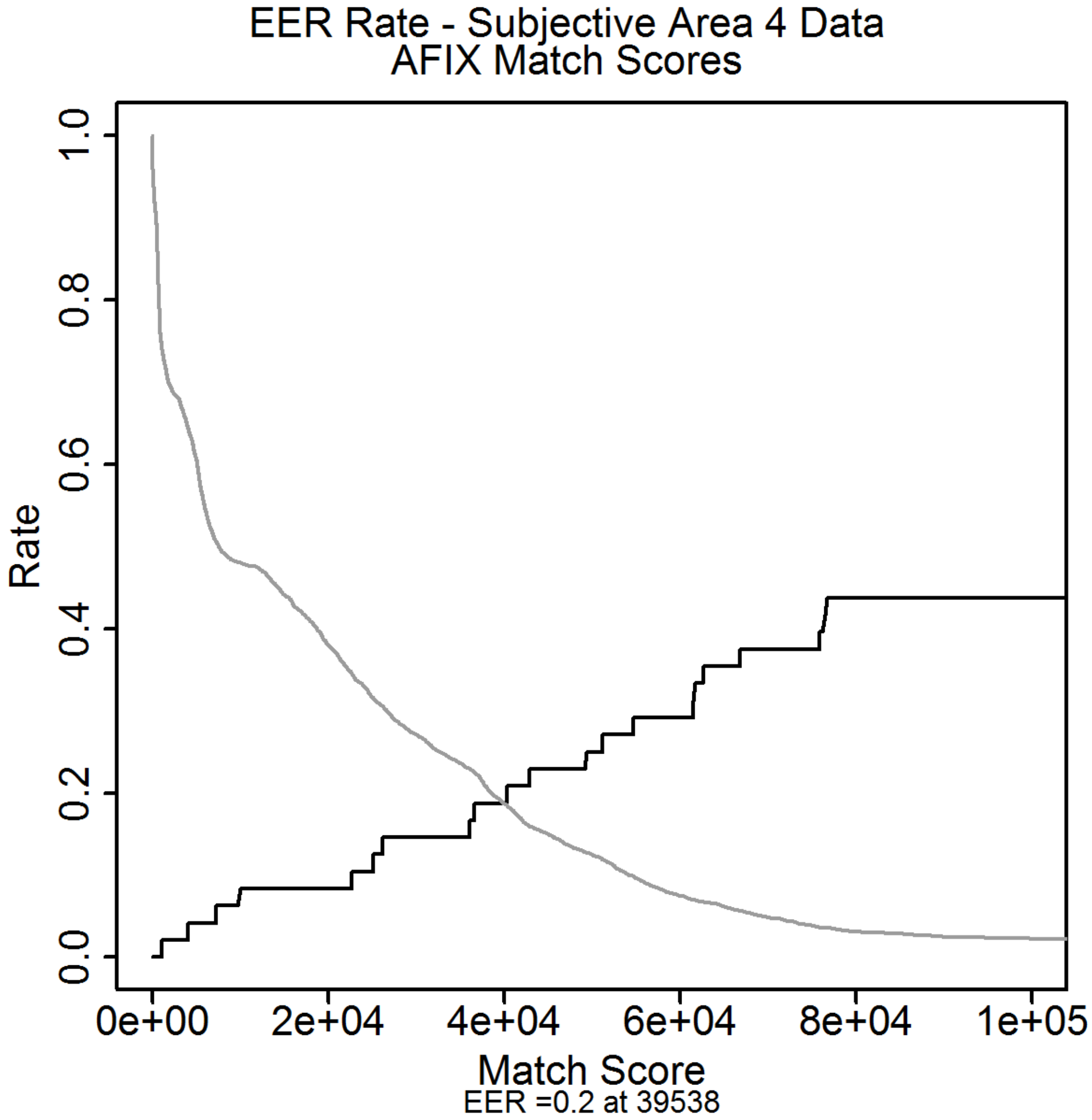


Figure D.70: This figure presents the EER plot from the subjective Area 4 images for the match scores.

D.7.2 Subjective Method: Automatic Extraction

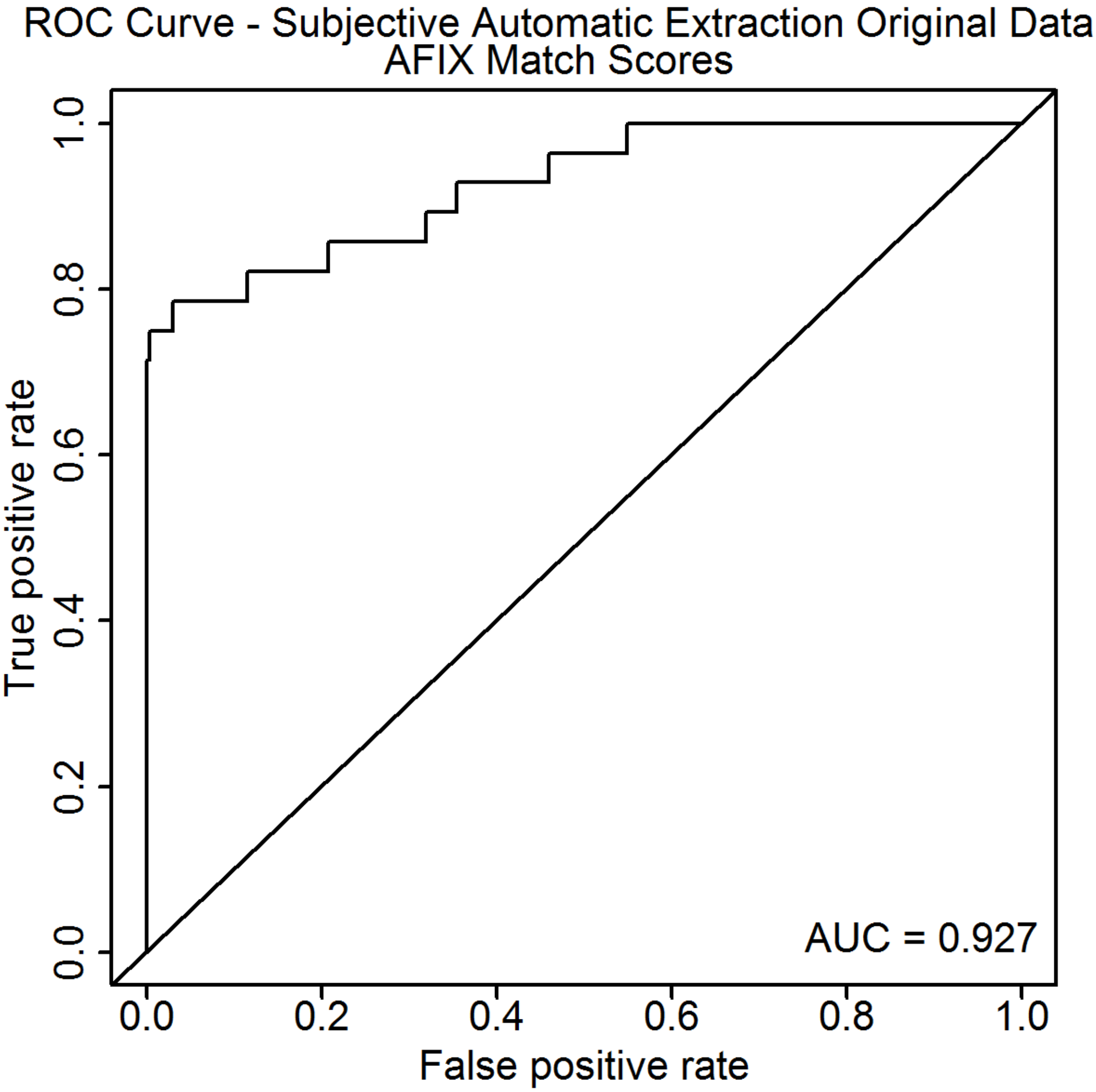


Figure D.71: This figure presents the ROC curve from the subjective original images for the match scores in the automatic extraction.

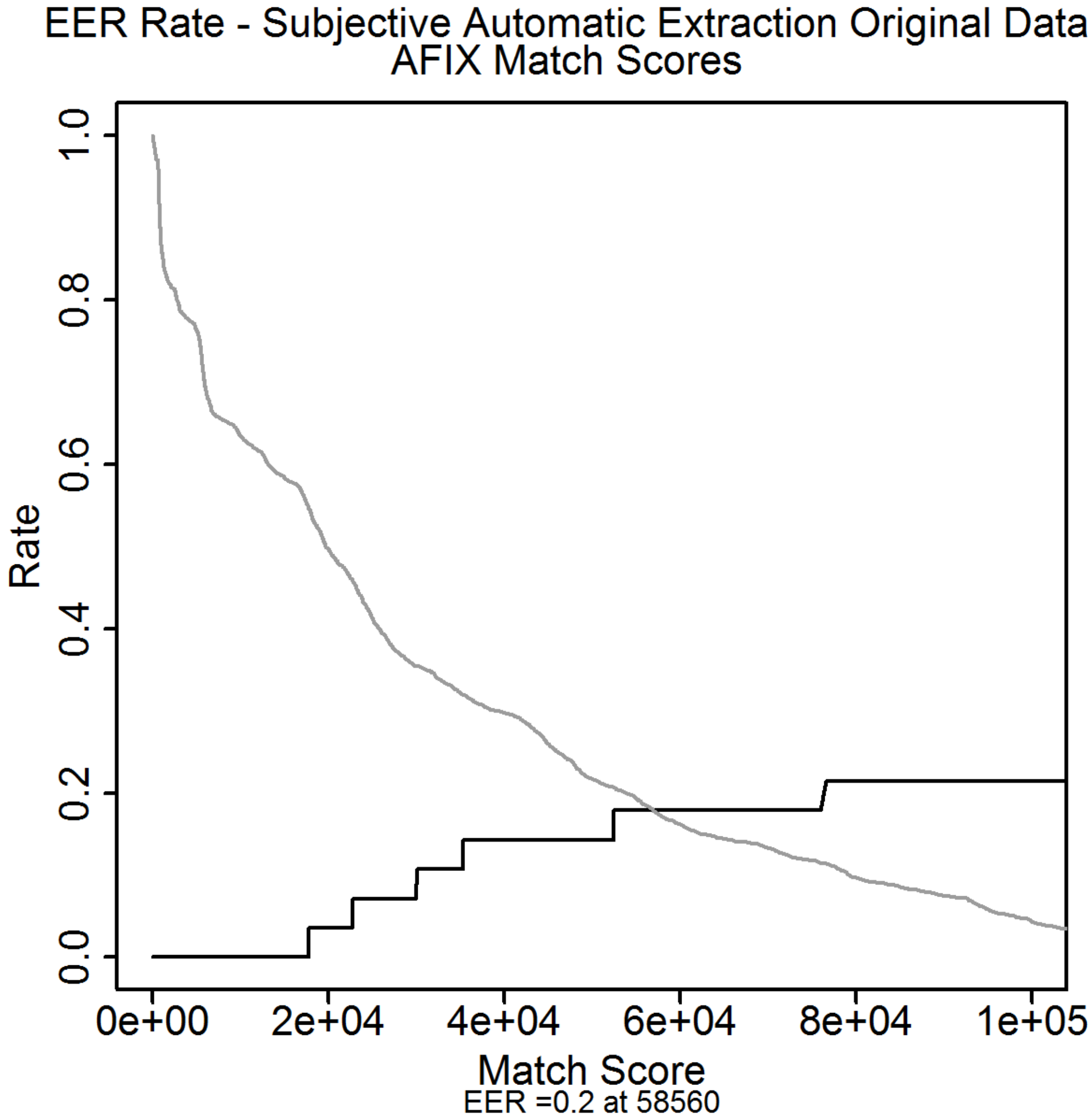


Figure D.72: This figure presents the EER plot from the subjective original images for the match scores in the automatic extraction.

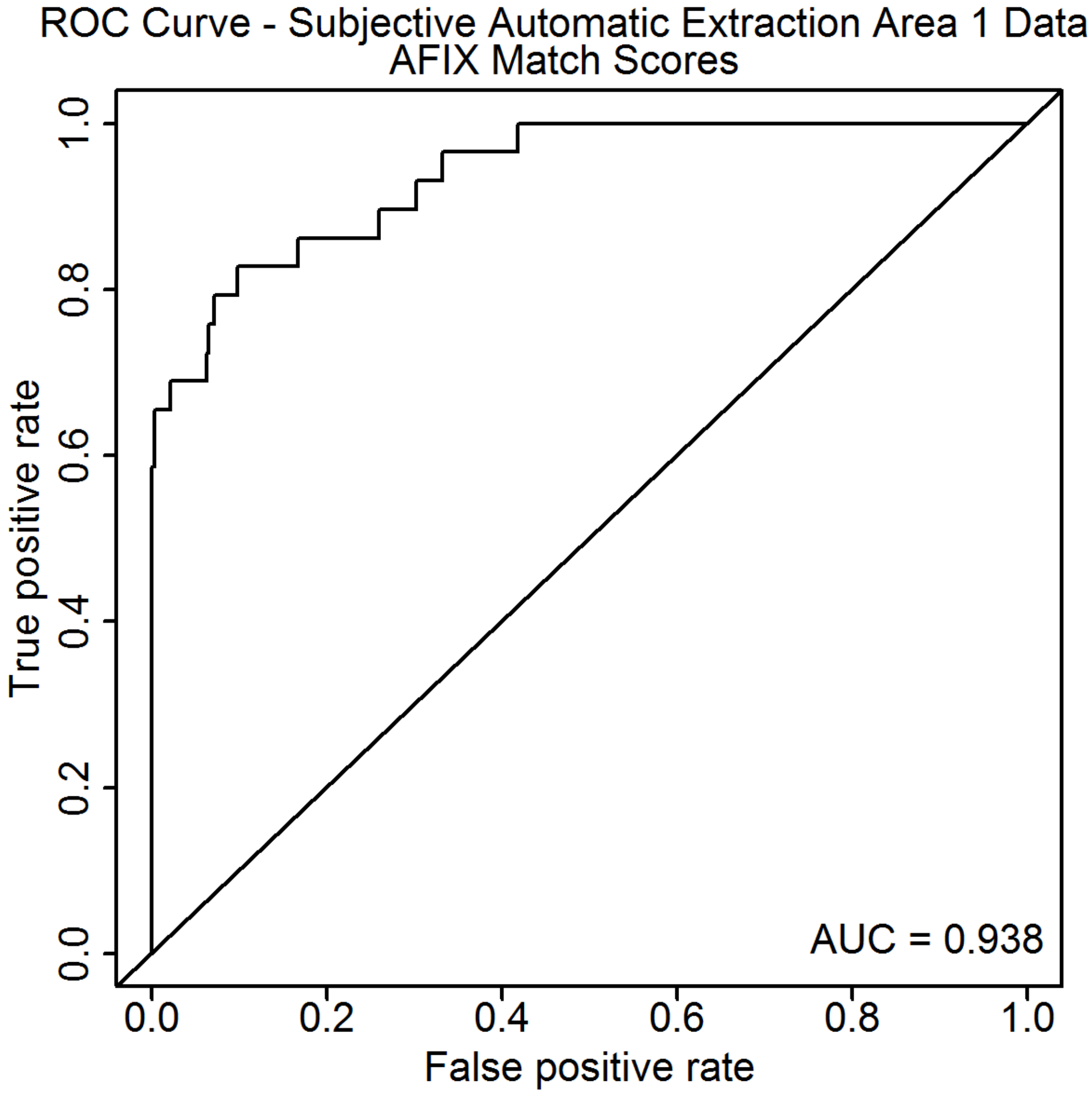


Figure D.73: This figure presents the ROC curve from the subjective Area 1 images for the match scores in the automatic extraction.

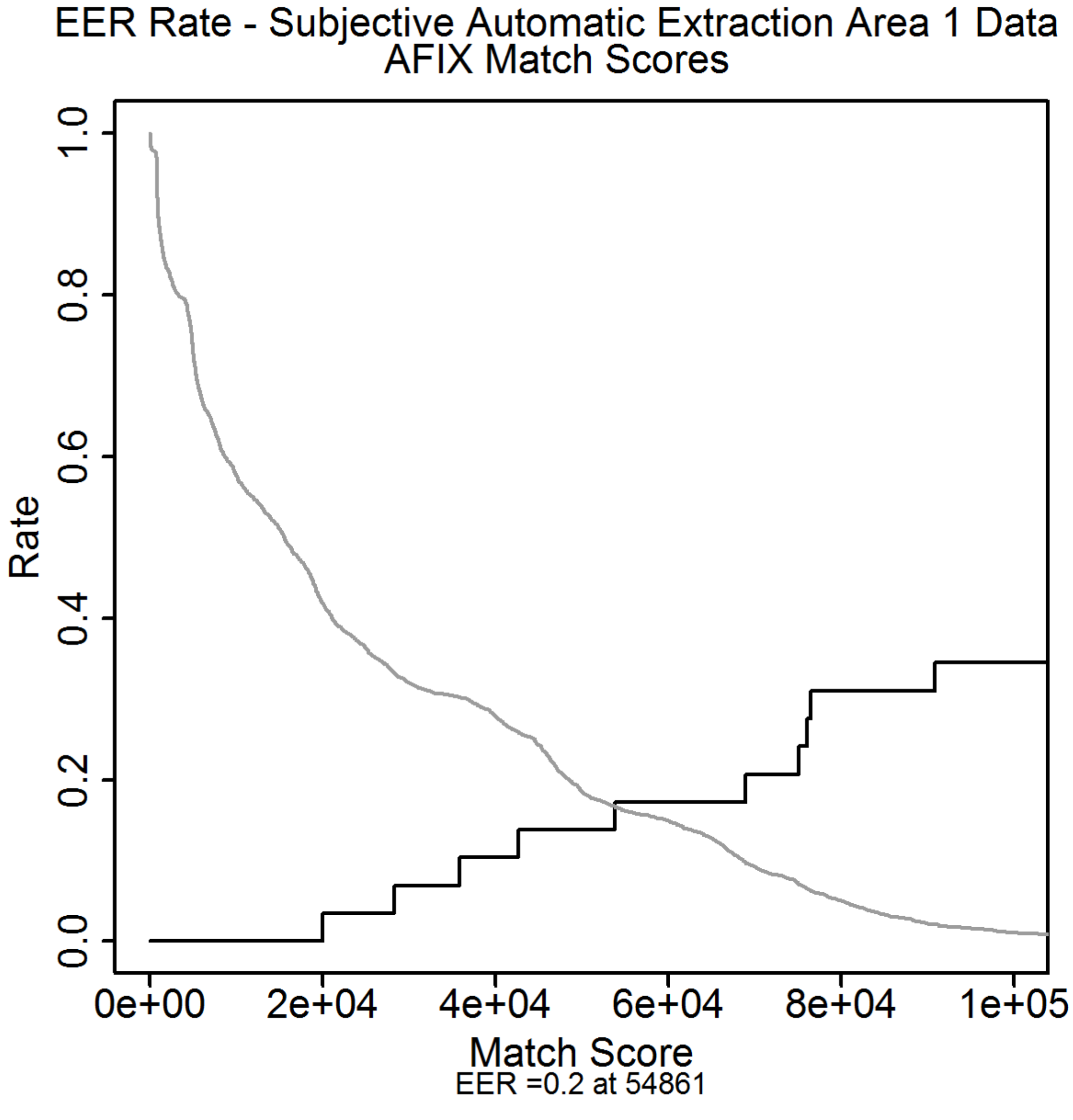


Figure D.74: This figure presents the EER plot from the subjective Area 1 images for the match scores in the automatic extraction.

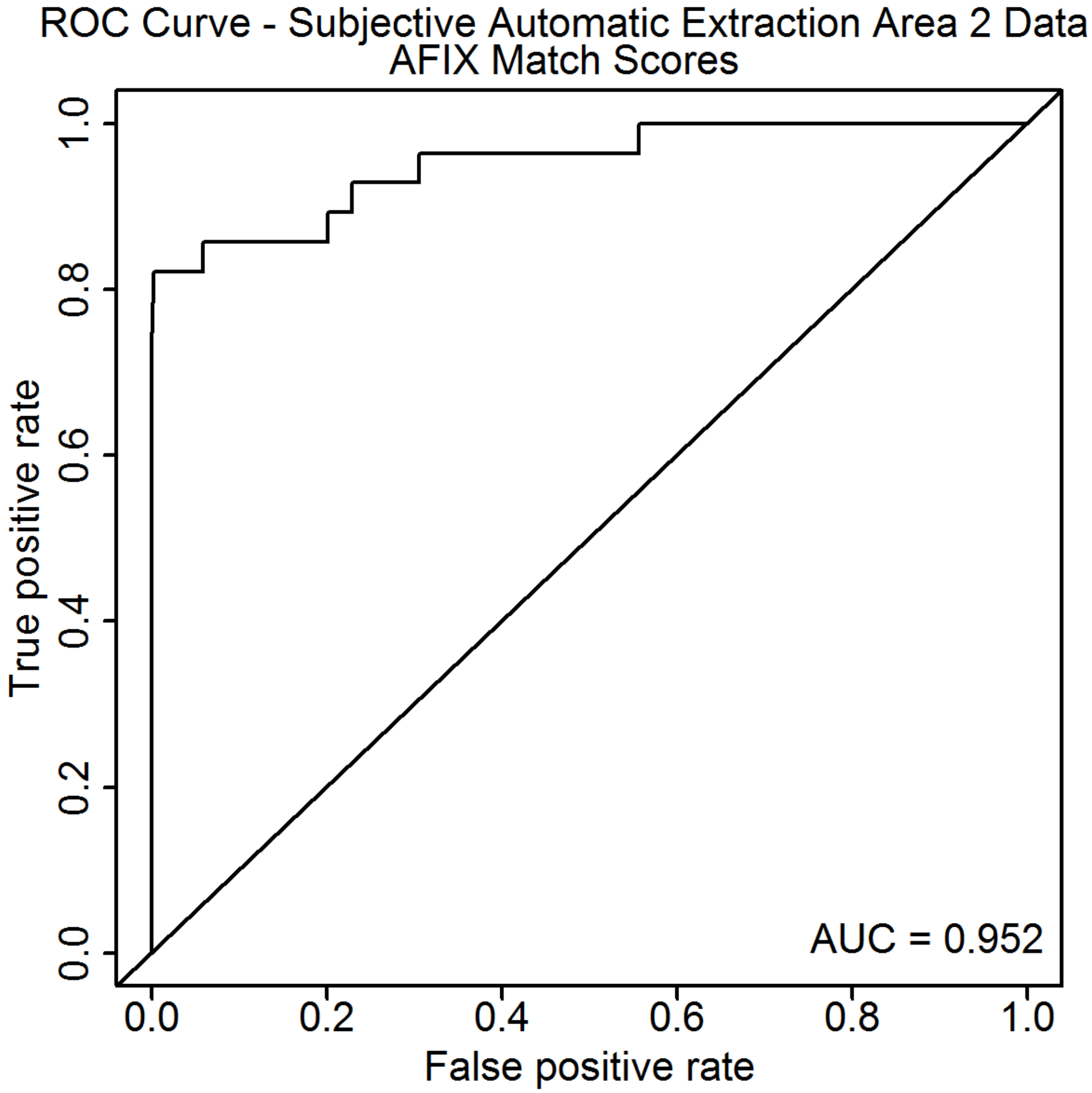


Figure D.75: This figure presents the ROC curve from the subjective Area 2 images for the match scores in the automatic extraction.

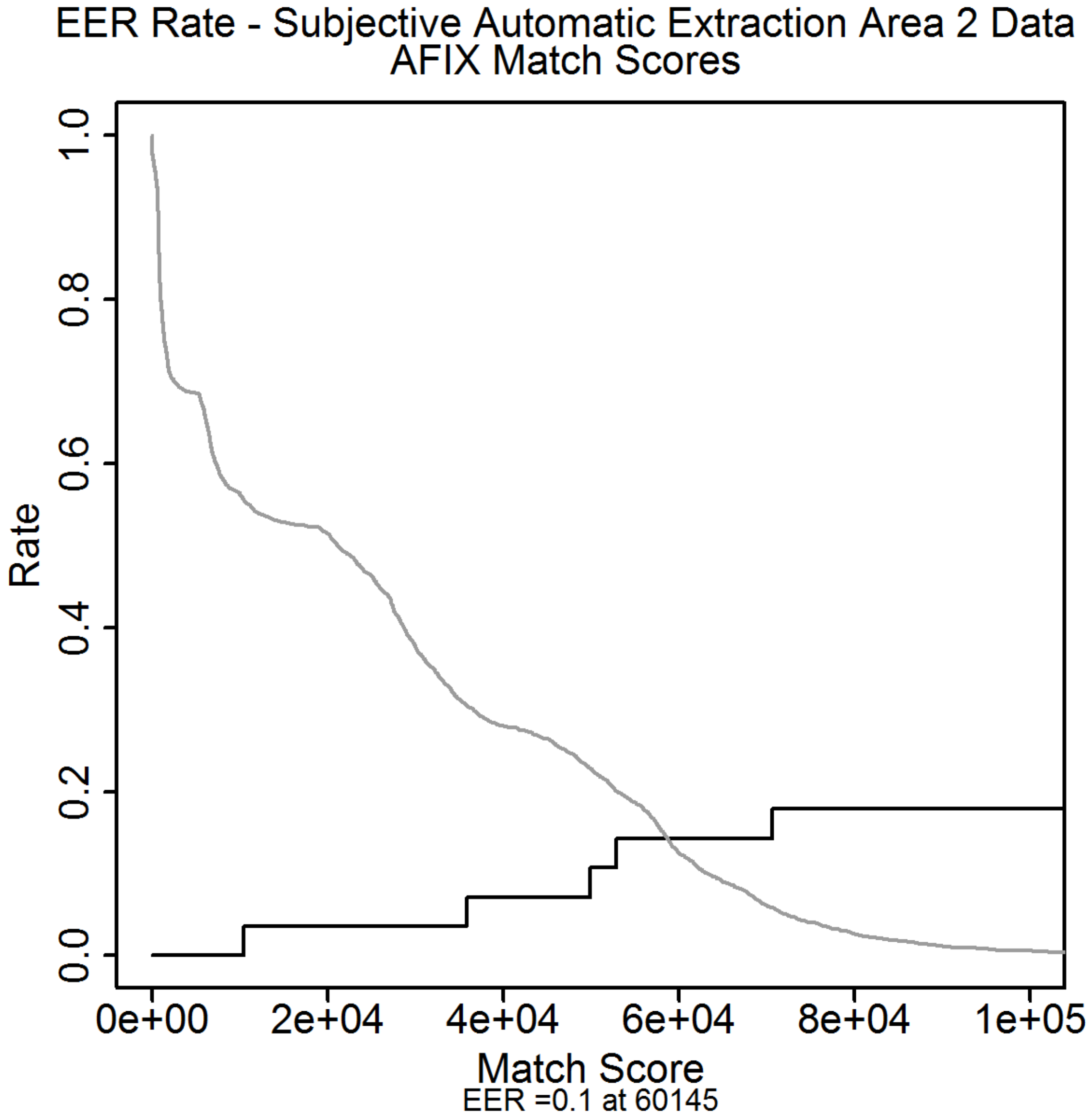


Figure D.76: This figure presents the EER plot from the subjective Area 2 images for the match scores in the automatic extraction.

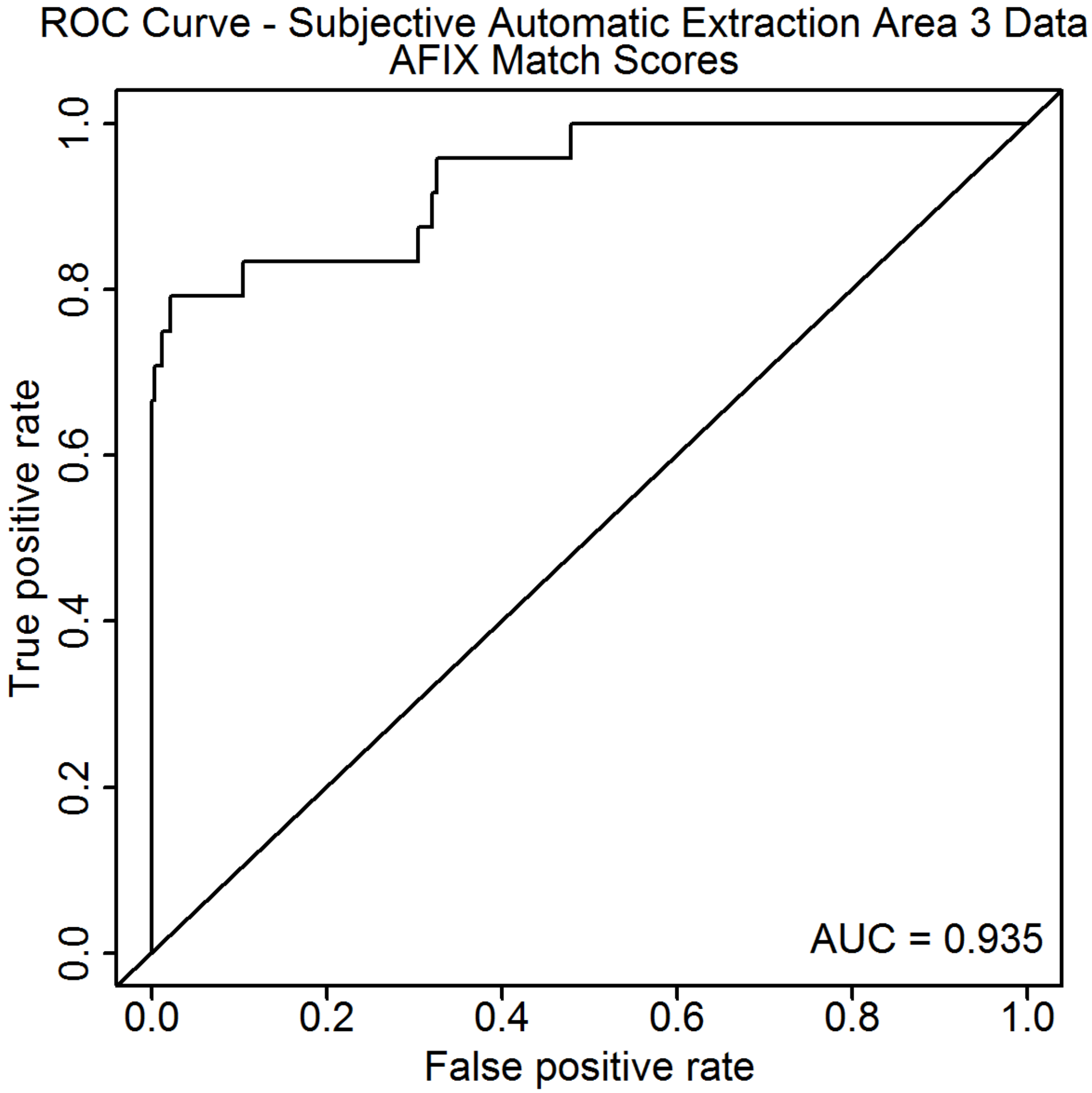


Figure D.77: This figure presents the ROC curve from the subjective Area 3 images for the match scores in the automatic extraction.

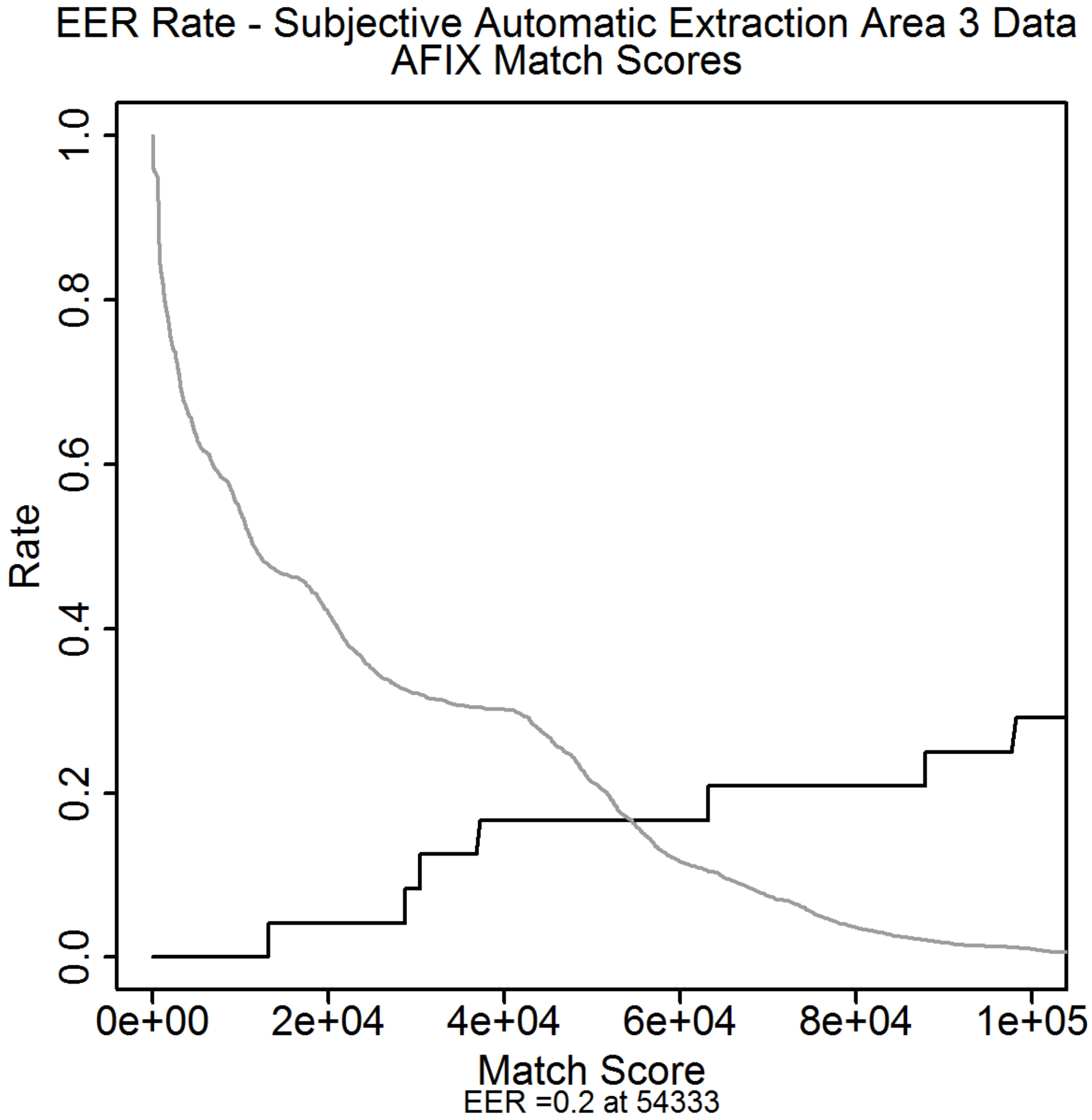


Figure D.78: This figure presents the EER plot from the subjective Area 3 images for the match scores in the automatic extraction.

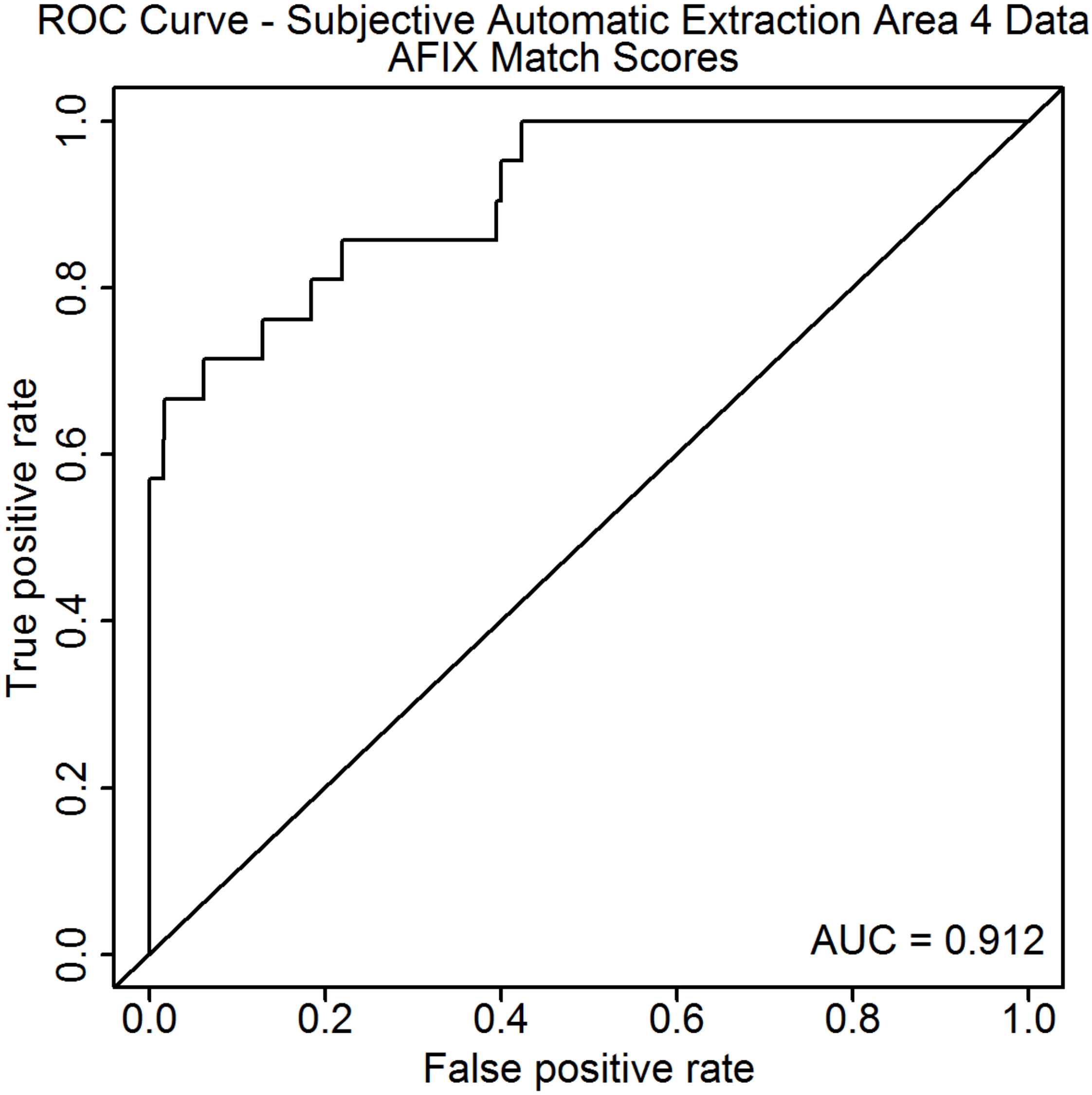


Figure D.79: This figure presents the ROC curve from the subjective Area 4 images for the match scores in the automatic extraction.

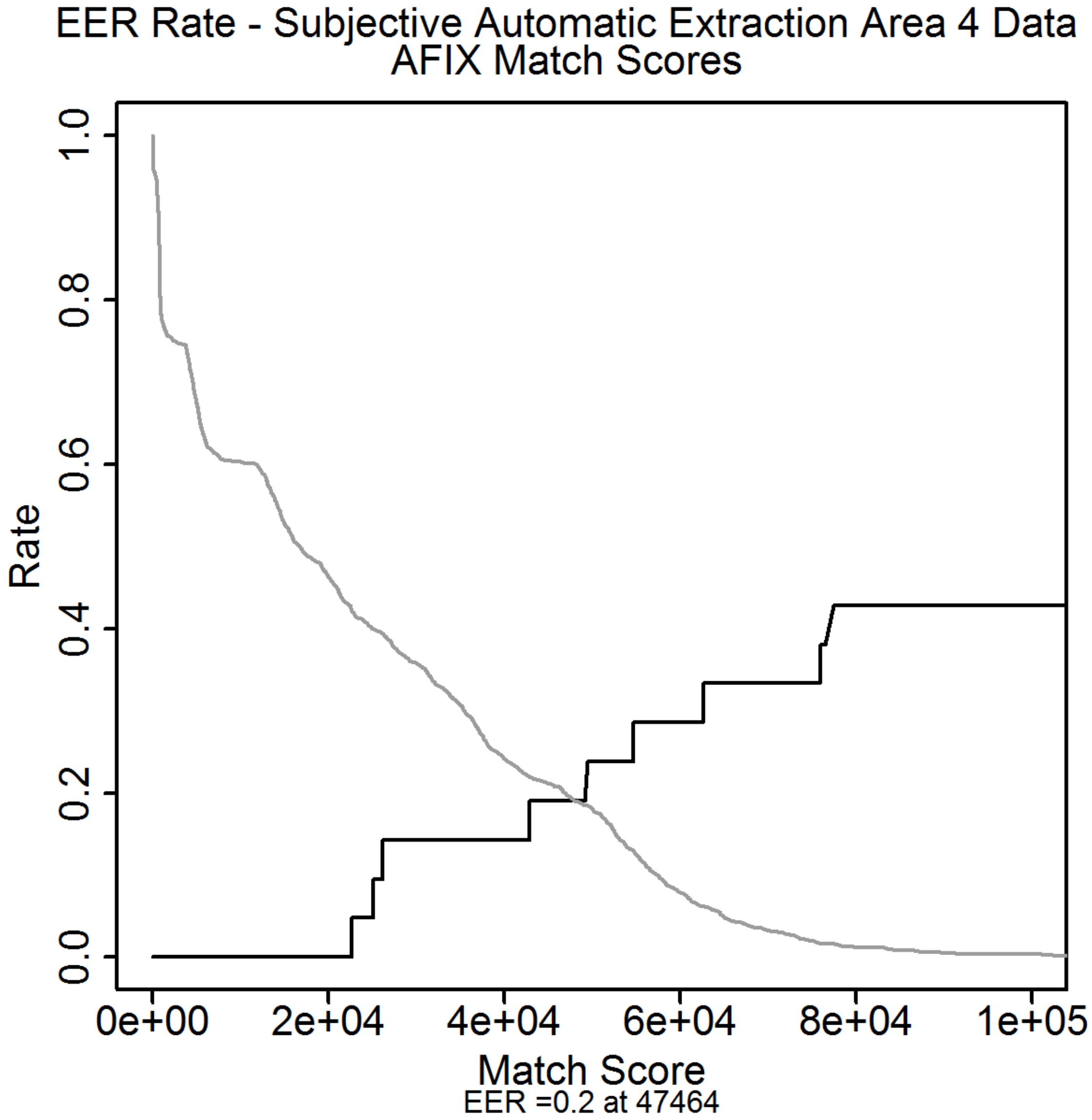


Figure D.80: This figure presents the EER plot from the subjective Area 4 images for the match scores in the automatic extraction.

D.7.3 Subjective Method: Manual Extraction

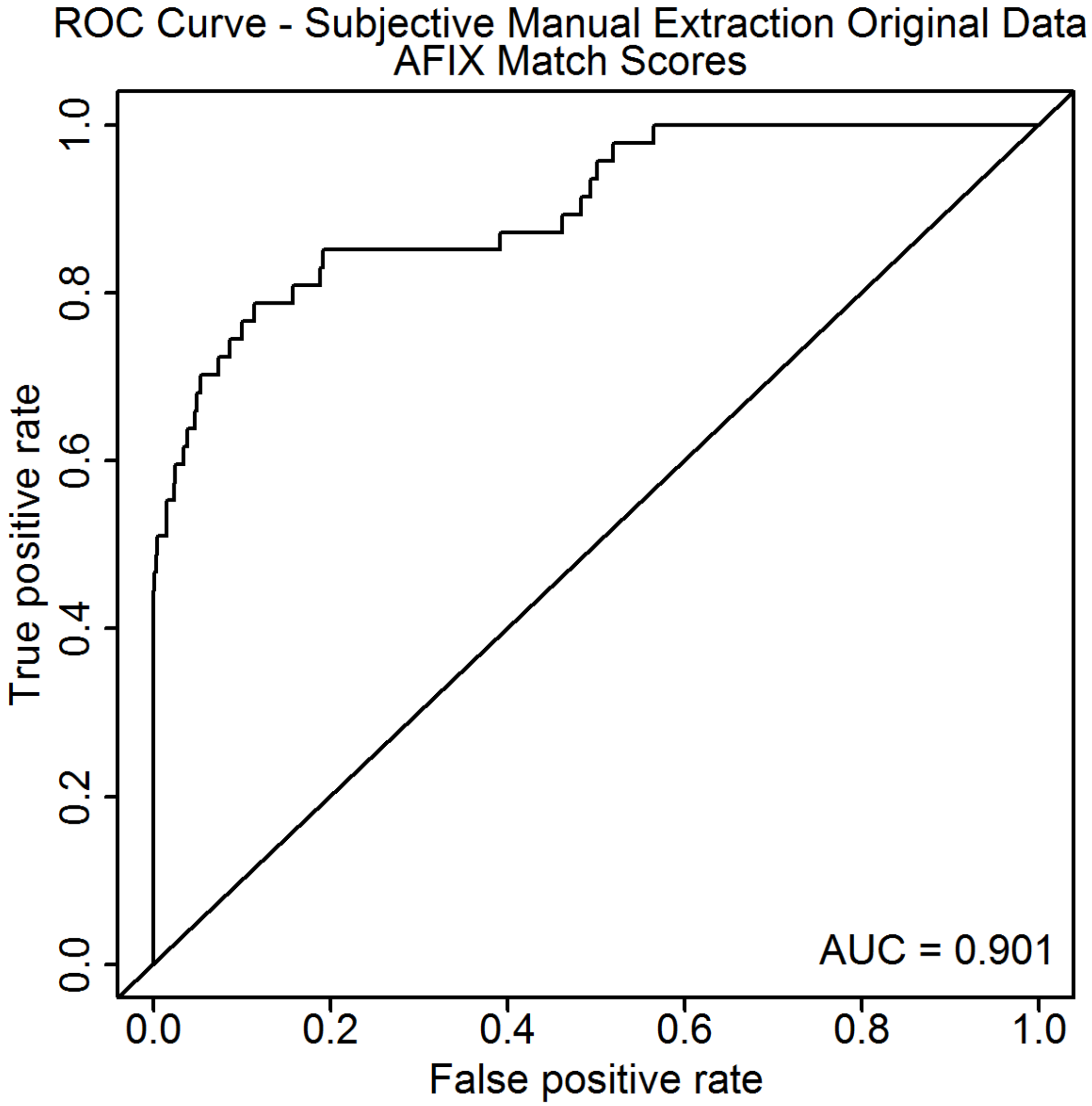


Figure D.81: This figure presents the ROC curve from the subjective original images for the match scores in the manual extraction.

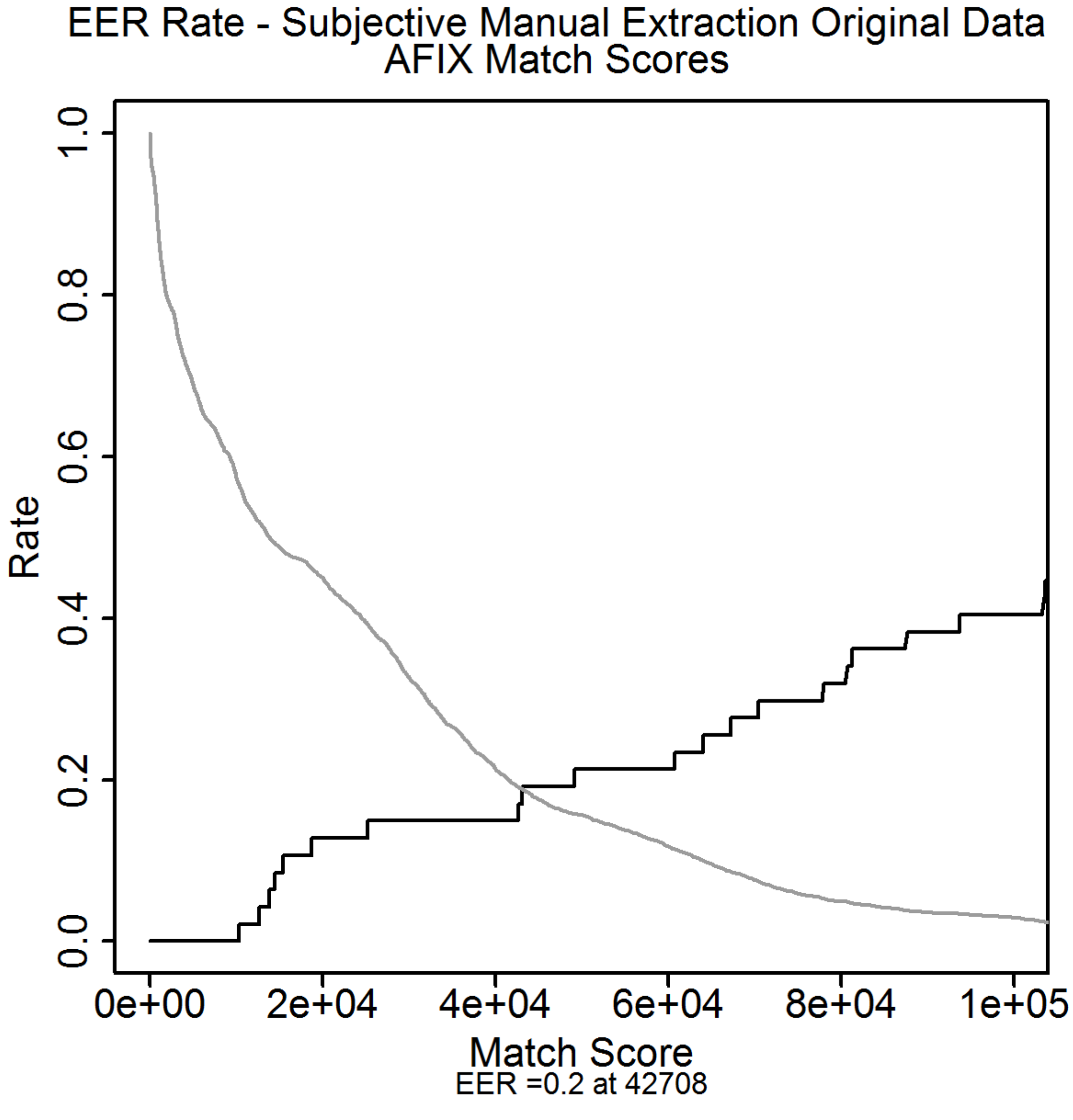


Figure D.82: This figure presents the EER plot from the subjective original images for the match scores in the manual extraction.

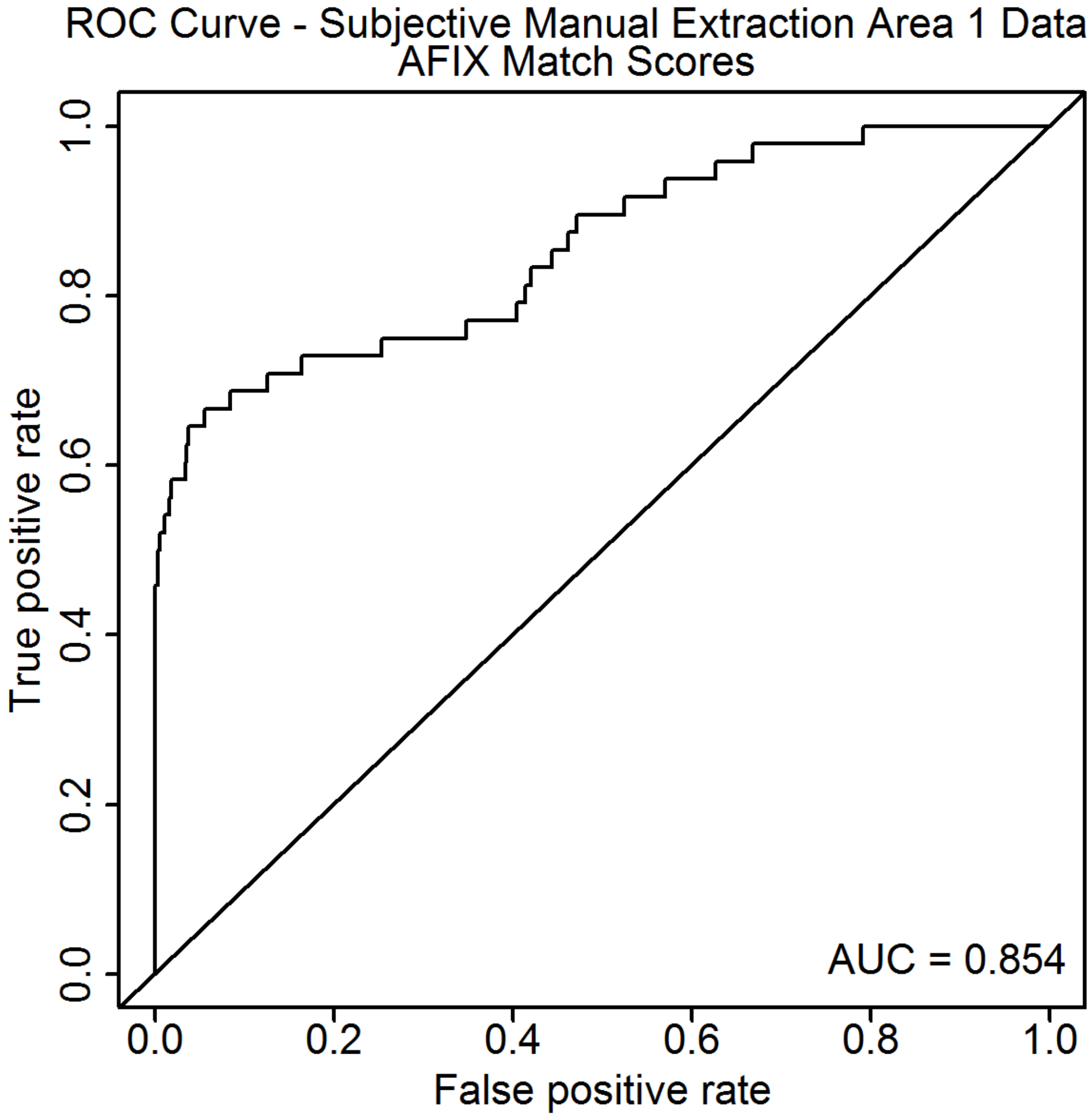


Figure D.83: This figure presents the ROC curve from the subjective Area 1 images for the match scores in the manual extraction.

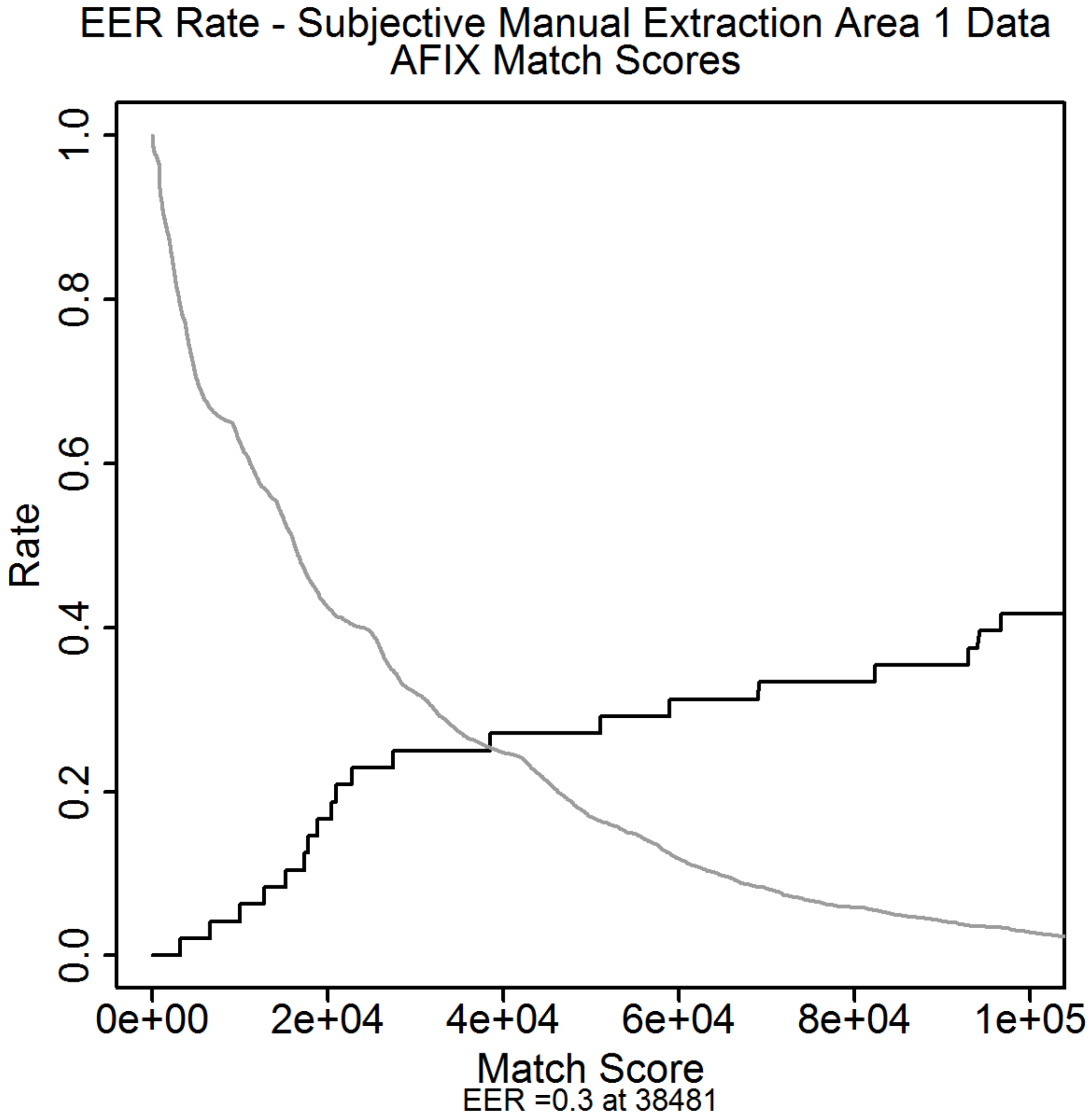


Figure D.84: This figure presents the EER plot from the subjective Area 1 images for the match scores in the manual extraction.

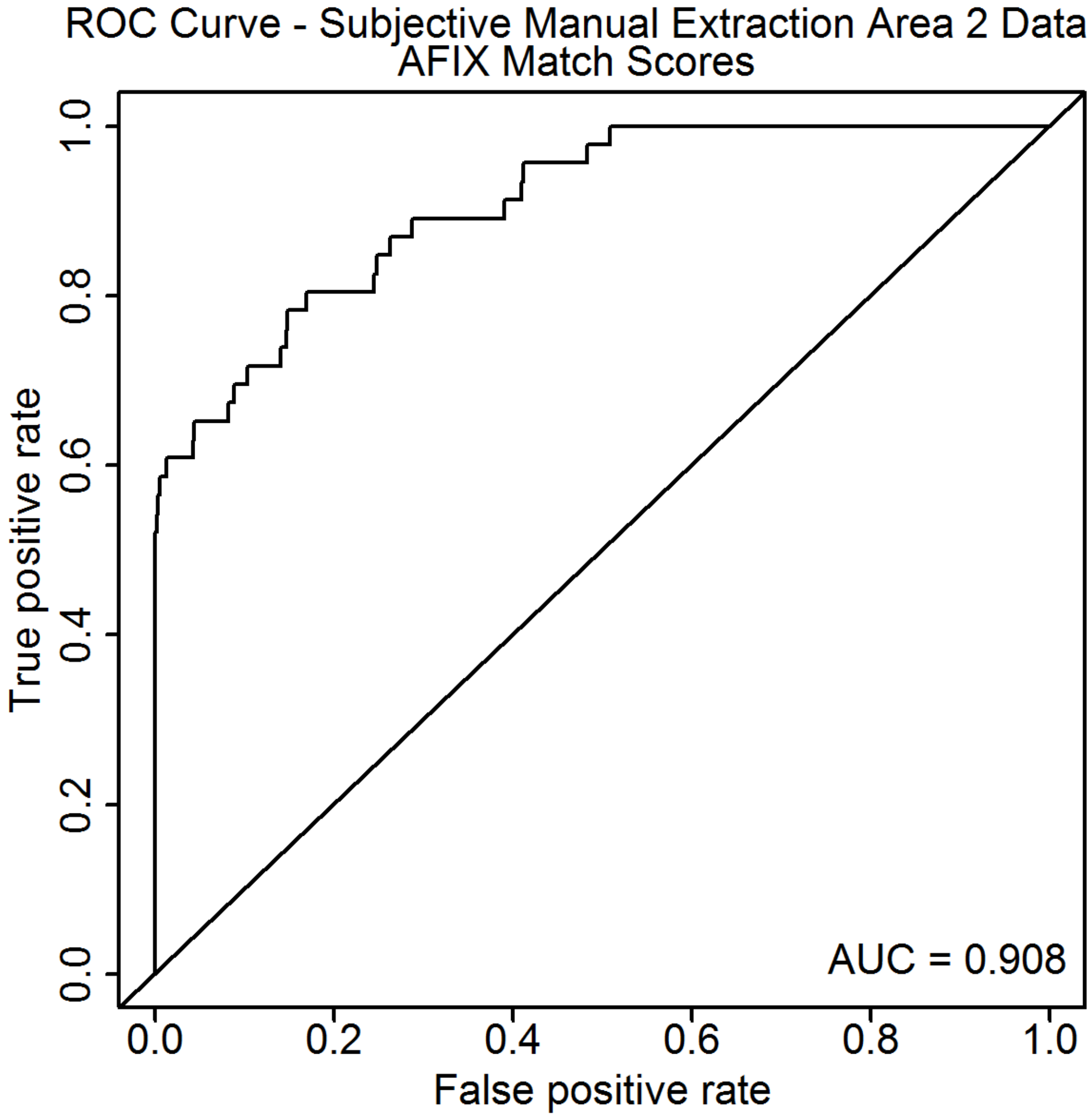


Figure D.85: This figure presents the ROC curve from the subjective Area 2 images for the match scores in the manual extraction.

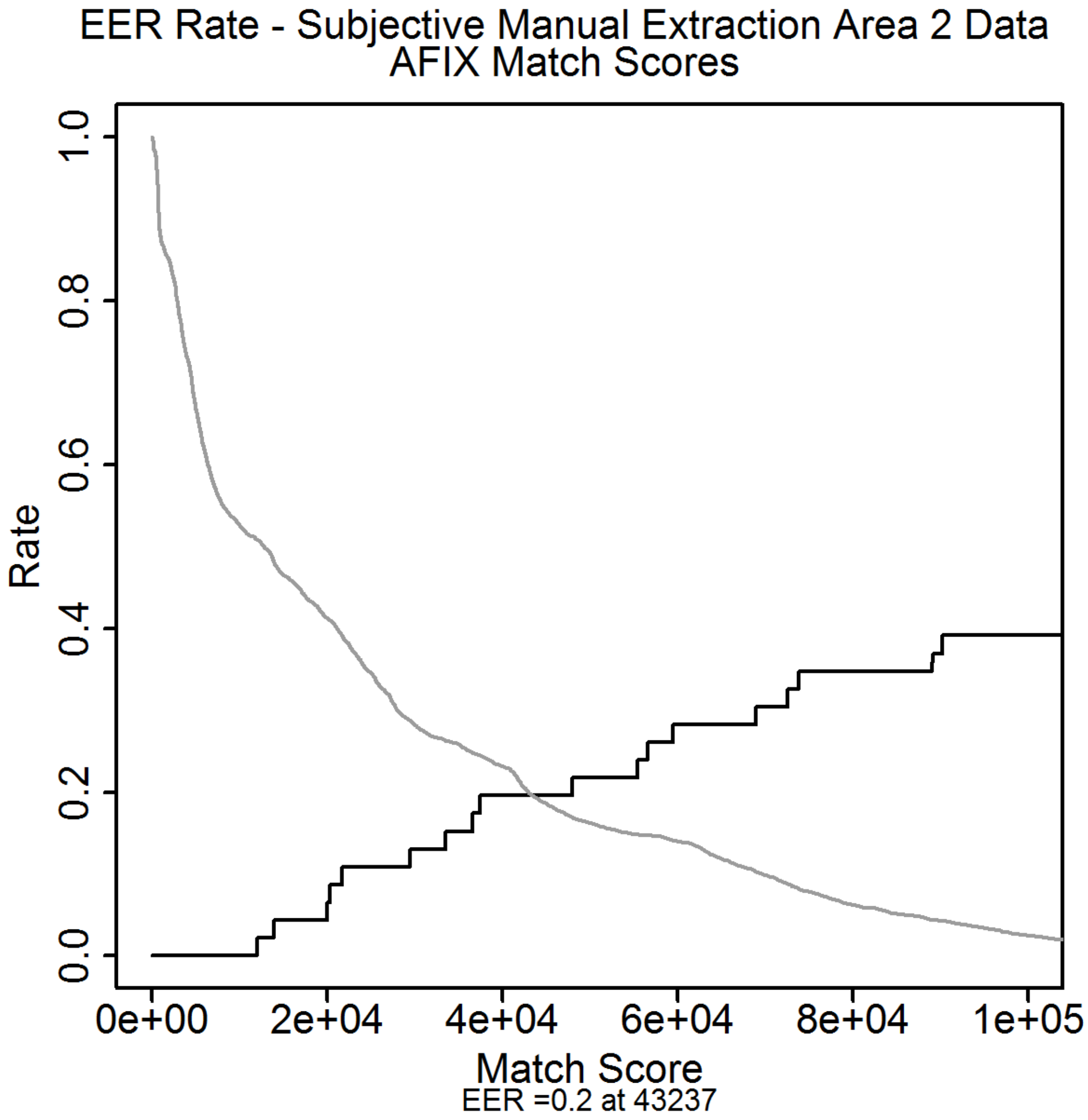


Figure D.86: This figure presents the EER plot from the subjective Area 2 images for the match scores in the manual extraction.

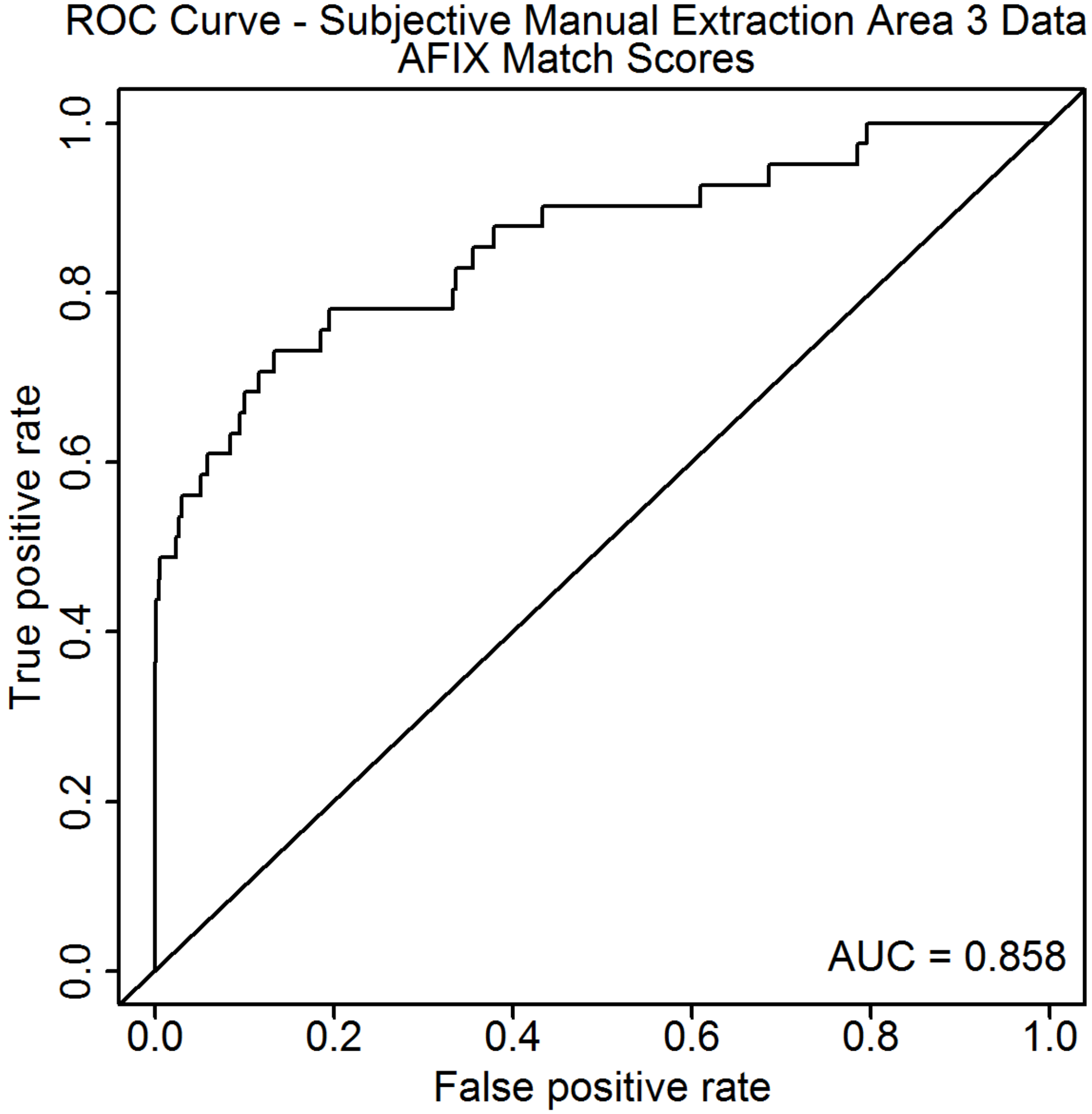


Figure D.87: This figure presents the ROC curve from the subjective Area 3 images for the match scores in the manual extraction.

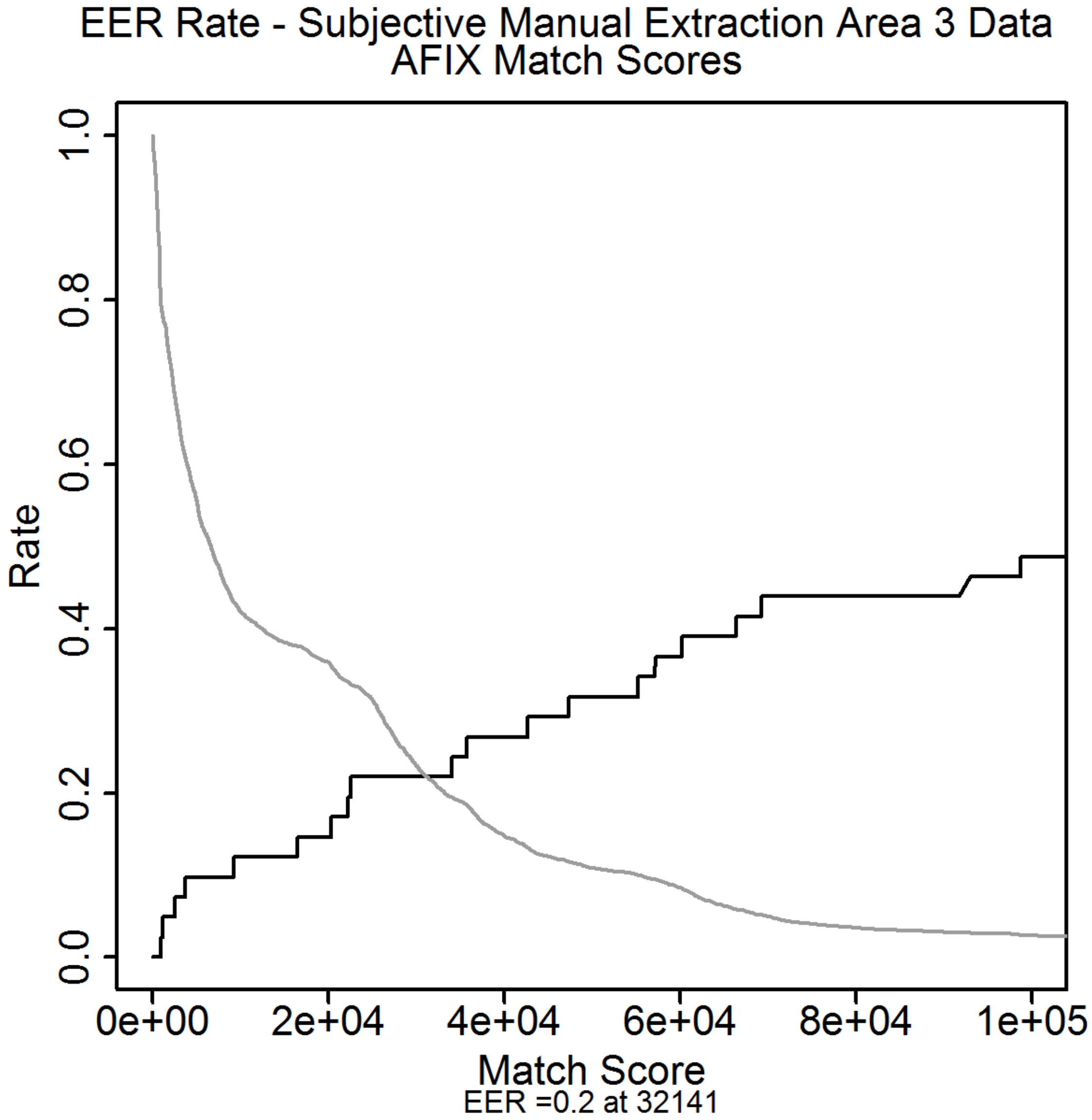


Figure D.88: This figure presents the EER plot from the subjective Area 3 images for the match scores in the manual extraction.

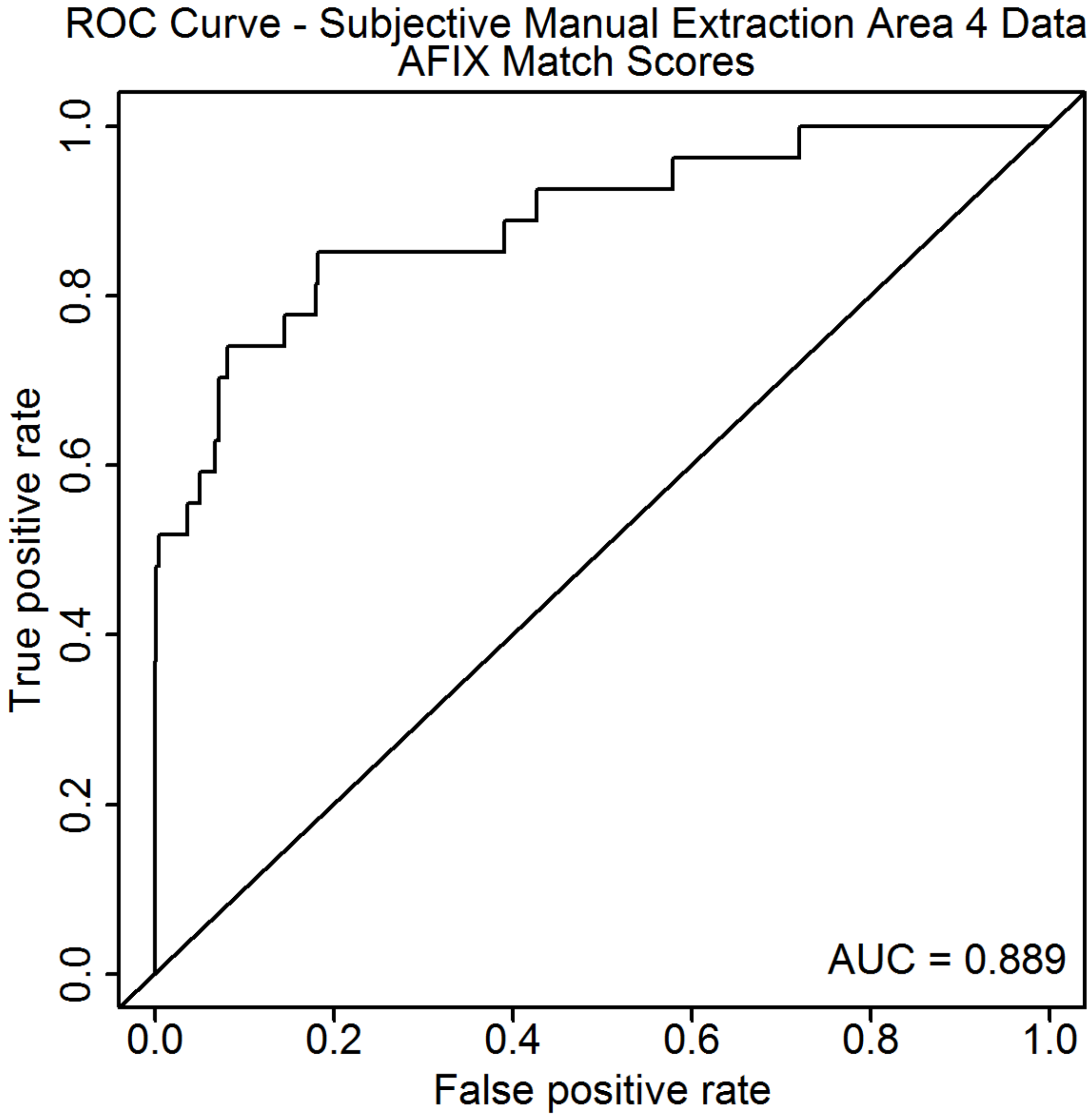


Figure D.89: This figure presents the ROC curve from the subjective Area 4 images for the match scores in the manual extraction.

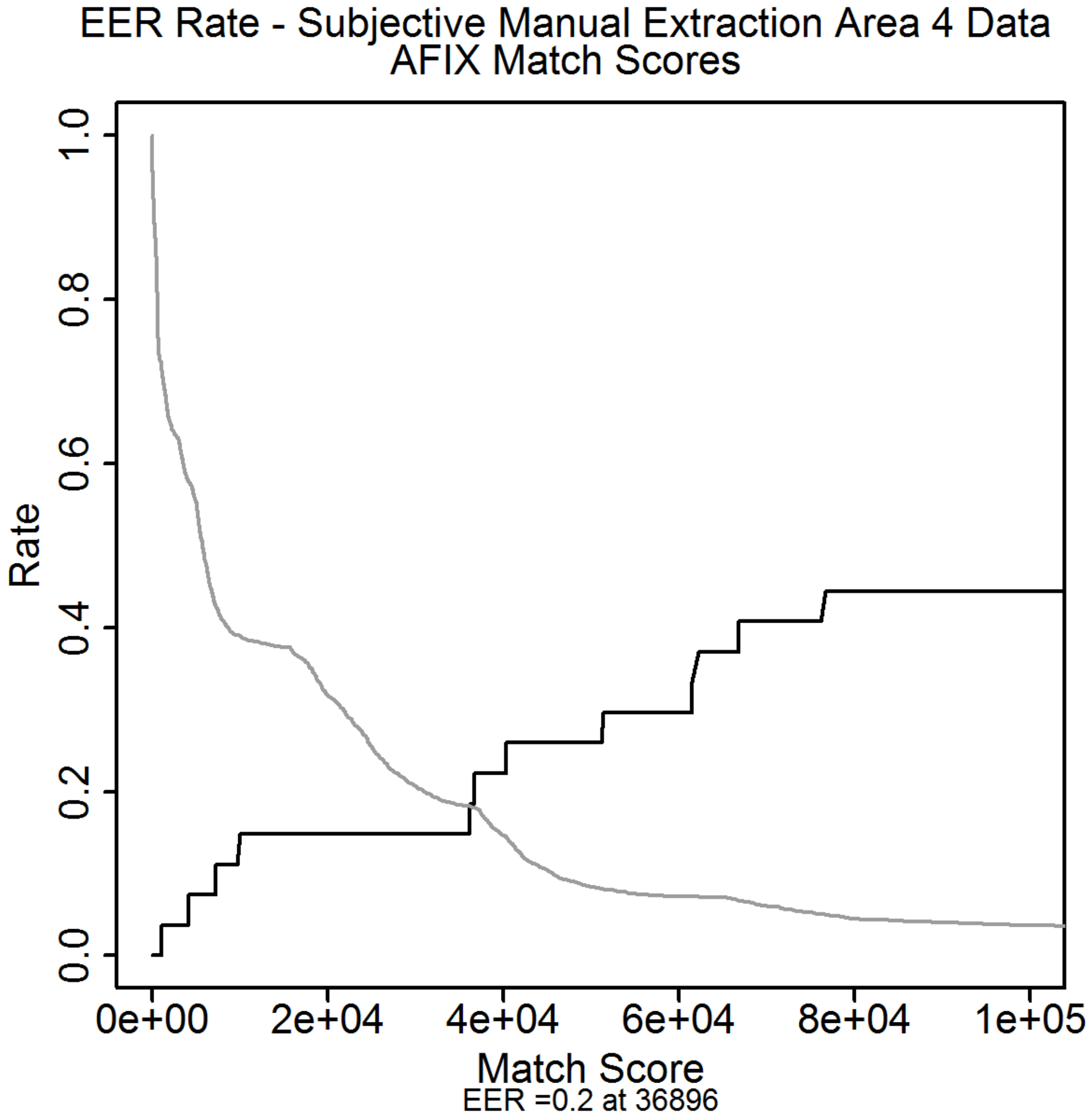


Figure D.90: This figure presents the EER plot from the subjective Area 4 images for the match scores in the manual extraction.

D.7.4 Objective Method: All Samples

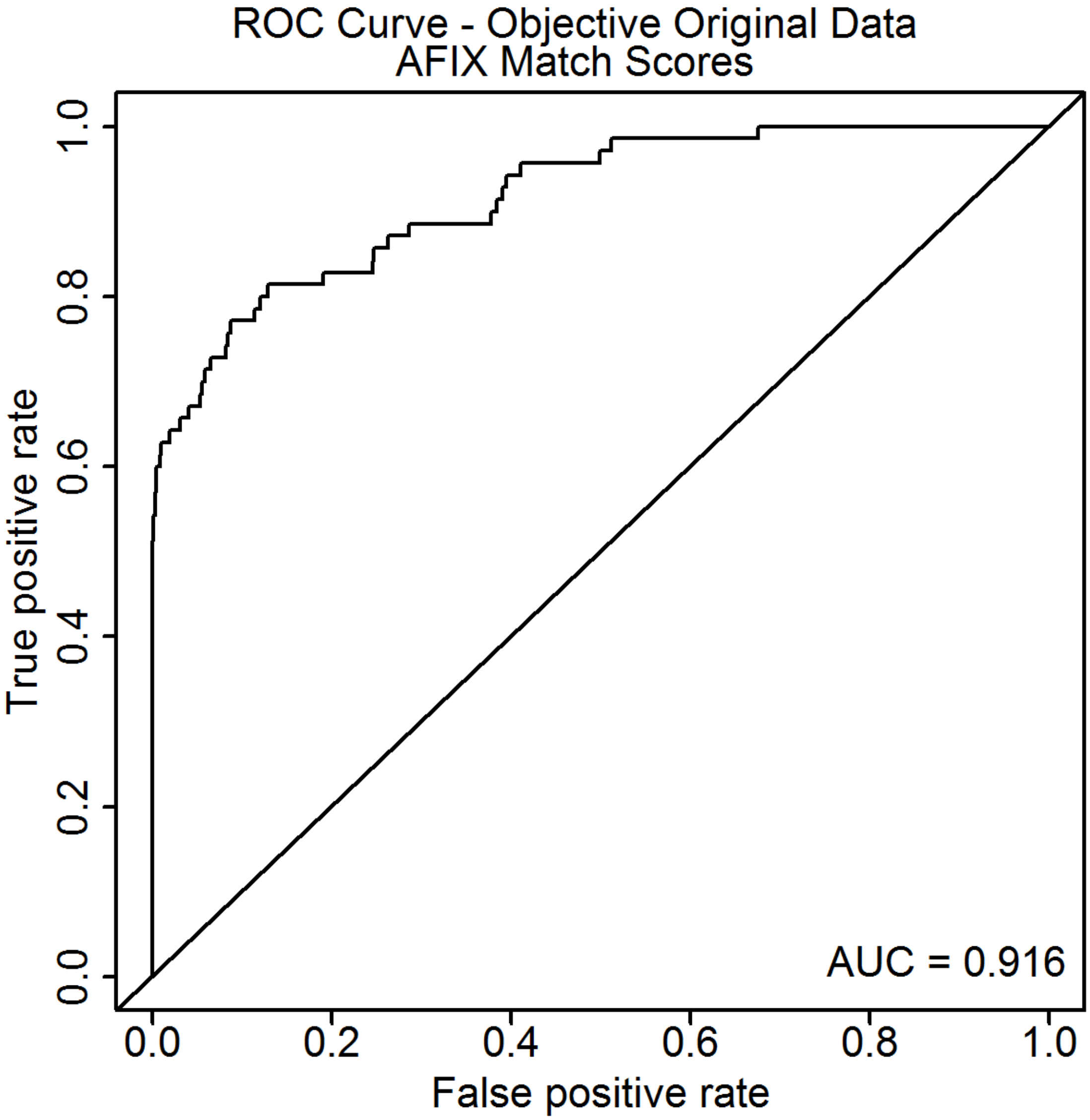


Figure D.91: This figure presents the ROC curve from the objective original images for the match scores in the automatic extraction.

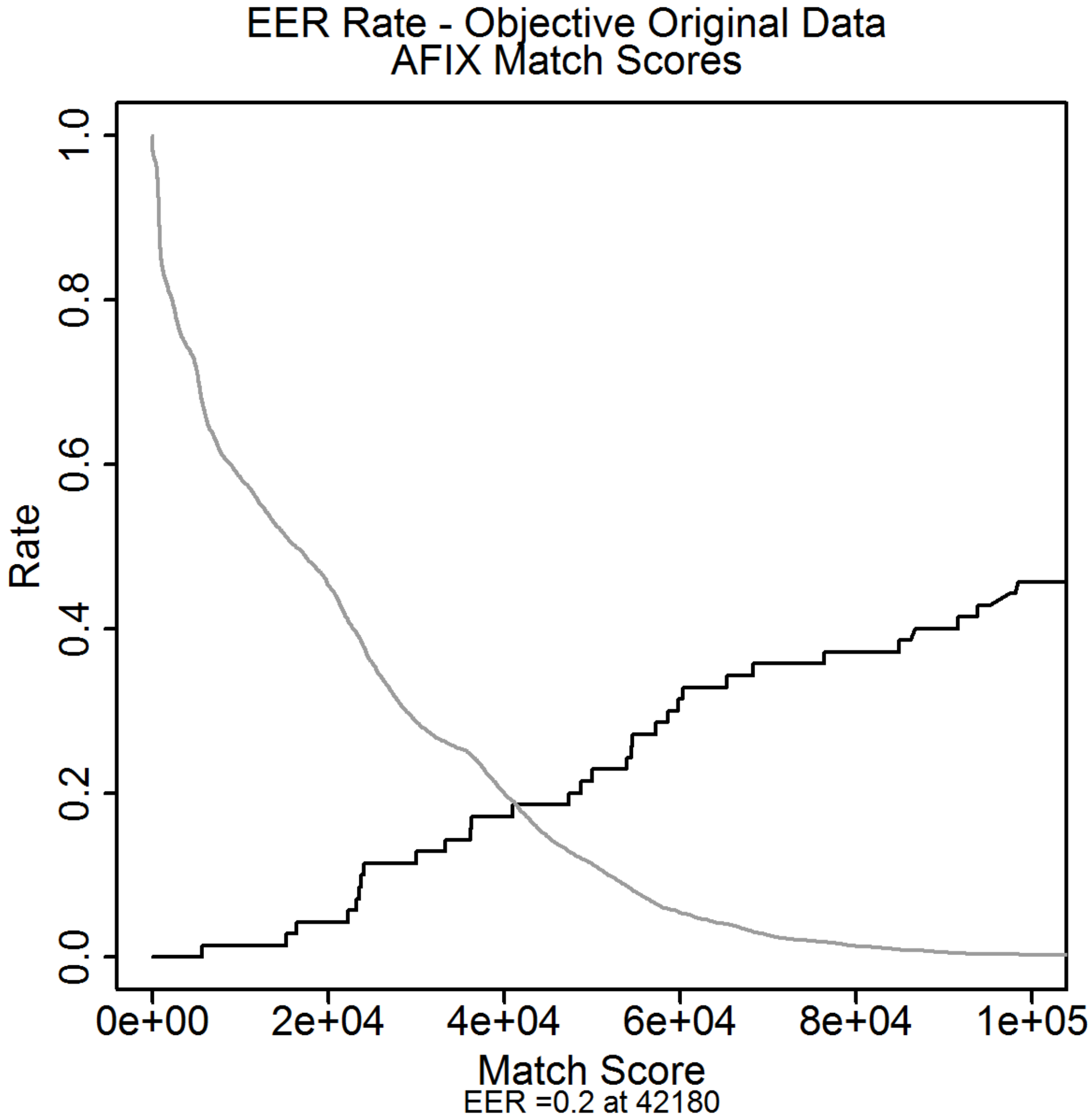


Figure D.92: This figure presents the EER plot from the objective original images for the match scores in the automatic extraction.

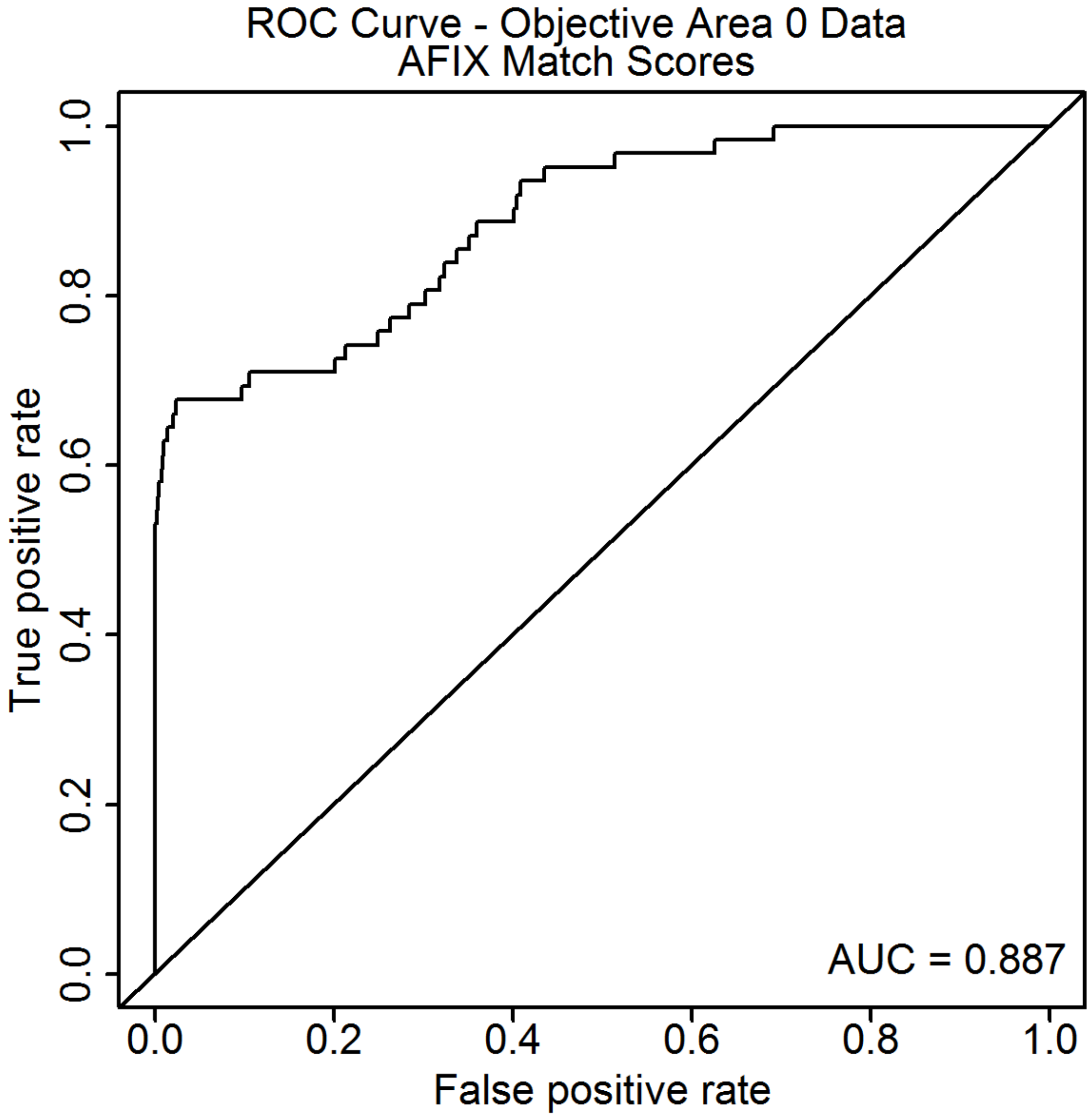


Figure D.93: This figure presents the ROC curve from the objective Area 0 images for the match scores in the automatic extraction.

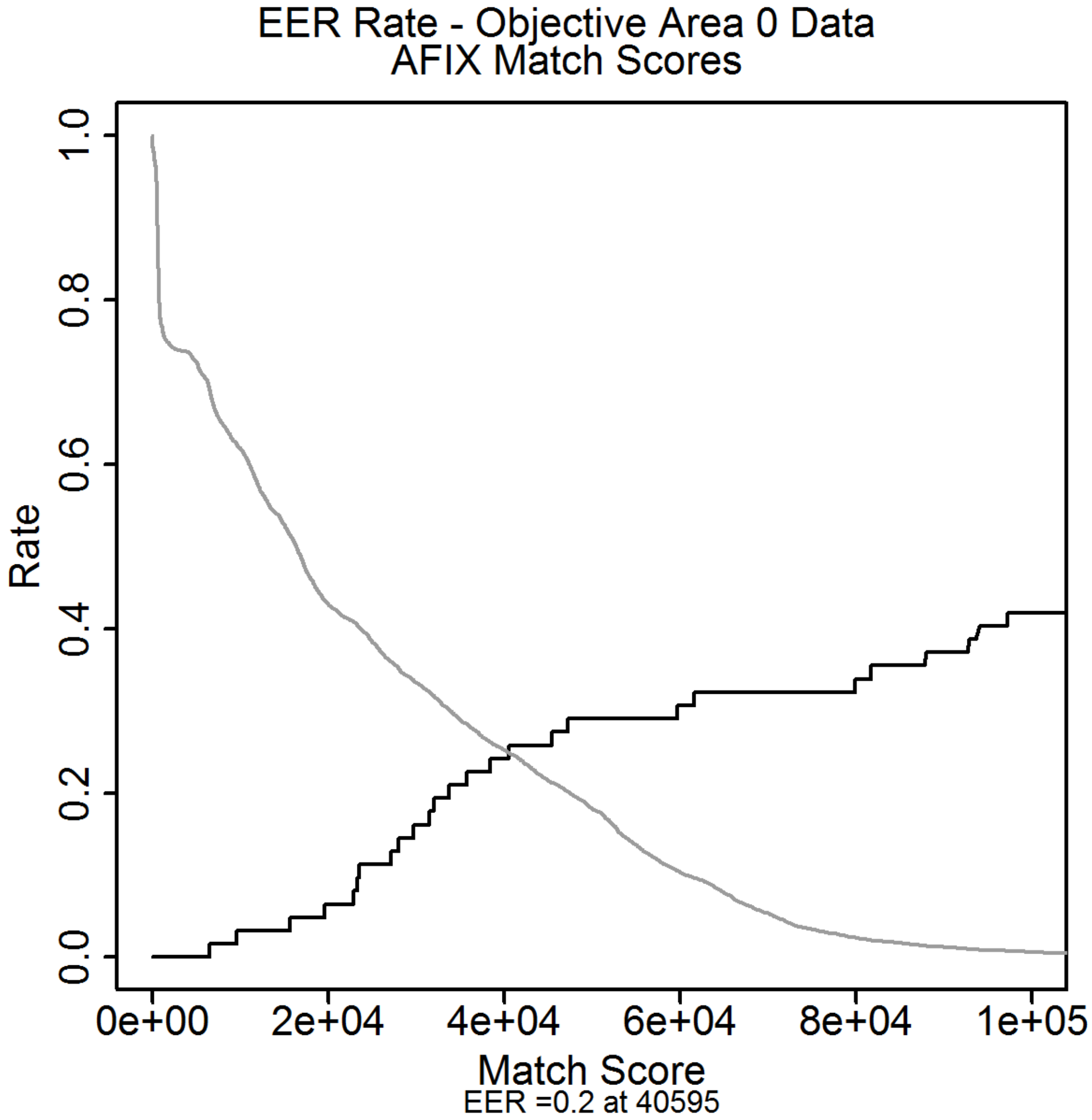


Figure D.94: This figure presents the EER plot from the subjective Area 1 images for the match scores in the automatic extraction.

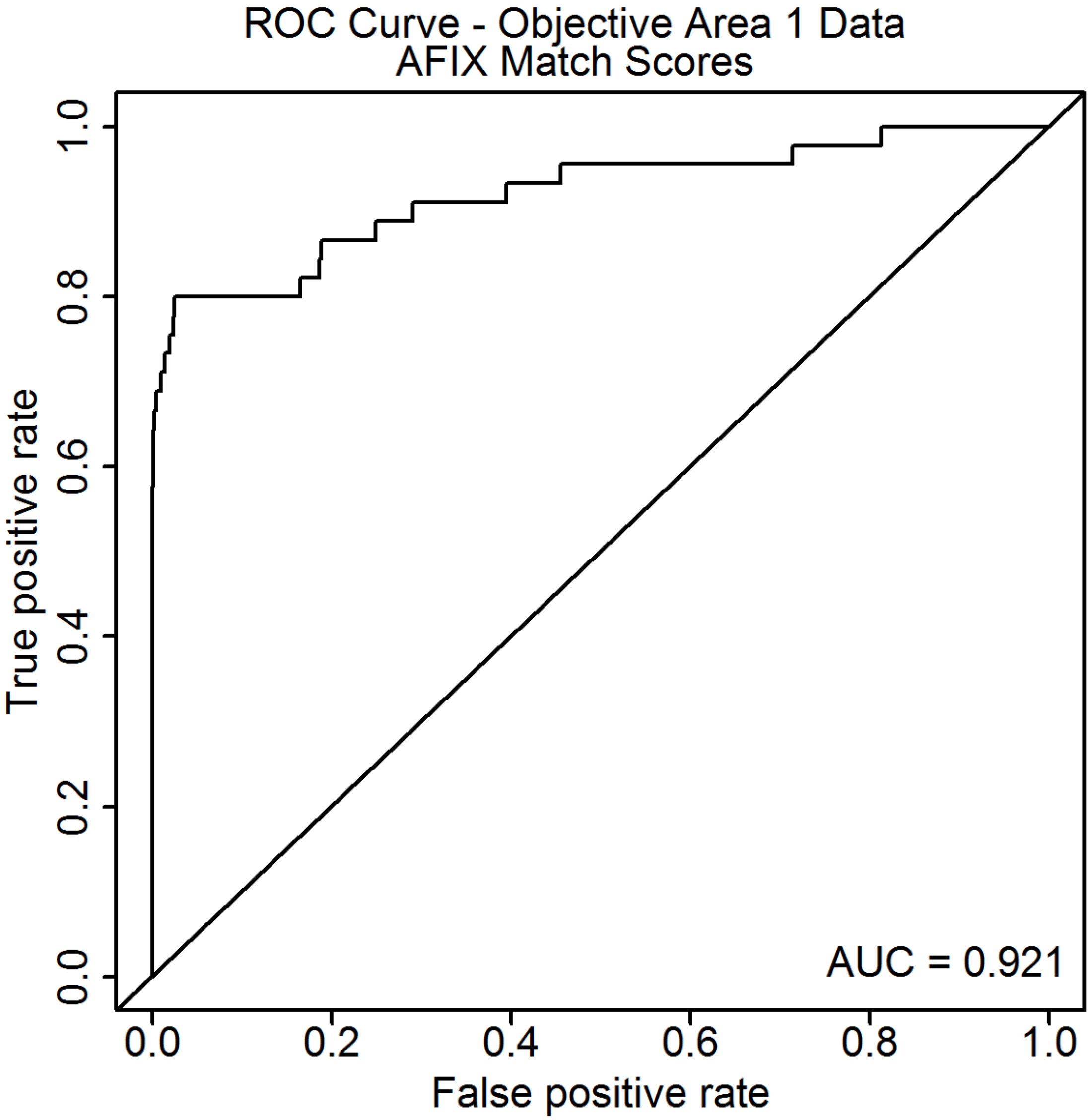


Figure D.95: This figure presents the ROC curve from the objective Area 1 images for the match scores in the automatic extraction.

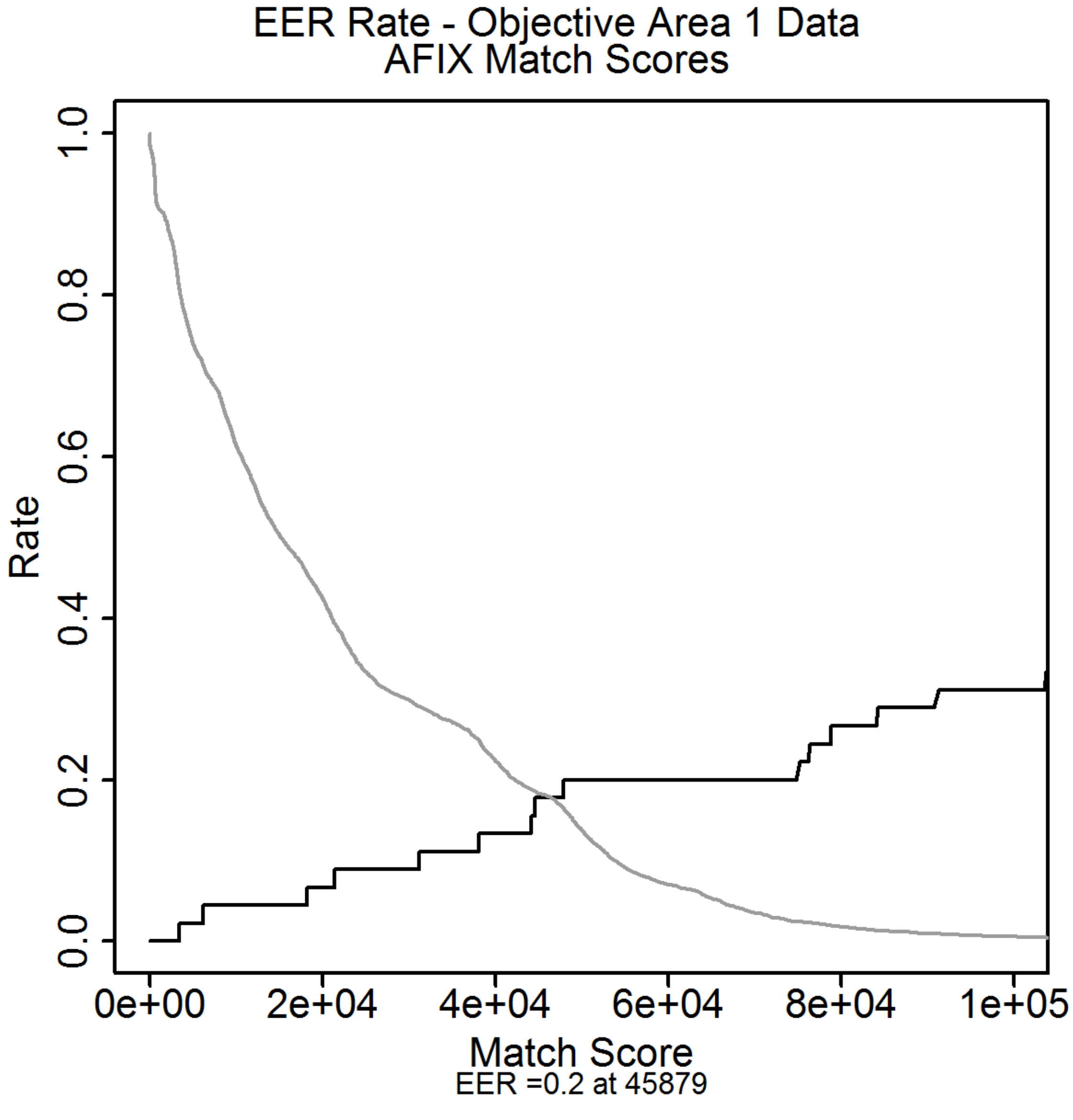


Figure D.96: This figure presents the EER plot from the objective Area 1 images for the match scores in the automatic extraction.

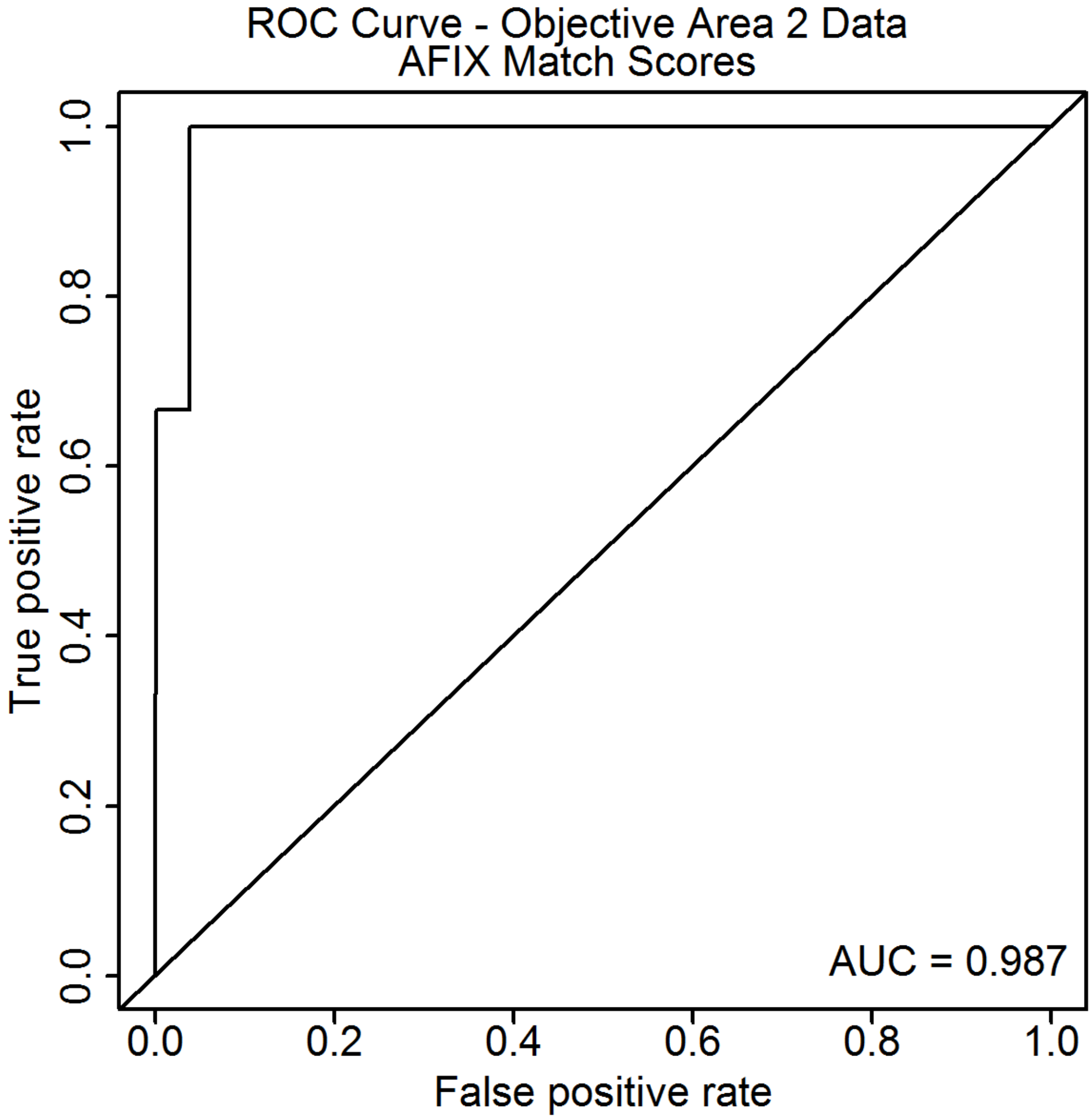


Figure D.97: This figure presents the ROC curve from the objective Area 2 images for the match scores in the automatic extraction.

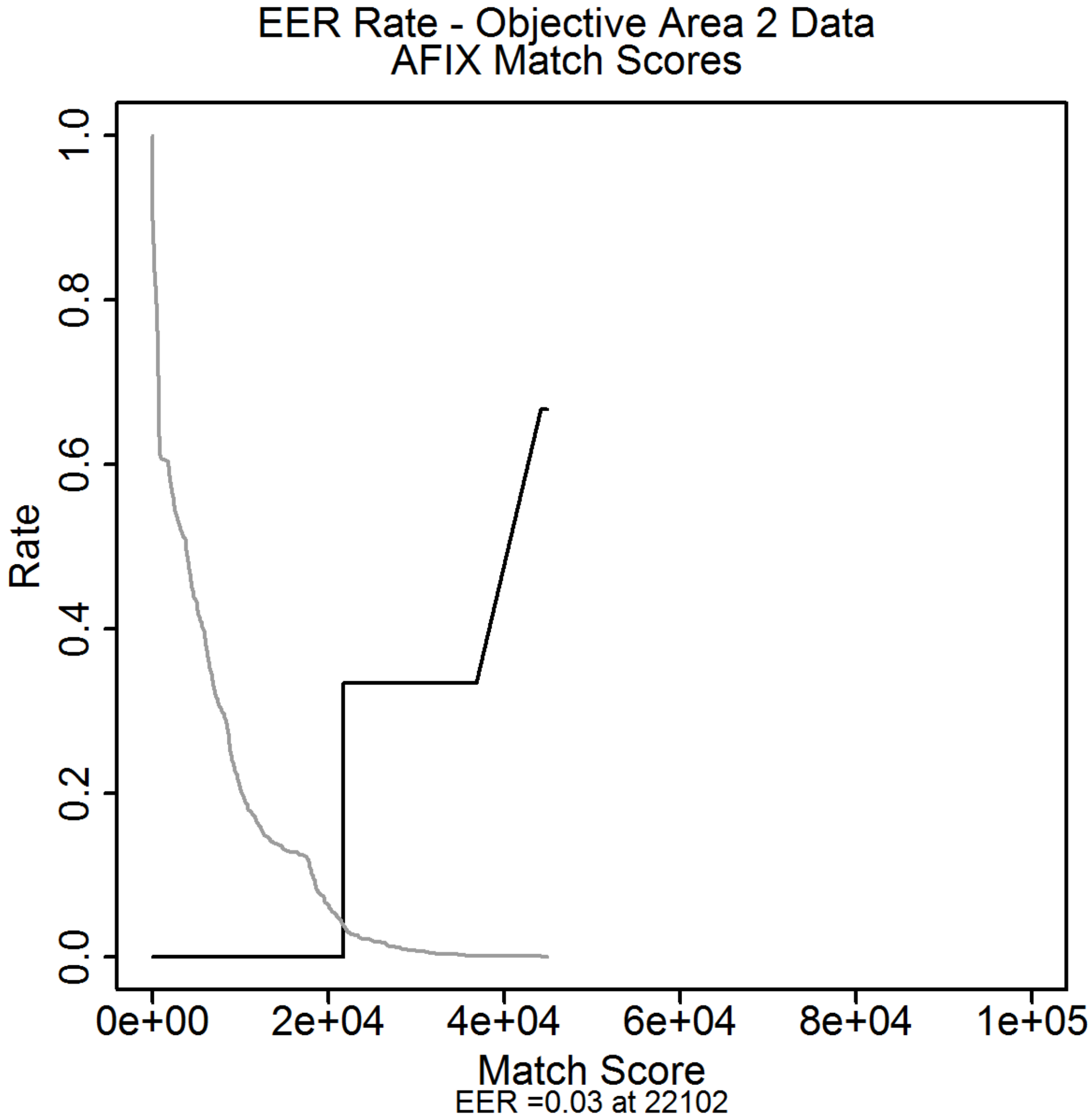


Figure D.98: This figure presents the EER plot from the objective Area 2 images for the match scores in the automatic extraction.

D.7.5 Objective Method: Automatic Extraction

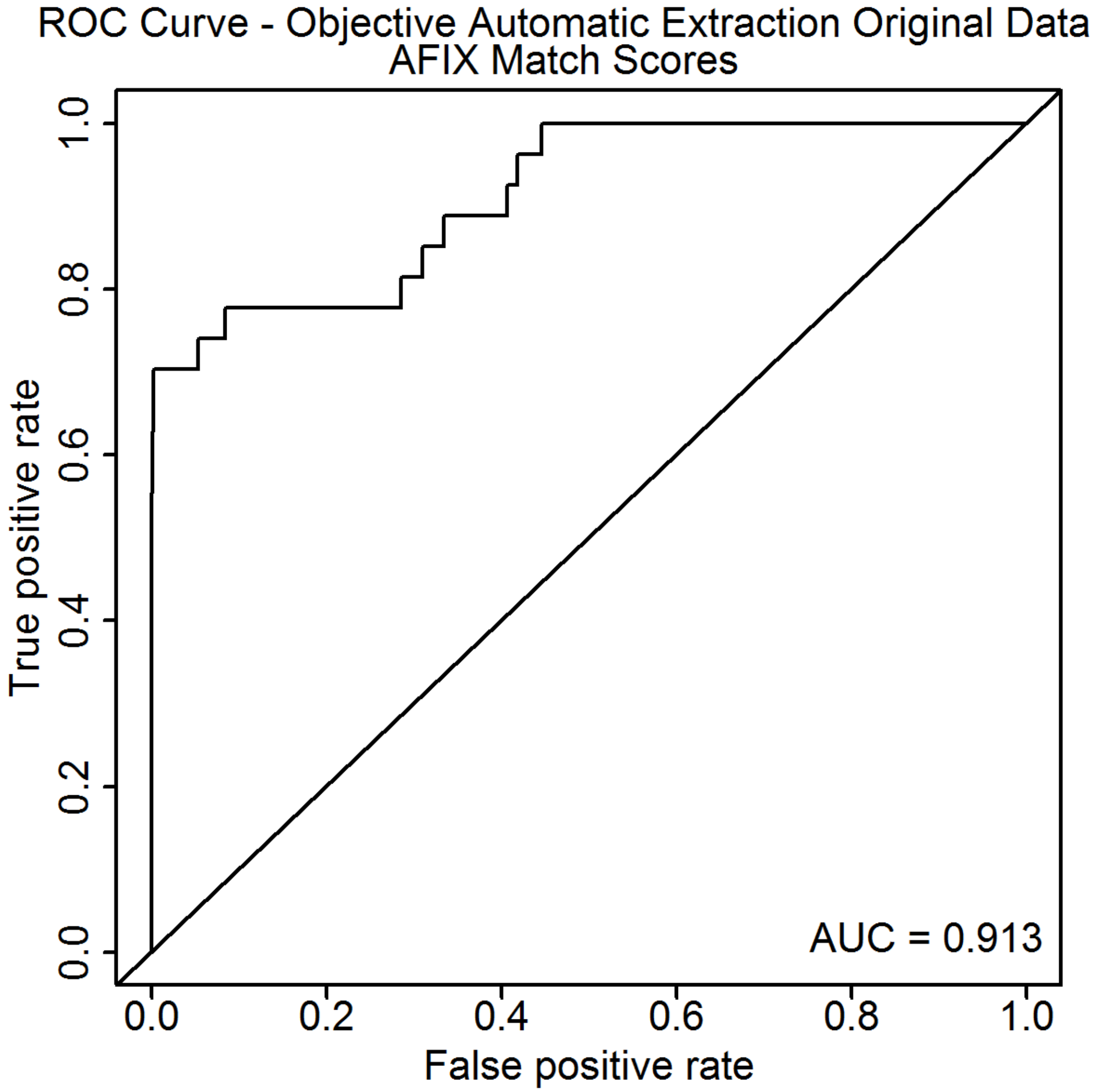


Figure D.99: This figure presents the ROC curve from the objective original images for the match scores in the automatic extraction.

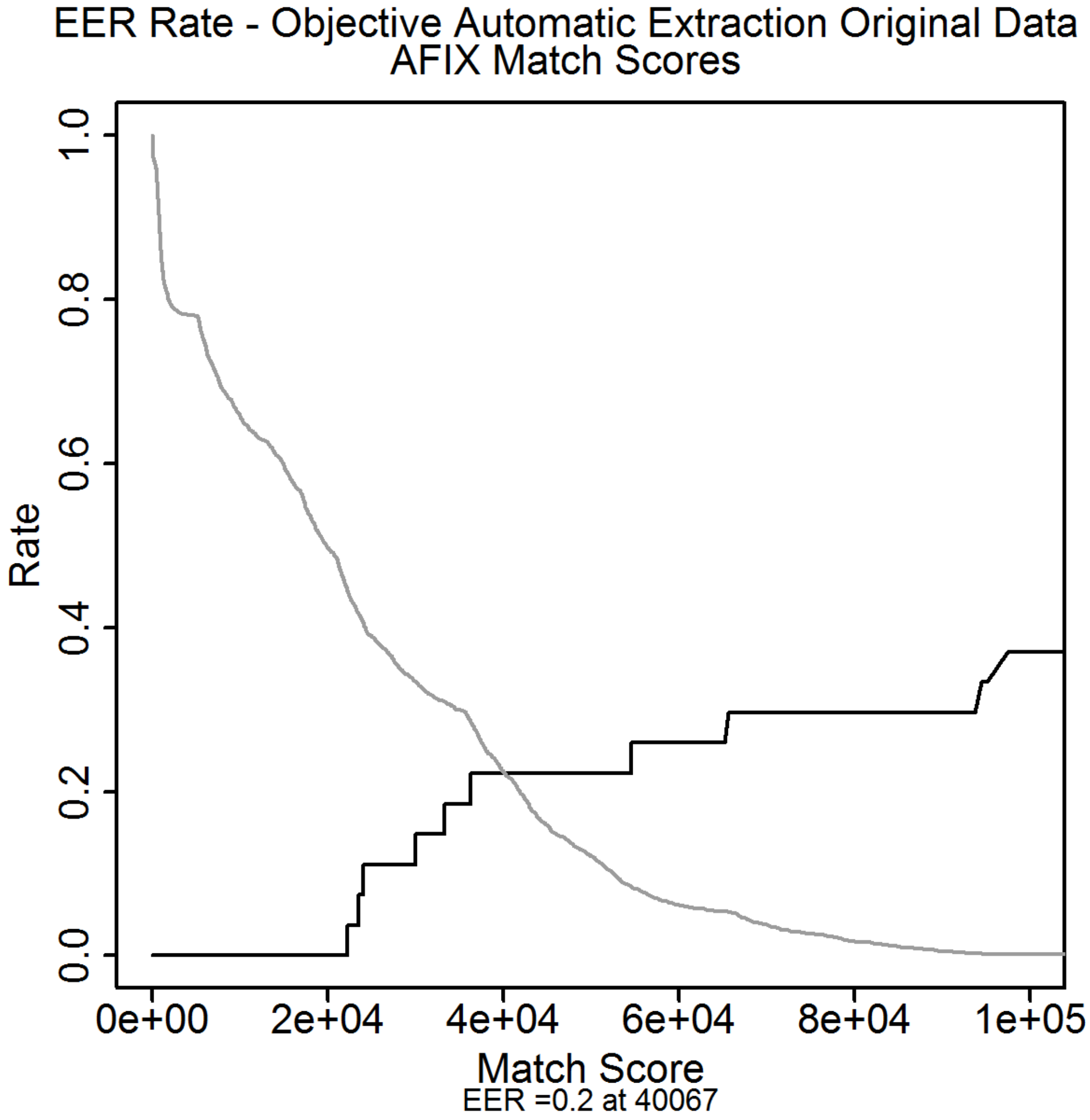


Figure D.100: This figure presents the EER plot from the objective original images for the match scores in the automatic extraction.

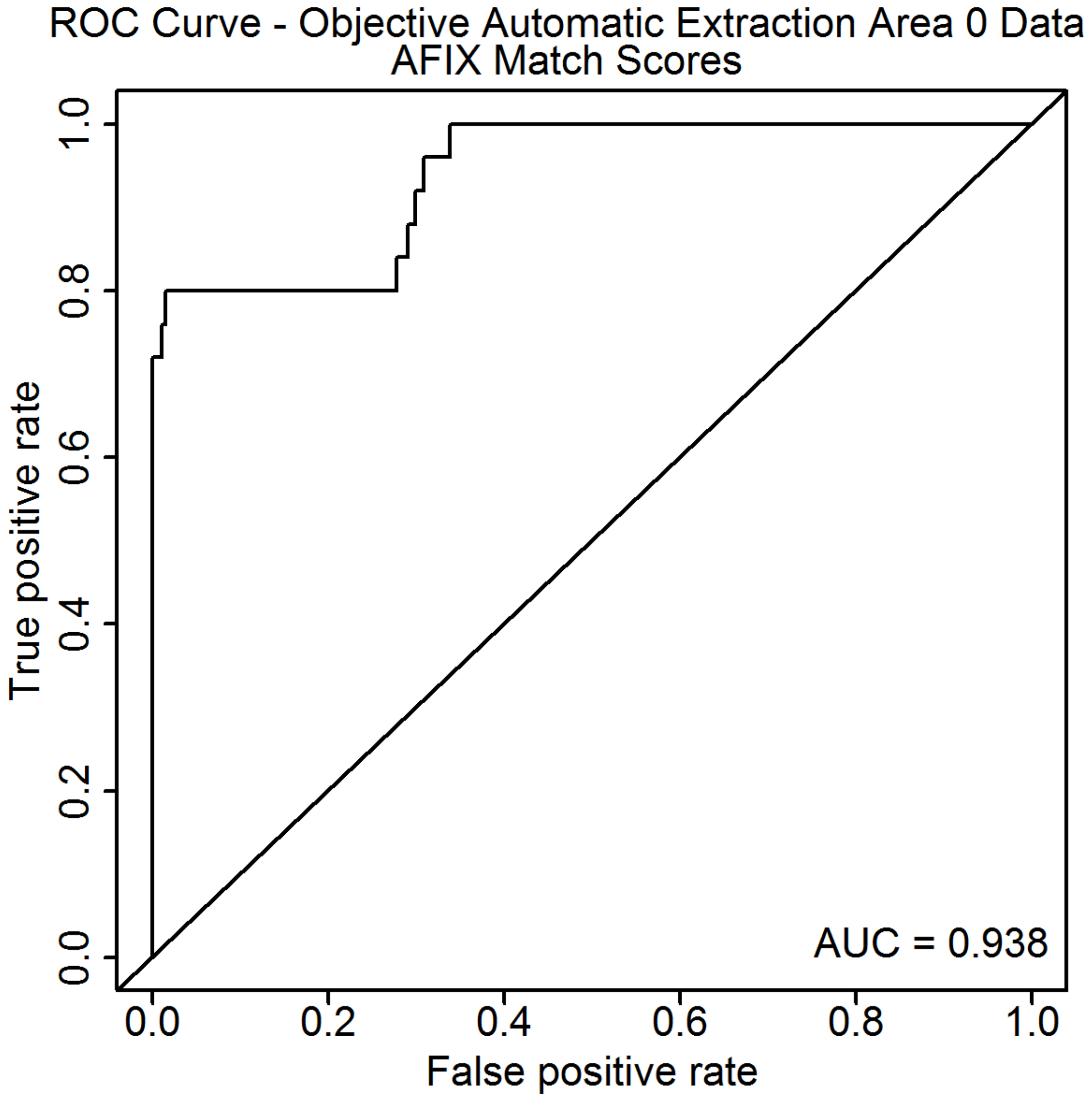


Figure D.101: This figure presents the ROC curve from the objective Area 0 images for the match scores in the automatic extraction.

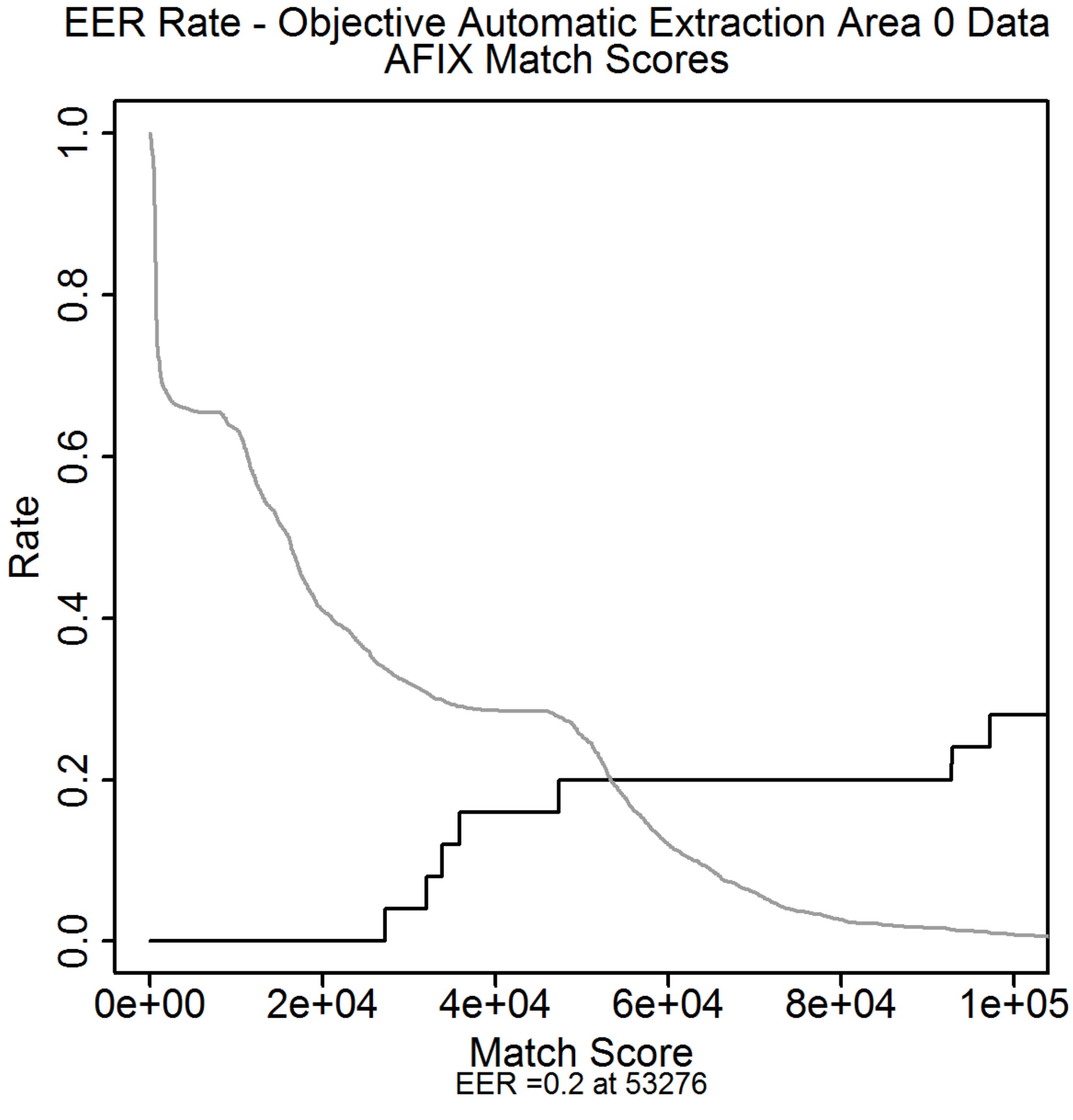


Figure D.102: This figure presents the EER plot from the objective Area 0 images for the match scores in the automatic extraction.

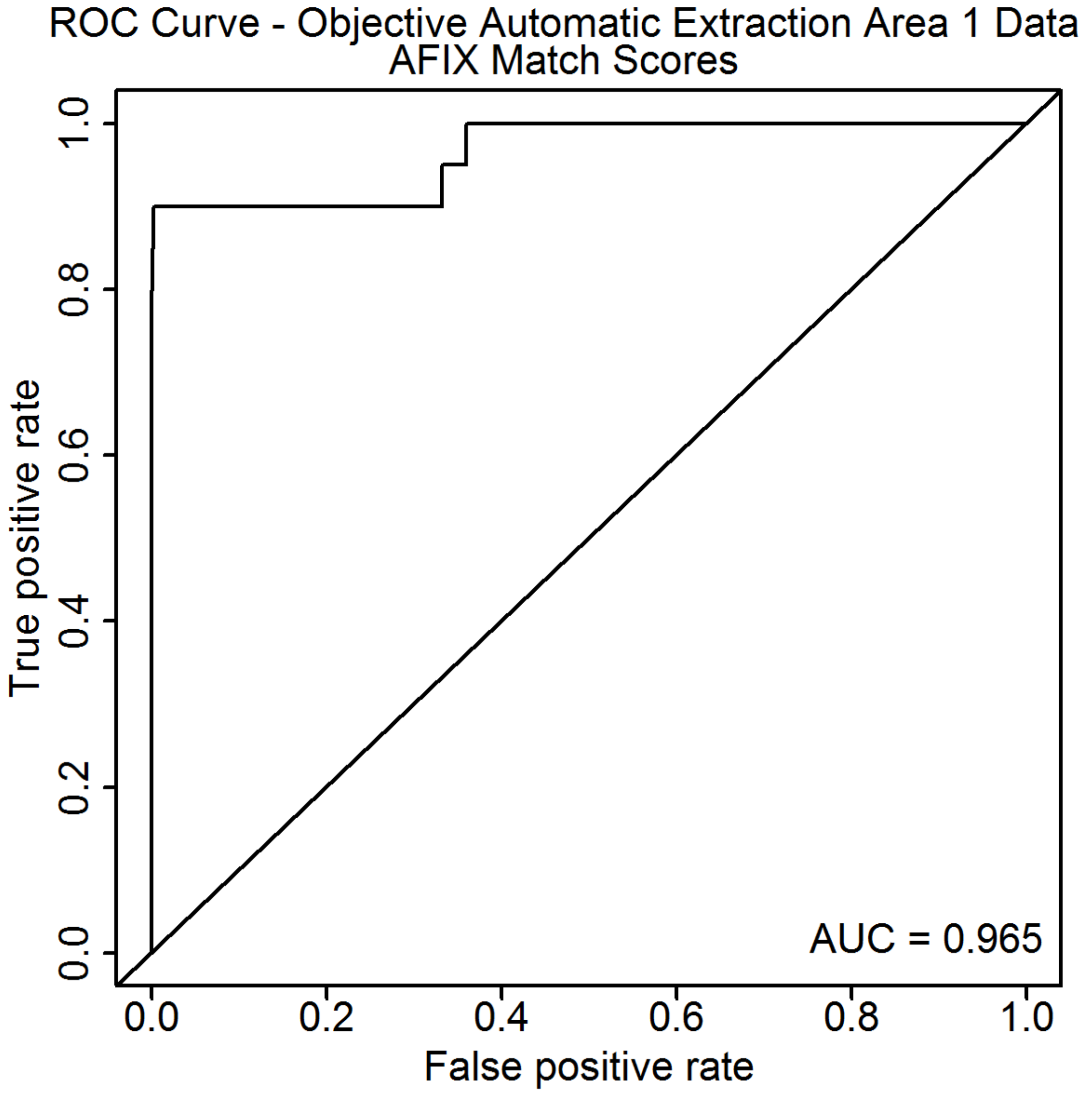


Figure D.103: This figure presents the ROC curve from the objective Area 1 images for the match scores in the automatic extraction.

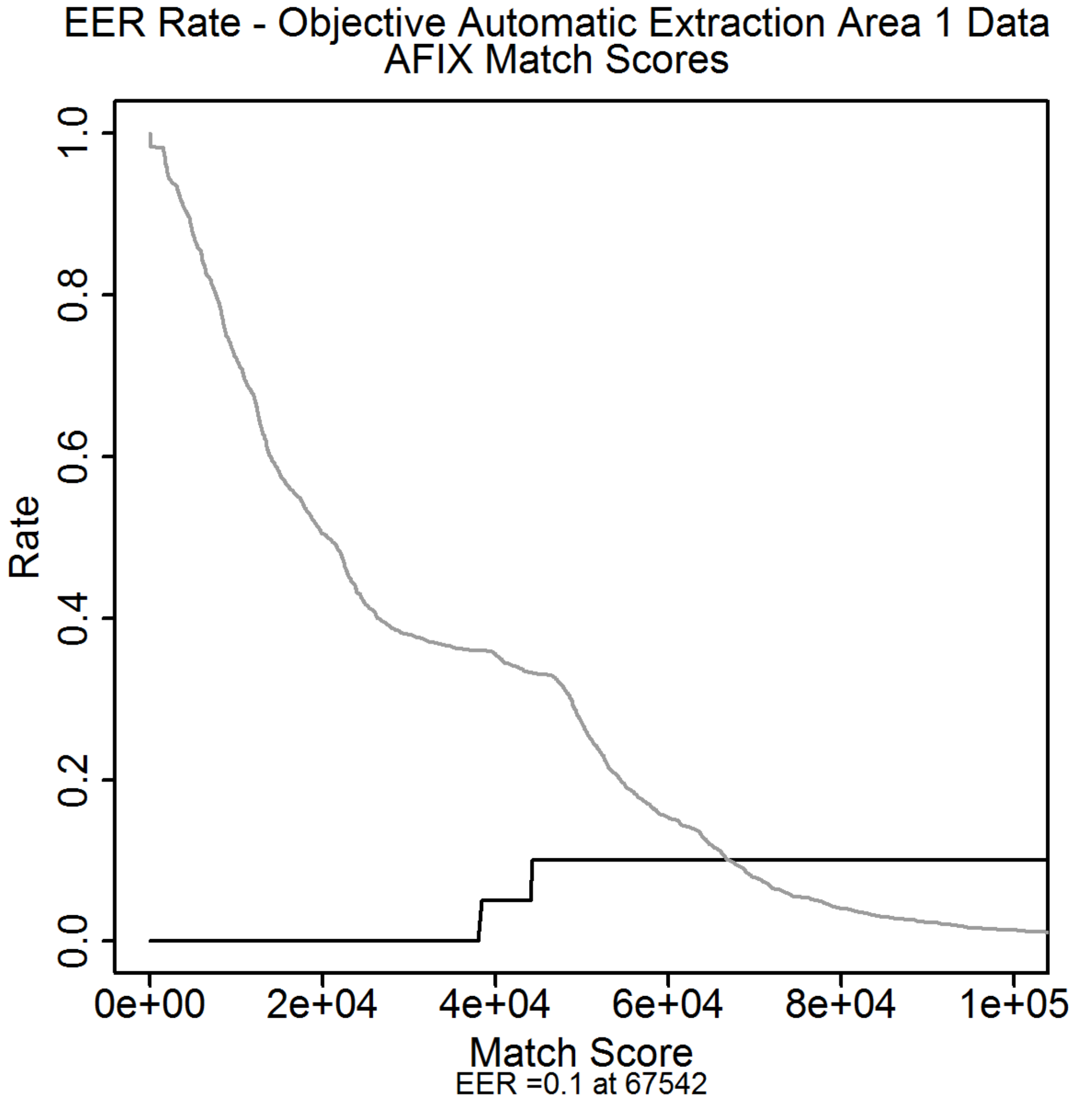


Figure D.104: This figure presents the EER plot from the objective Area 1 images for the match scores in the automatic extraction.

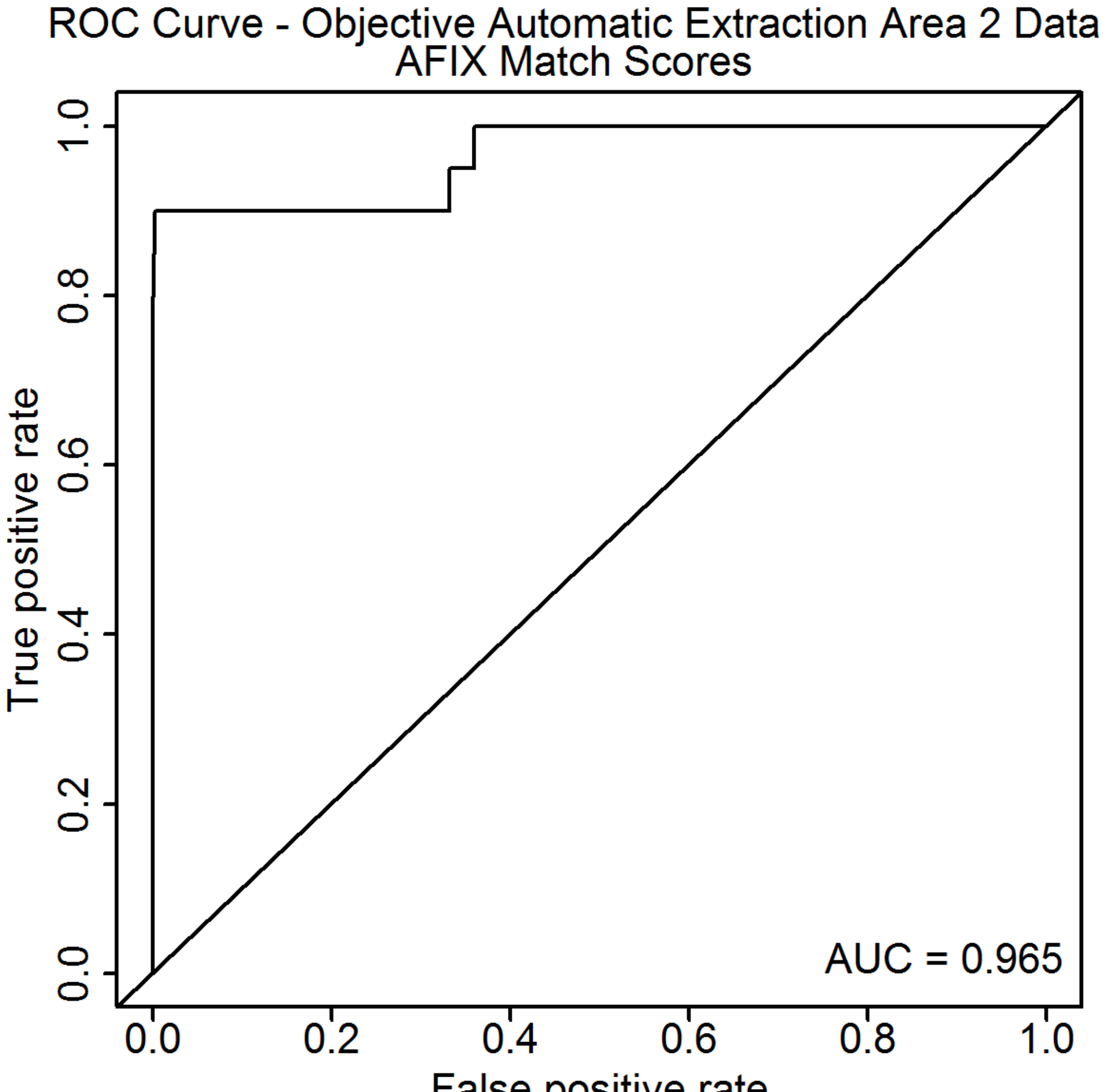


Figure D.105: This figure presents the ROC curve from the objective Area 2 images for the match scores in the automatic extraction.

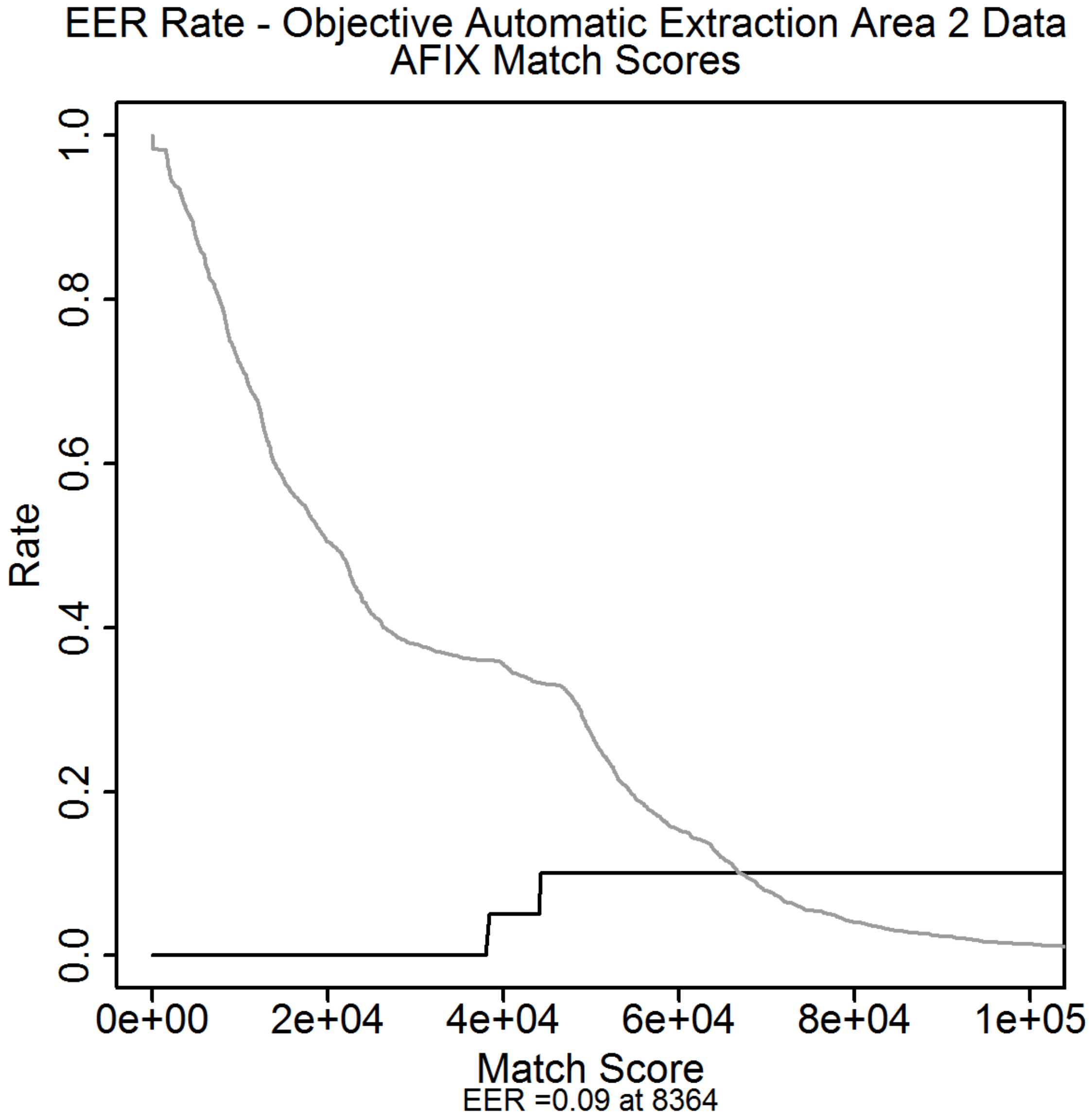


Figure D.106: This figure presents the EER plot from the objective Area 2 images for the match scores in the automatic extraction.

D.7.6 Objective Method: Manual Extraction

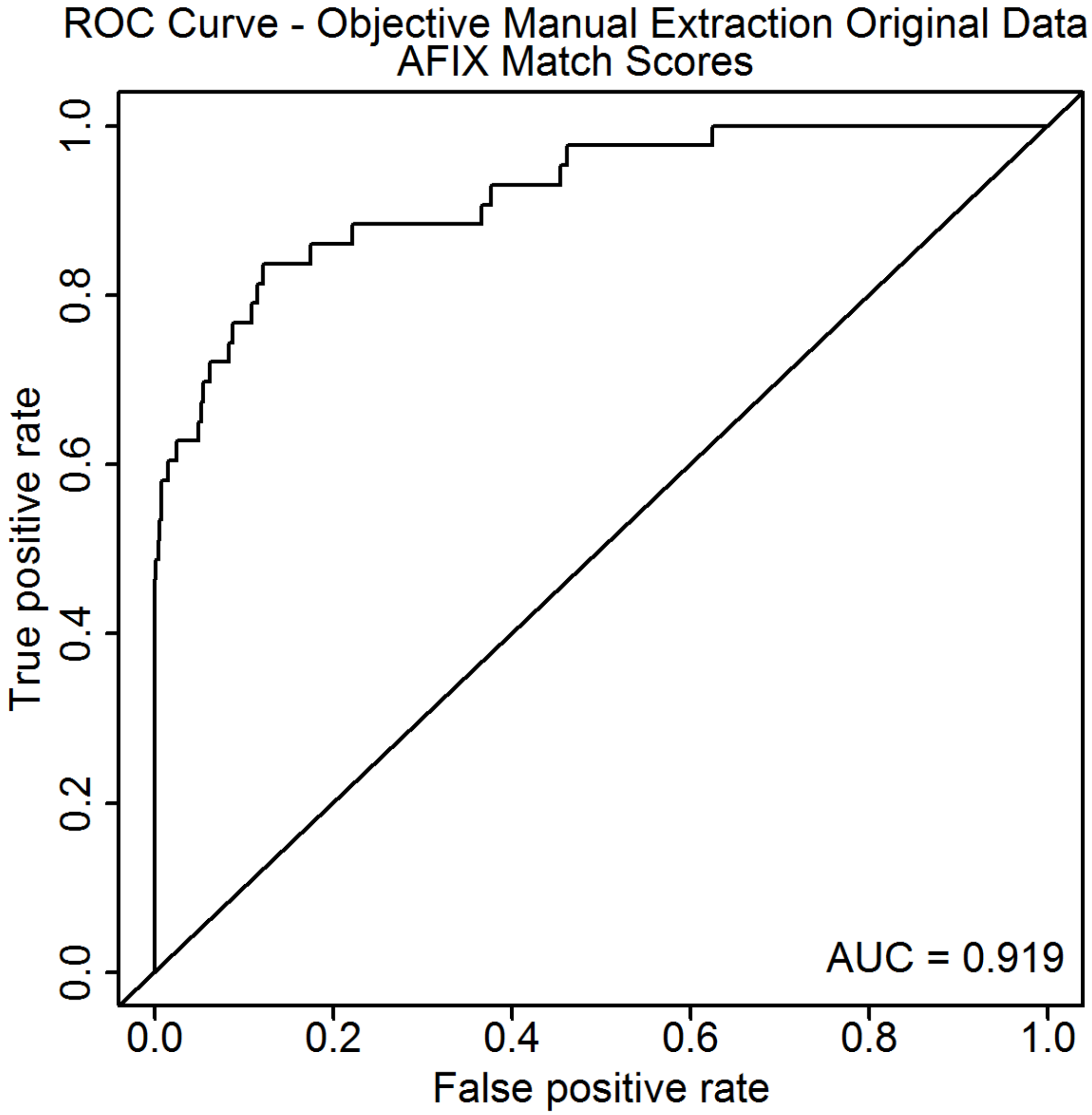


Figure D.107: This figure presents the ROC curve from the objective Area original images for the match scores in the manual extraction.

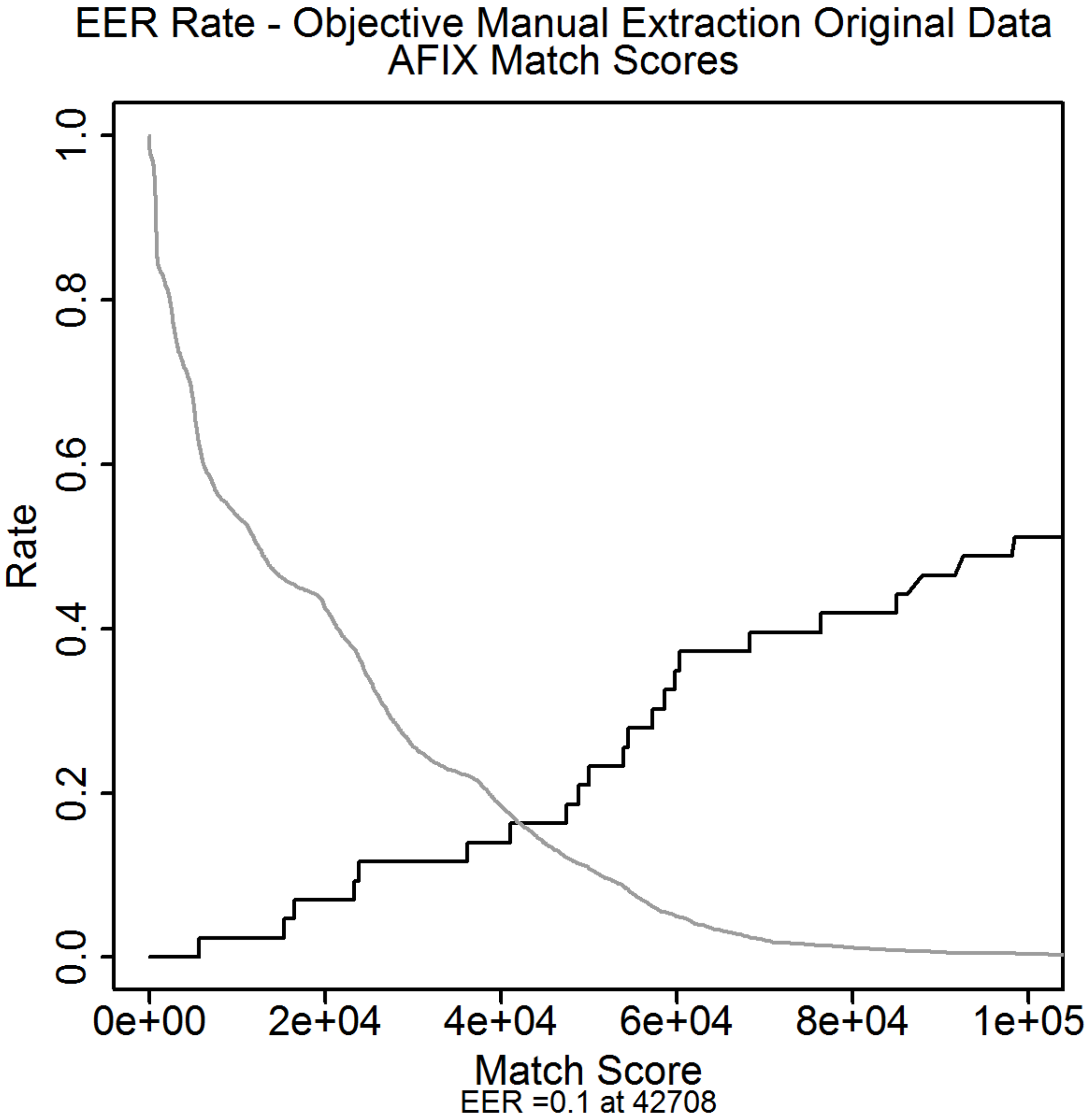


Figure D.108: This figure presents the EER plot from the objective original images for the match scores in the manual extraction.

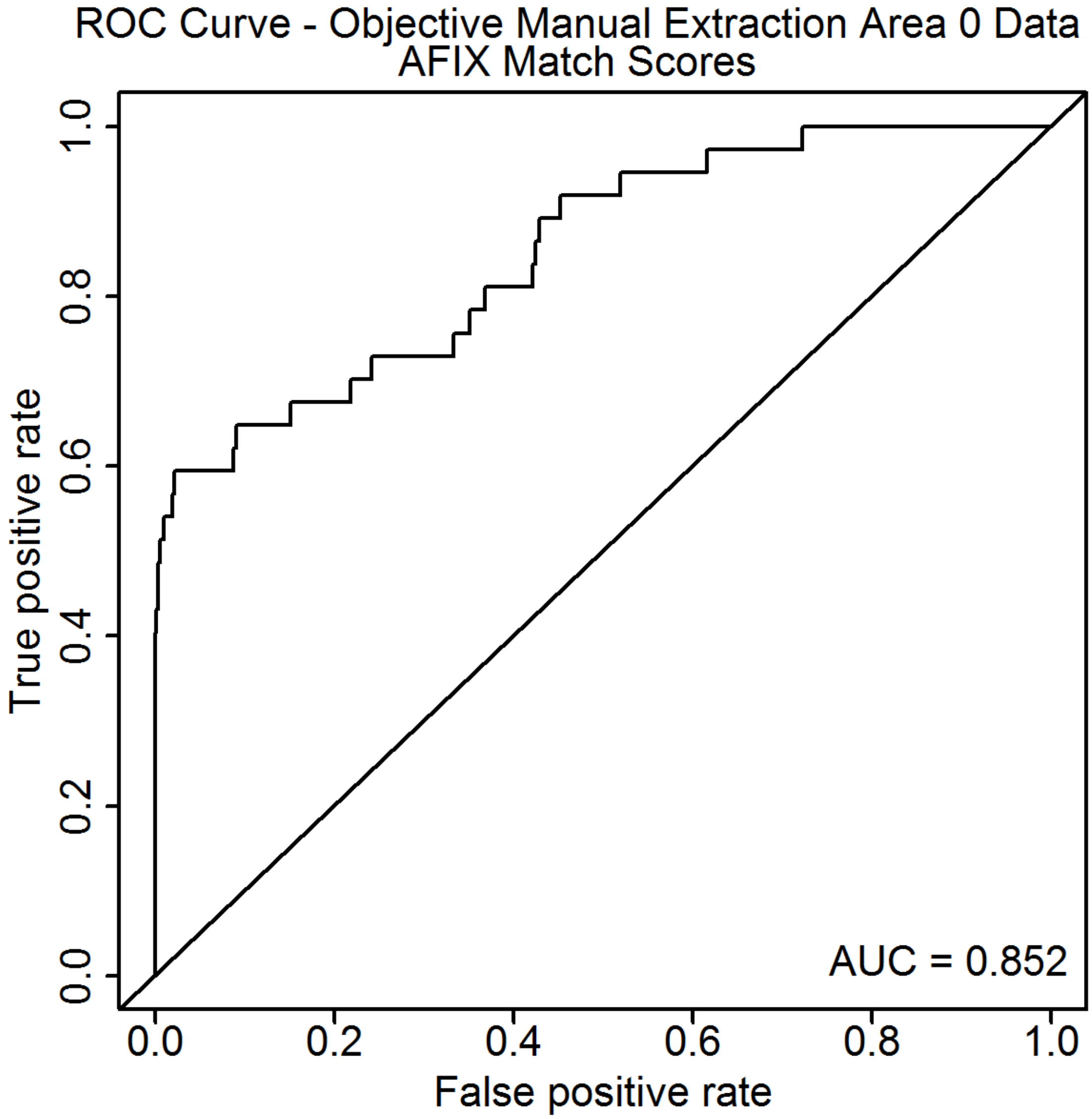


Figure D.109: This figure presents the ROC curve from the objective Area 0 images for the match scores in the manual extraction.

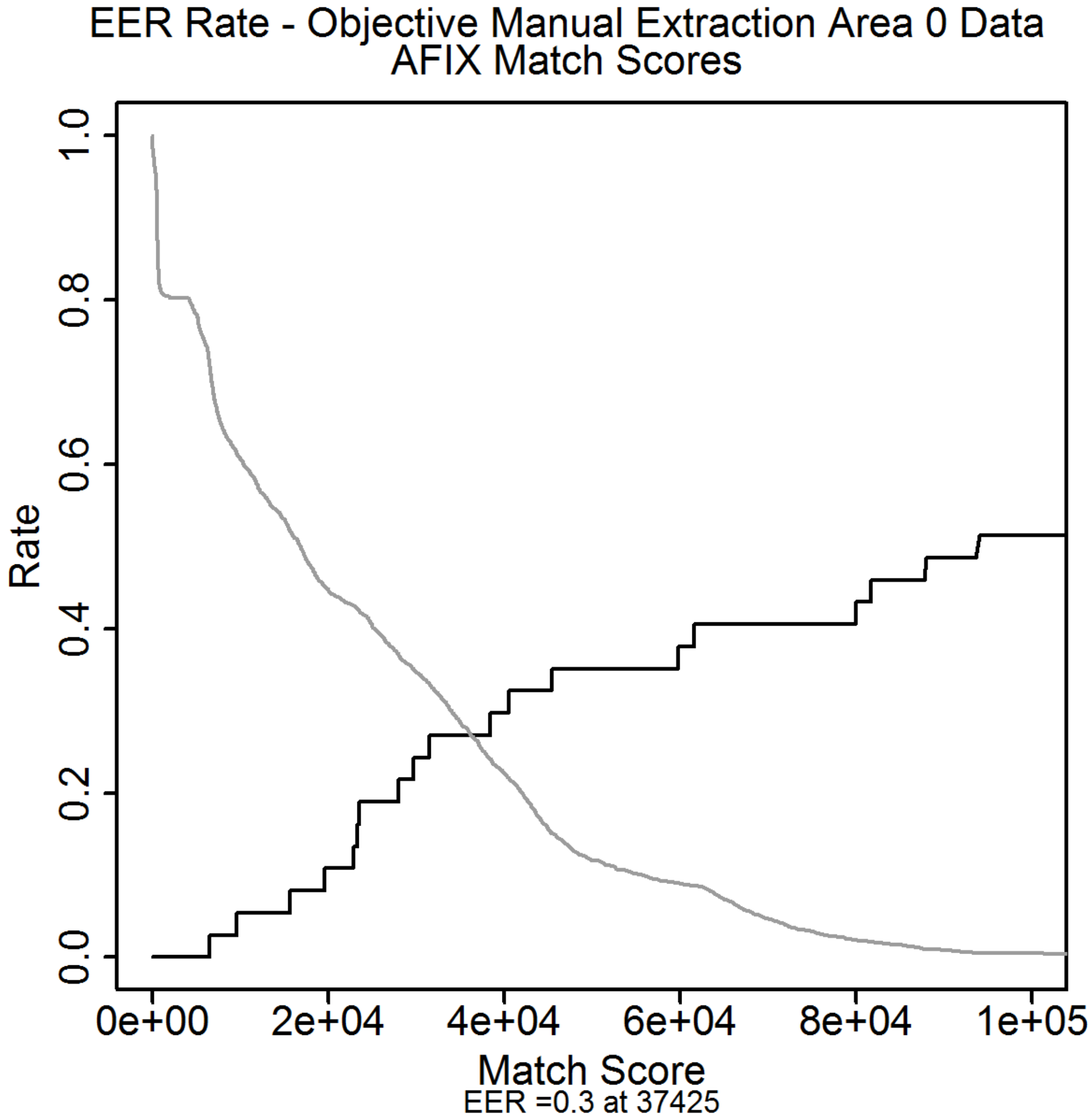


Figure D.110: This figure presents the EER plot from the objective Area 0 images for the match scores in the manual extraction.

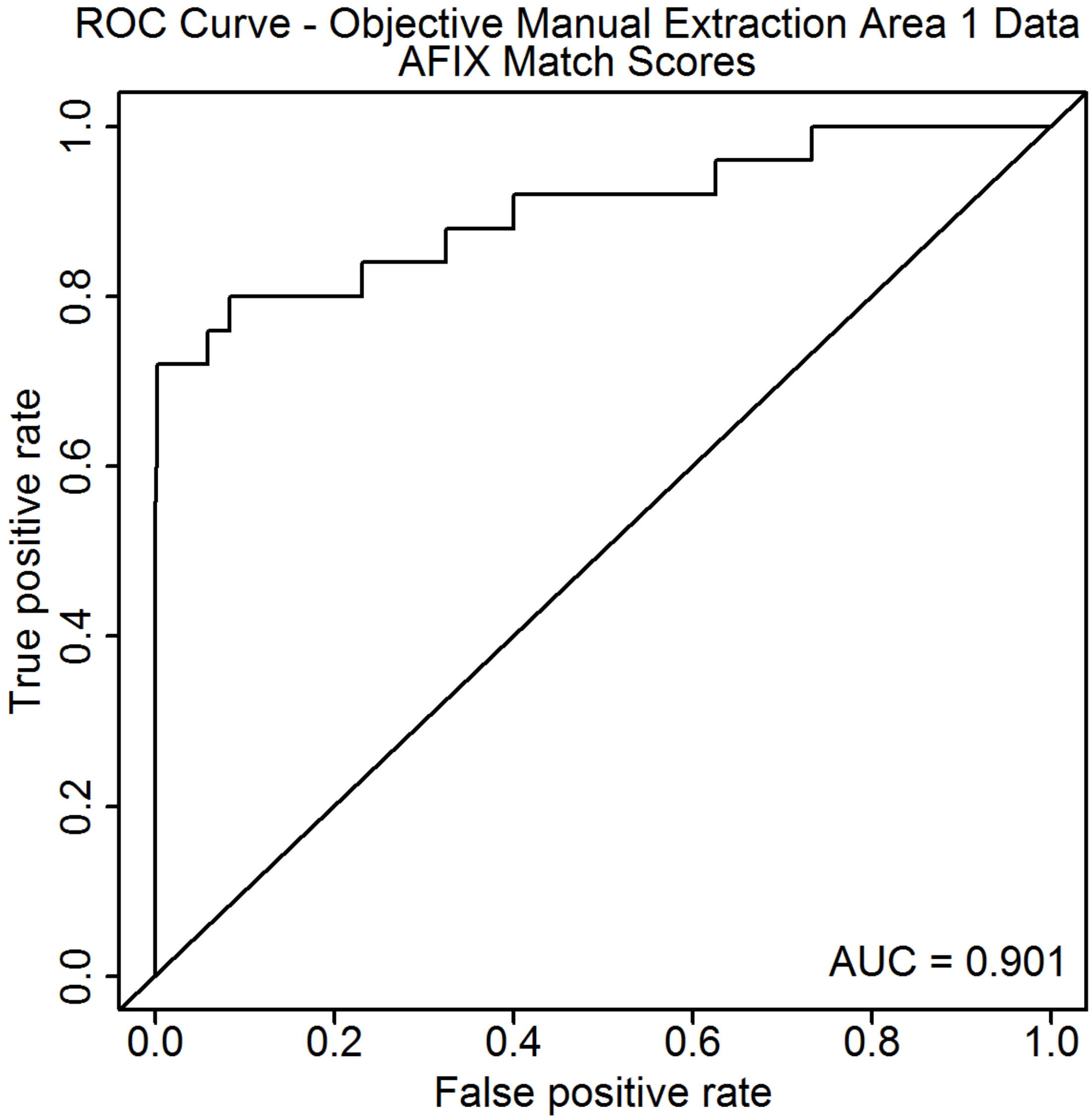


Figure D.111: This figure presents the ROC curve from the objective Area 1 images for the match scores in the manual extraction.

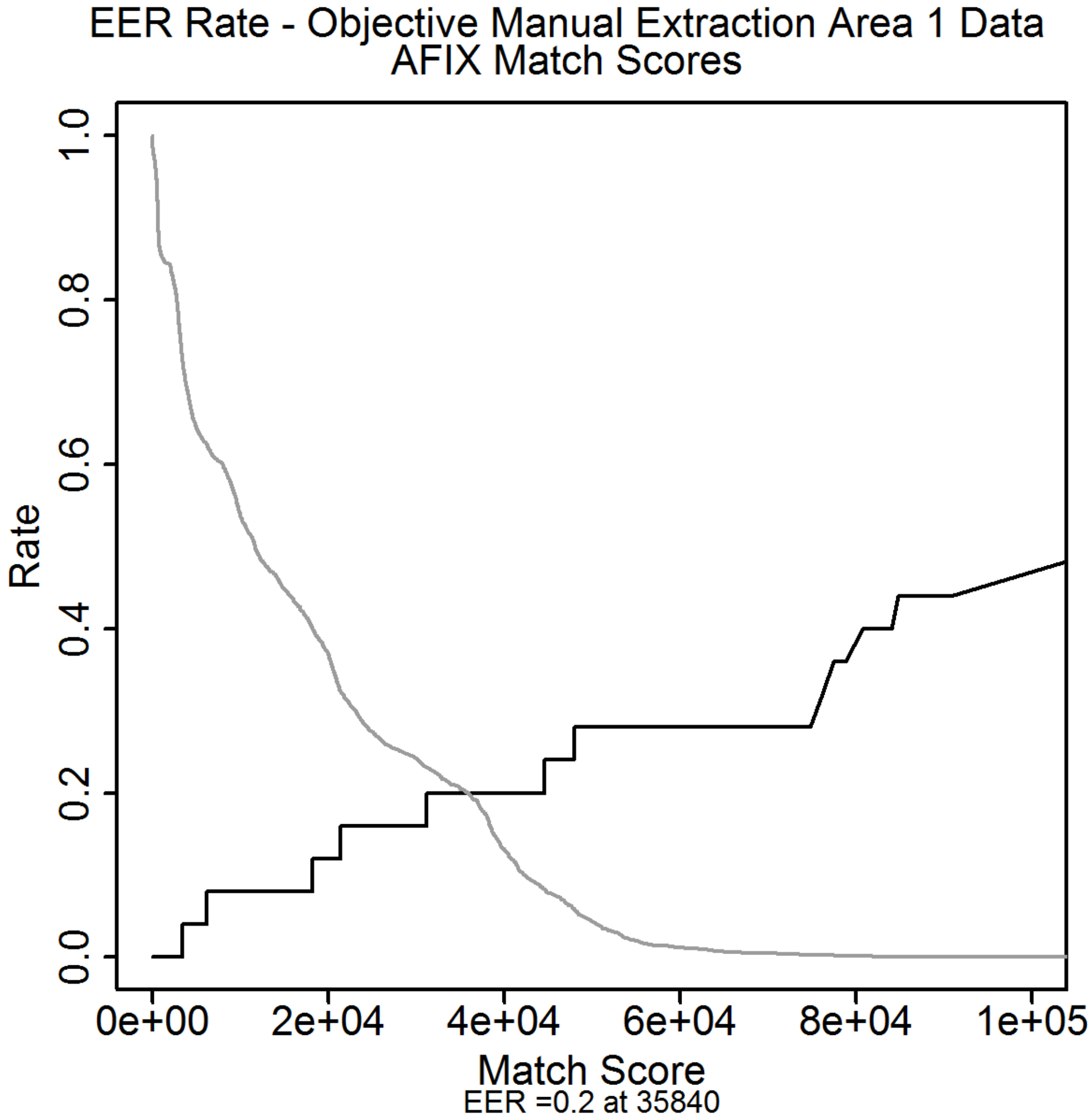


Figure D.112: This figure presents the EER plot from the objective Area 1 images for the match scores in the manual extraction.

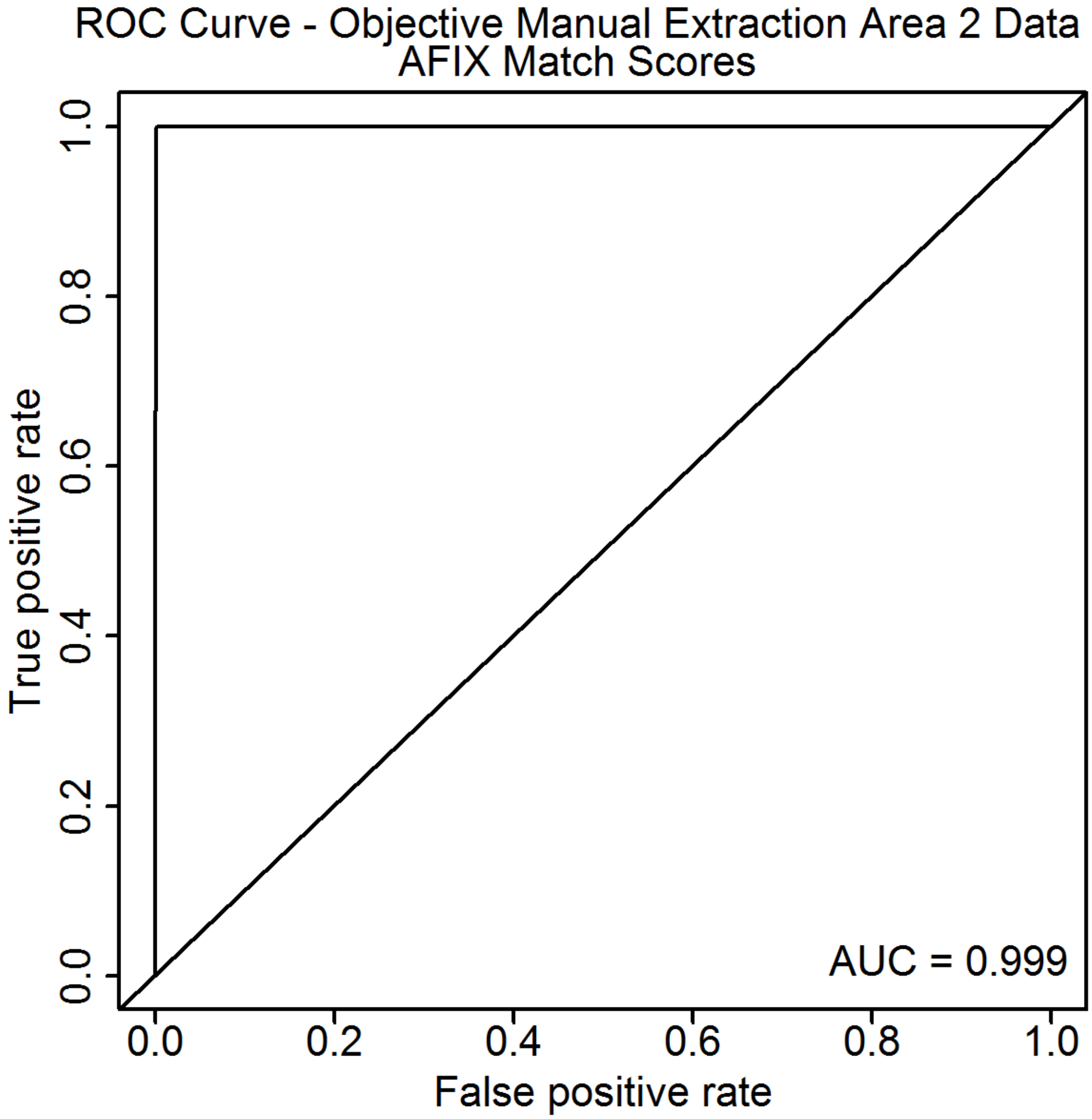


Figure D.113: This figure presents the ROC curve from the objective Area 2 images for the match scores in the manual extraction.

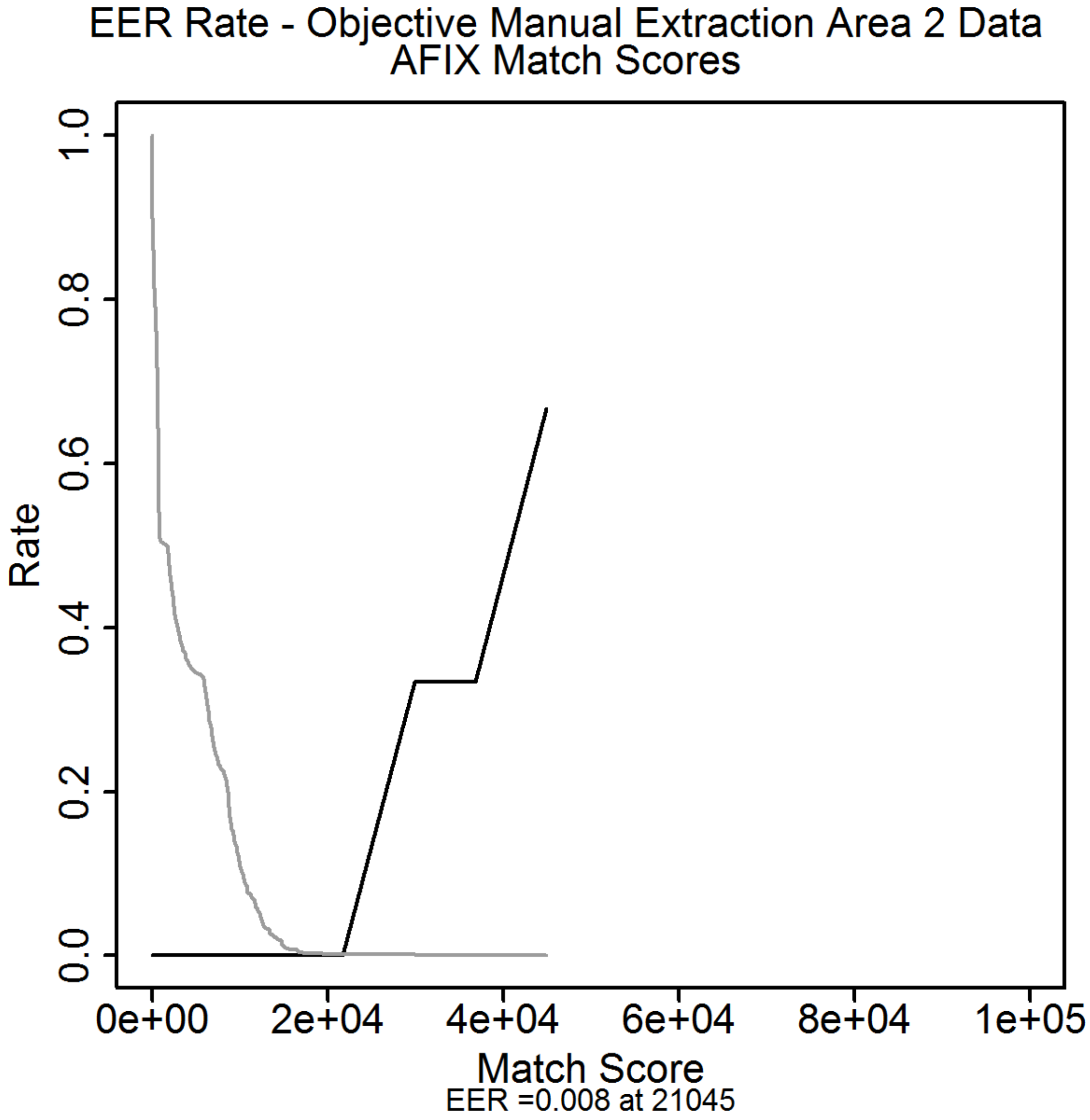


Figure D.114: This figure presents the EER plot from the objective Area 2 images for the match scores in the manual extraction.

D.8 ROC Curves-Rank

D.8.1 Subjective Method: All Samples

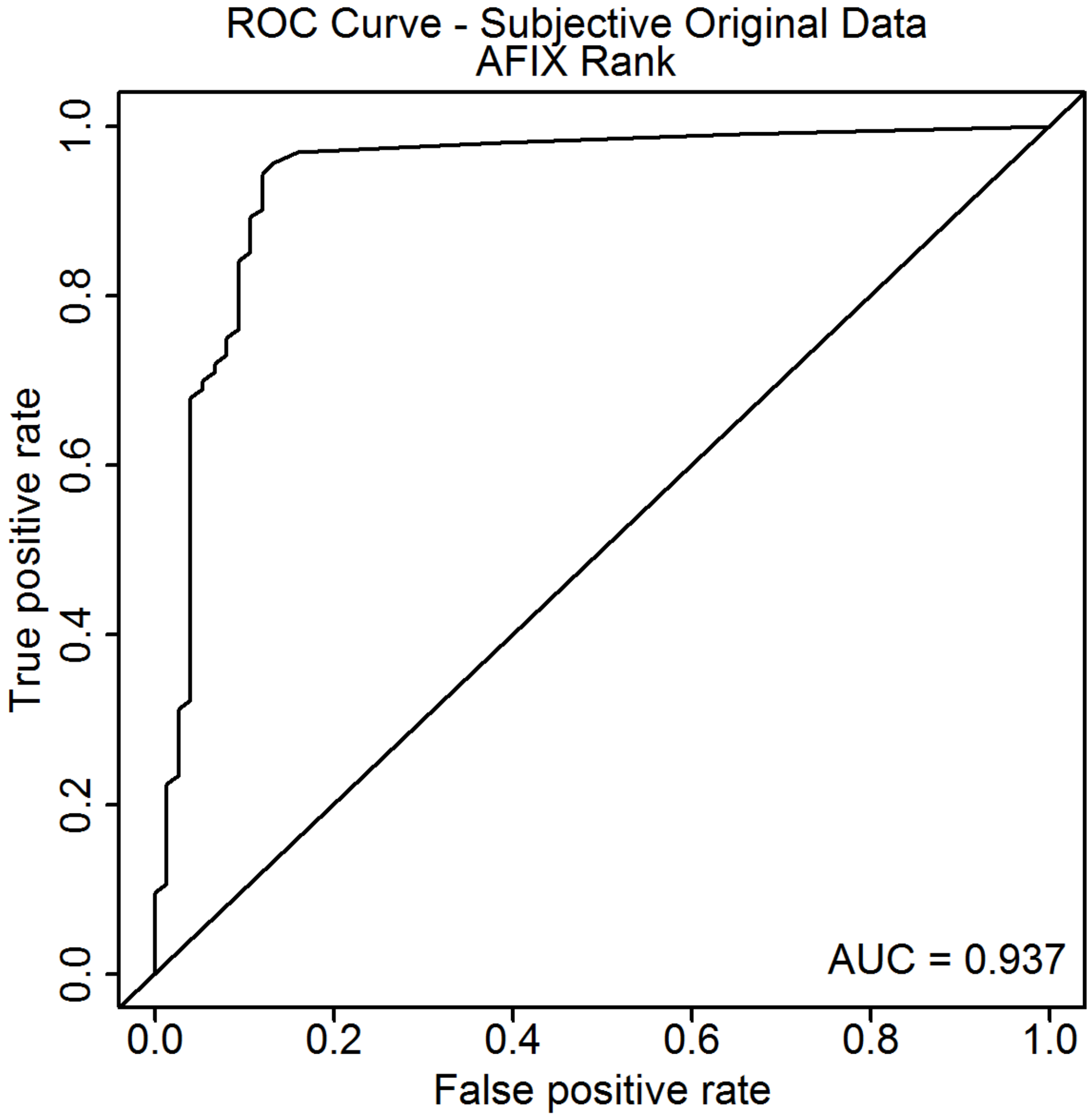


Figure D.115: This figure presents the ROC curve from the subjective original images for the ranks.

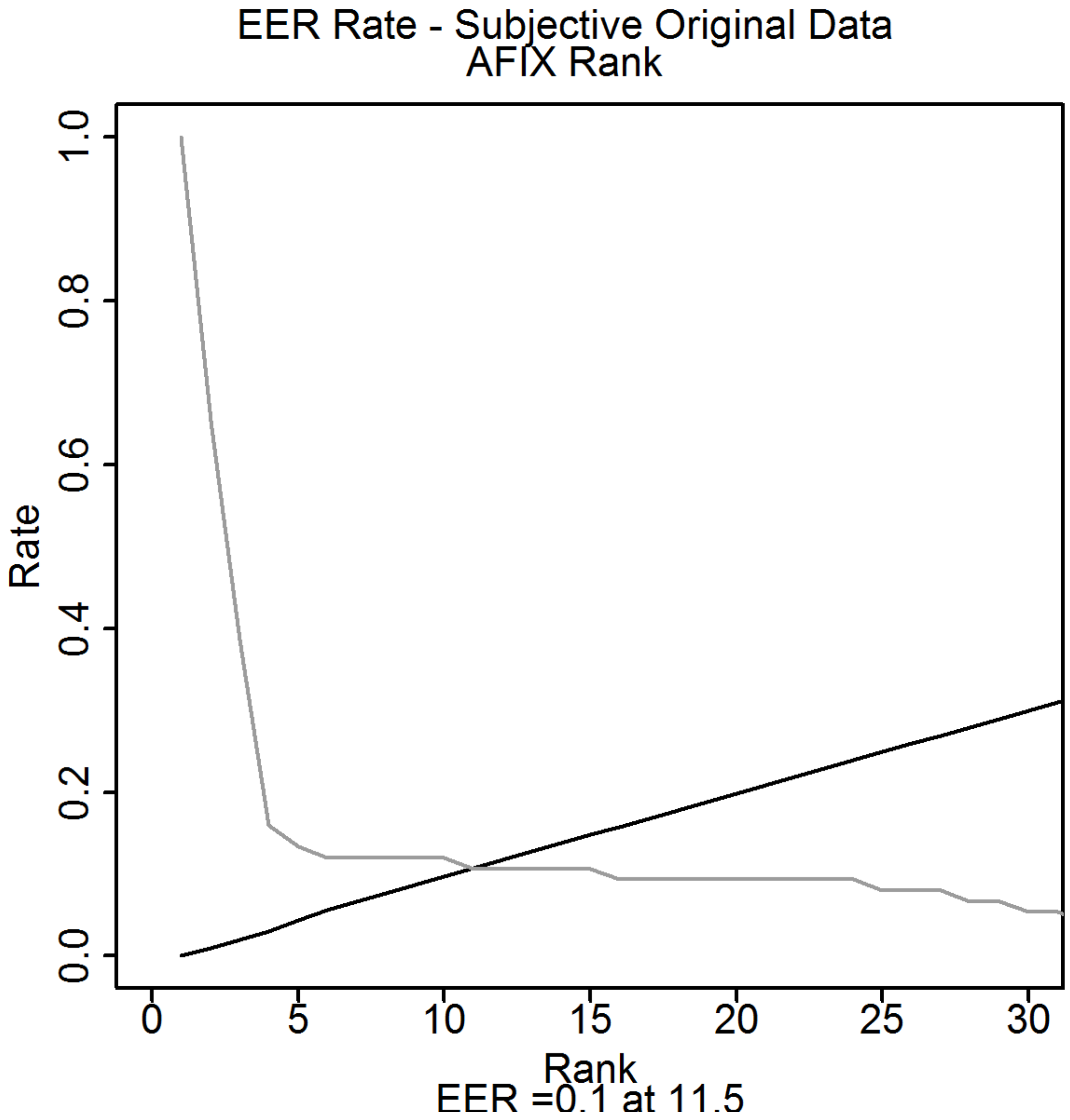


Figure D.116: This figure presents the EER plot from the subjective original images for the ranks.

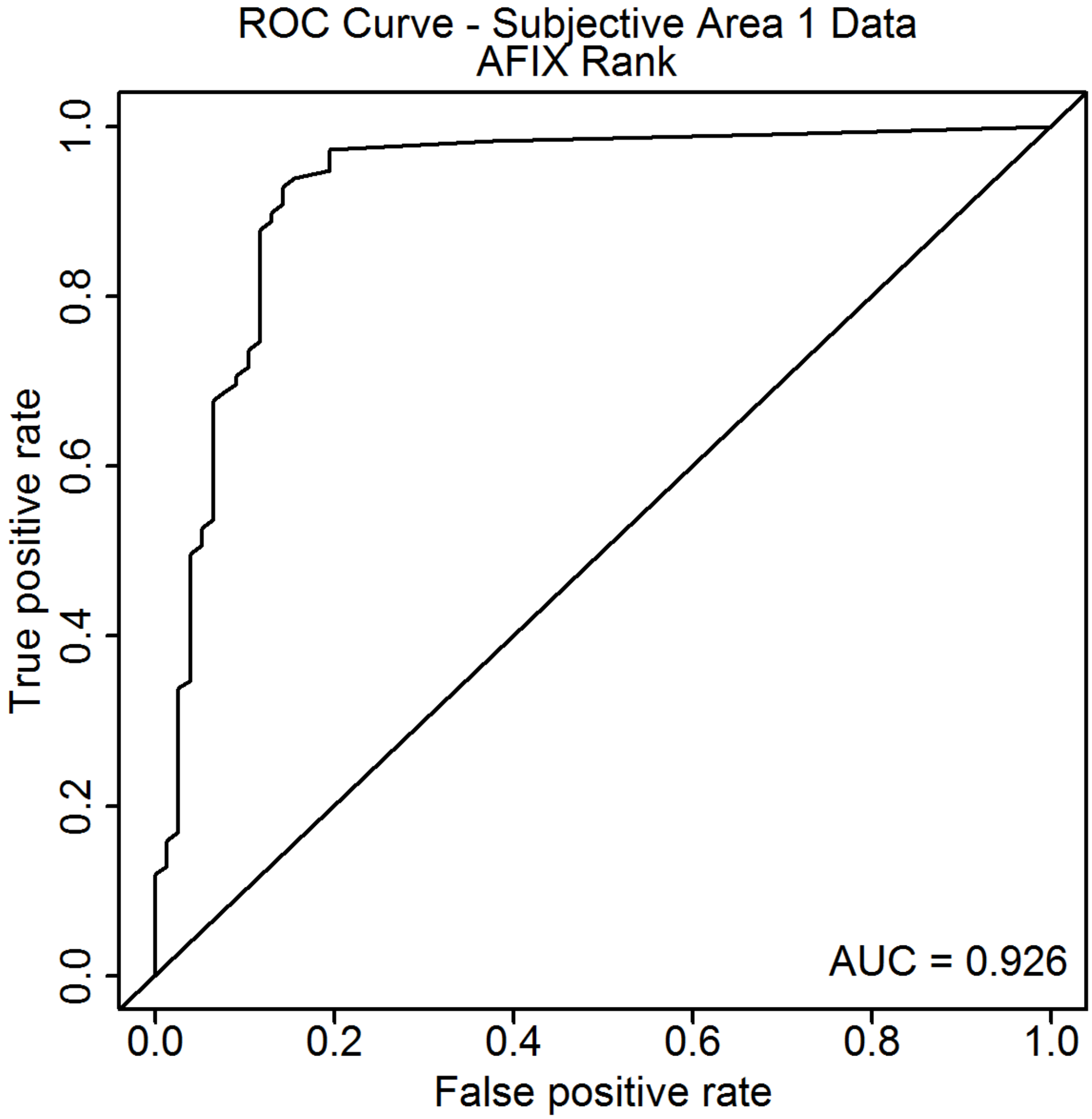


Figure D.117: This figure presents the ROC curve from the subjective Area 1 images for the ranks.

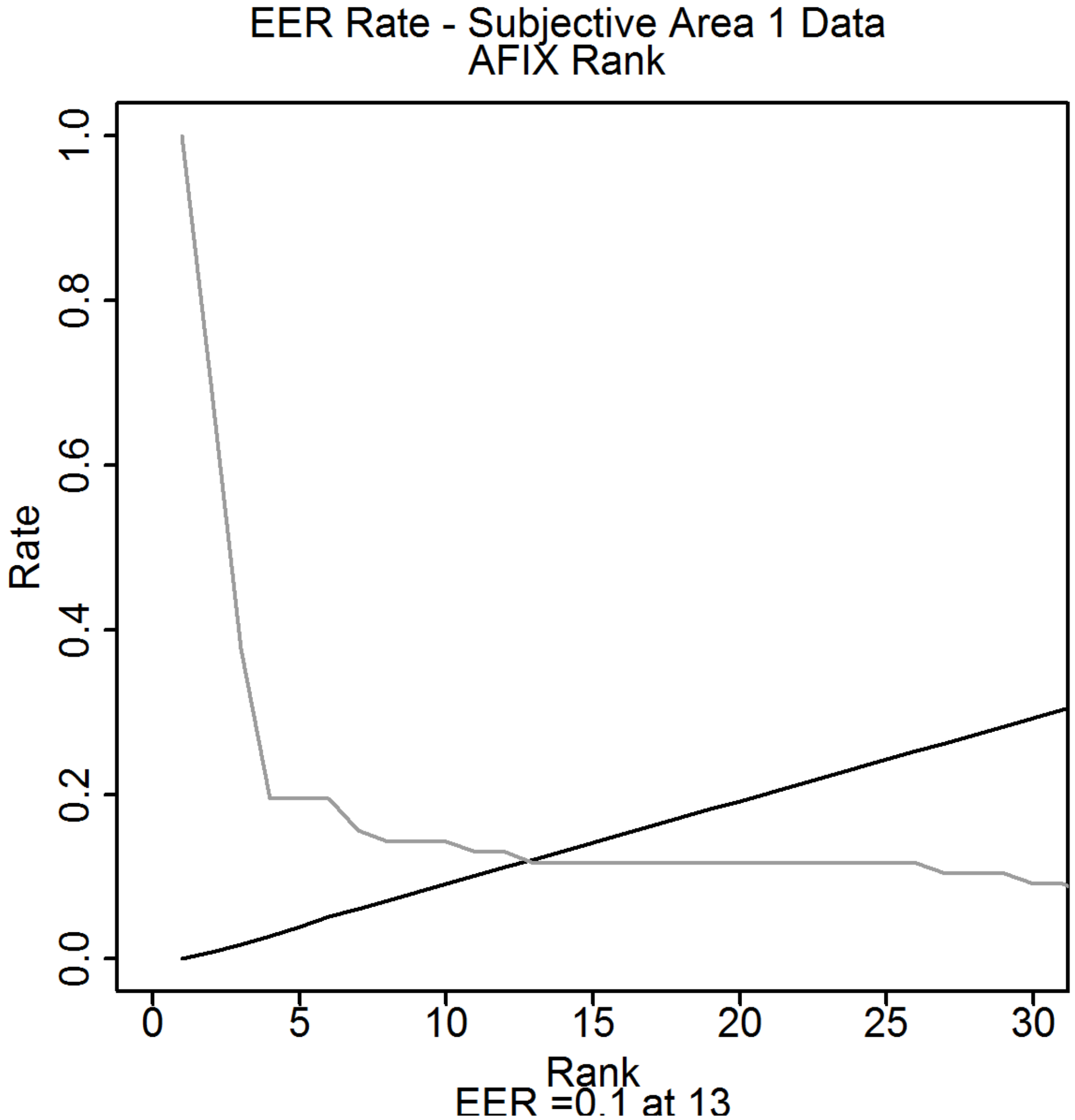


Figure D.118: This figure presents the EER plot from the subjective Area 1 images for the ranks.

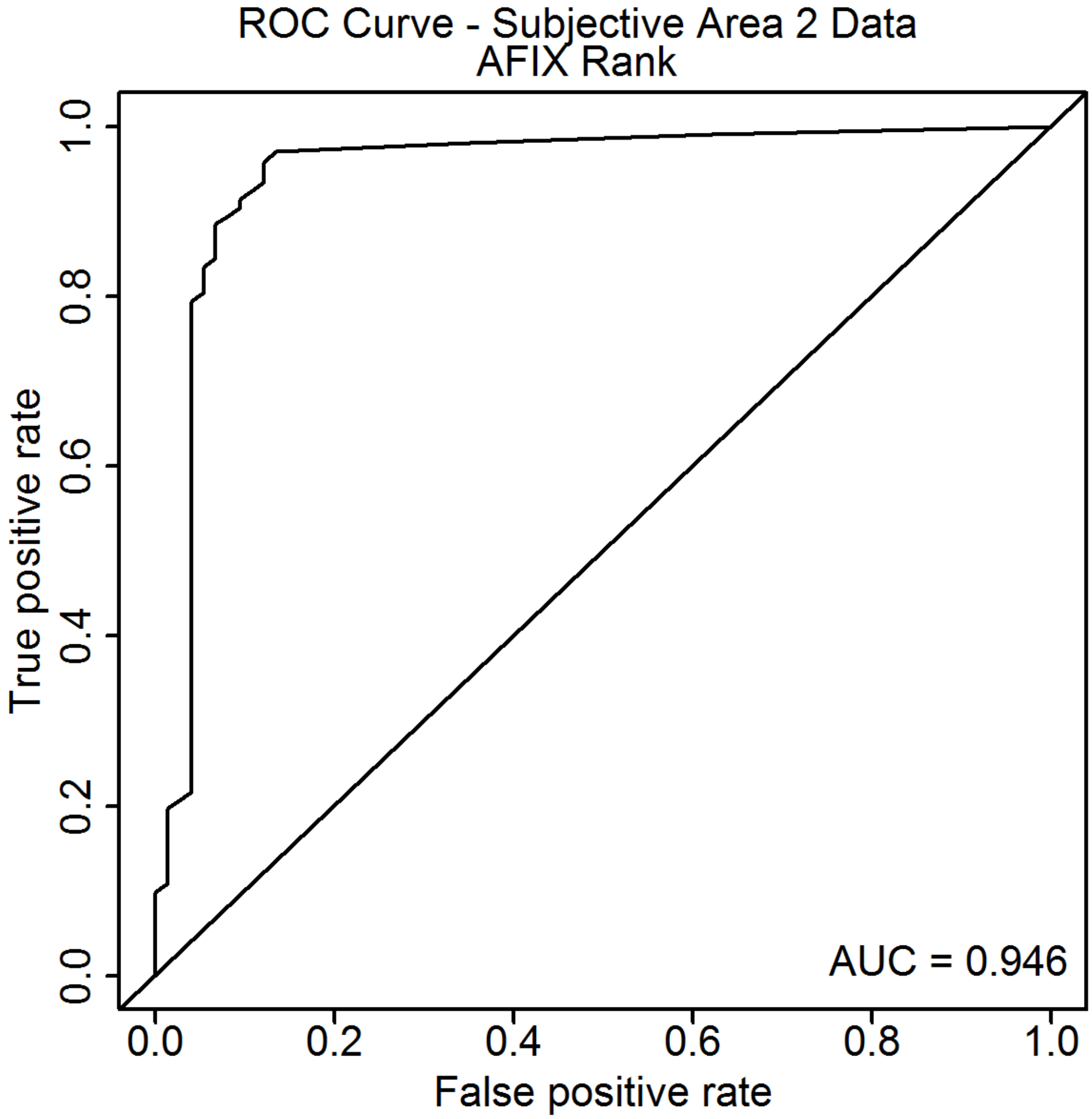


Figure D.119: This figure presents the ROC curve from the subjective Area 2 images for the ranks.

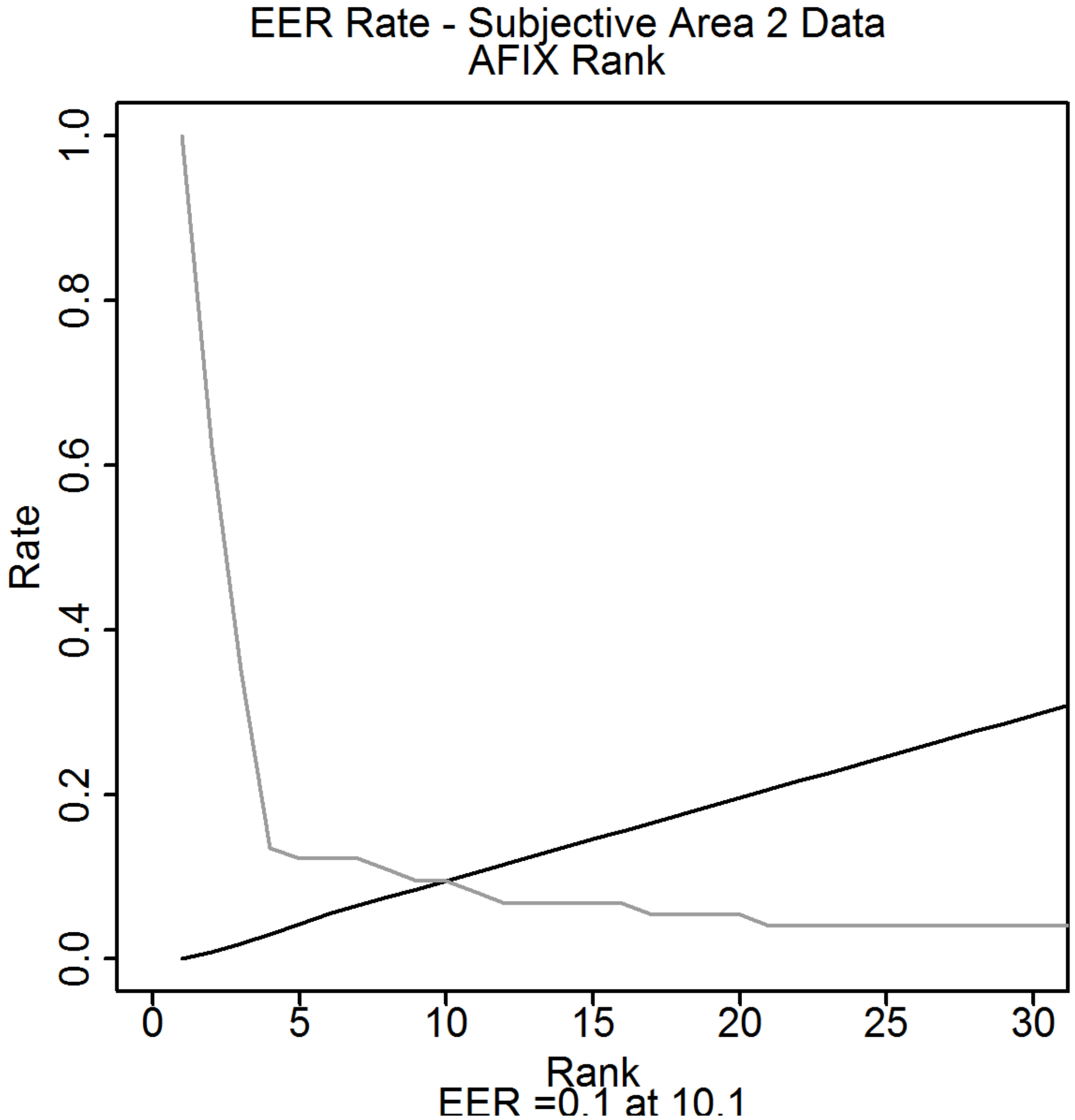


Figure D.120: This figure presents the EER plot from the subjective Area 2 images for the ranks.

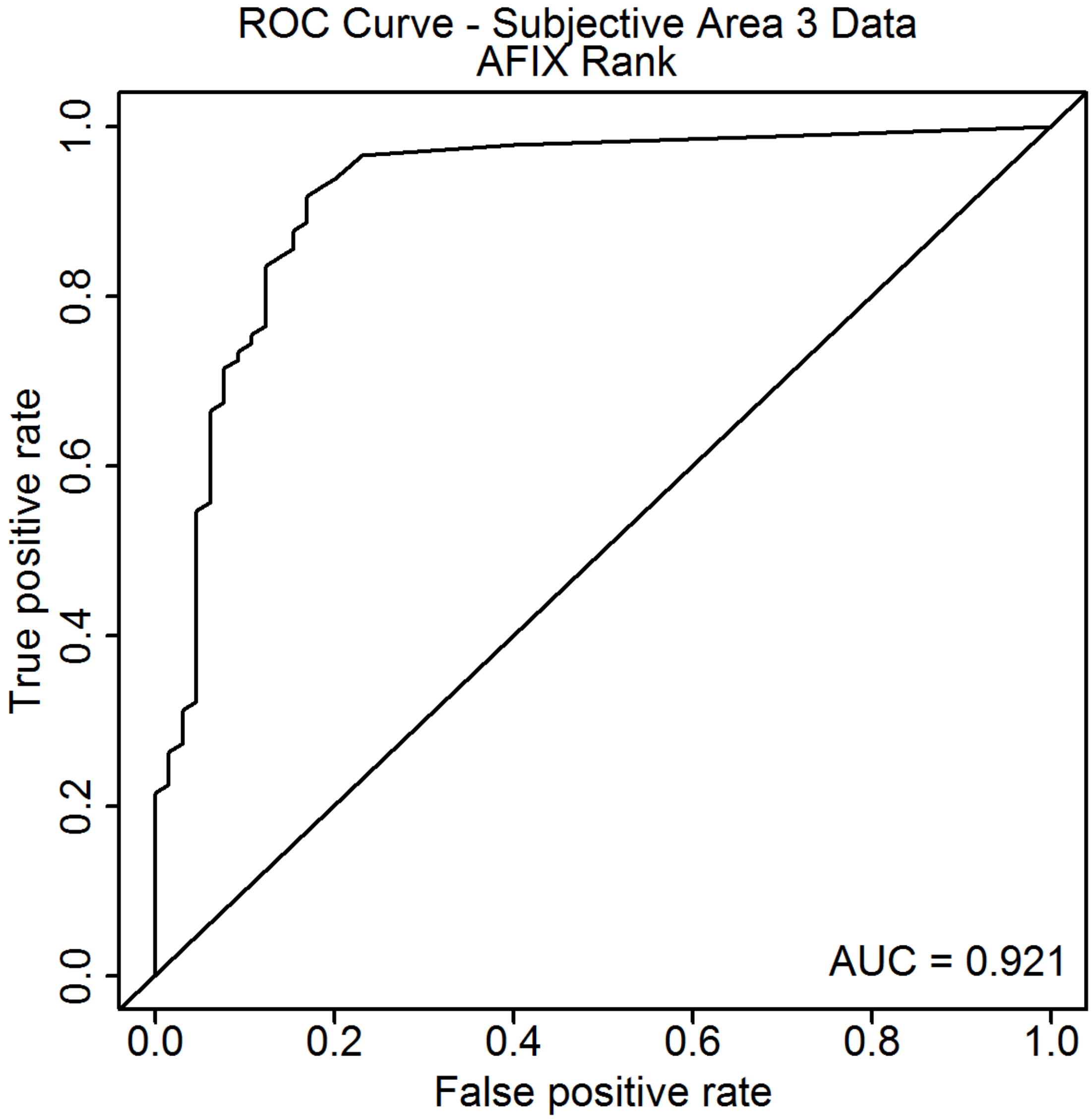


Figure D.121: This figure presents the ROC curve from the subjective Area 3 images for the ranks.

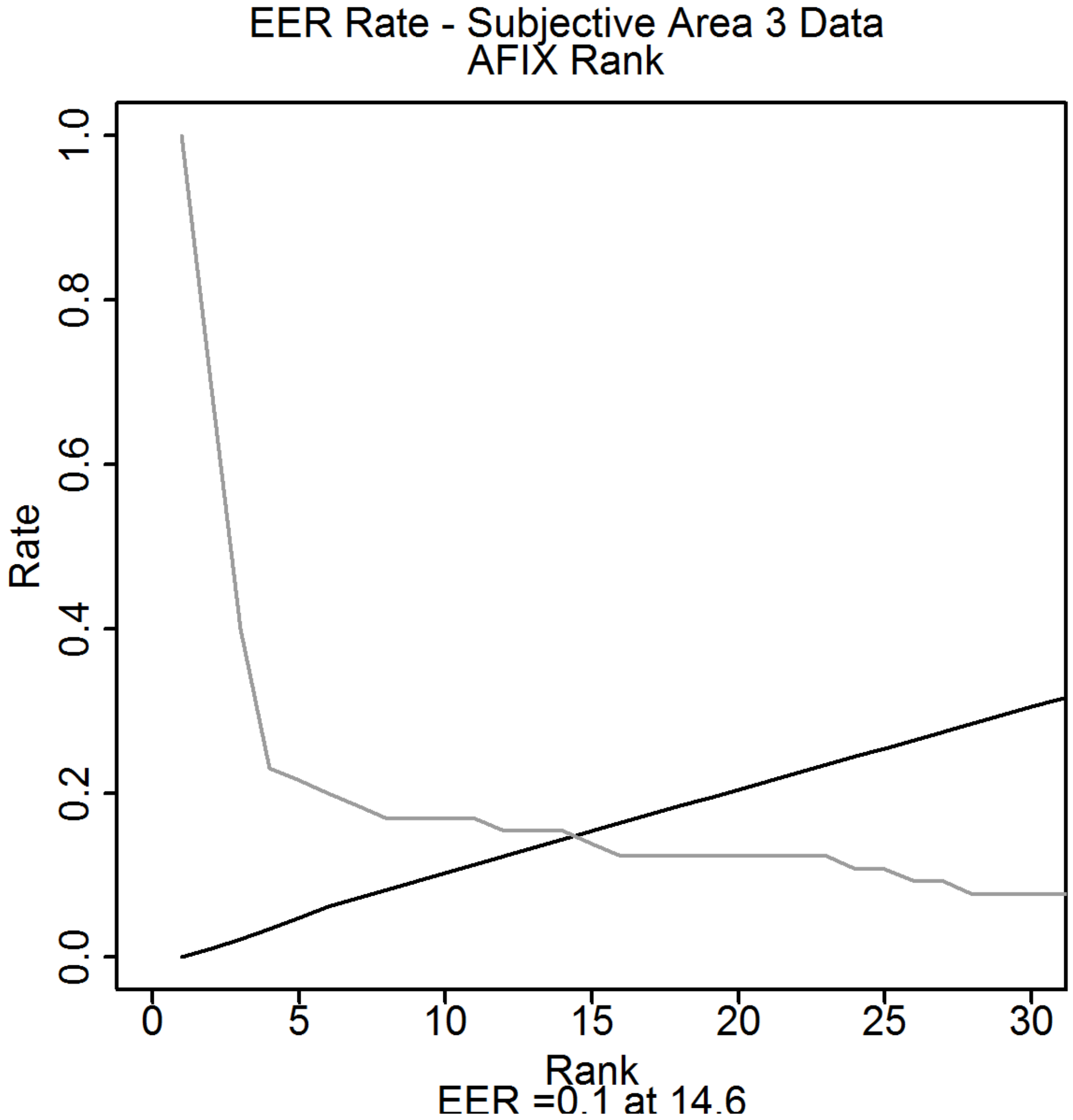


Figure D.122: This figure presents the EER plot from the subjective Area 3 images for the ranks.

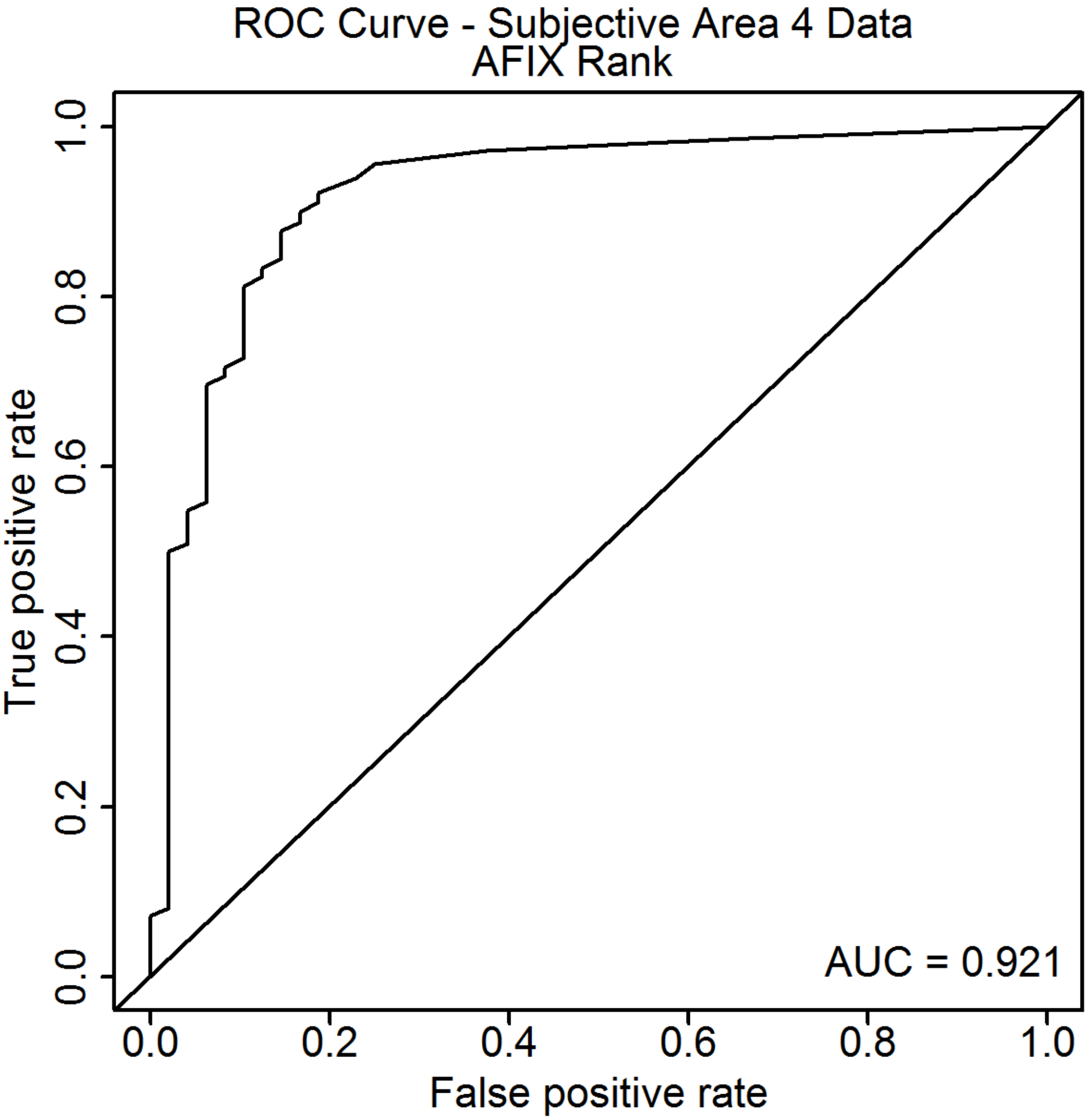


Figure D.123: This figure presents the ROC curve from the subjective Area 4 images for the ranks.

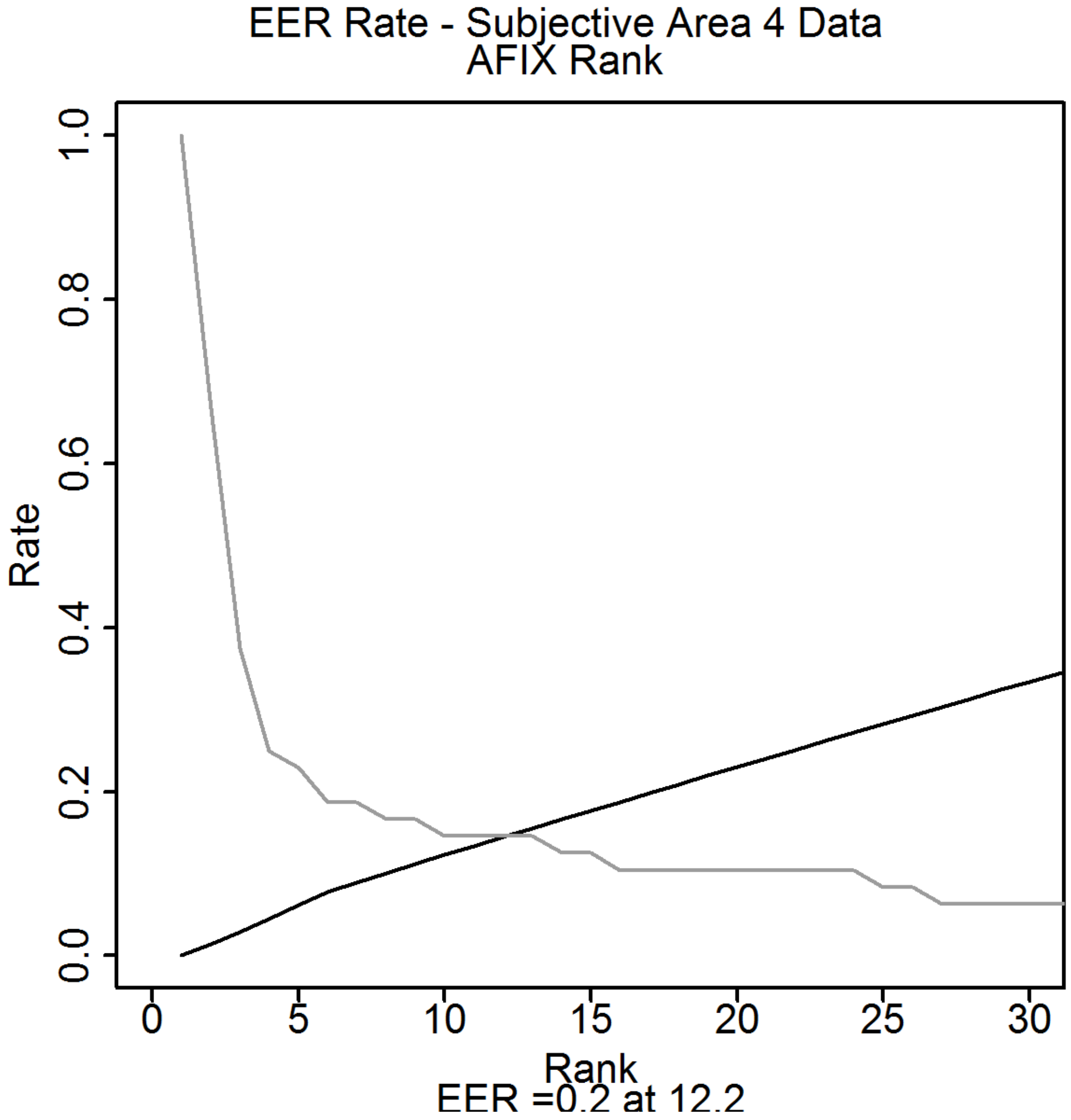


Figure D.124: This figure presents the EER plot from the subjective Area 4 images for the ranks.

D.8.2 Subjective Method: Automatic Extraction

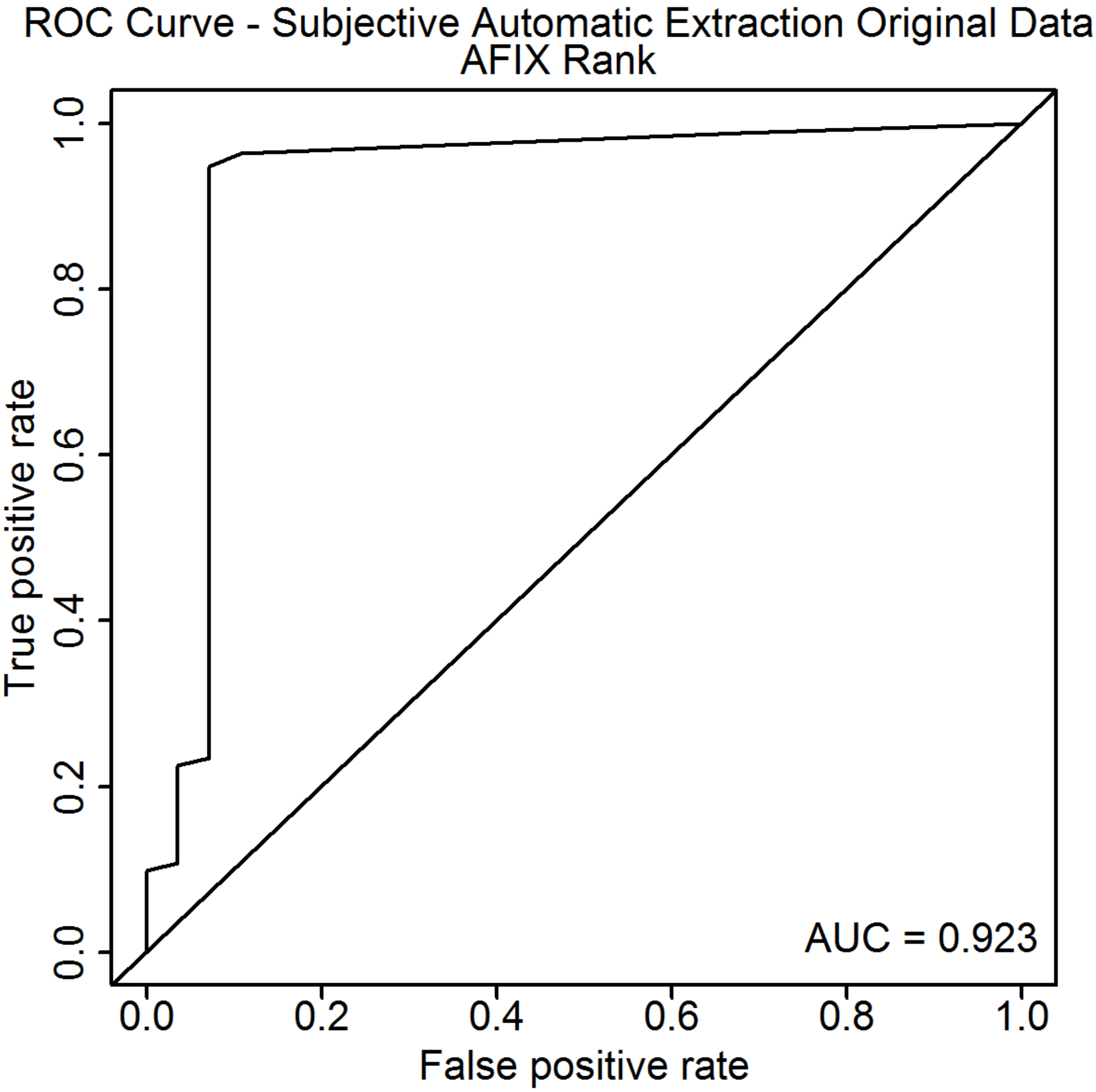


Figure D.125: This figure presents the ROC curve from the subjective original images for the ranks in the automatic extraction.

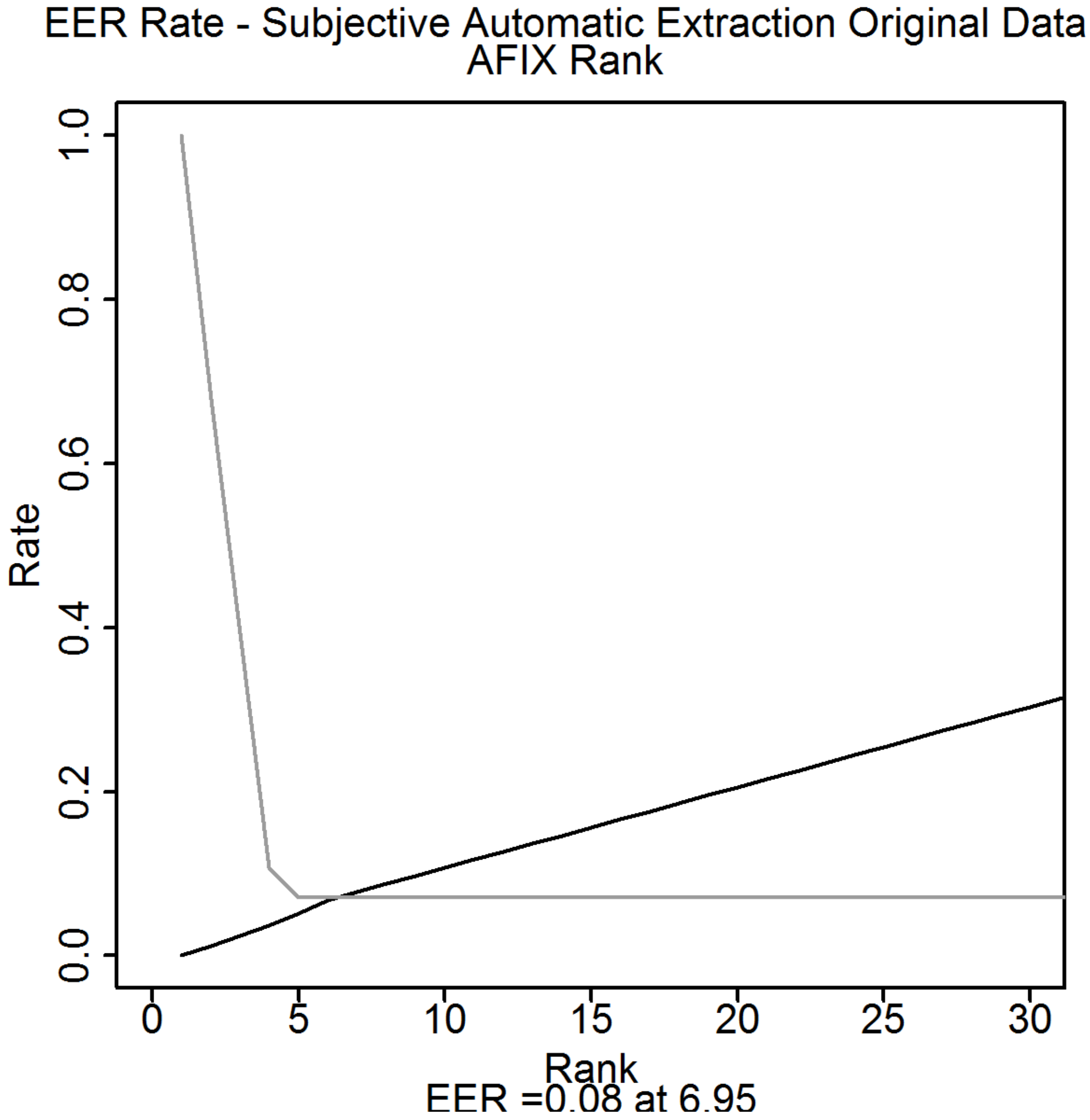


Figure D.126: This figure presents the EER plot from the subjective original images for the ranks in the automatic extraction.

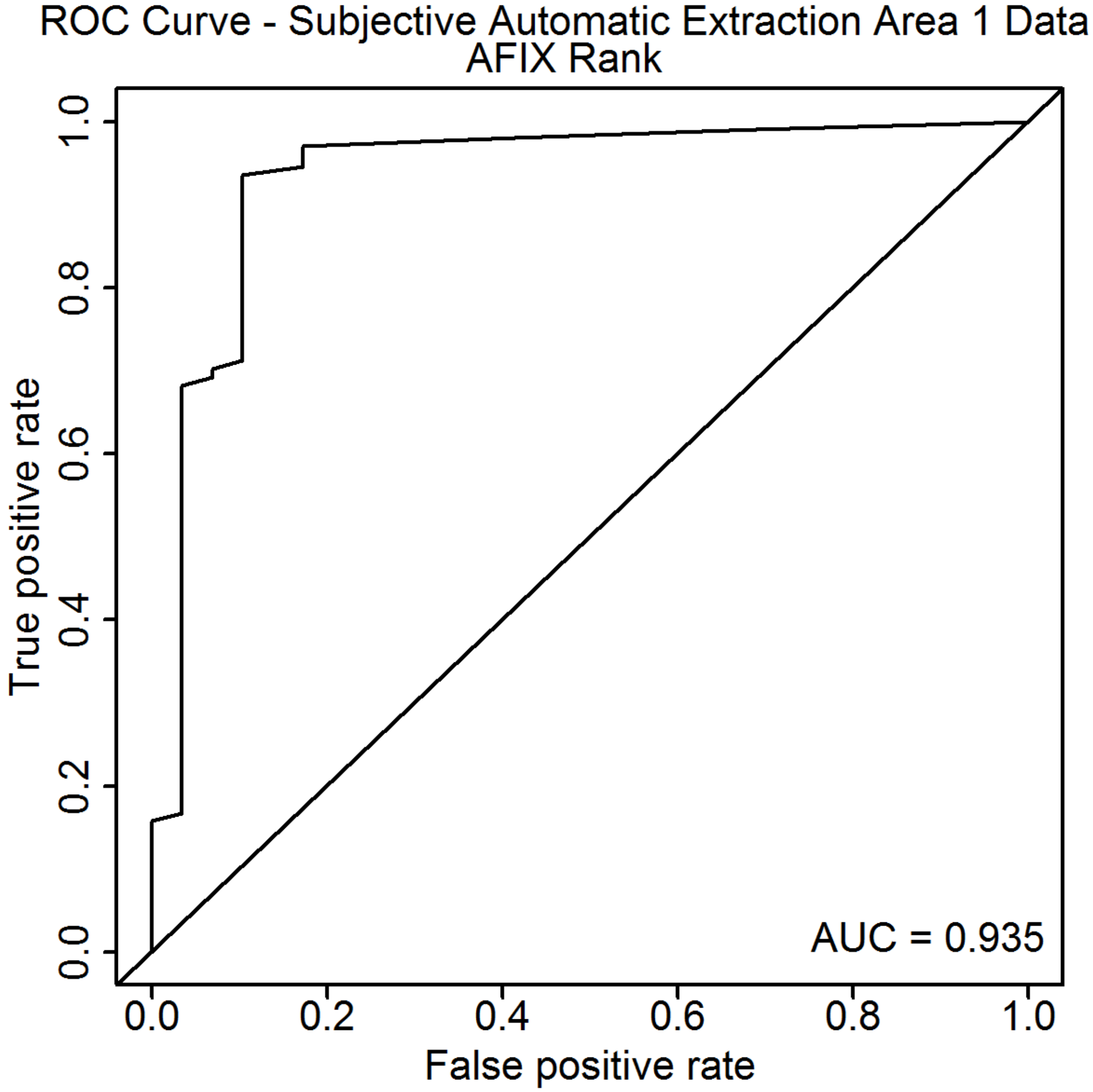


Figure D.127: This figure presents the ROC curve from the subjective Area 1 images for the ranks in the automatic extraction.

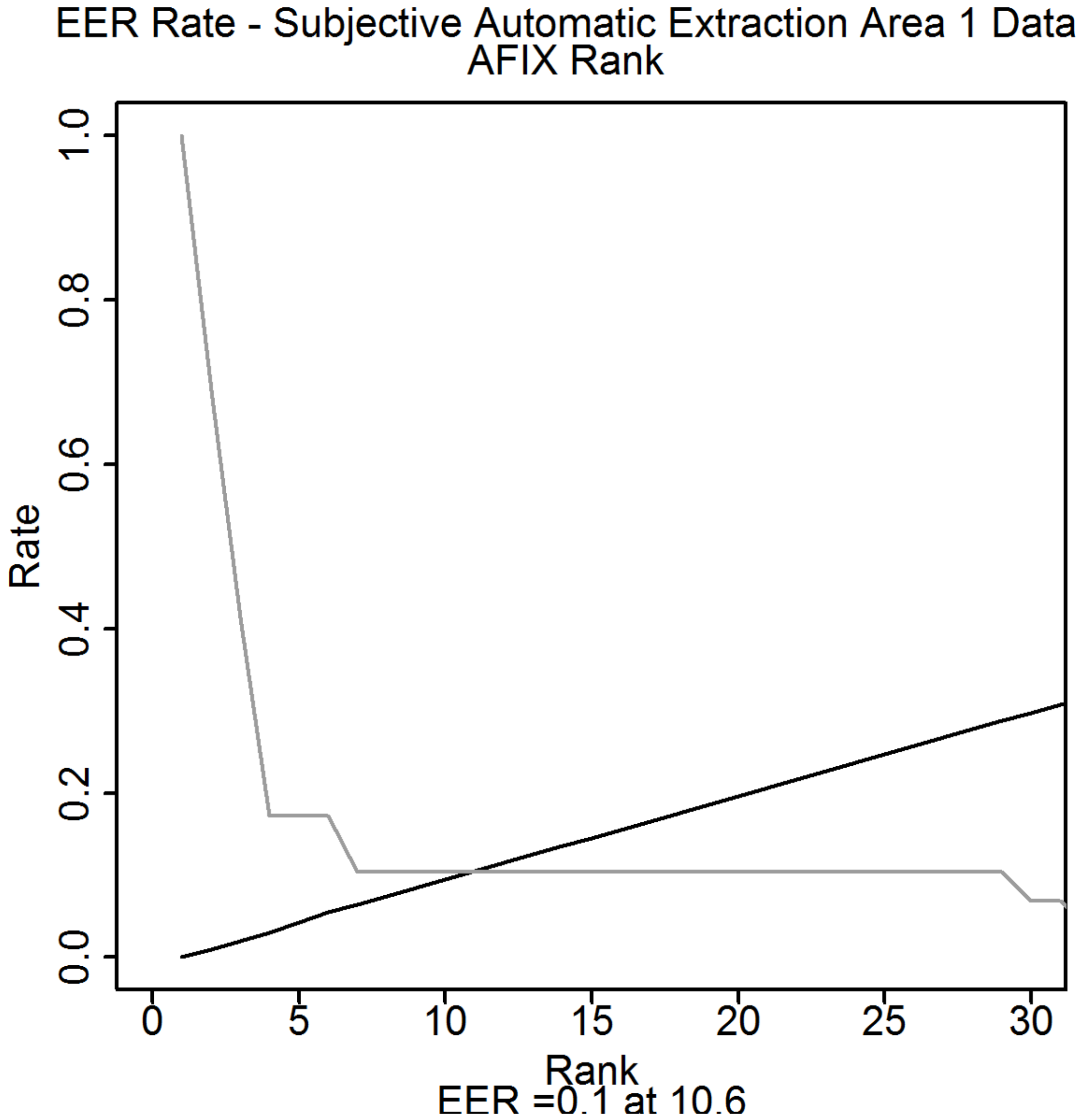


Figure D.128: This figure presents the EER plot from the subjective Area 1 images for the ranks in the automatic extraction.

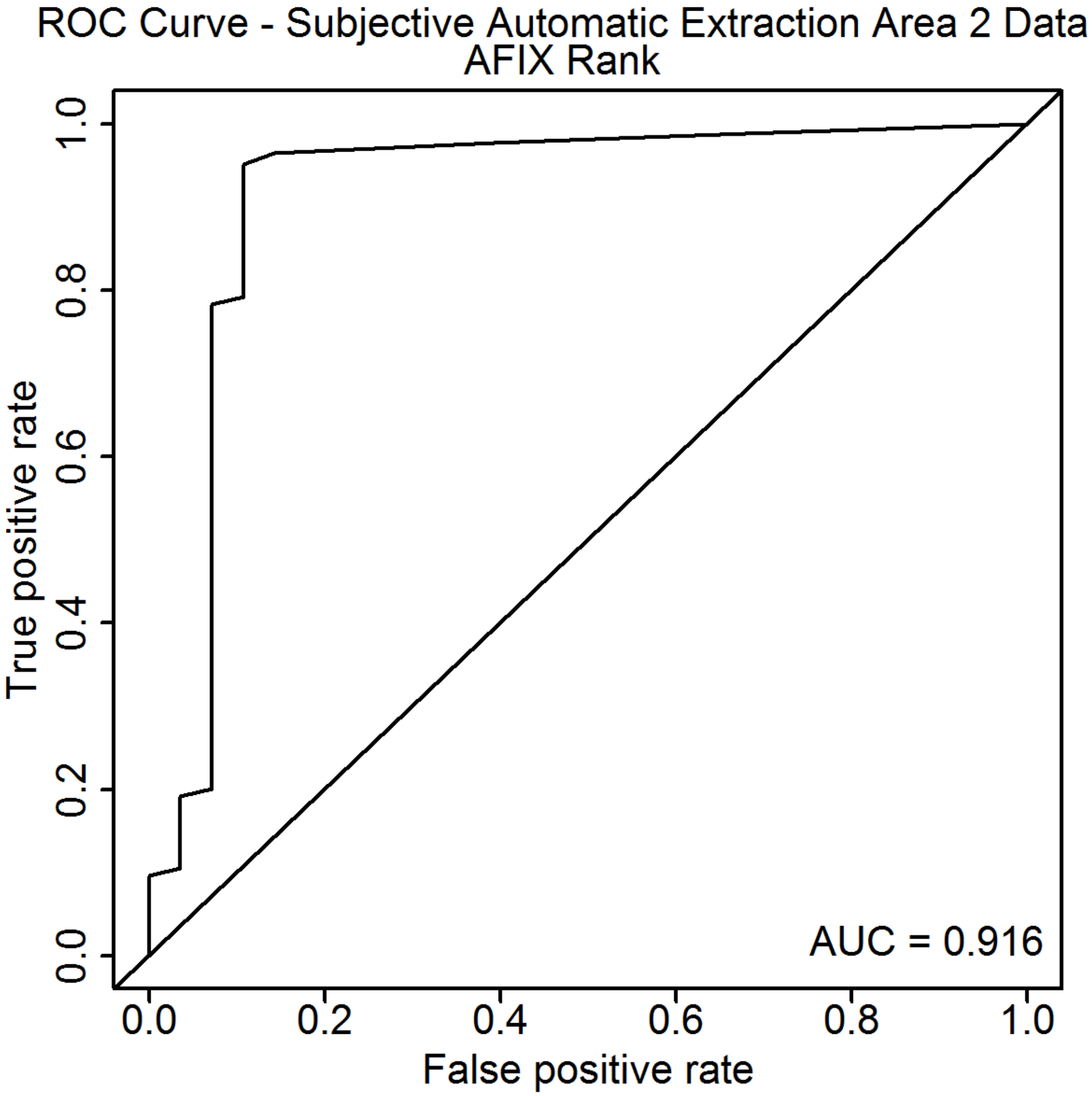


Figure D.129: This figure presents the ROC curve from the subjective Area 2 images for the ranks in the automatic extraction.

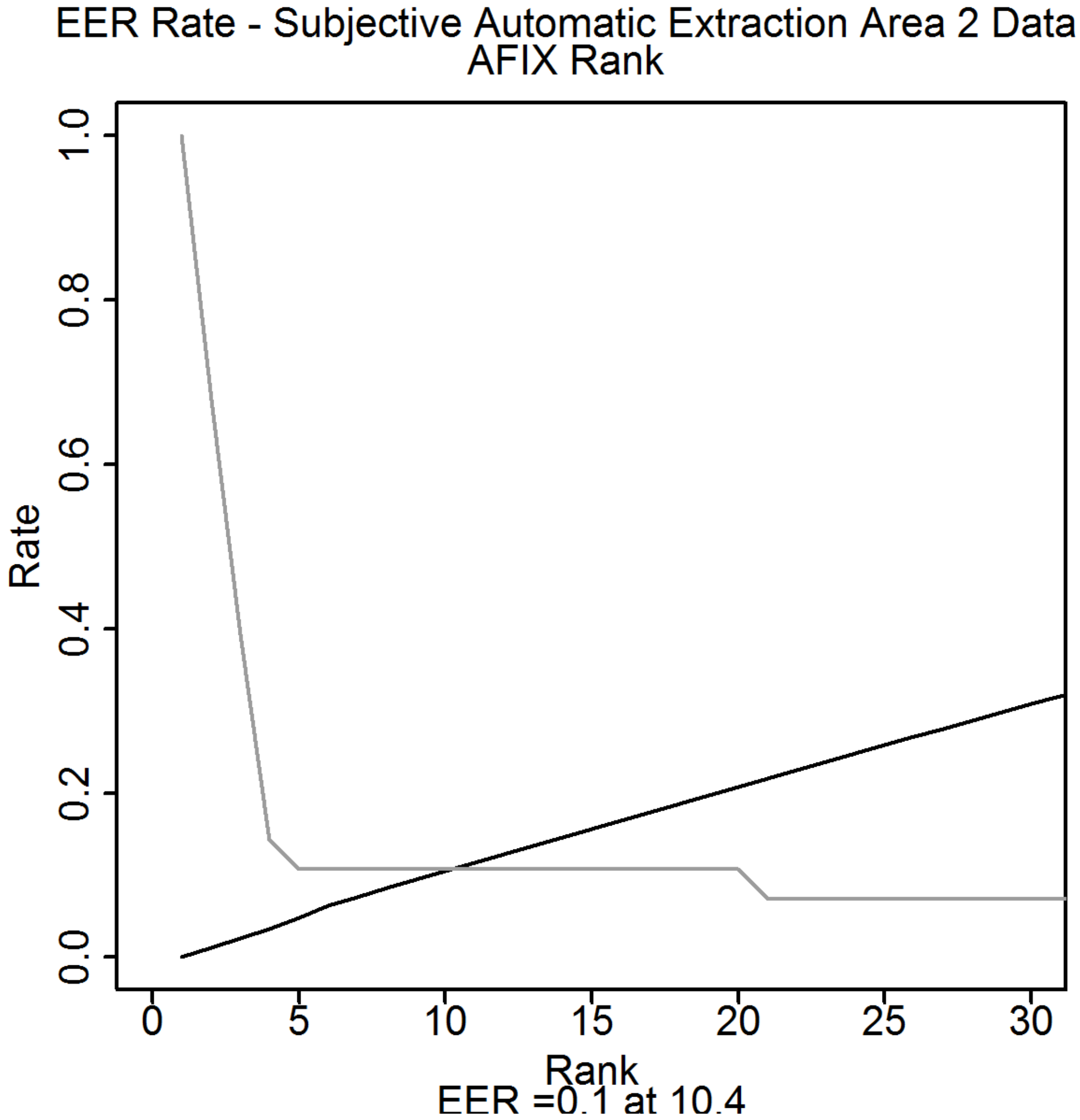


Figure D.130: This figure presents the EER plot from the subjective Area 2 images for the ranks in the automatic extraction.

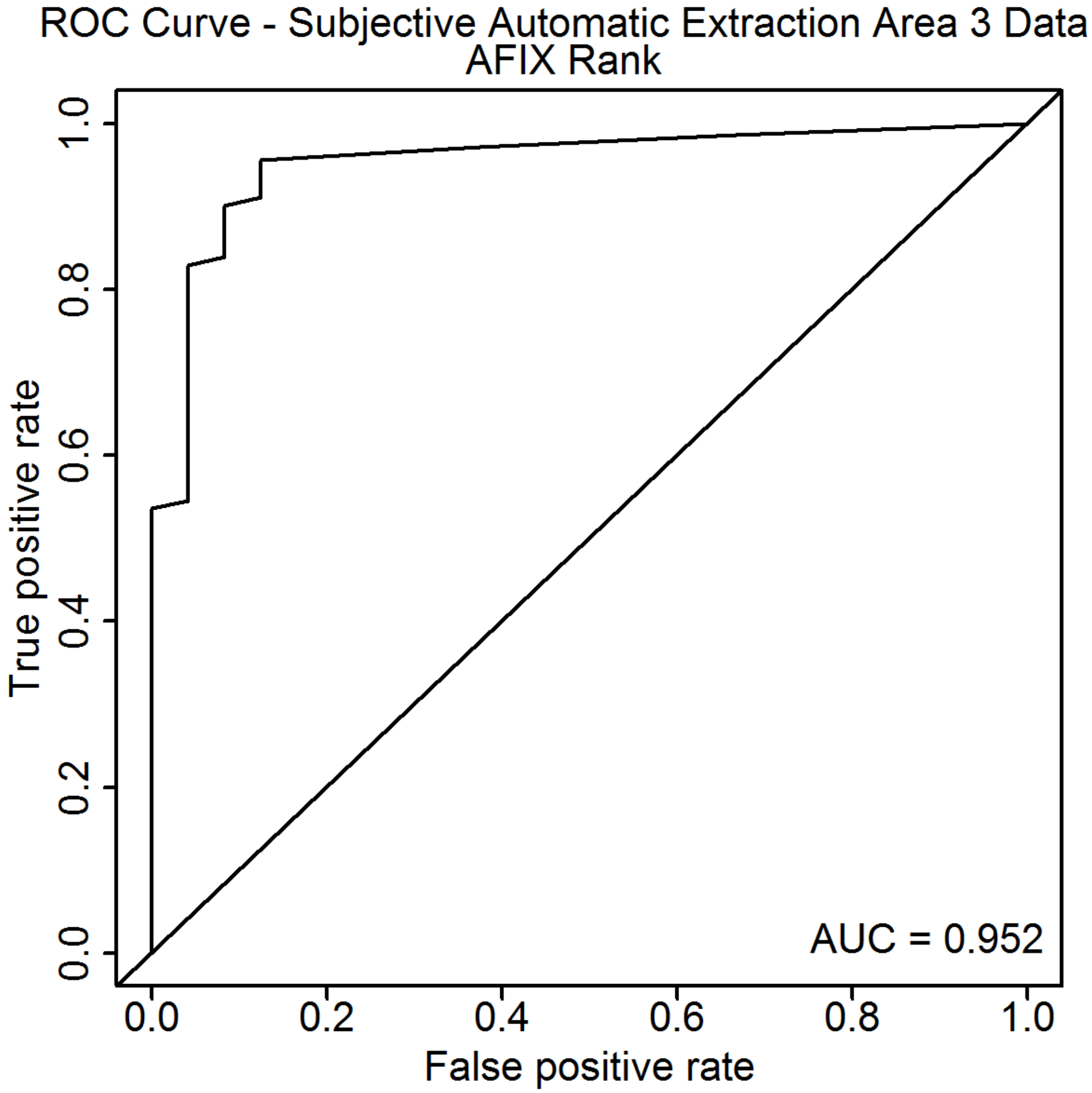


Figure D.131: This figure presents the ROC curve from the subjective Area 3 images for the ranks in the automatic extraction.

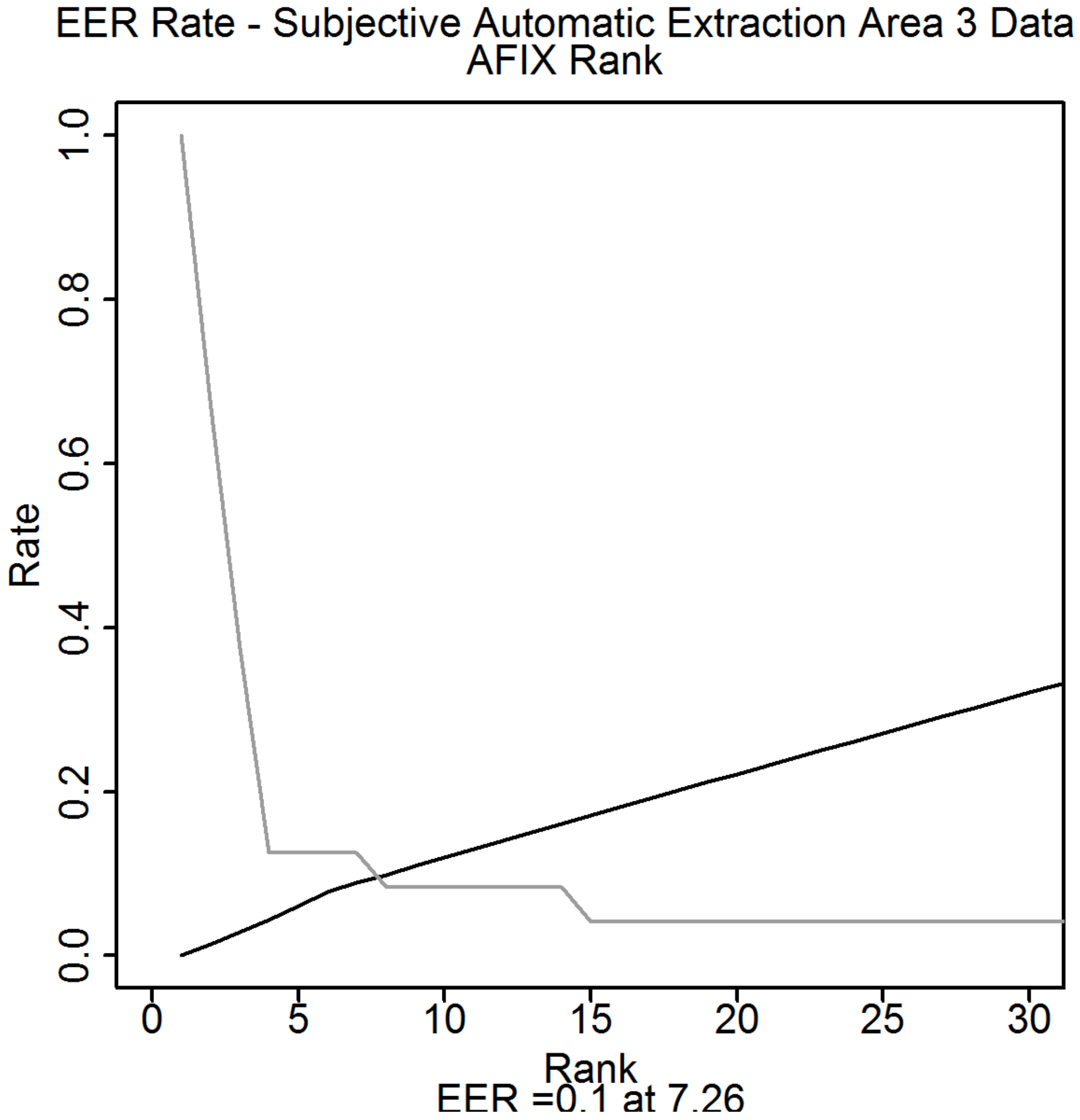


Figure D.132: This figure presents the EER plot from the subjective Area 3 images for the ranks in the automatic extraction.

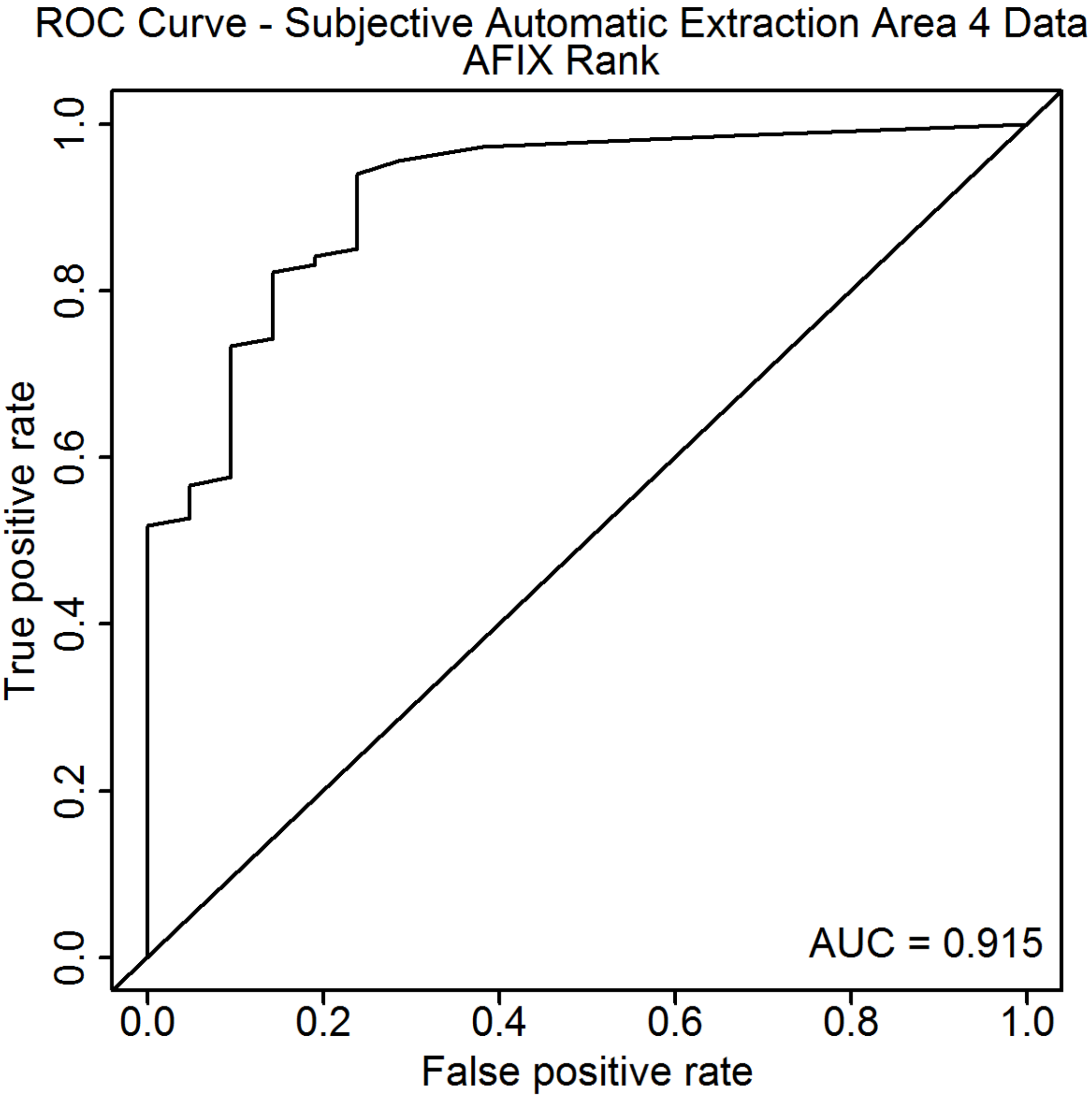


Figure D.133: This figure presents the ROC curve from the subjective Area 4 images for the ranks in the automatic extraction.

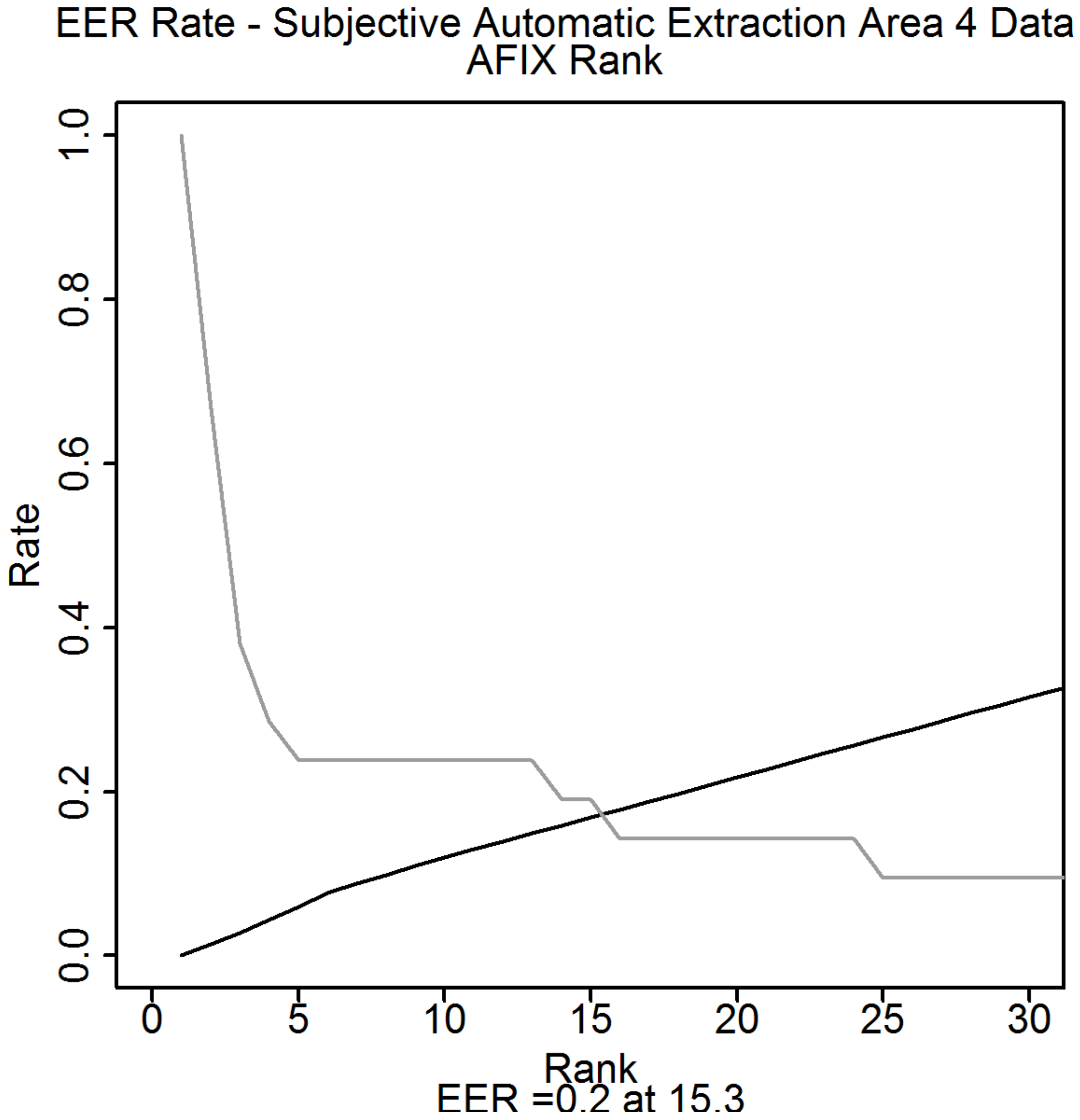


Figure D.134: This figure presents the EER plot from the subjective Area 4 images for the ranks in the automatic extraction.

D.8.3 Subjective Method: Manual Extraction

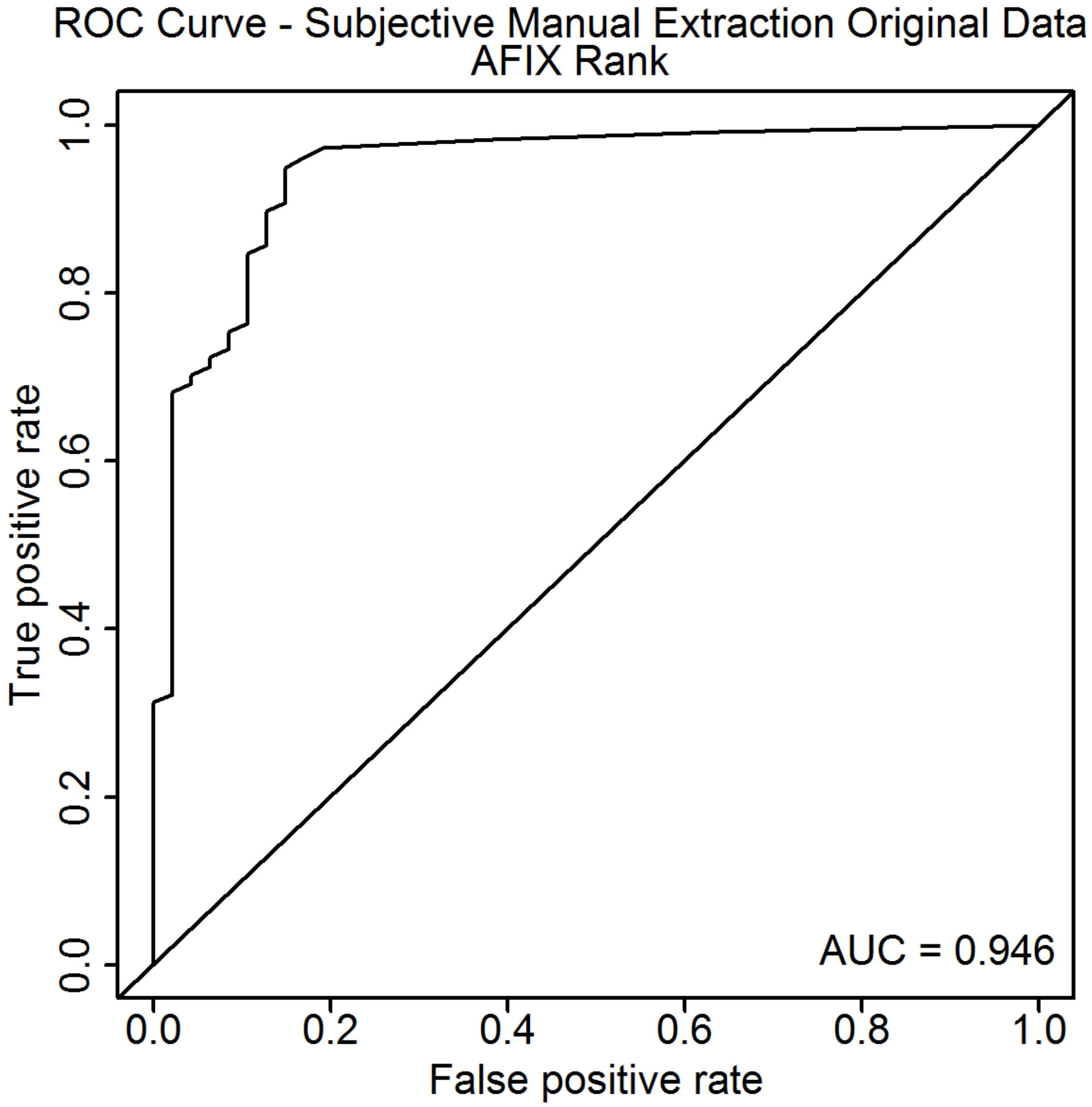


Figure D.135: This figure presents the ROC curve from the subjective original images for the ranks in the manual extraction.

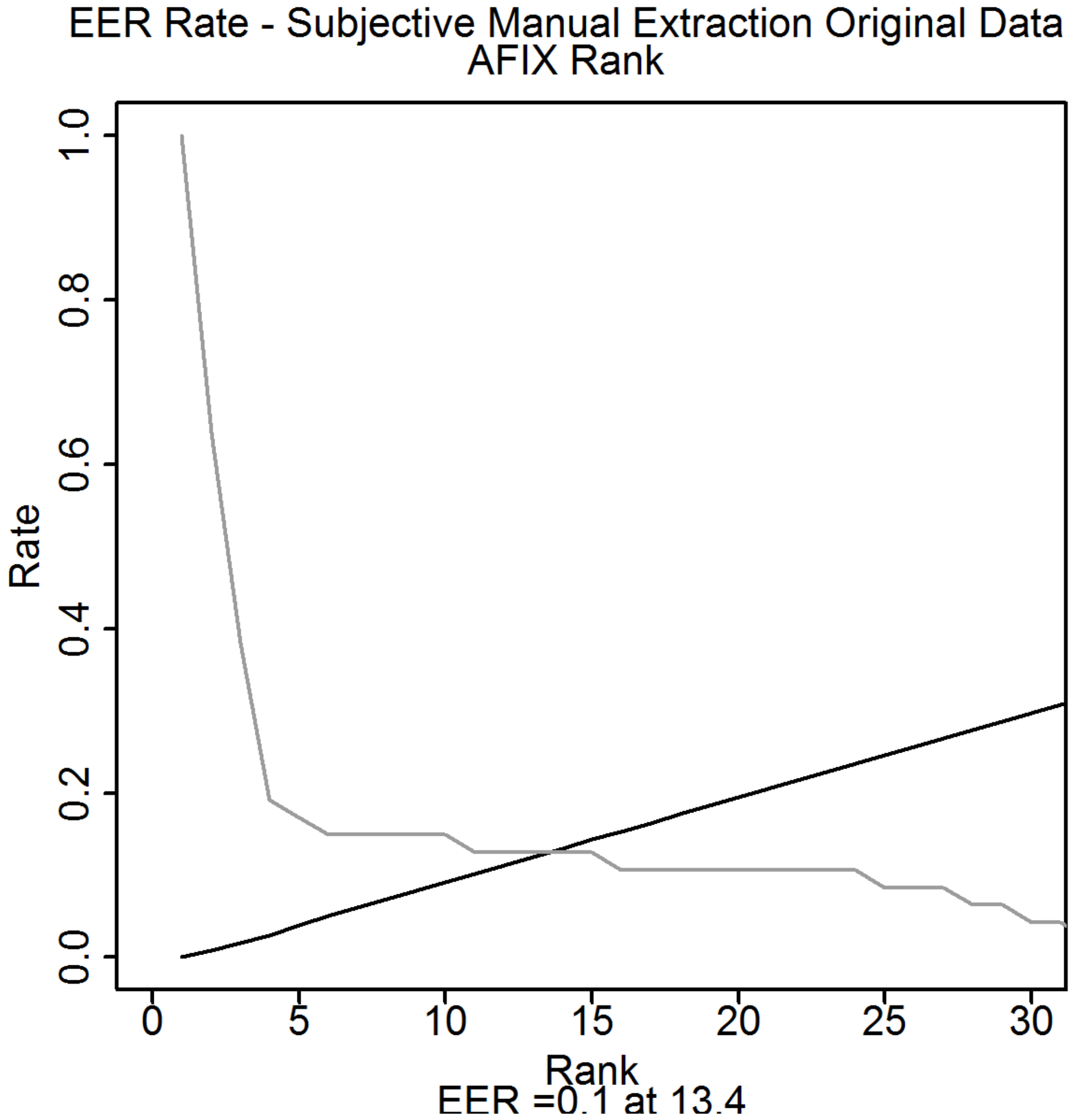


Figure D.136: This figure presents the EER plot from the subjective original images for the ranks in the manual extraction.

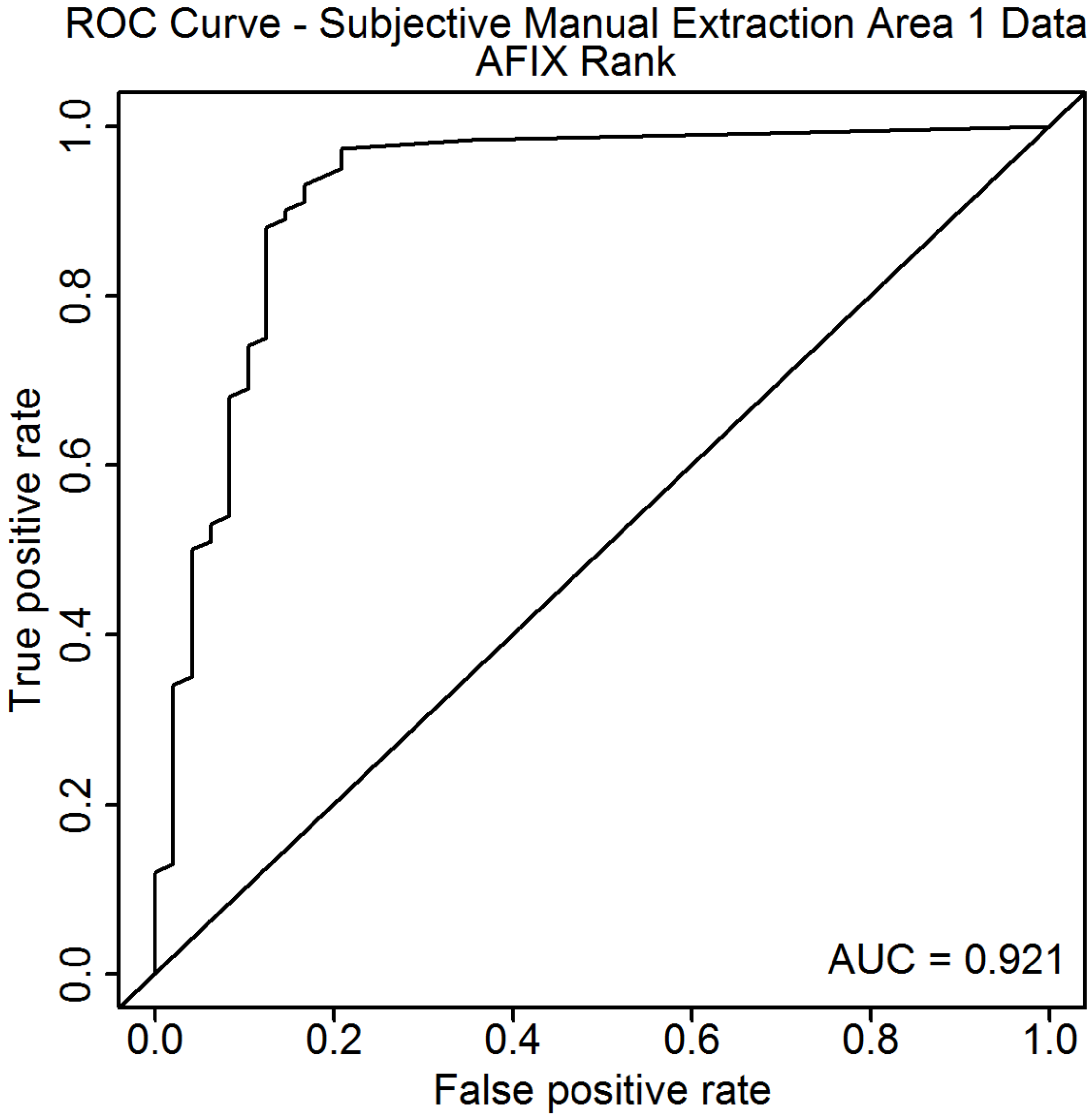


Figure D.137: This figure presents the ROC curve from the subjective Area 1 images for the ranks in the manual extraction.

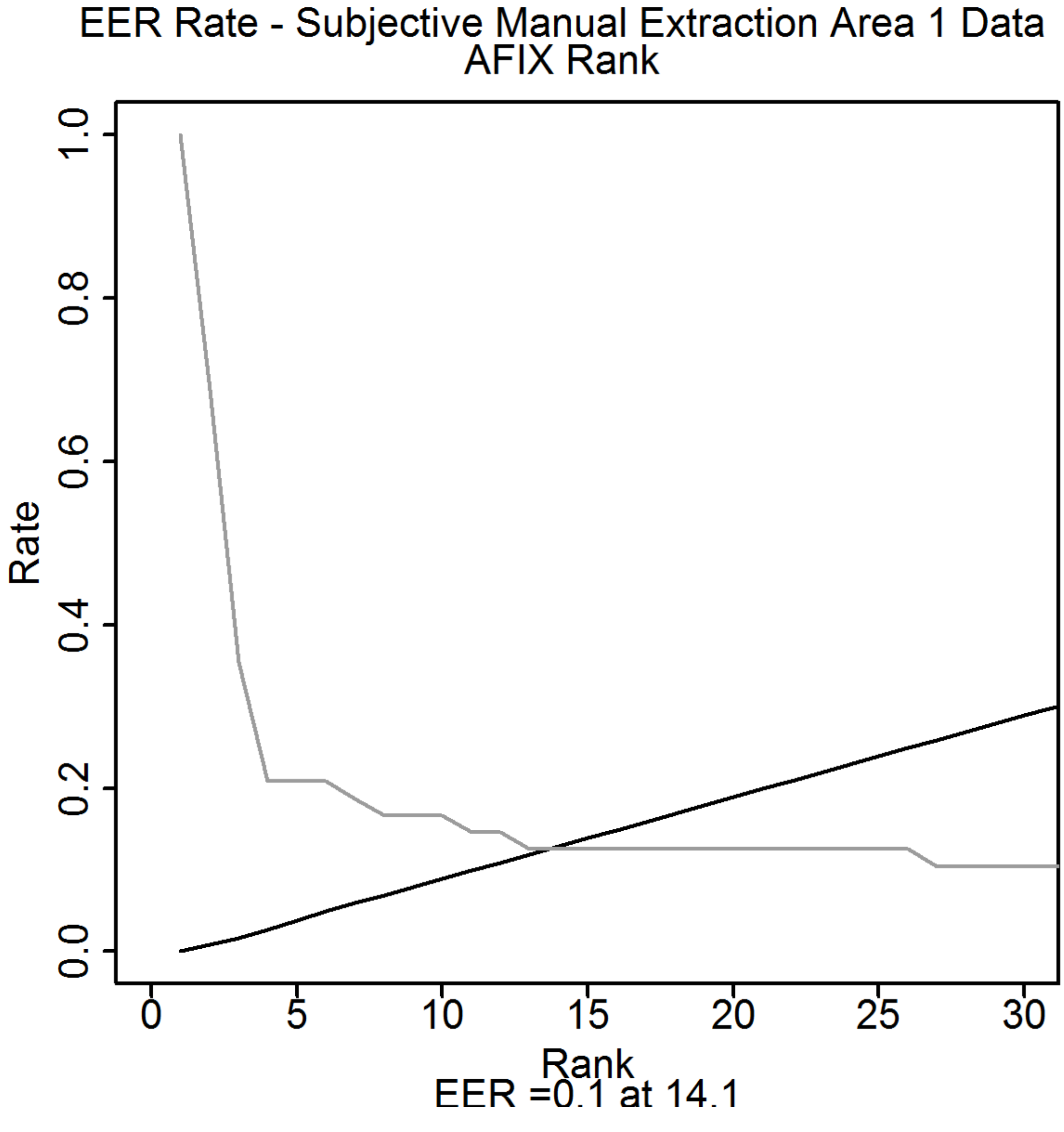


Figure D.138: This figure presents the EER plot from the subjective Area 1 images for the ranks in the manual extraction.

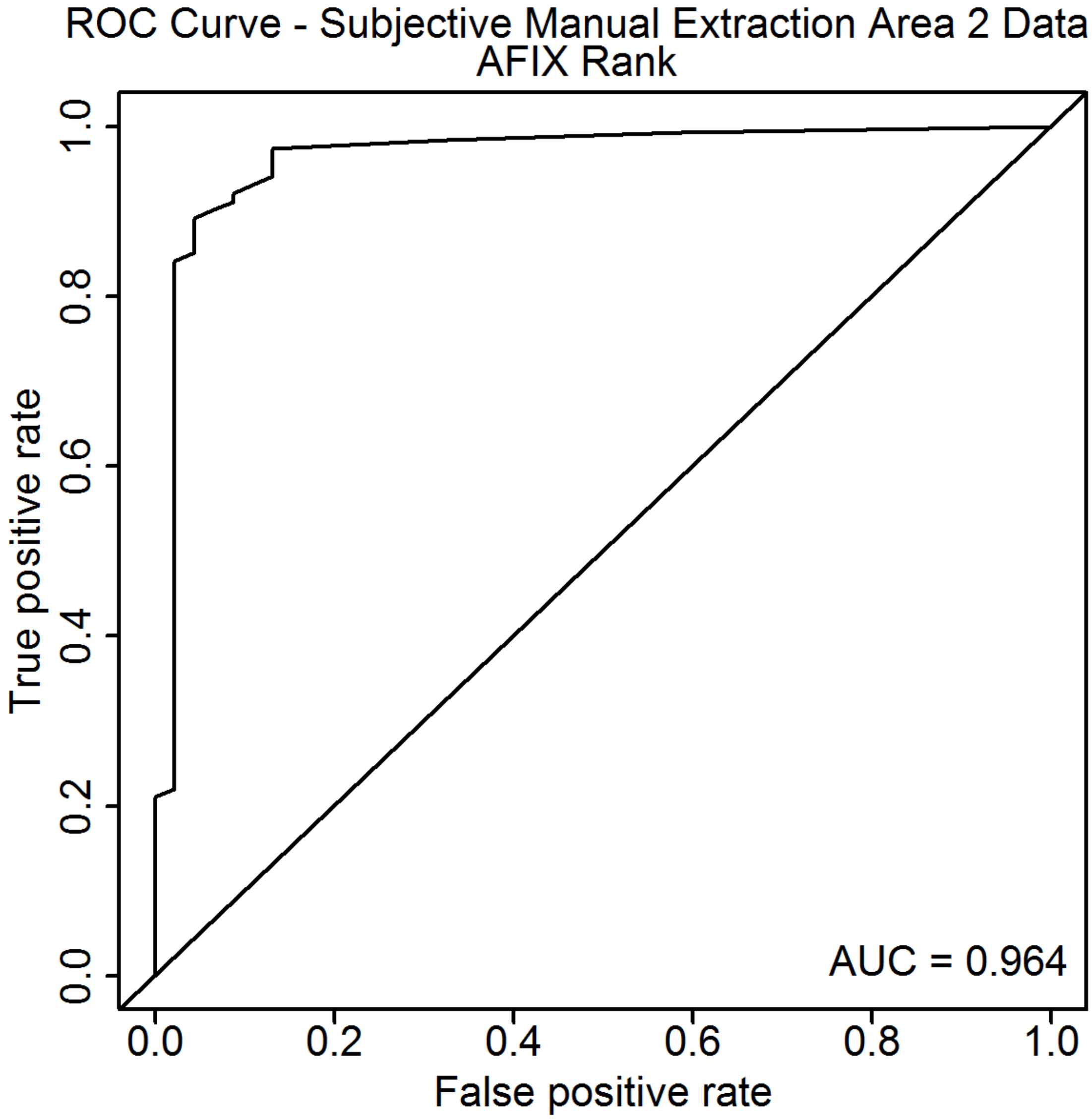


Figure D.139: This figure presents the ROC curve from the subjective Area 2 images for the ranks in the manual extraction.

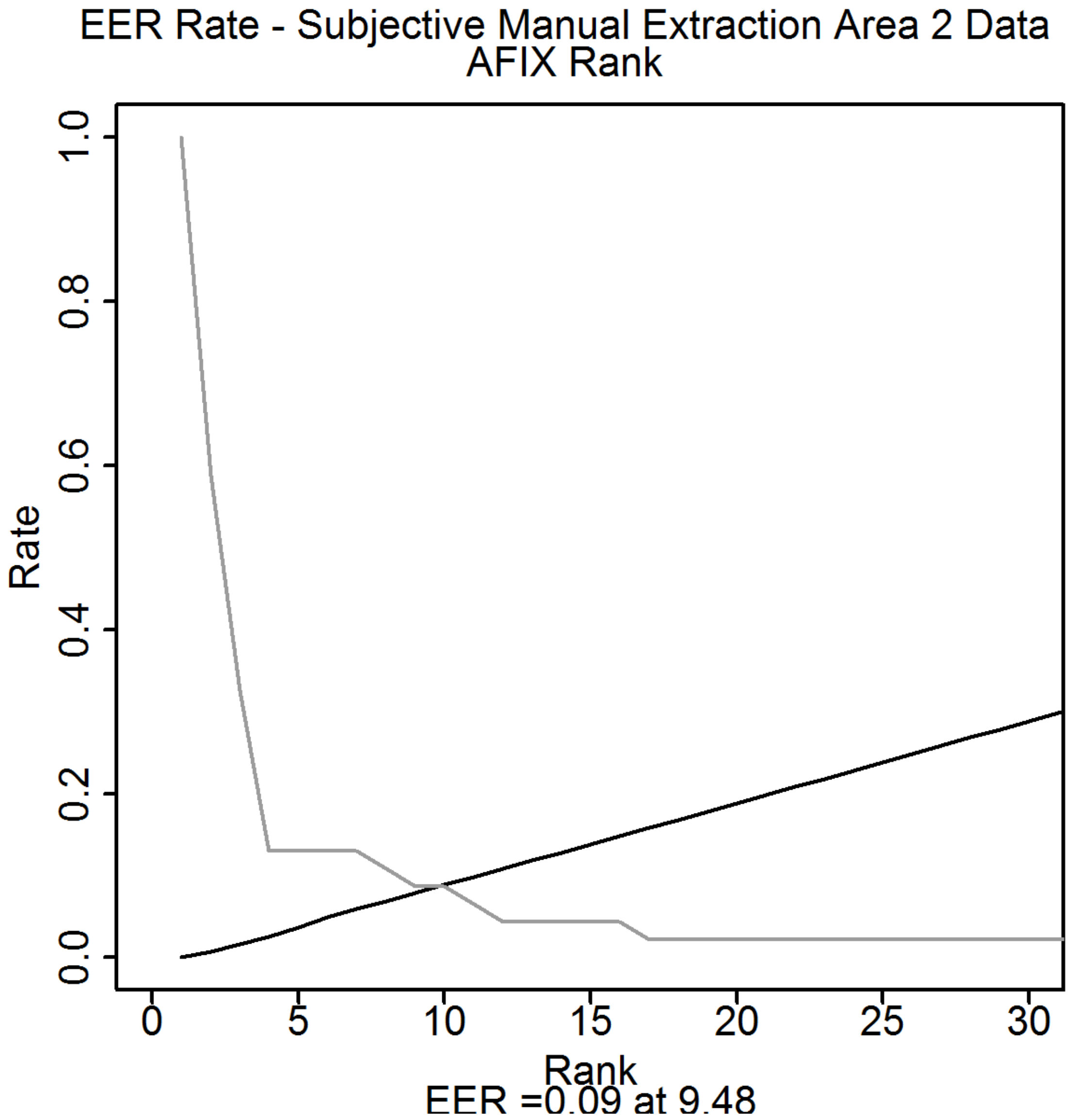


Figure D.140: This figure presents the EER plot from the subjective Area 2 images for the ranks in the manual extraction.

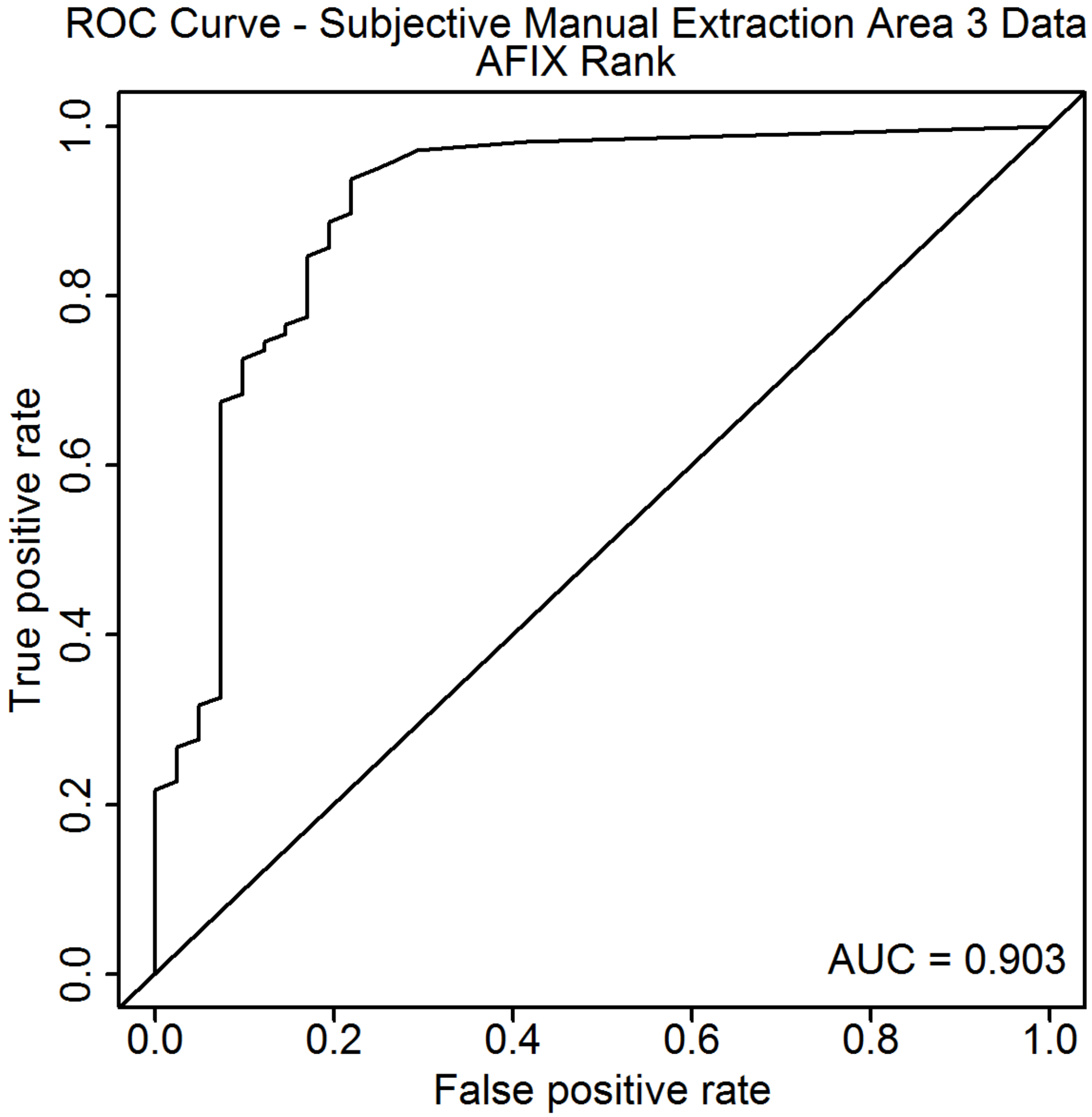


Figure D.141: This figure presents the ROC curve from the subjective Area 3 images for the ranks in the manual extraction.

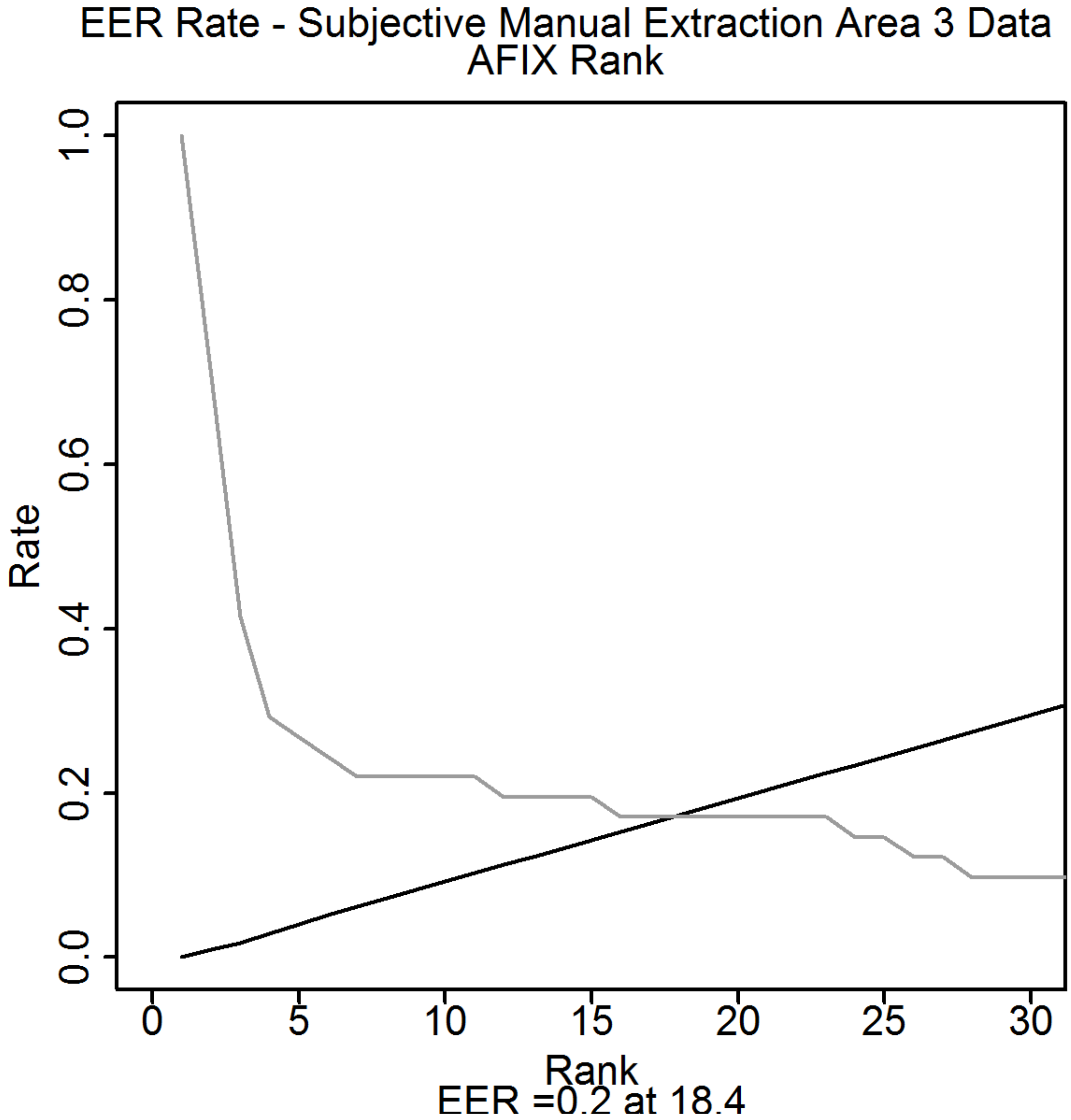


Figure D.142: This figure presents the EER plot from the subjective Area 3 images for the ranks in the manual extraction.

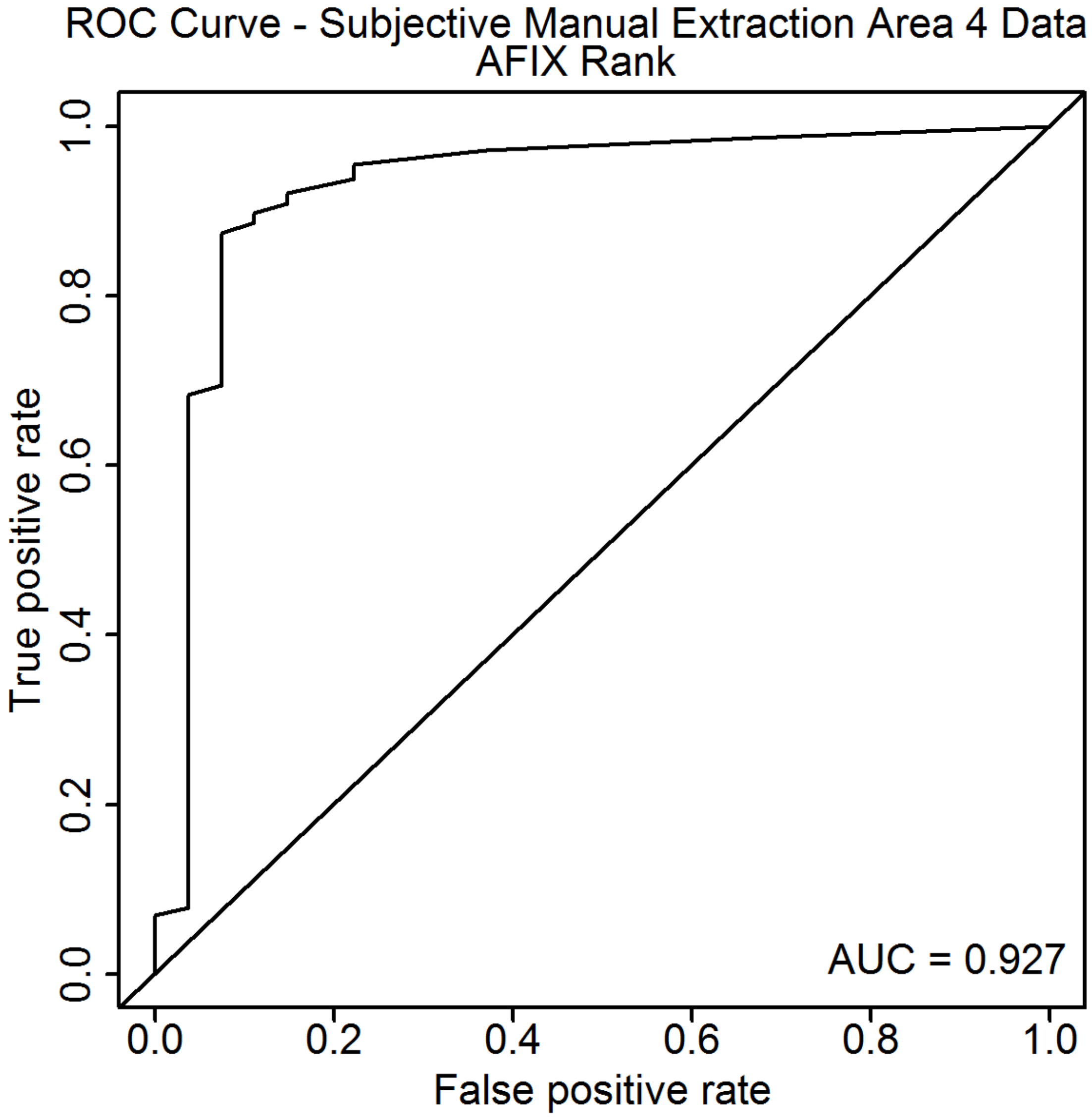


Figure D.143: This figure presents the ROC curve from the subjective Area 4 images for the ranks in the manual extraction.

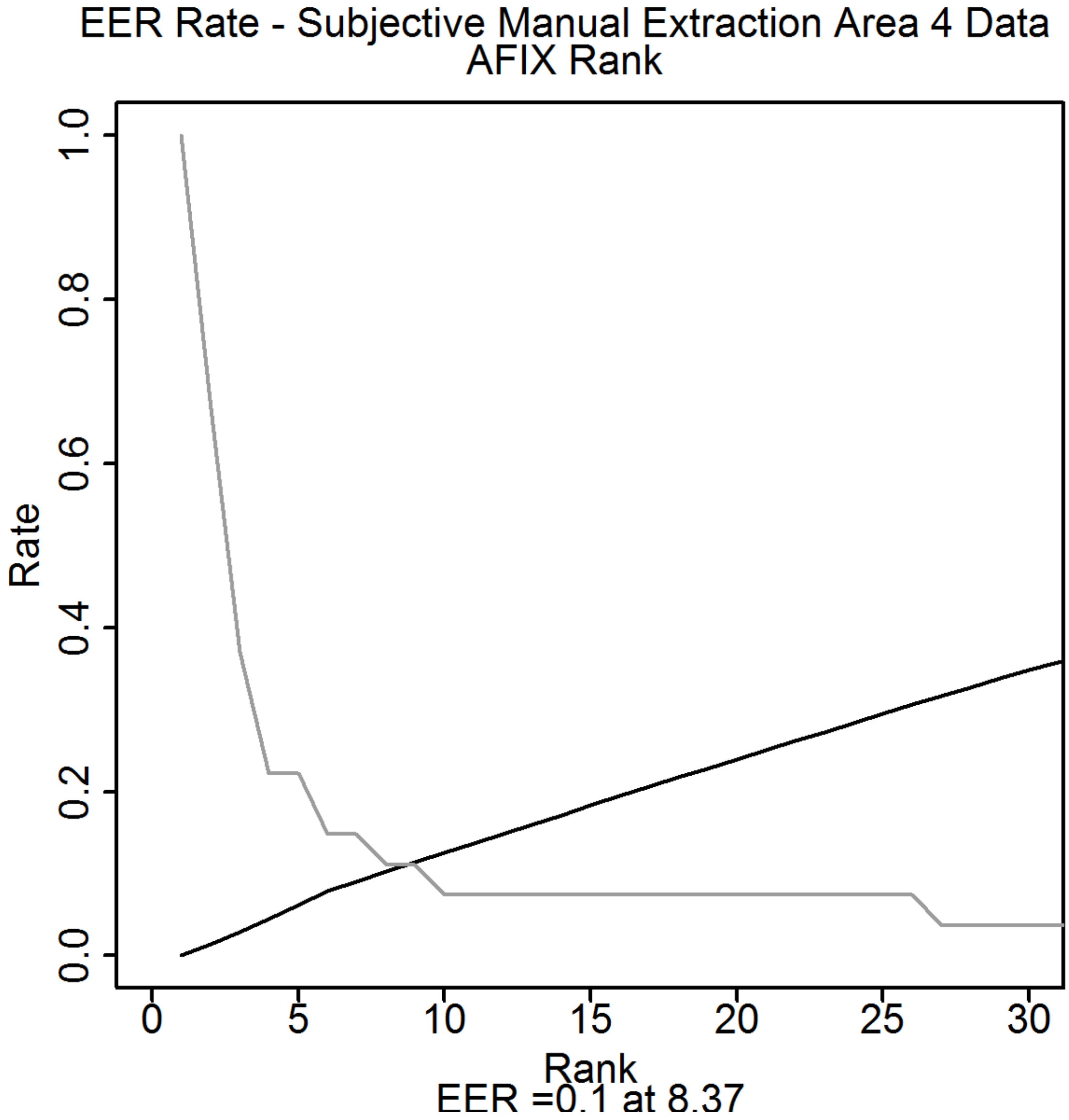


Figure D.144: This figure presents the EER plot from the subjective Area 4 images for the ranks in the manual extraction.

D.8.4 Objective Method: All Samples

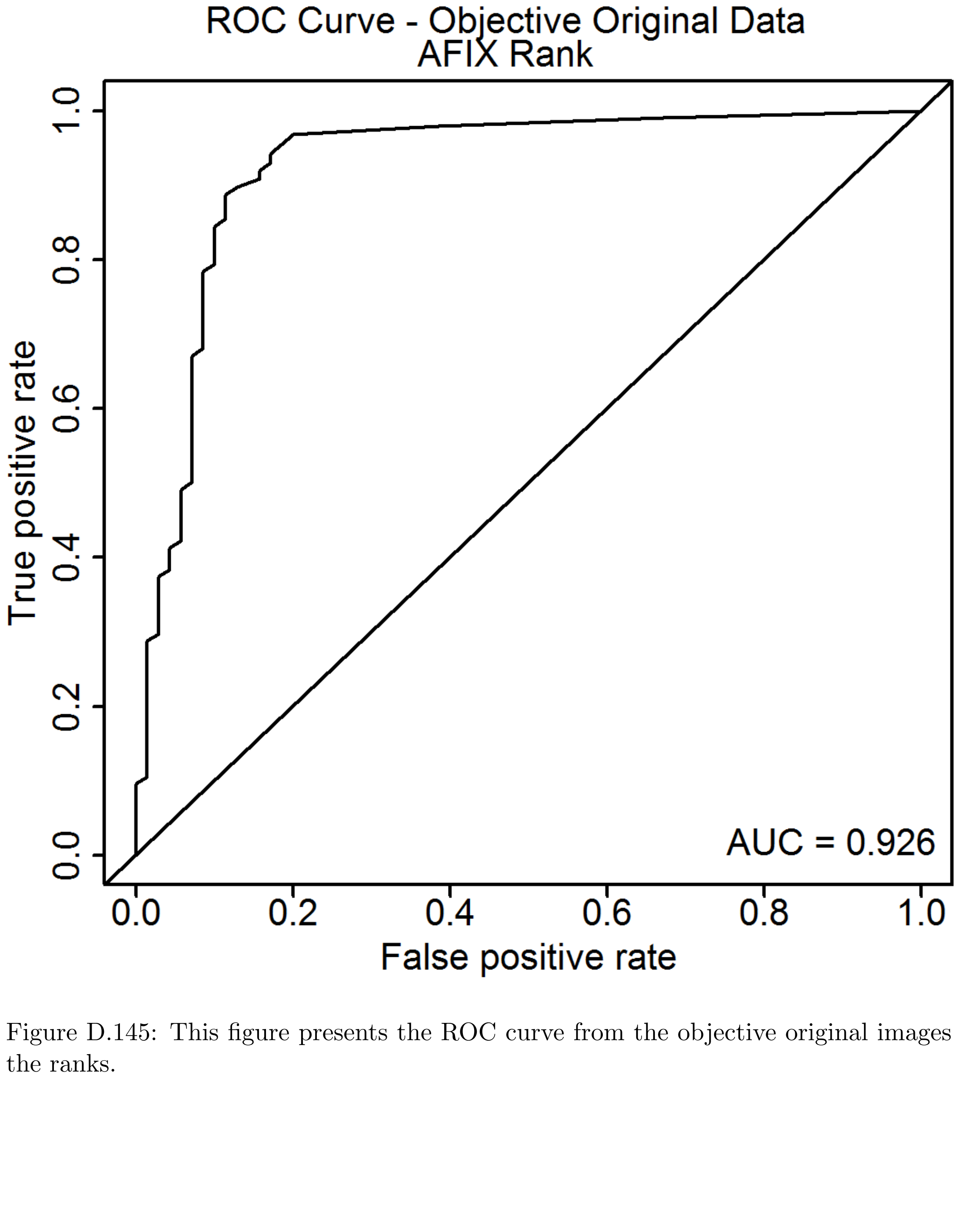


Figure D.145: This figure presents the ROC curve from the objective original images for the ranks.

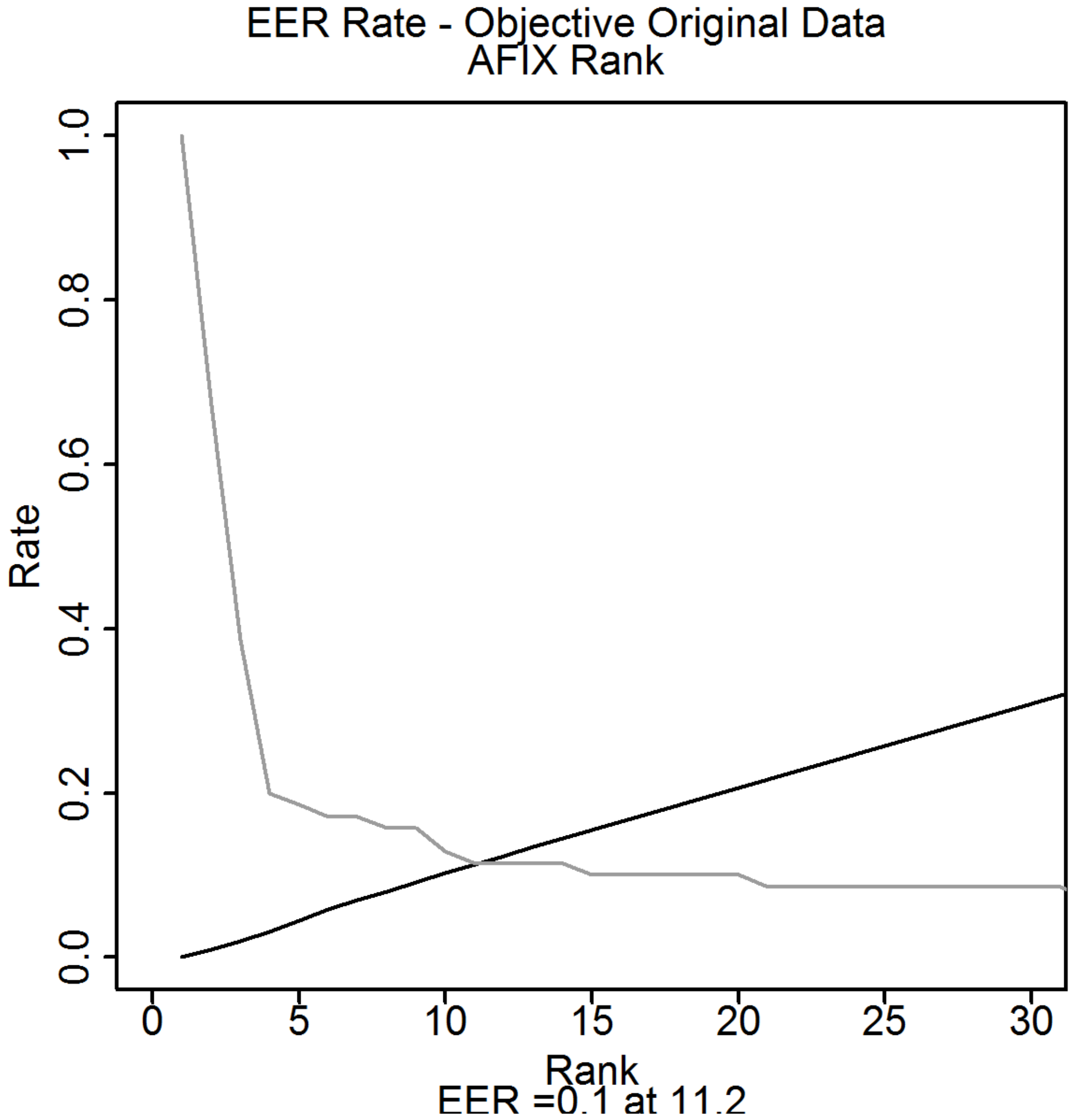


Figure D.146: This figure presents the EER plot from the objective original images for the ranks.

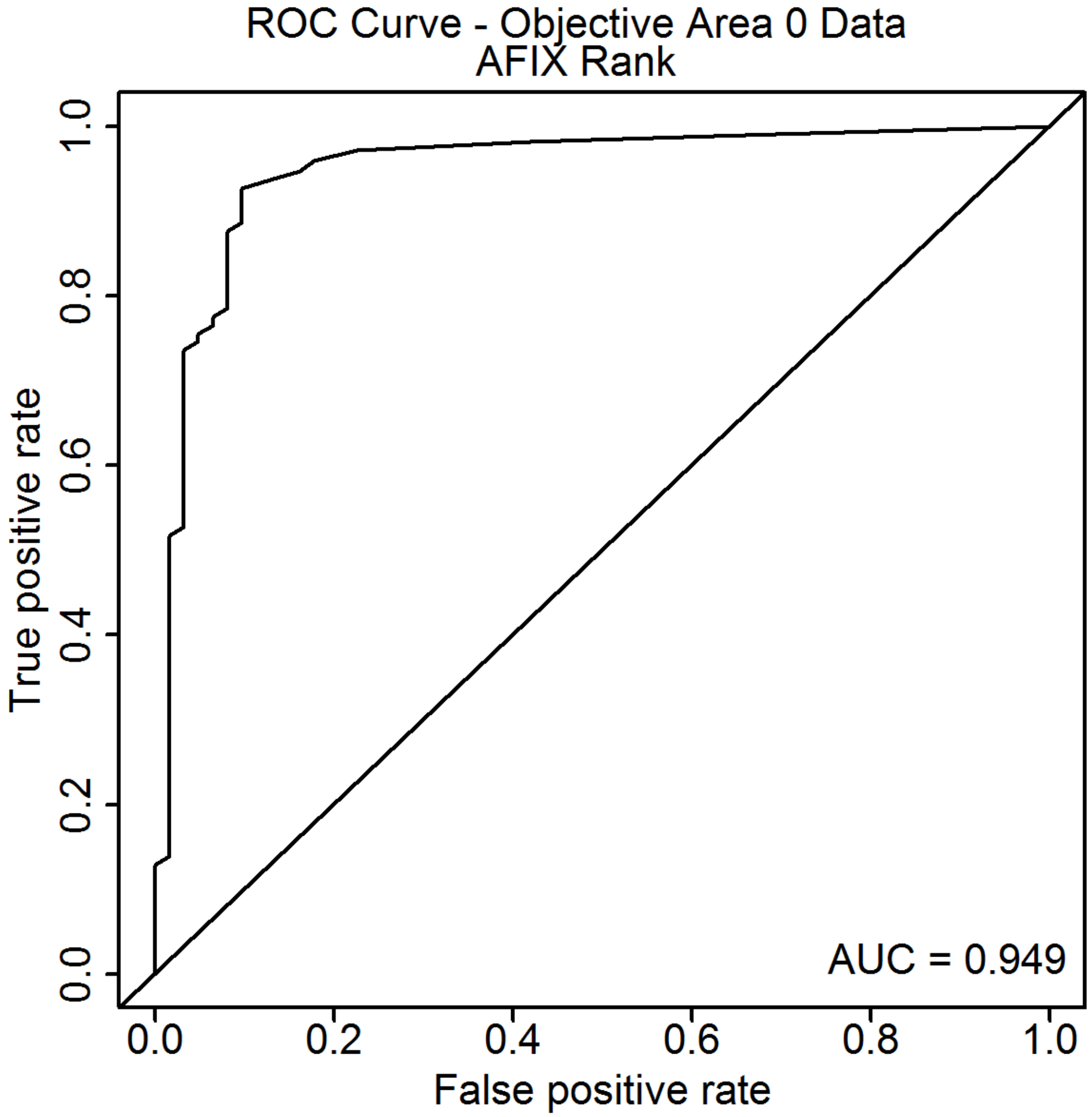


Figure D.147: This figure presents the ROC curve from the objective Area 0 images for the ranks.

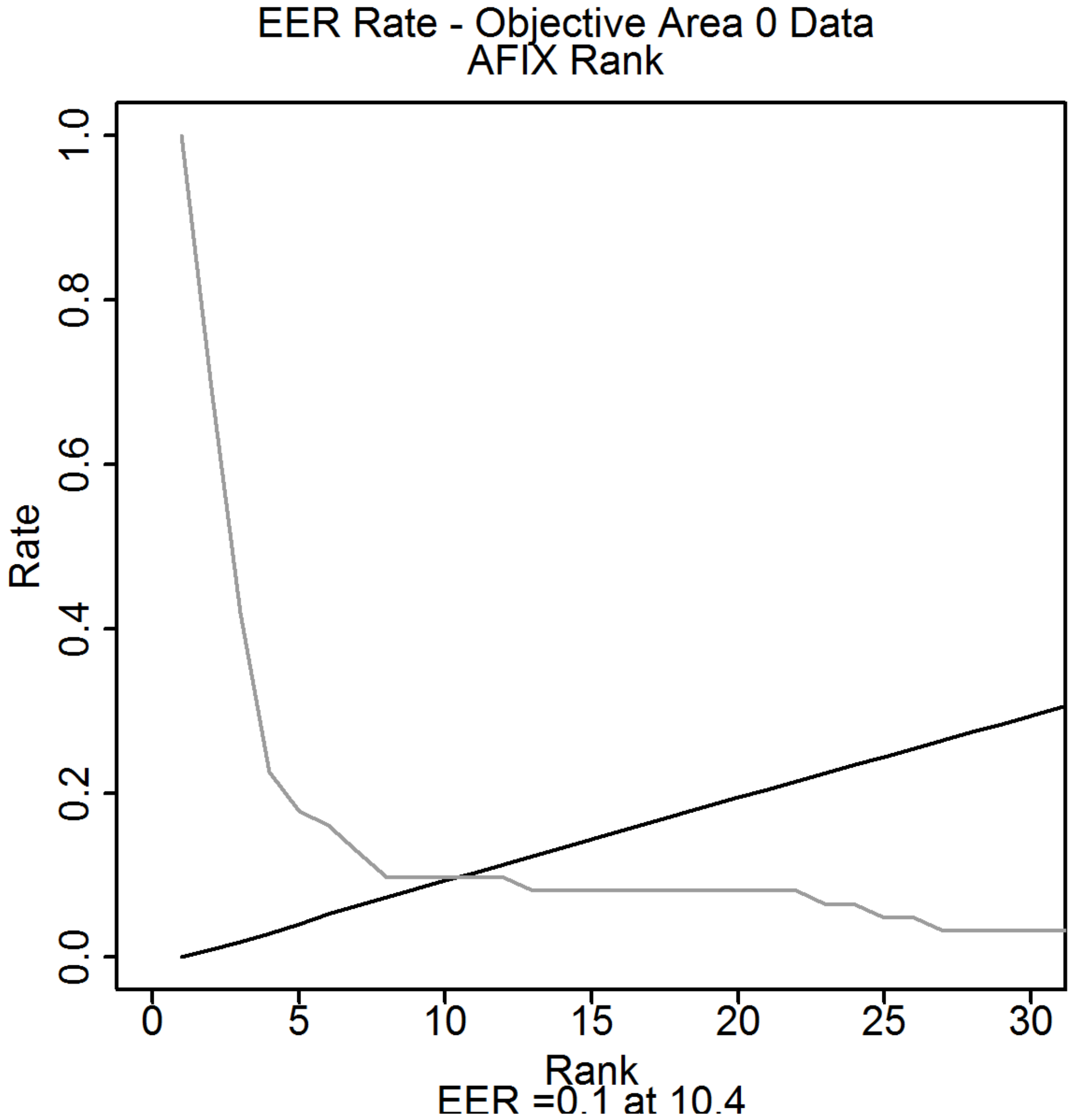


Figure D.148: This figure presents the EER plot from the subjective Area 0 images for the ranks.

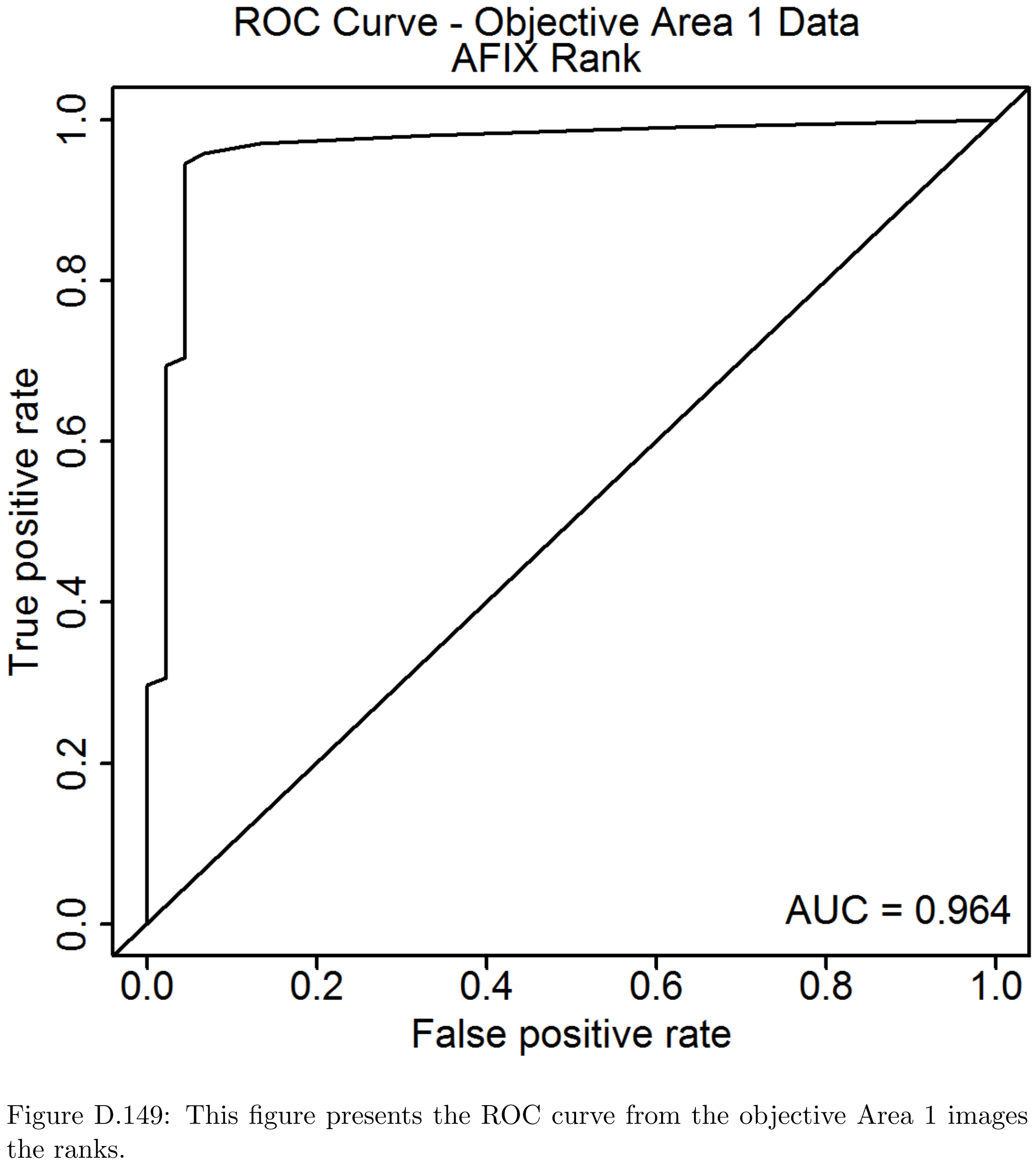


Figure D.149: This figure presents the ROC curve from the objective Area 1 images for the ranks.

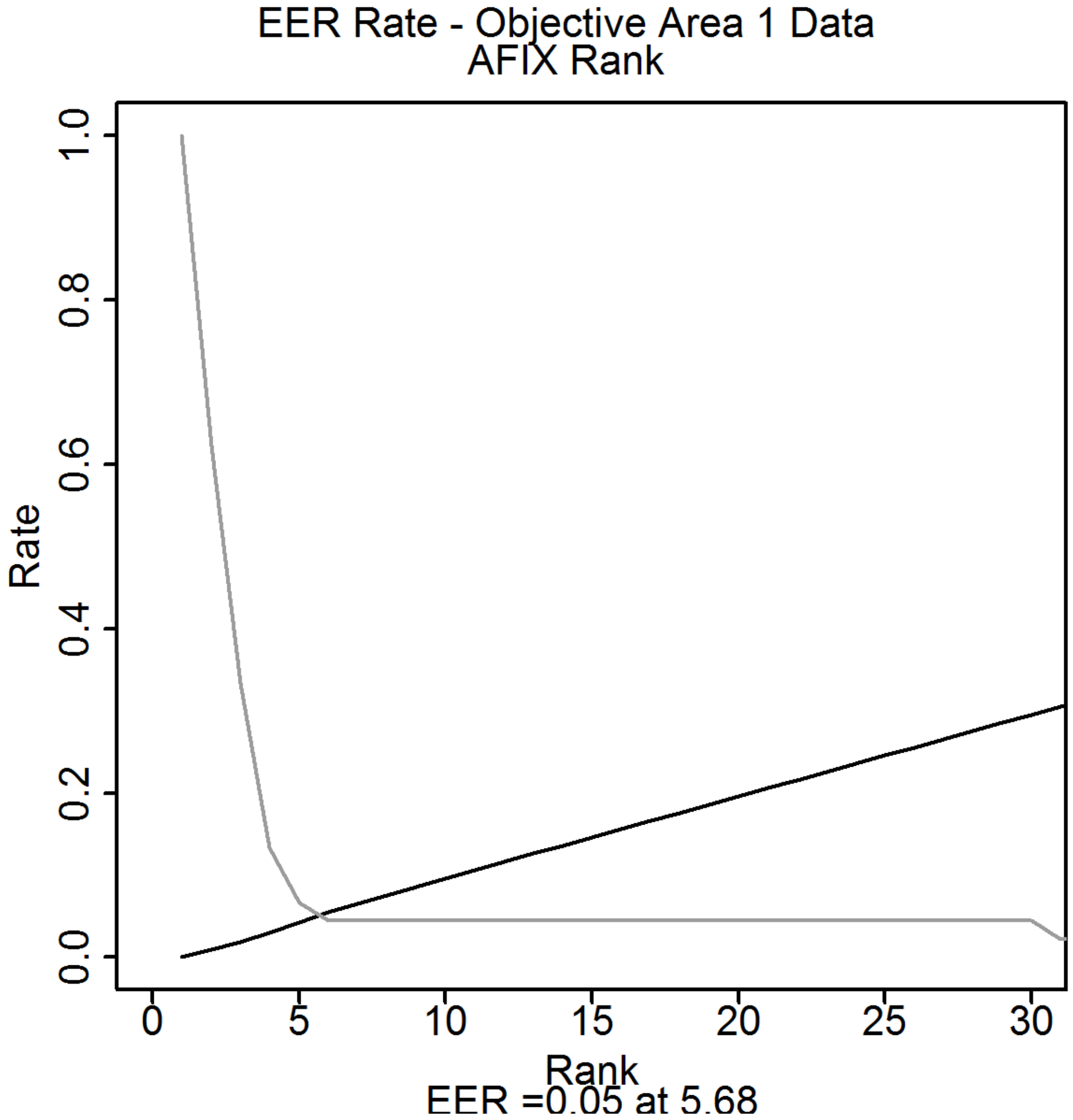


Figure D.150: This figure presents the EER plot from the objective Area 1 images for the ranks.

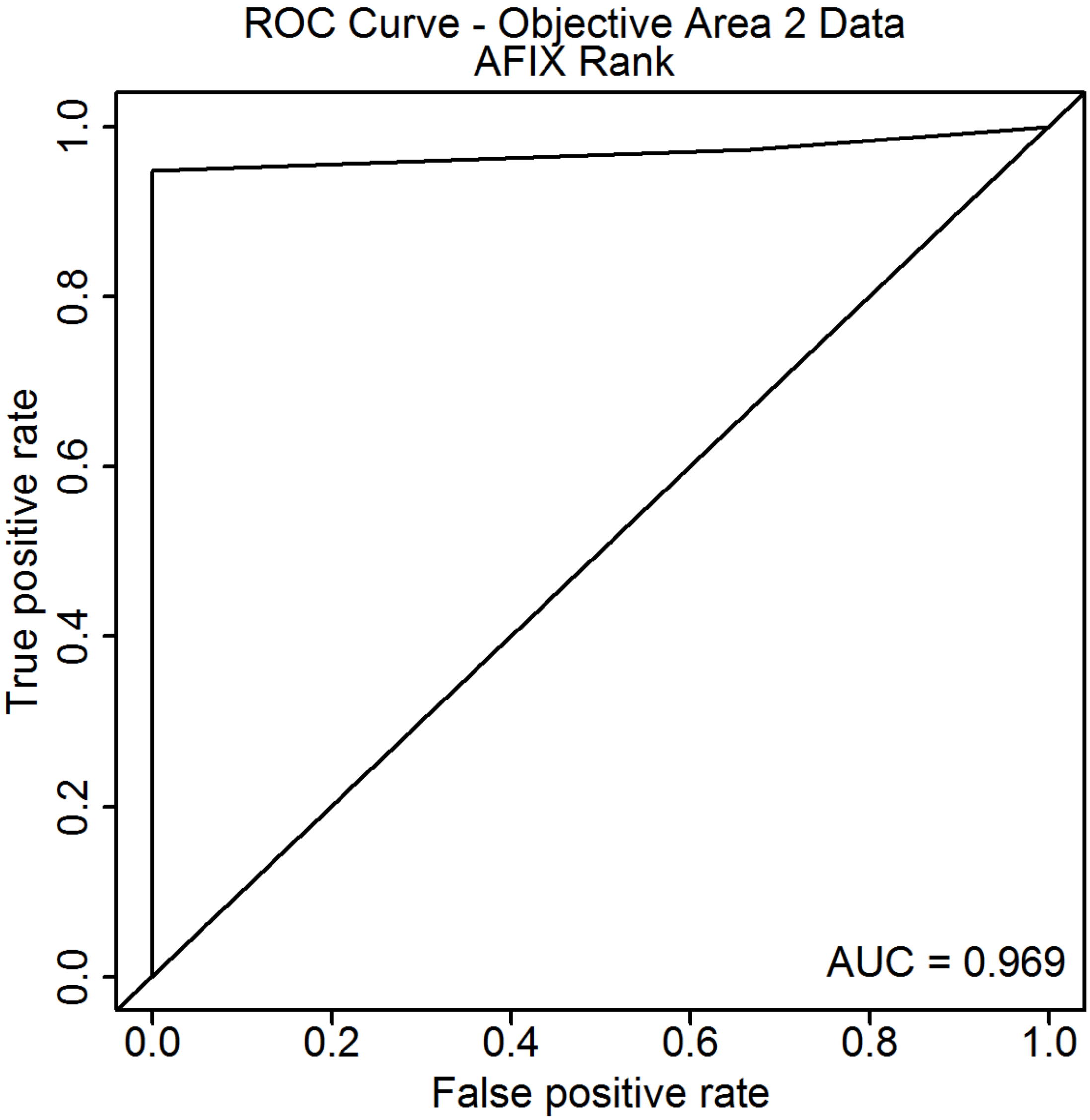


Figure D.151: This figure presents the ROC curve from the objective Area 2 images for the ranks.

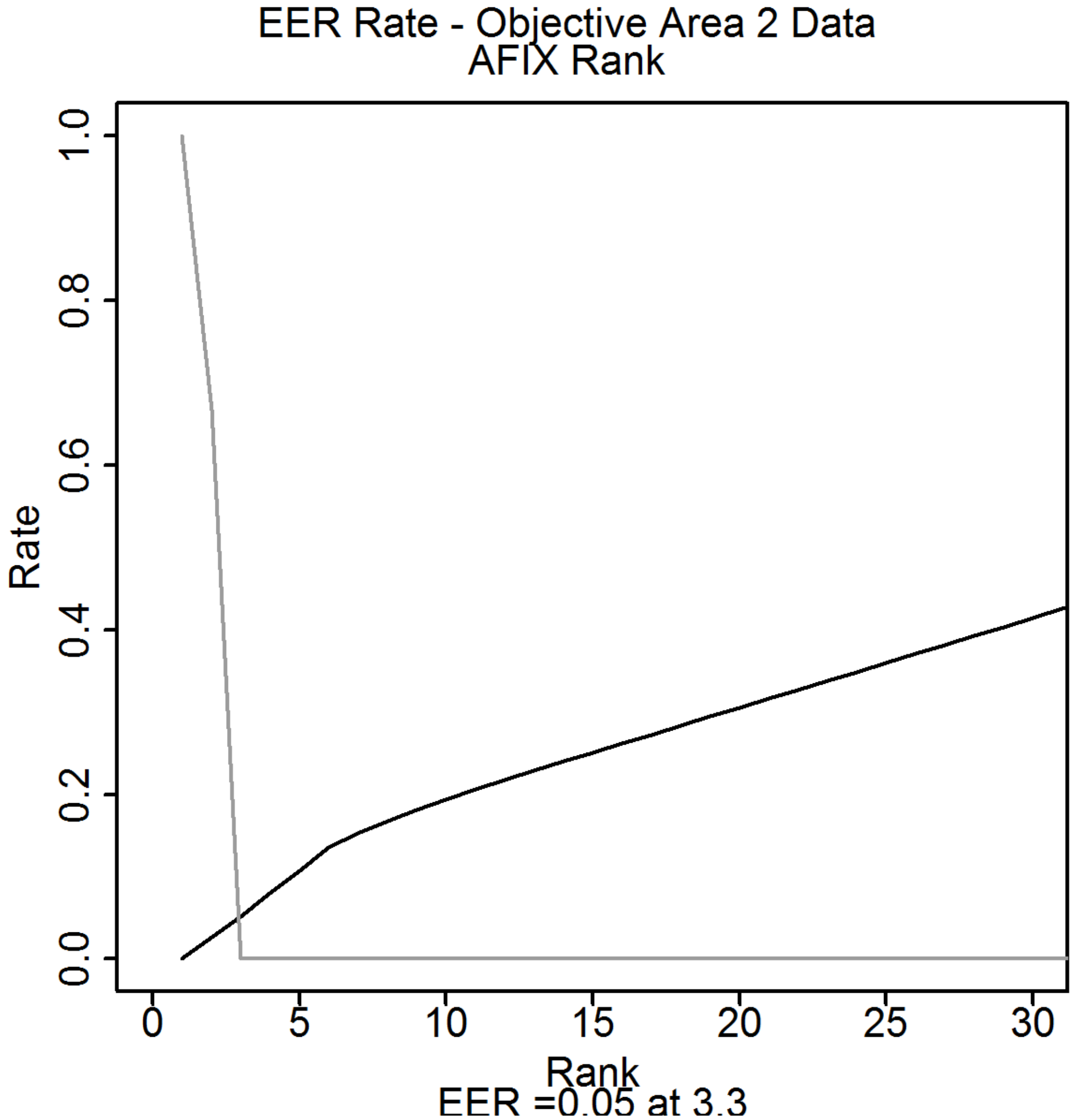


Figure D.152: This figure presents the EER plot from the objective Area 2 images for the ranks.

D.8.5 Objective Method: Automatic Extraction

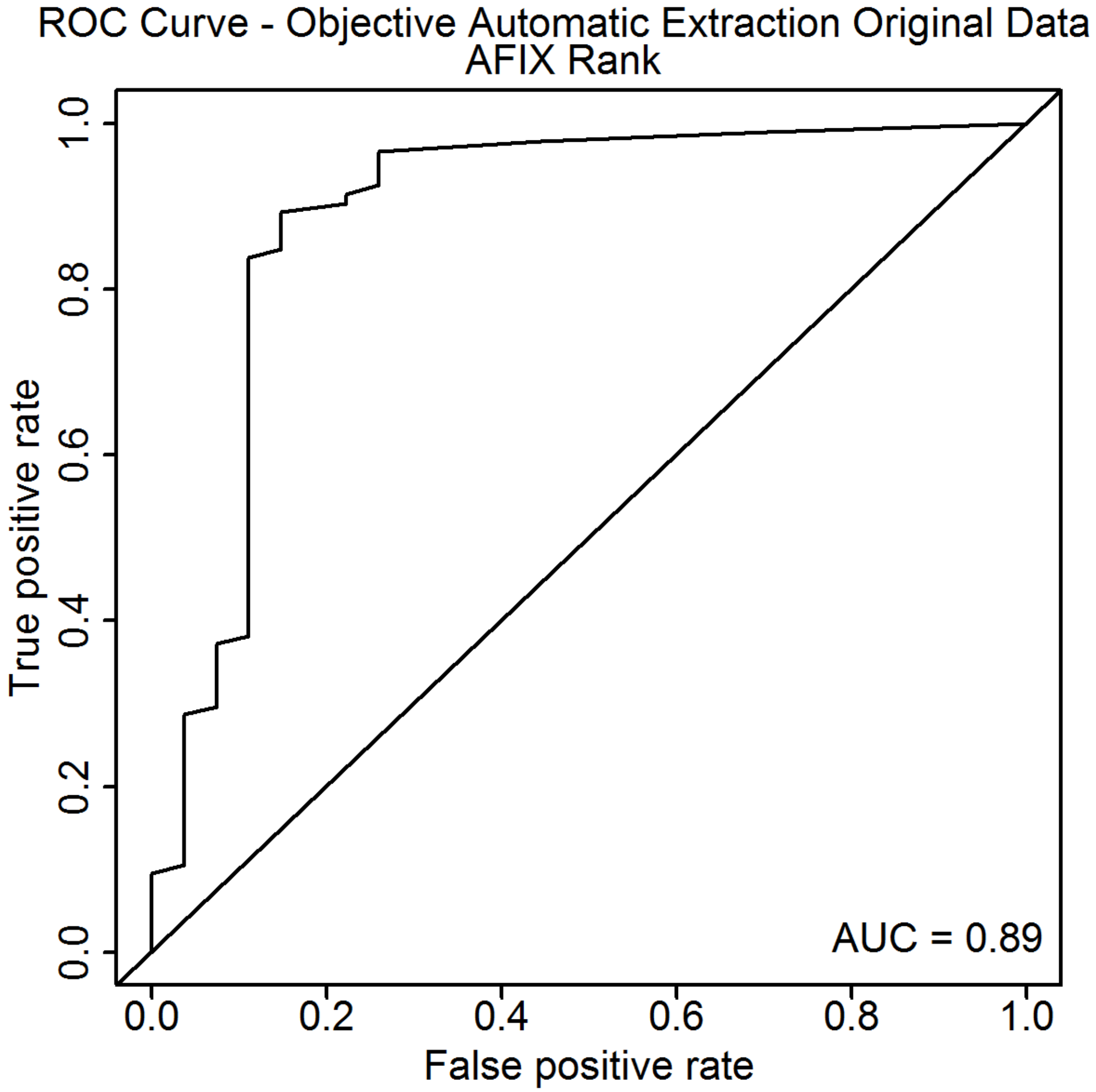


Figure D.153: This figure presents the ROC curve from the objective original images for the ranks in the automatic extraction.

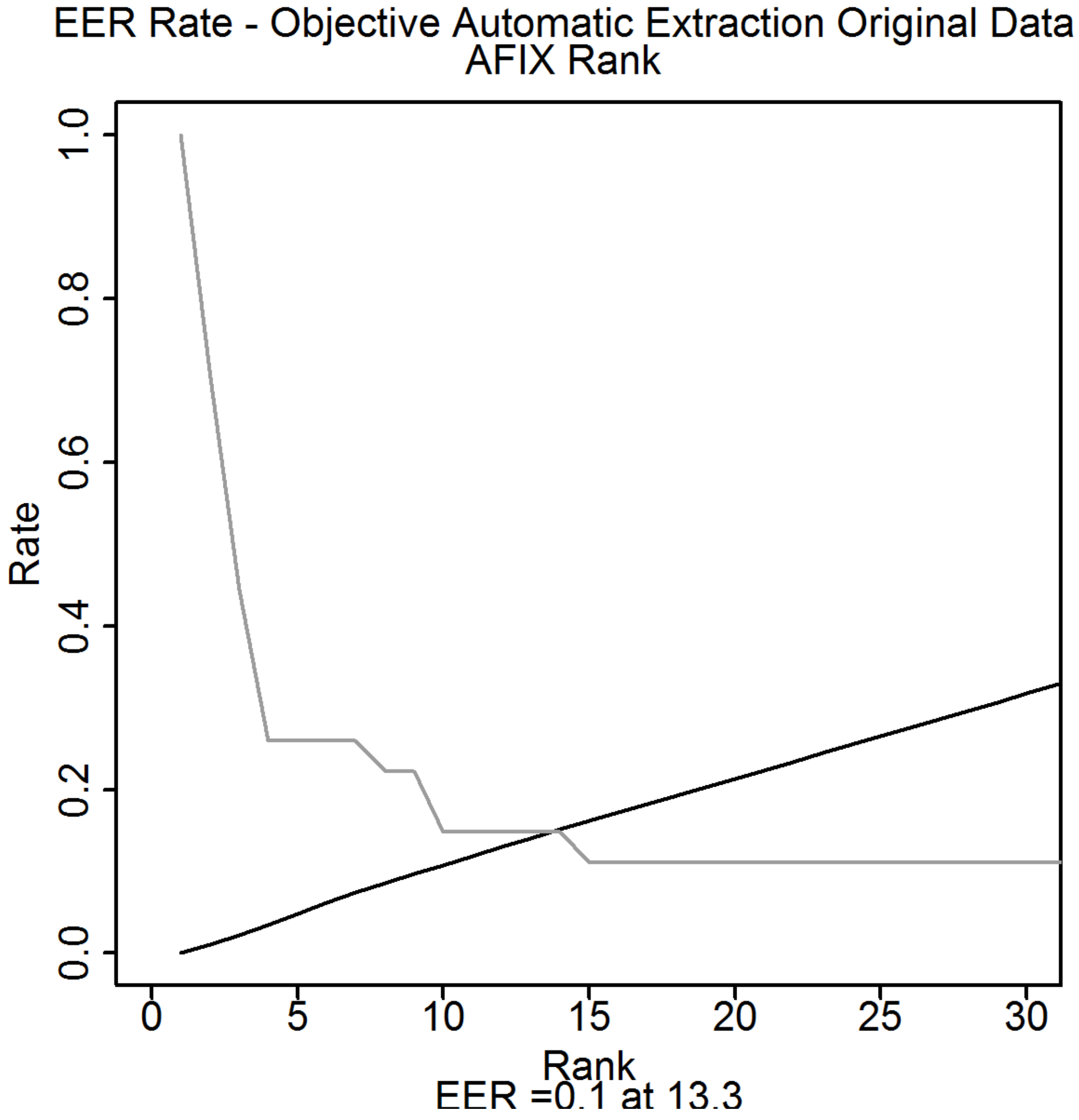


Figure D.154: This figure presents the EER plot from the objective original images for the ranks in the automatic extraction.

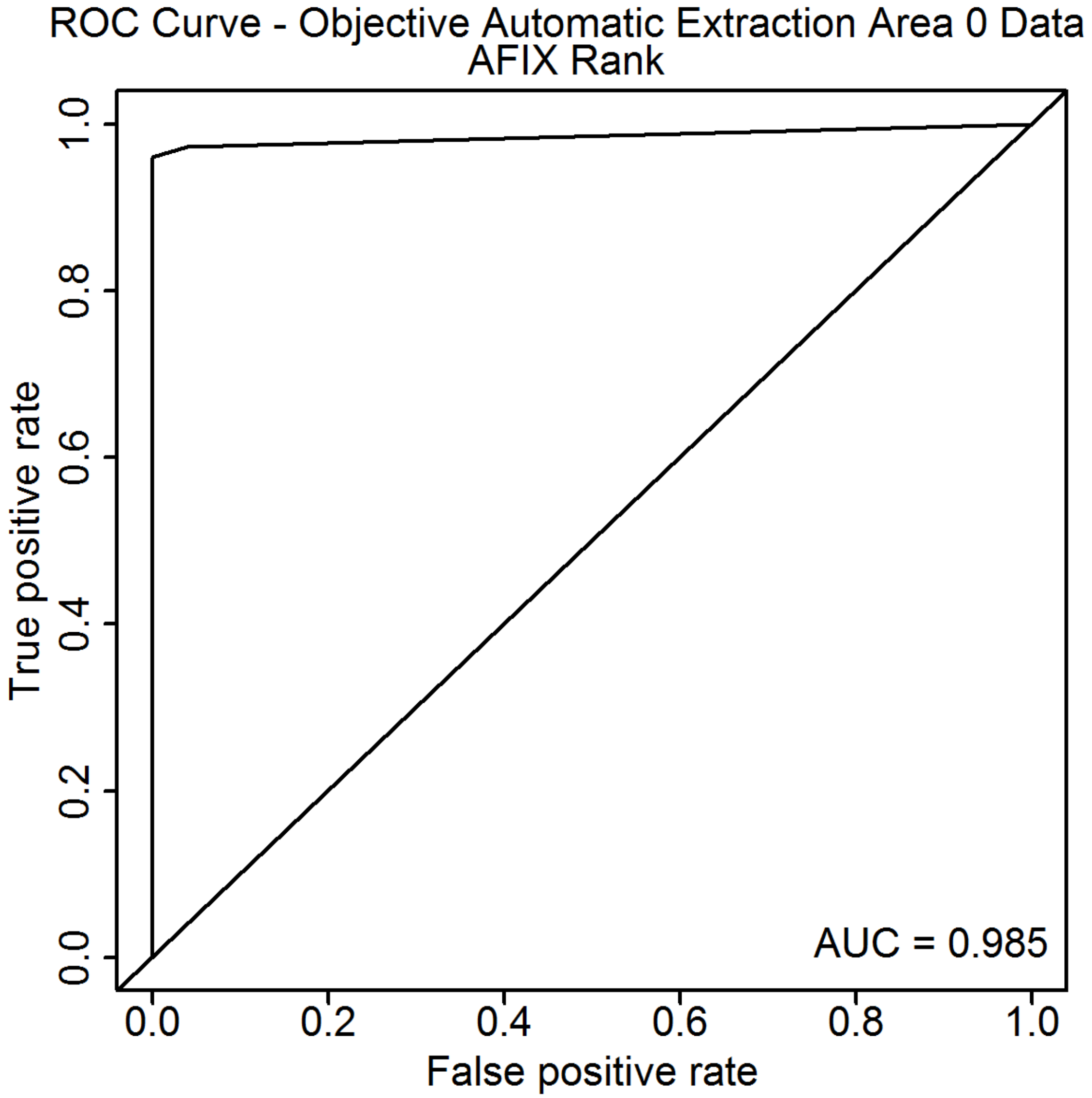


Figure D.155: This figure presents the ROC curve from the objective Area 0 images for the ranks in the automatic extraction.

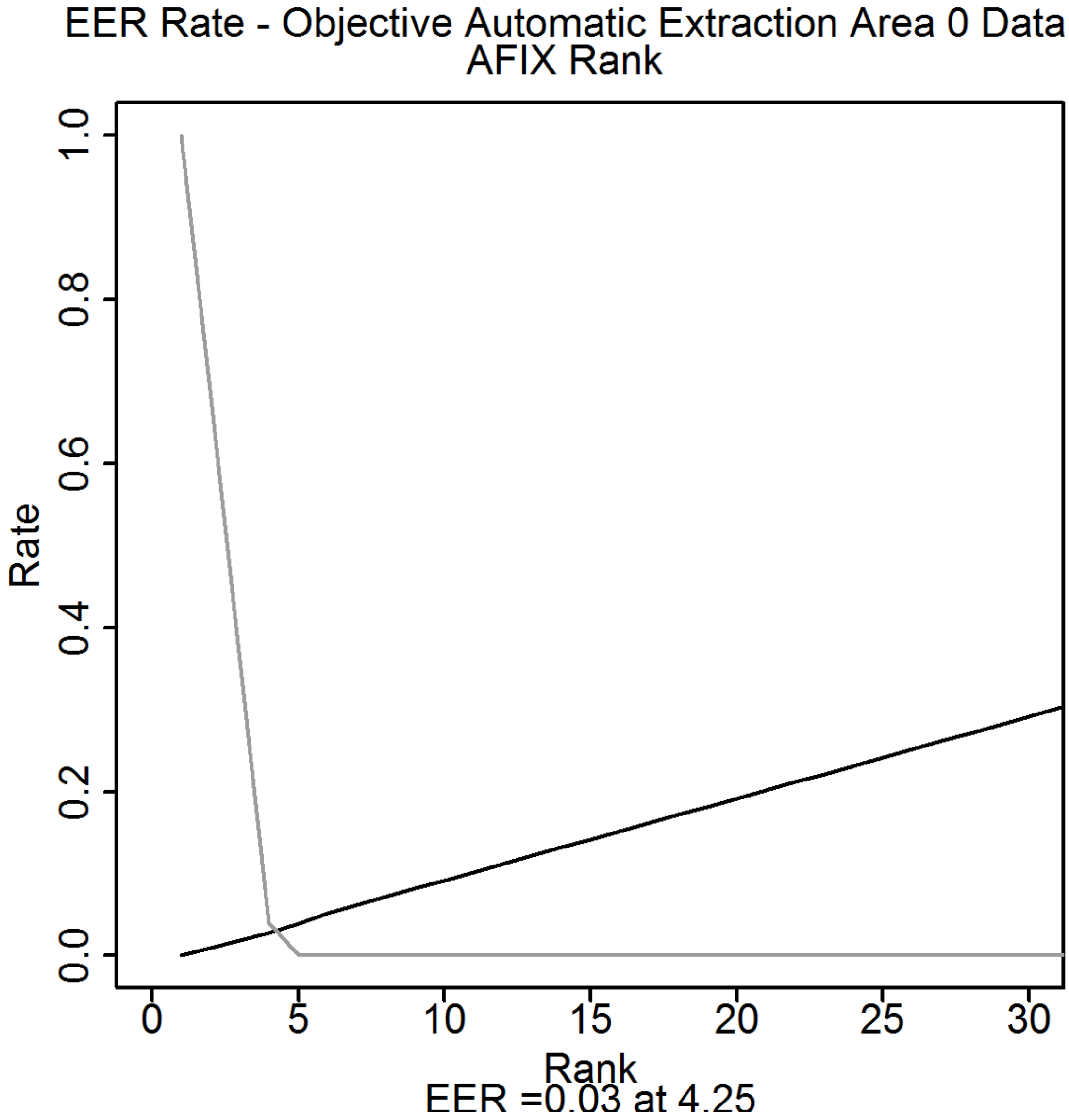


Figure D.156: This figure presents the EER plot from the objective Area 0 images for the ranks in the automatic extraction.

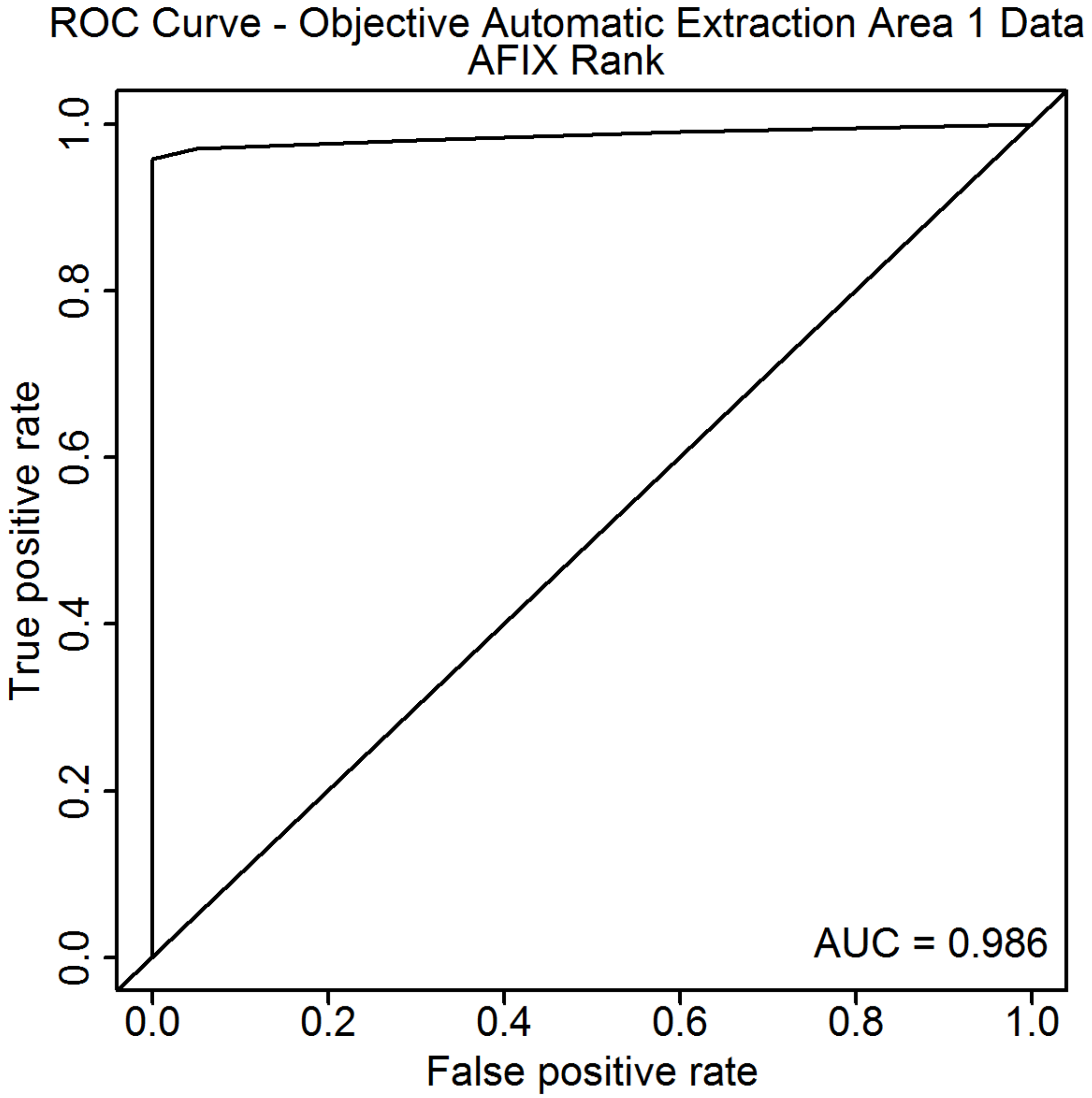


Figure D.157: This figure presents the ROC curve from the objective Area 1 images for the ranks in the automatic extraction.

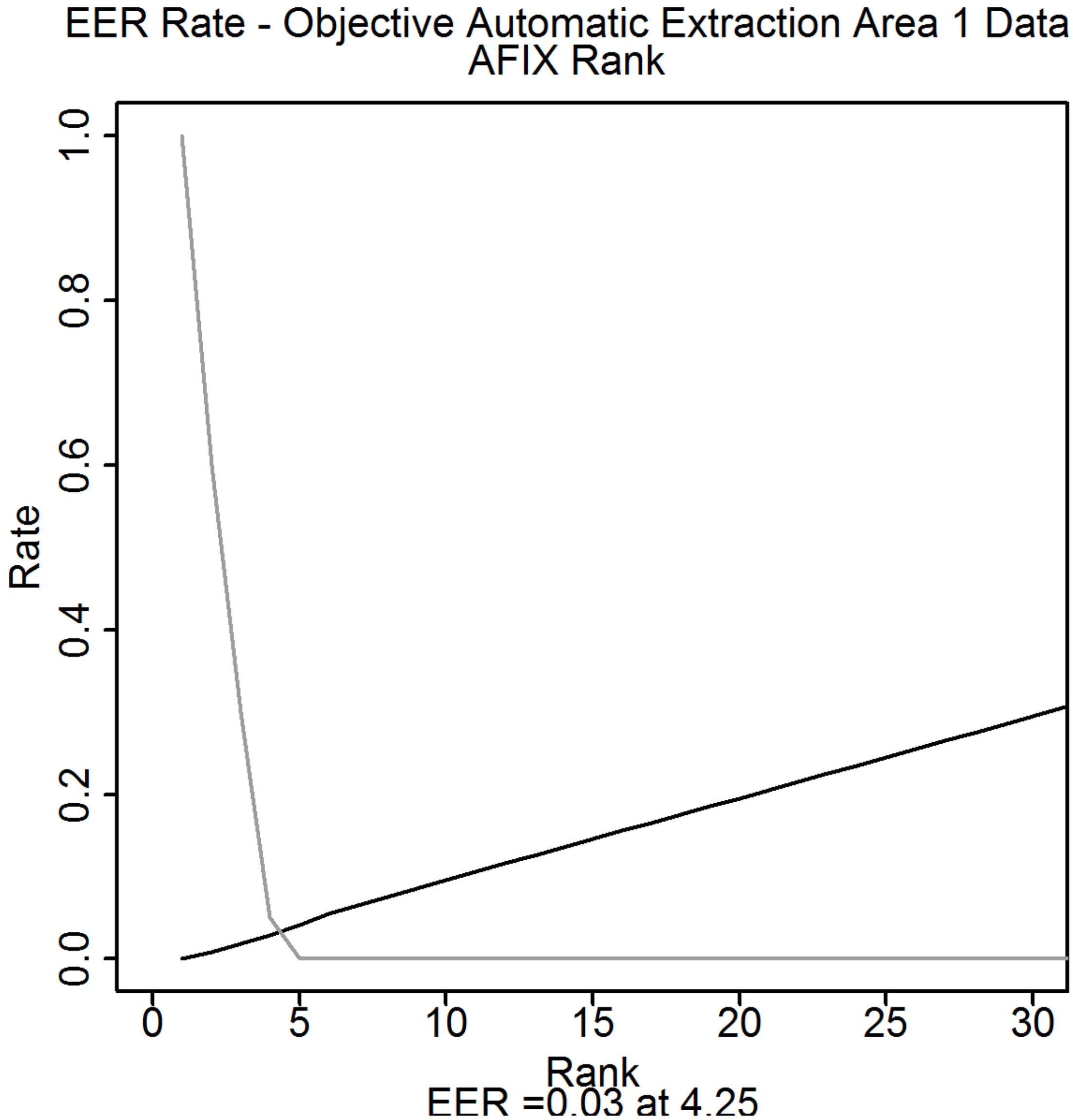


Figure D.158: This figure presents the EER plot from the objective Area 1 images for the ranks in the automatic extraction.

D.8.6 Objective Method: Manual Extraction

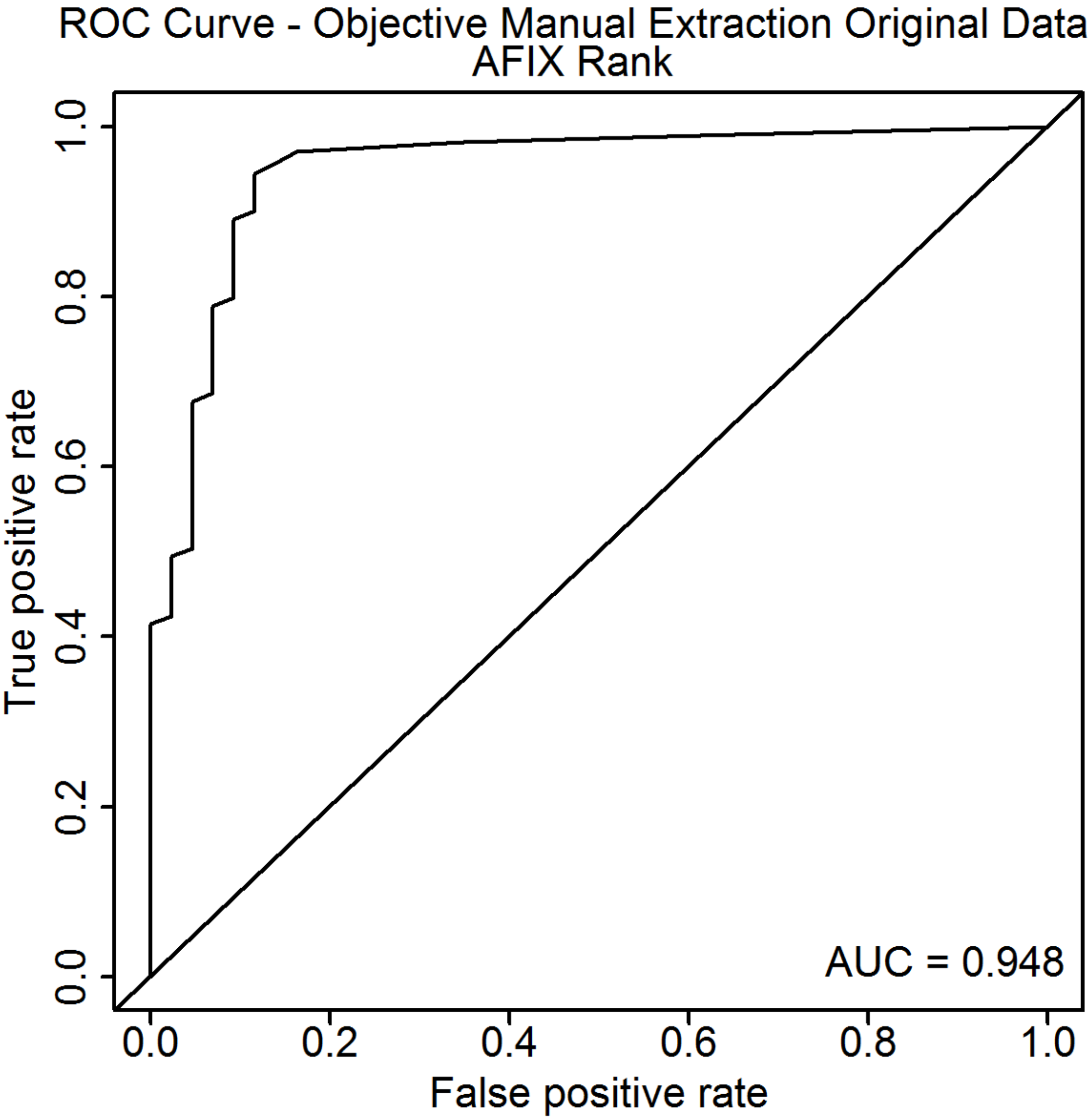


Figure D.159: This figure presents the ROC curve from the objective original images for the ranks in the manual extraction.

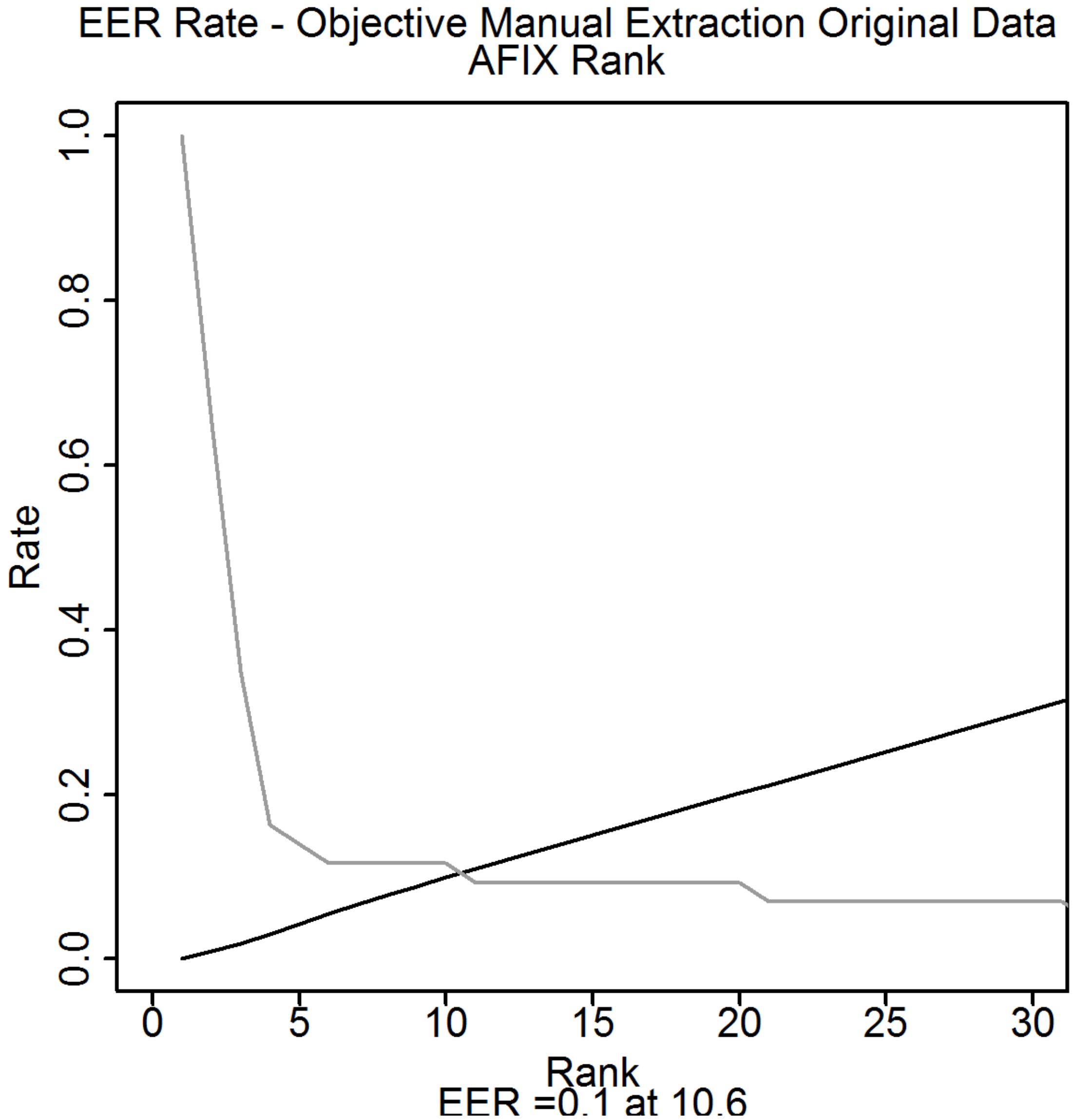


Figure D.160: This figure presents the EER plot from the objective original images for the ranks in the manual extraction.

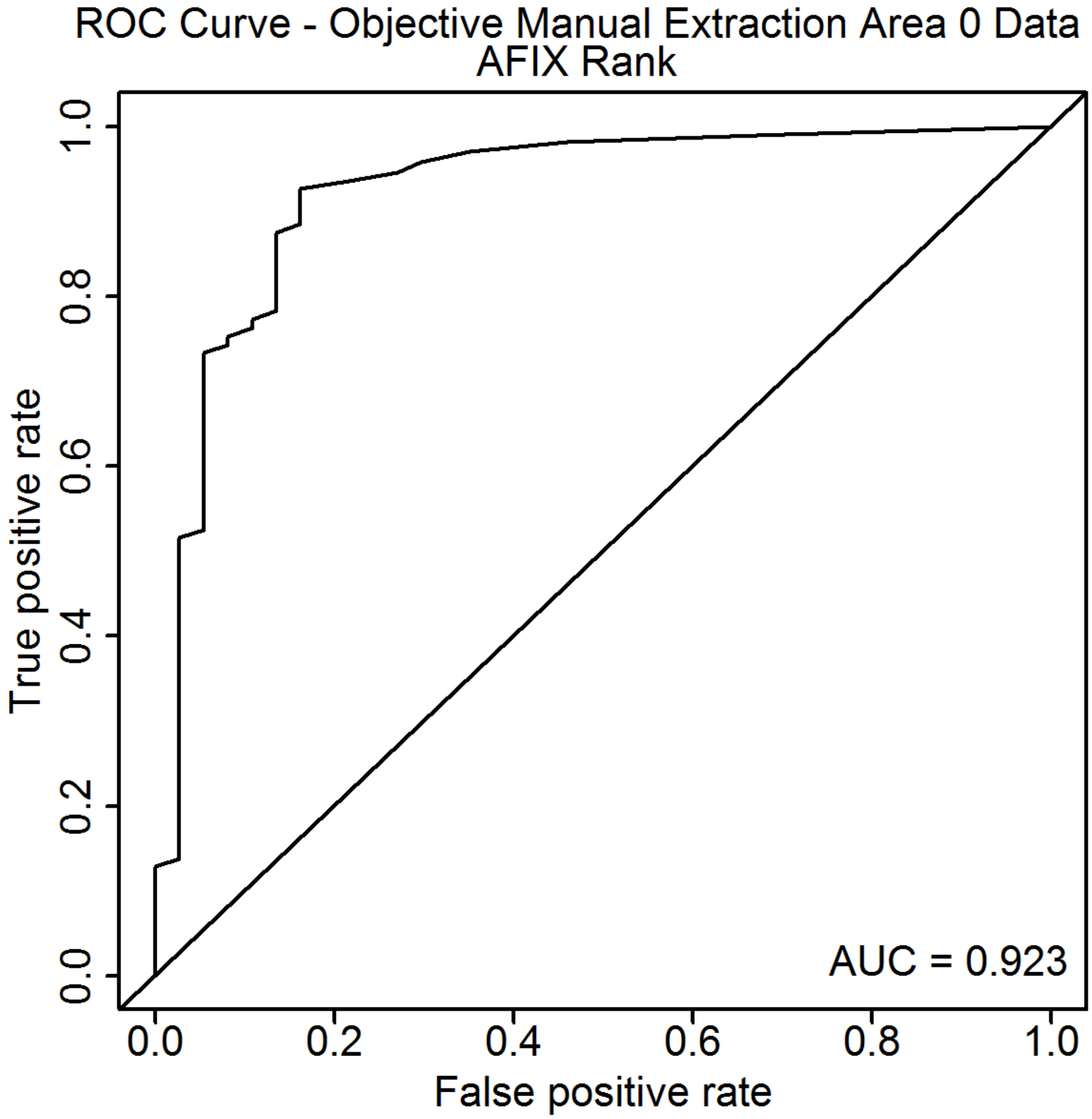


Figure D.161: This figure presents the ROC curve from the objective Area 0 images for the ranks in the manual extraction.

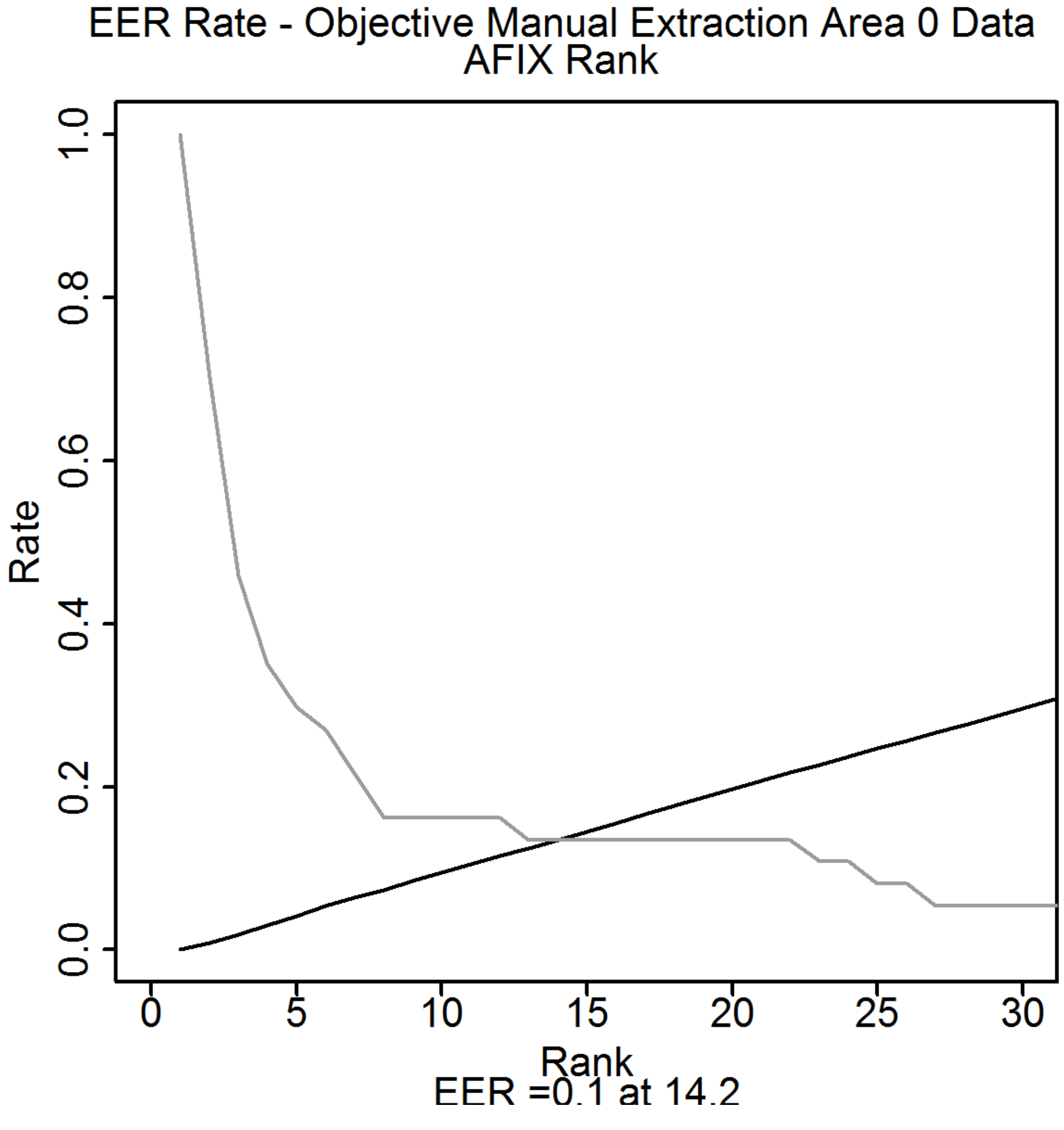


Figure D.162: This figure presents the EER plot from the objective Area 0 images for the ranks in the manual extraction.

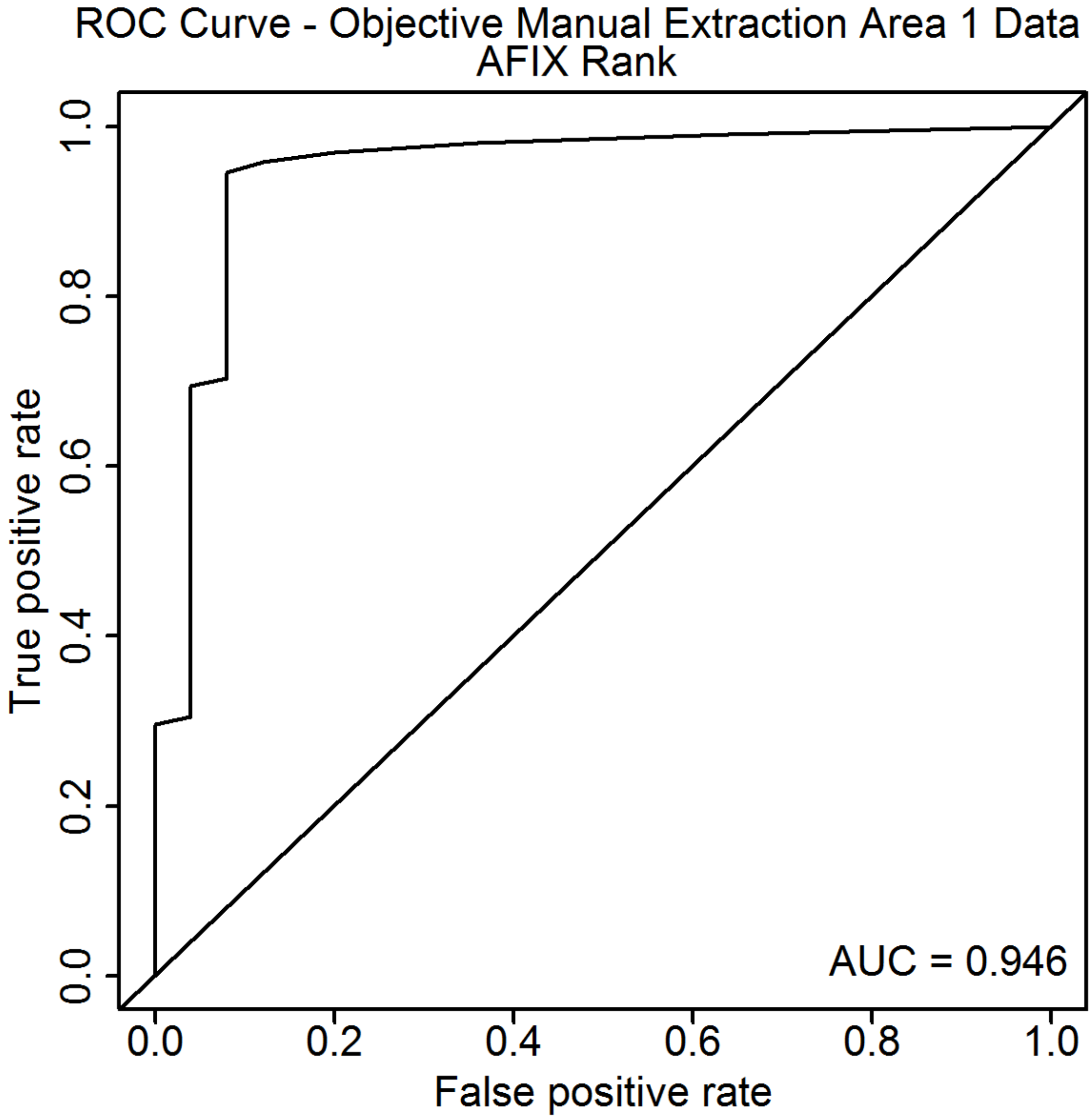


Figure D.163: This figure presents the ROC curve from the objective Area 1 images for the ranks in the manual extraction.

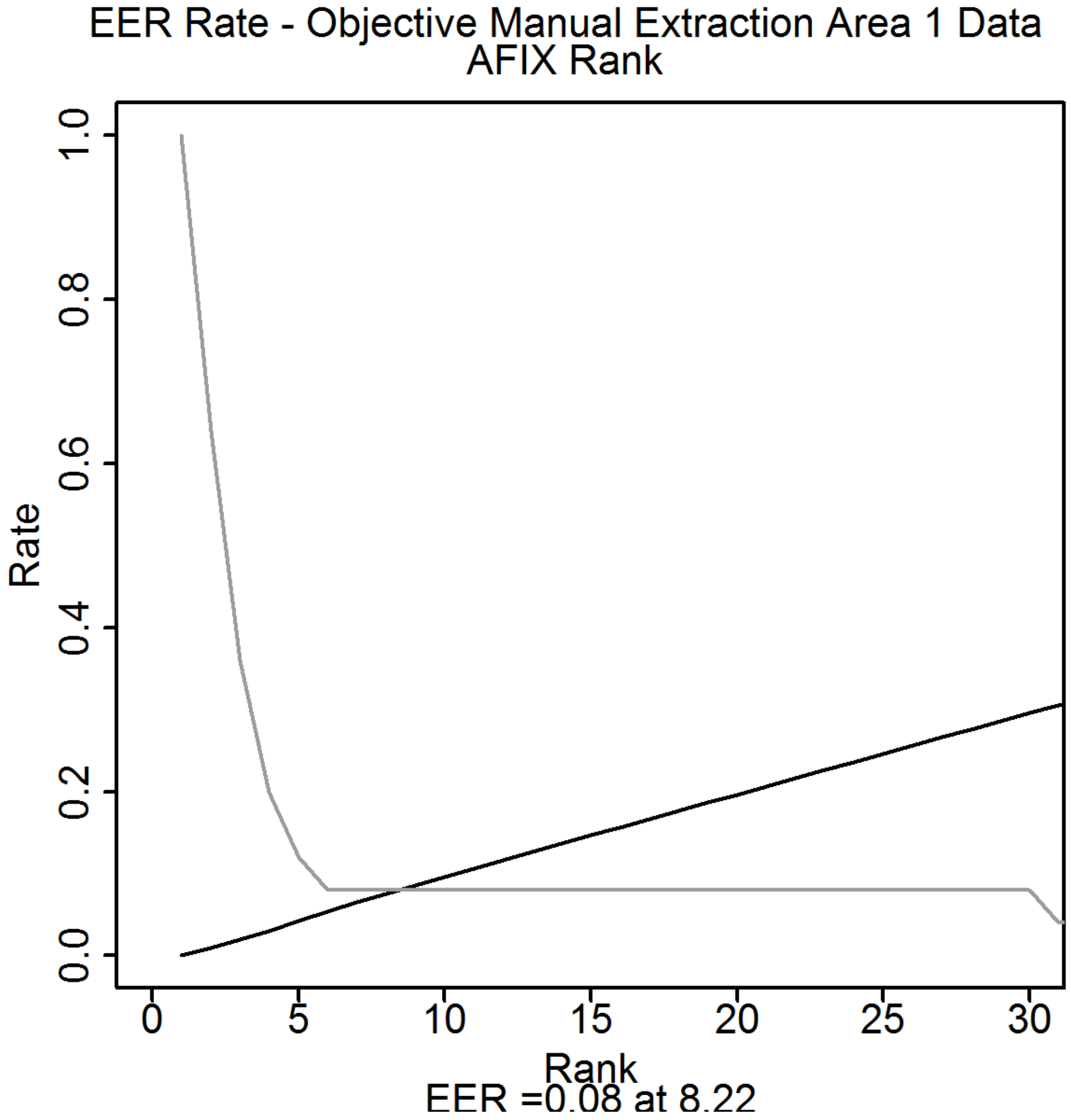


Figure D.164: This figure presents the EER plot from the objective Area 1 images for the ranks in the manual extraction.

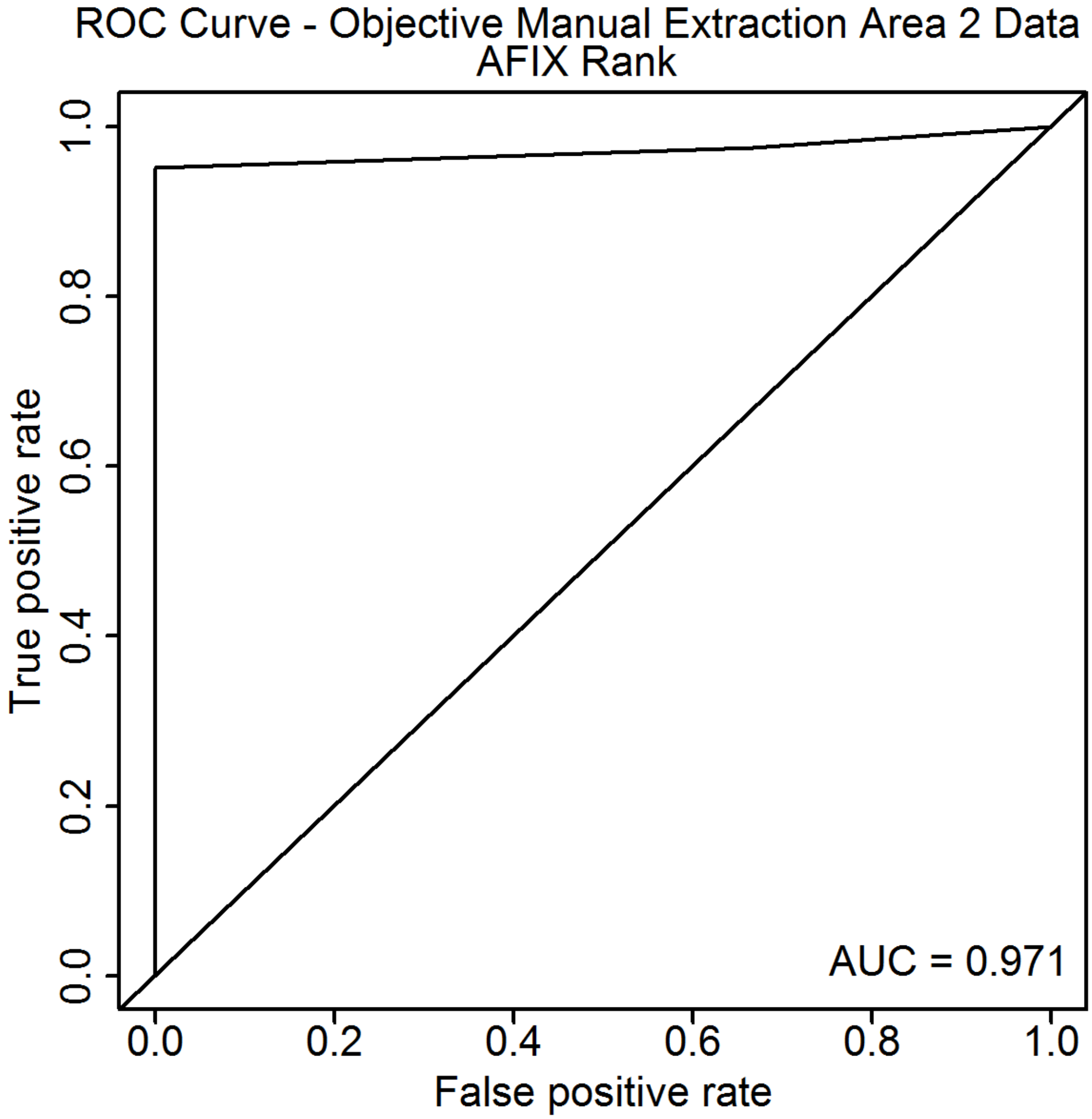


Figure D.165: This figure presents the ROC curve from the objective Area 2 images for the ranks in the manual extraction.

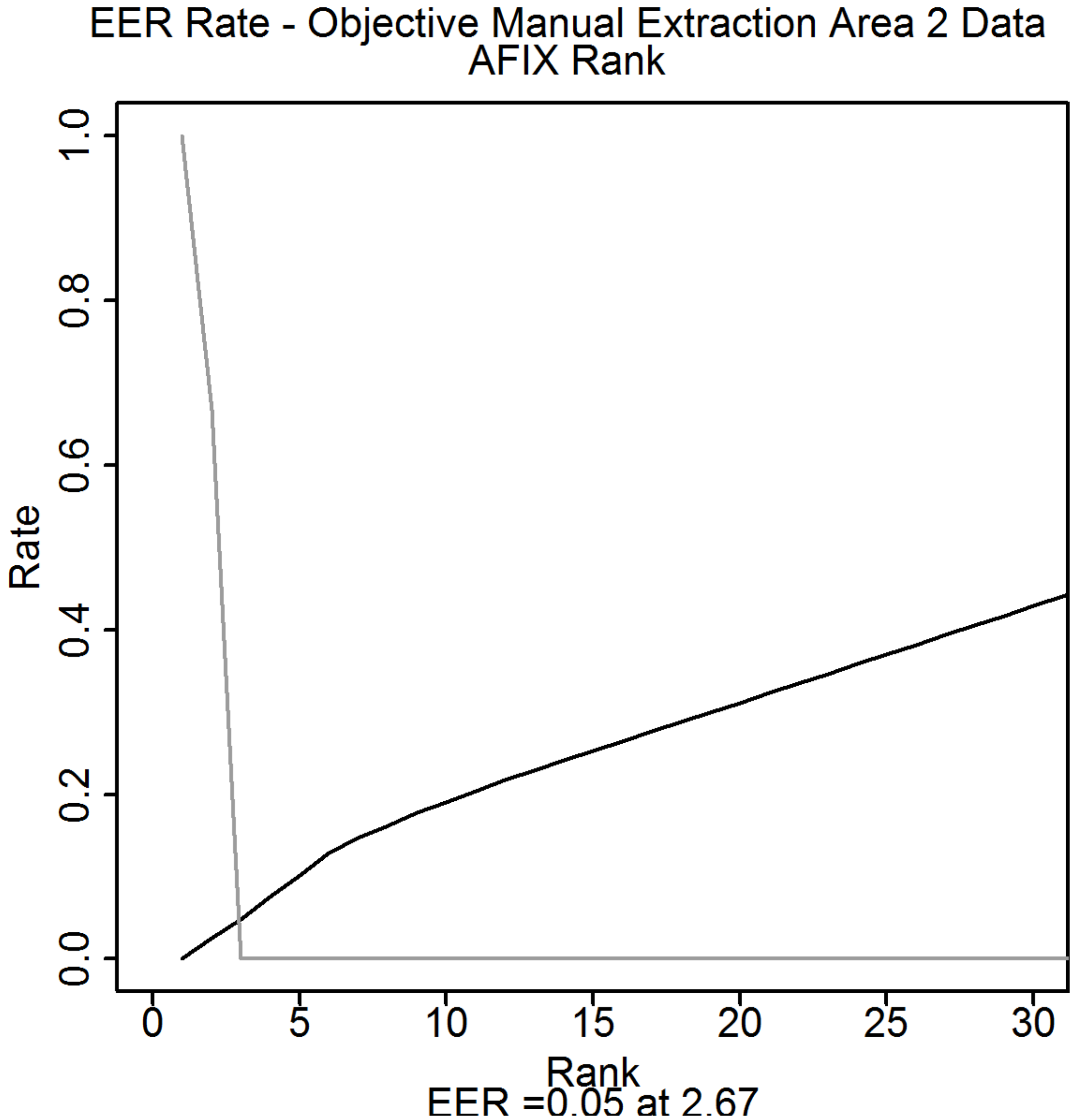


Figure D.166: This figure presents the EER plot from the objective Area 2 images for the ranks in the manual extraction.

E. BOZORTH3 Plots

E.1 Rank versus Match Score Scatter Plots

E.1.1 Subjective Method

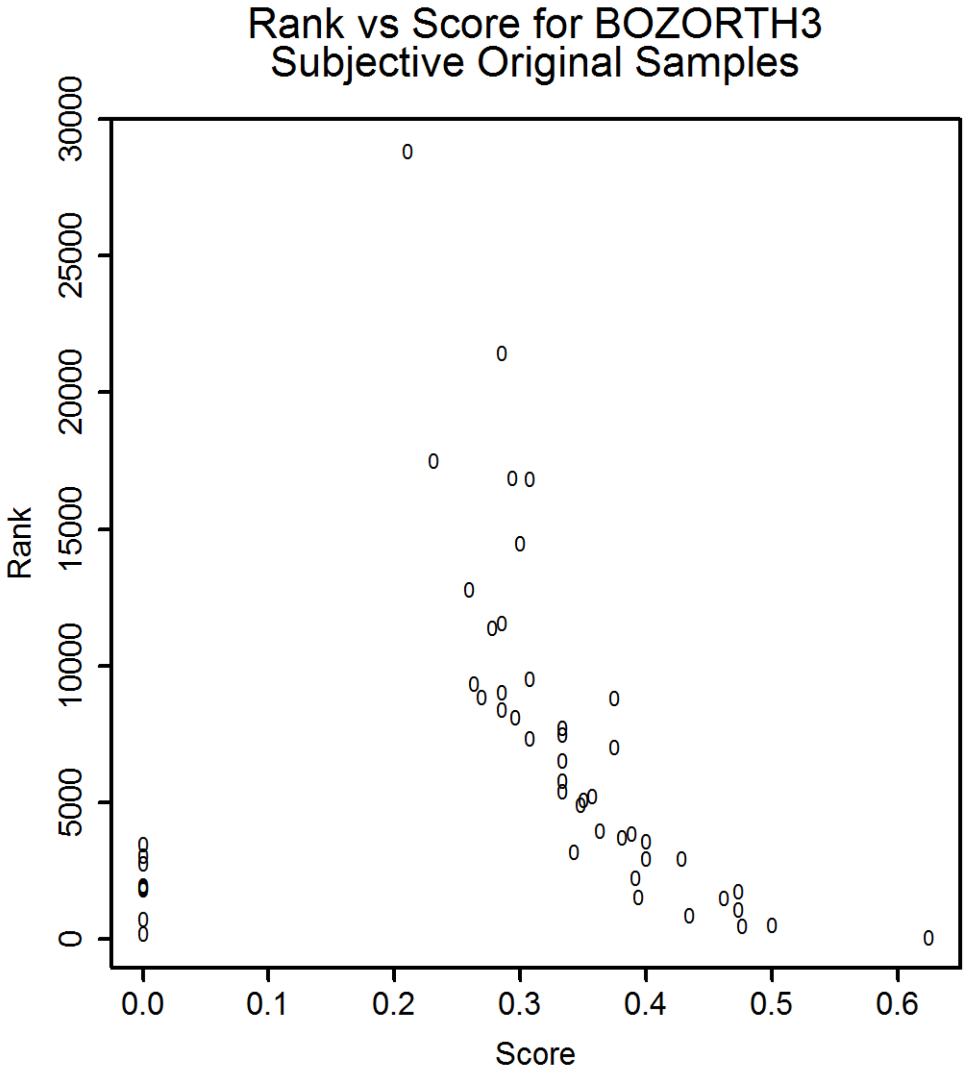


Figure E.1: The scatter plot for the match score versus the rank for the subjective original images.

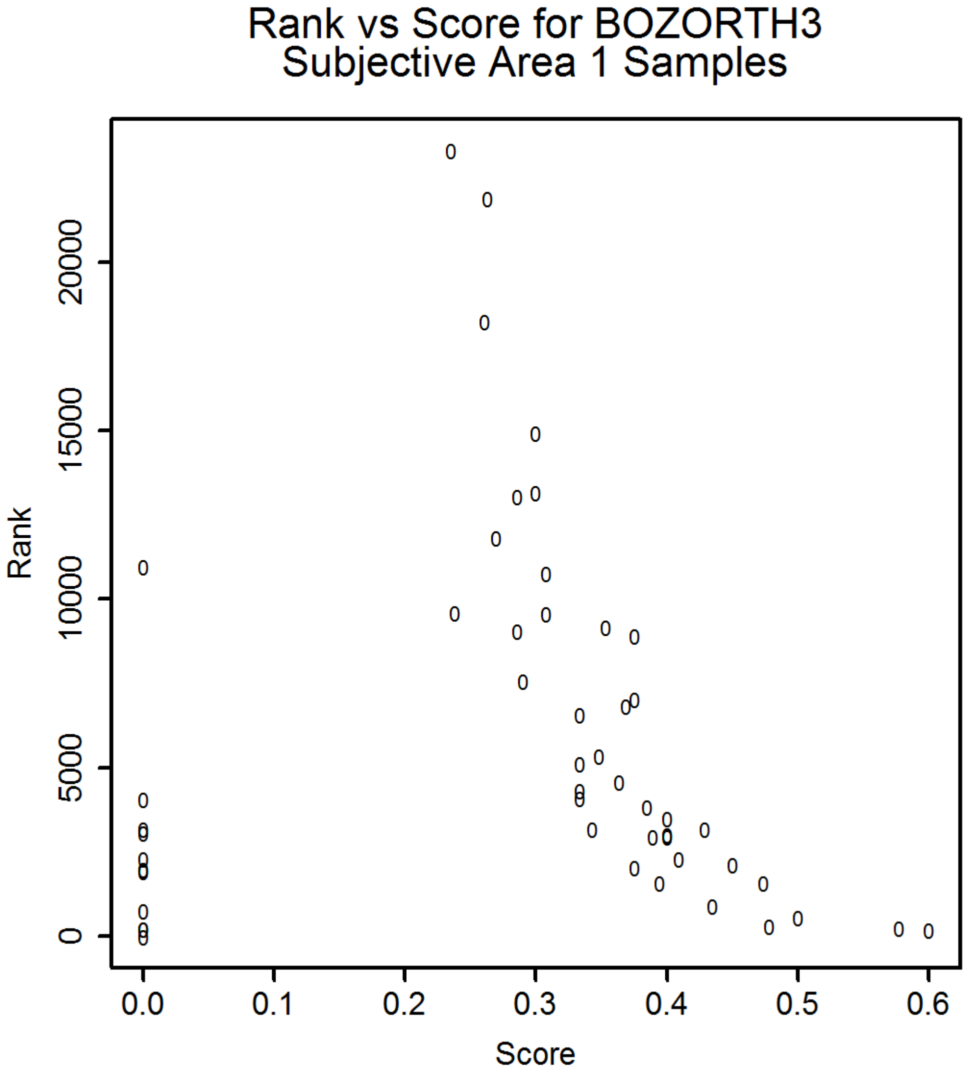


Figure E.2: The scatter plot for the match score versus the rank for the subjective Area 1 images.

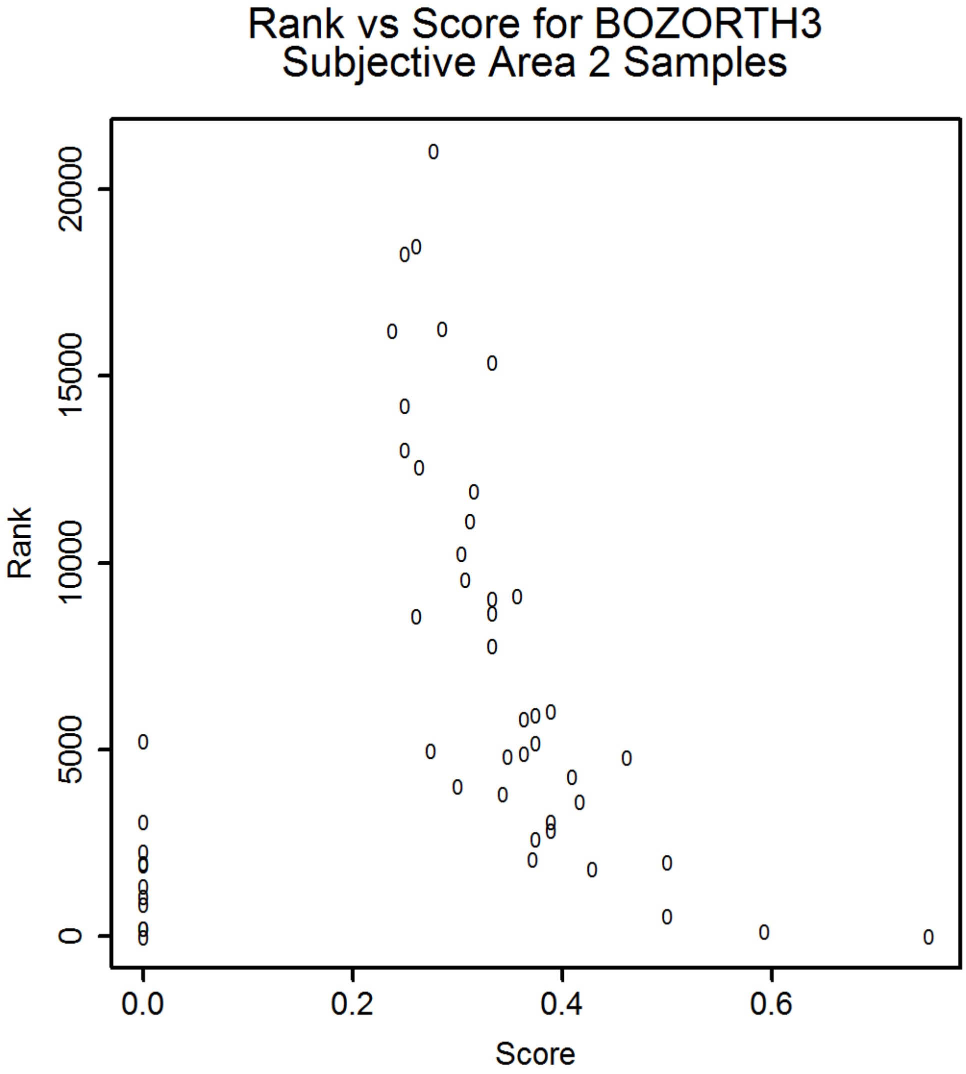


Figure E.3: The scatter plot for the match score versus the rank for the subjective Area 2 images.

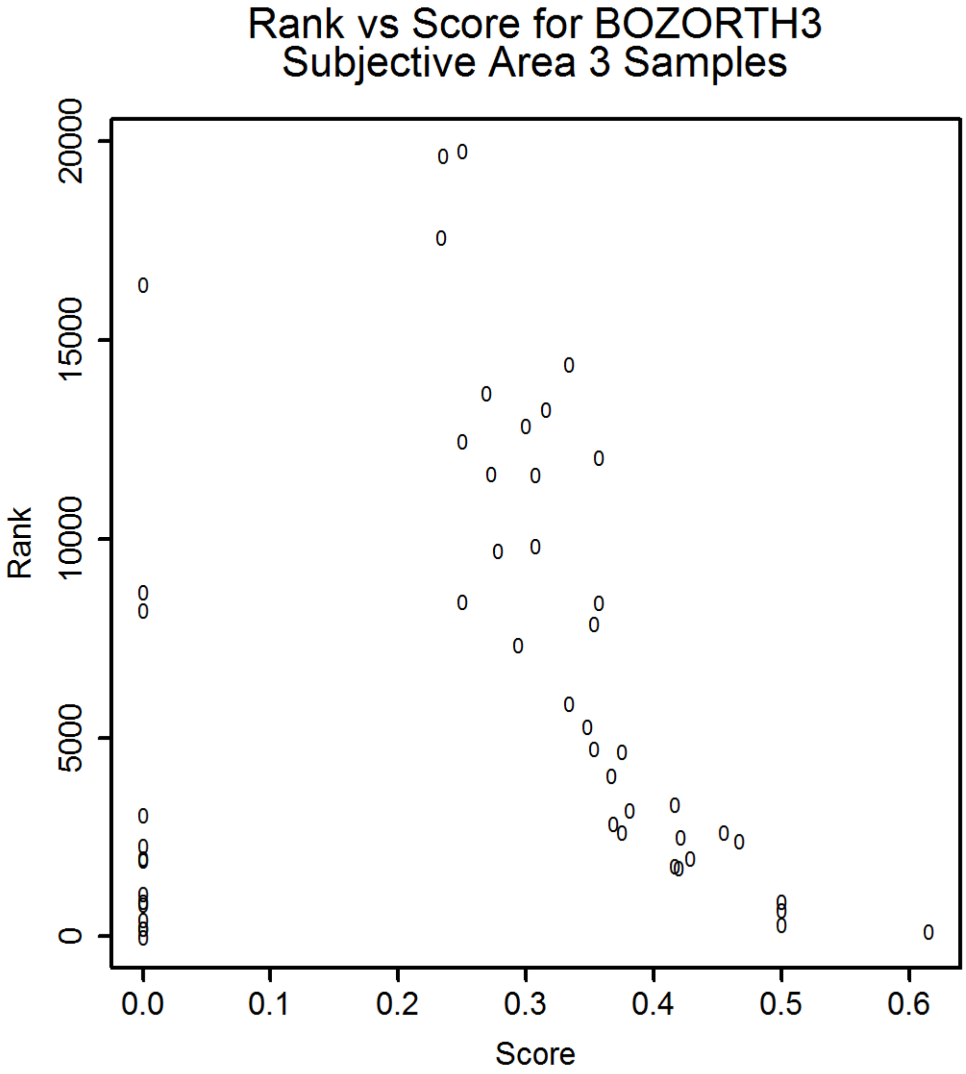


Figure E.4: The scatter plot for the match score versus the rank for the subjective Area 3 images.

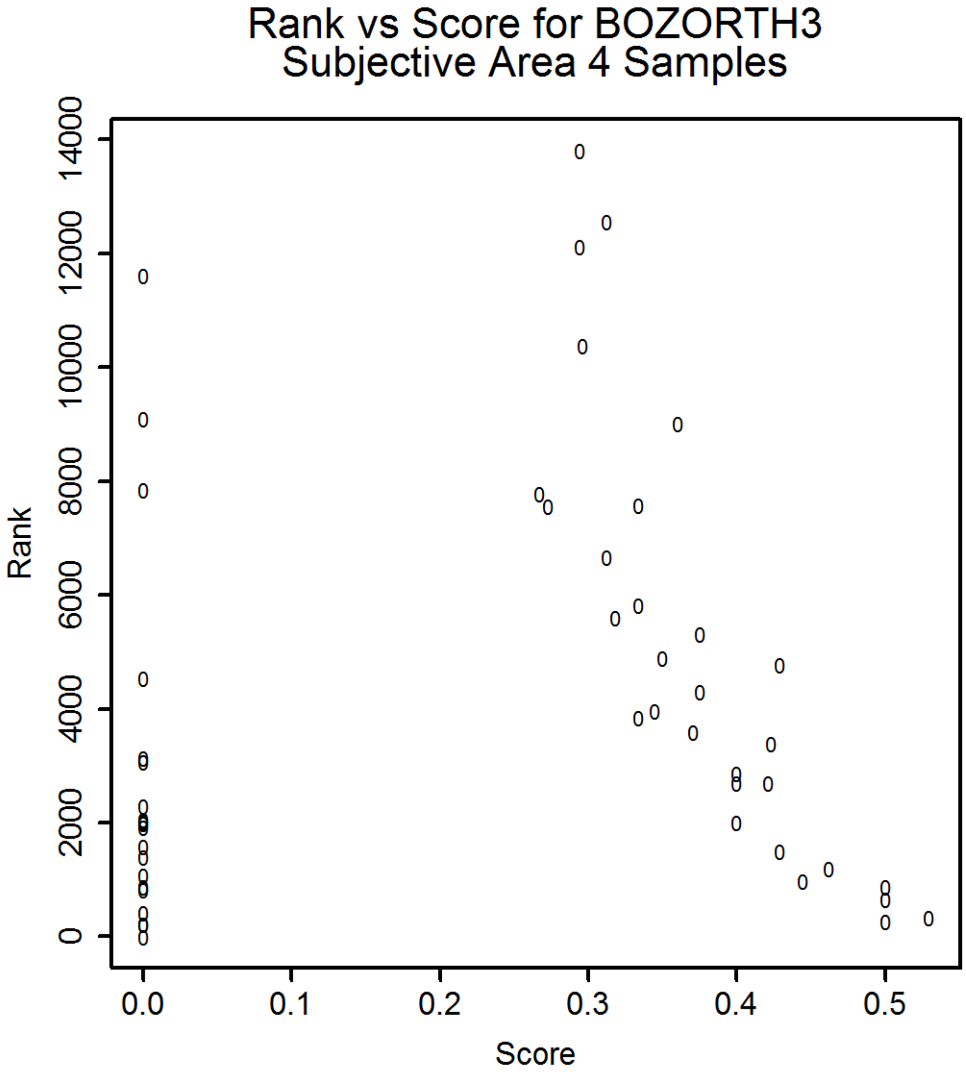


Figure E.5: The scatter plot for the match score versus the rank for the subjective Area 4 images.

E.1.2 Objective Method

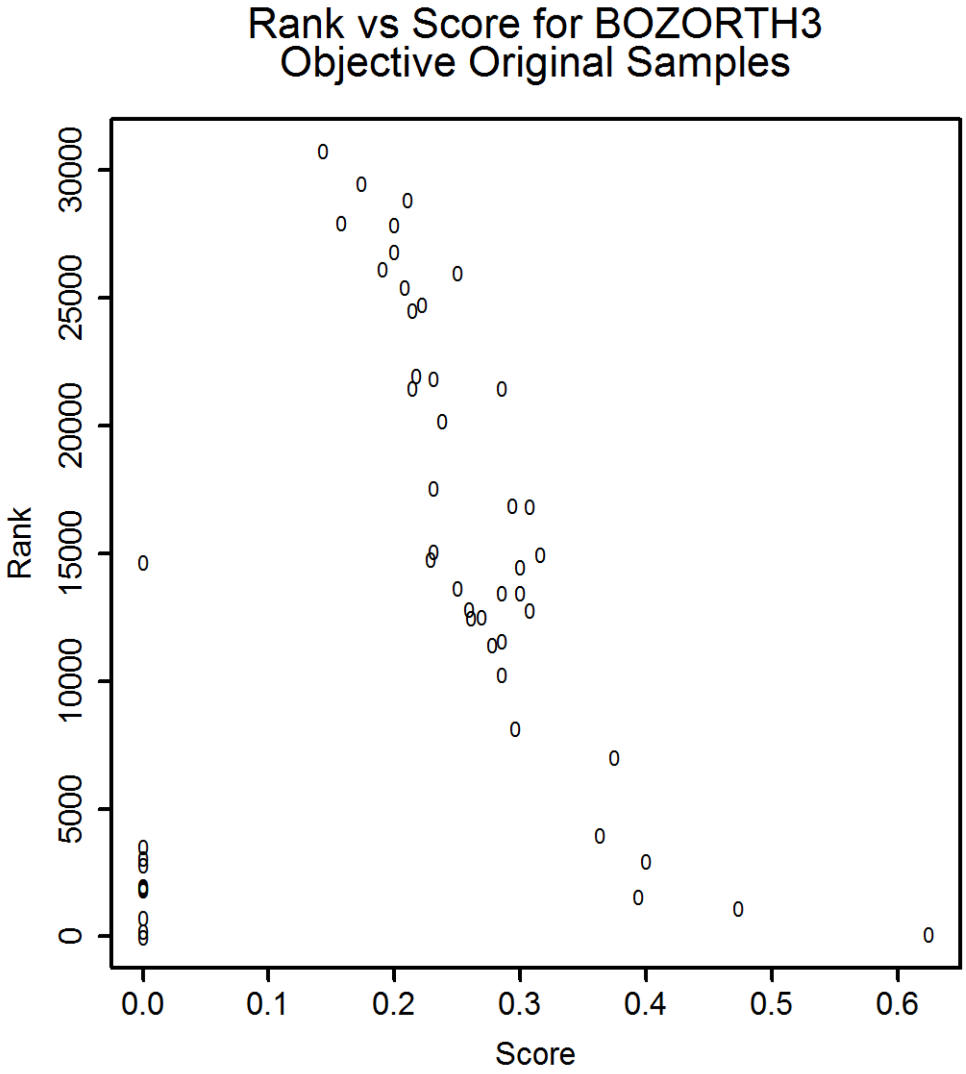


Figure E.6: The scatter plot for the match score versus the rank for the objective original images.

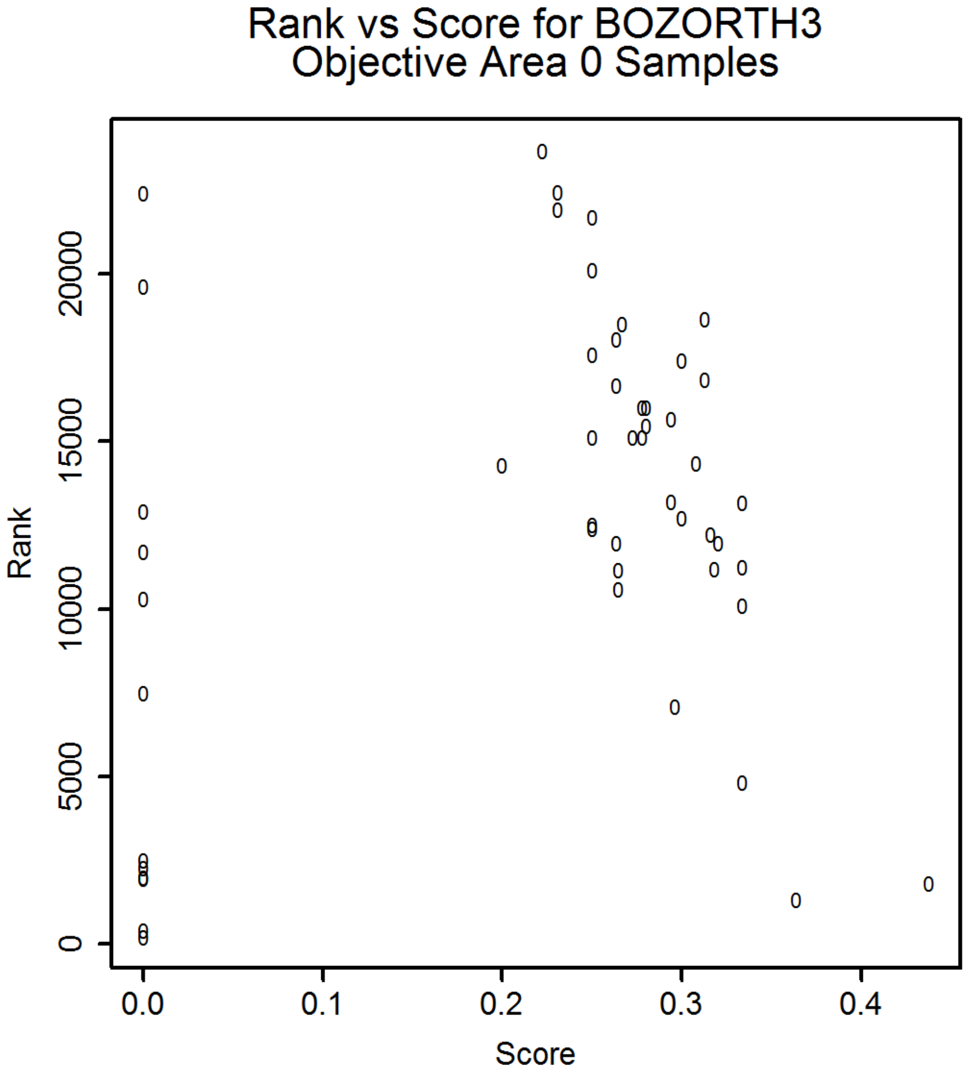


Figure E.7: The scatter plot for the match score versus the rank for the objective Area 0 images.

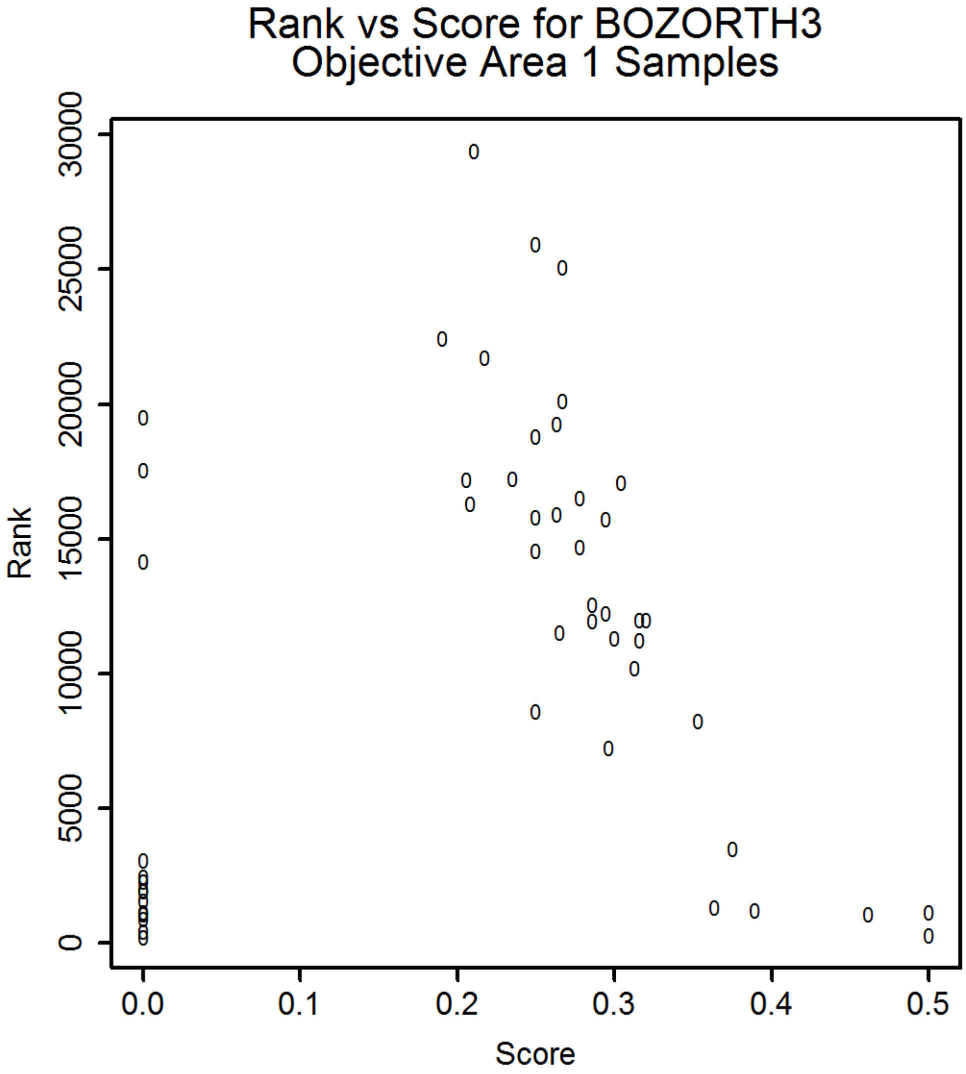


Figure E.8: The scatter plot for the match score versus the rank for the objective Area 1 images.

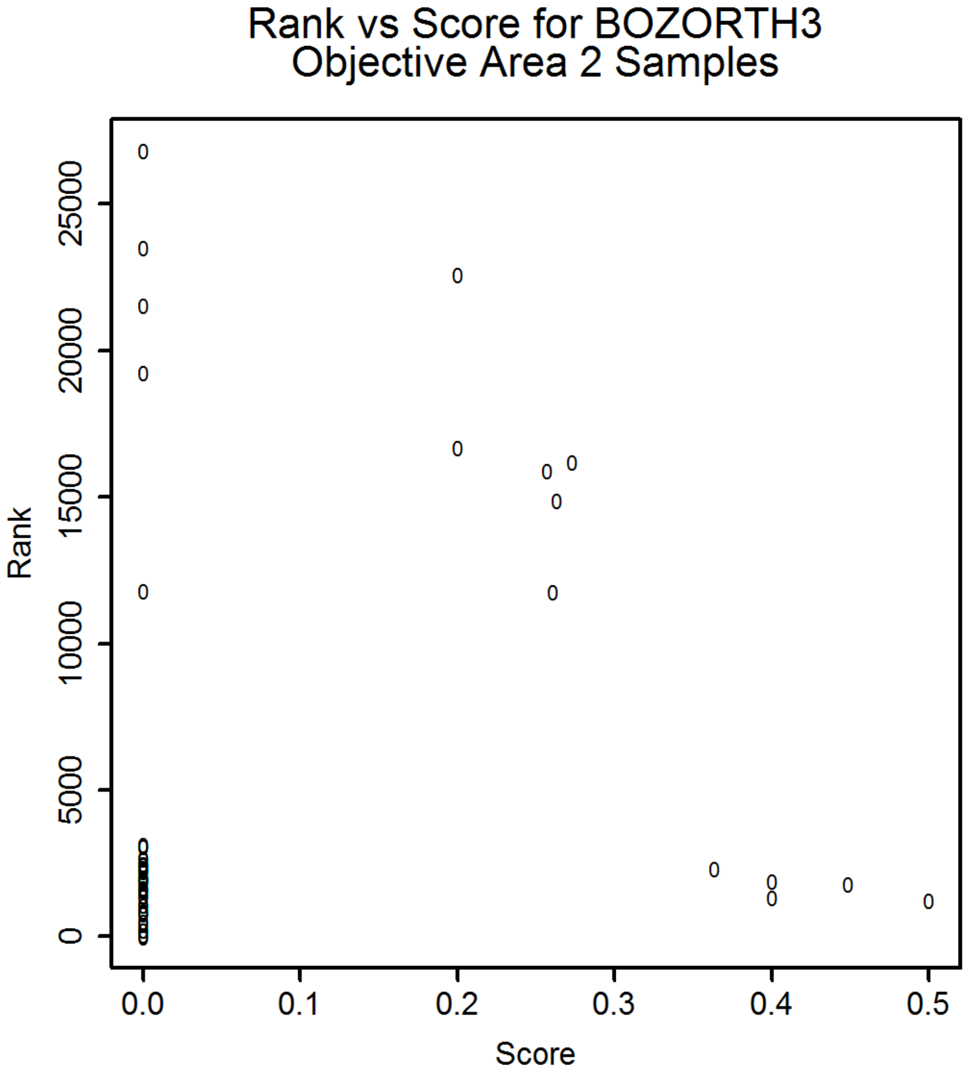


Figure E.9: The scatter plot for the match score versus the rank for the objective Area 2 images.

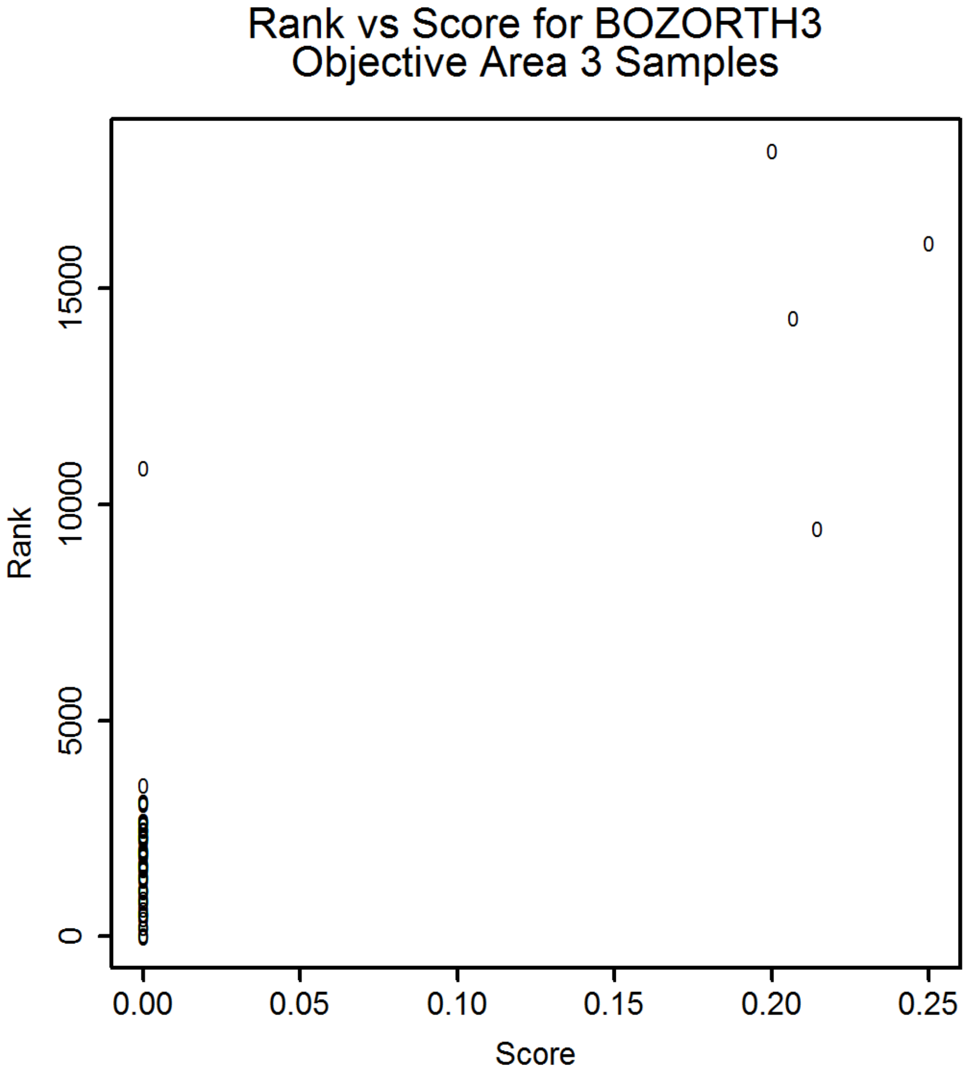


Figure E.10: The scatter plot for the match score versus the rank for the objective Area 3 images.

E.2 ROC Curves: Match Scores

E.2.1 Subjective Method

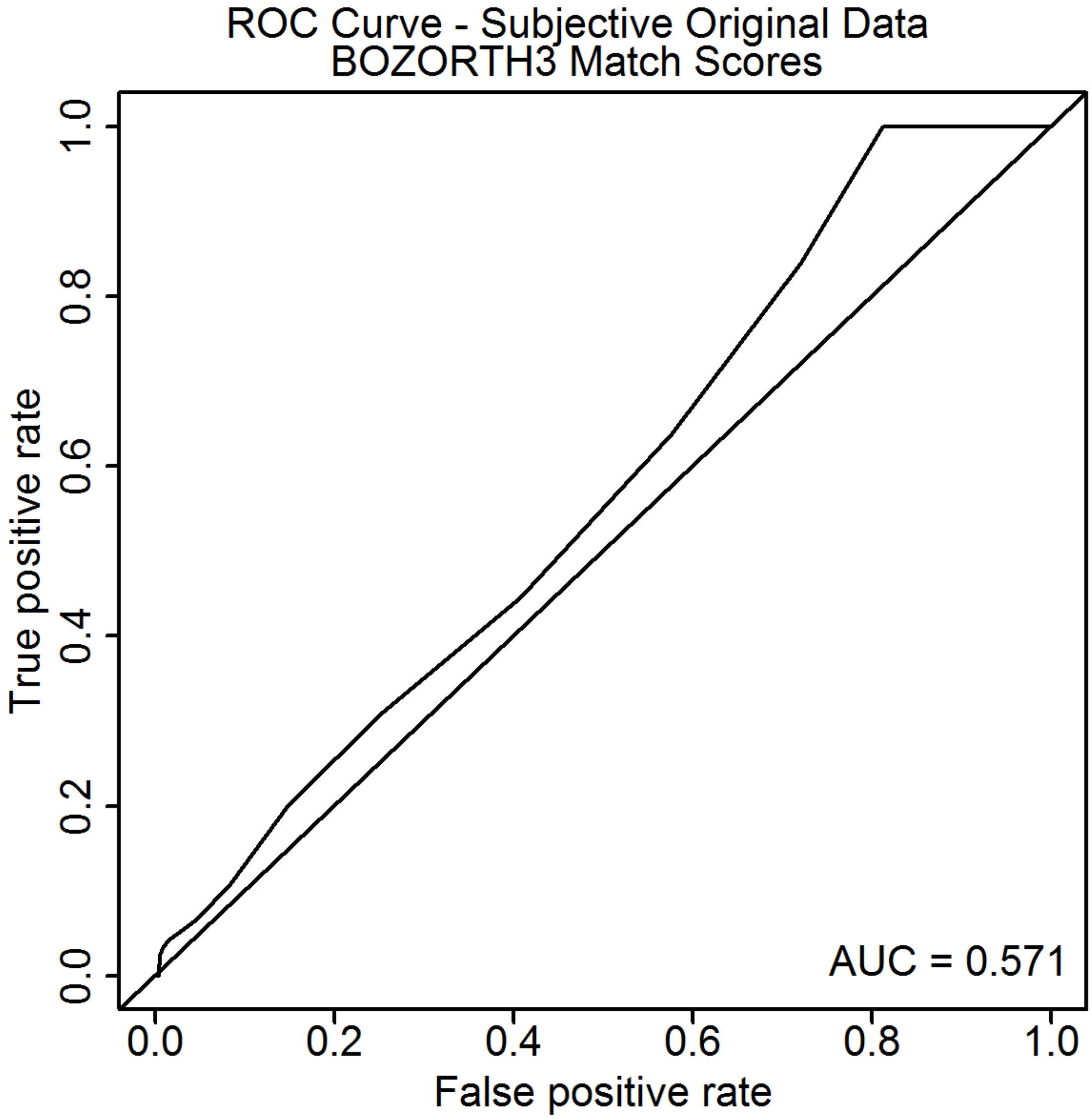


Figure E.11: This figure presents the ROC curve from the subjective original images for the match scores.

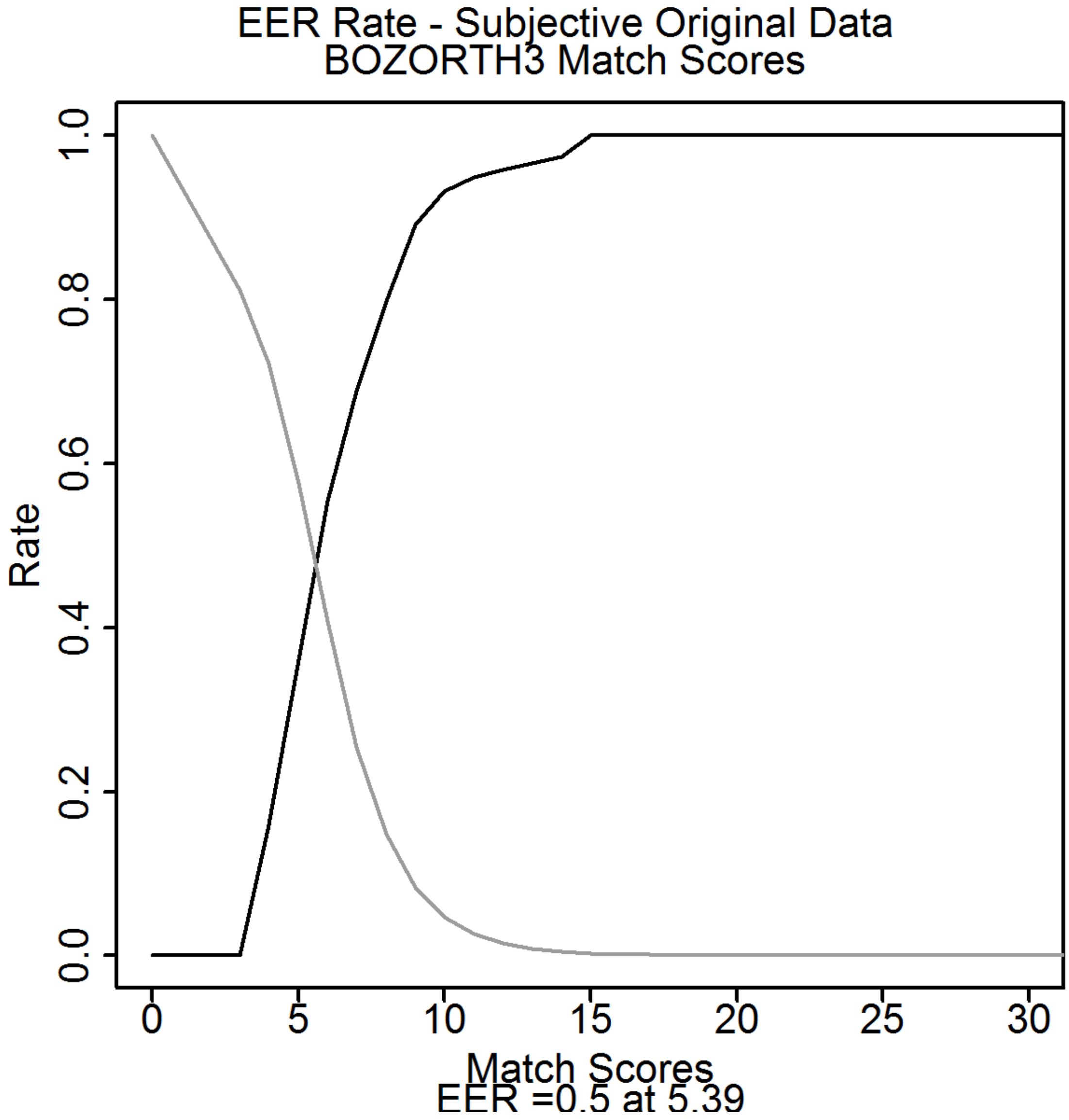


Figure E.12: This figure presents the EER plot from the subjective original images for the match scores.

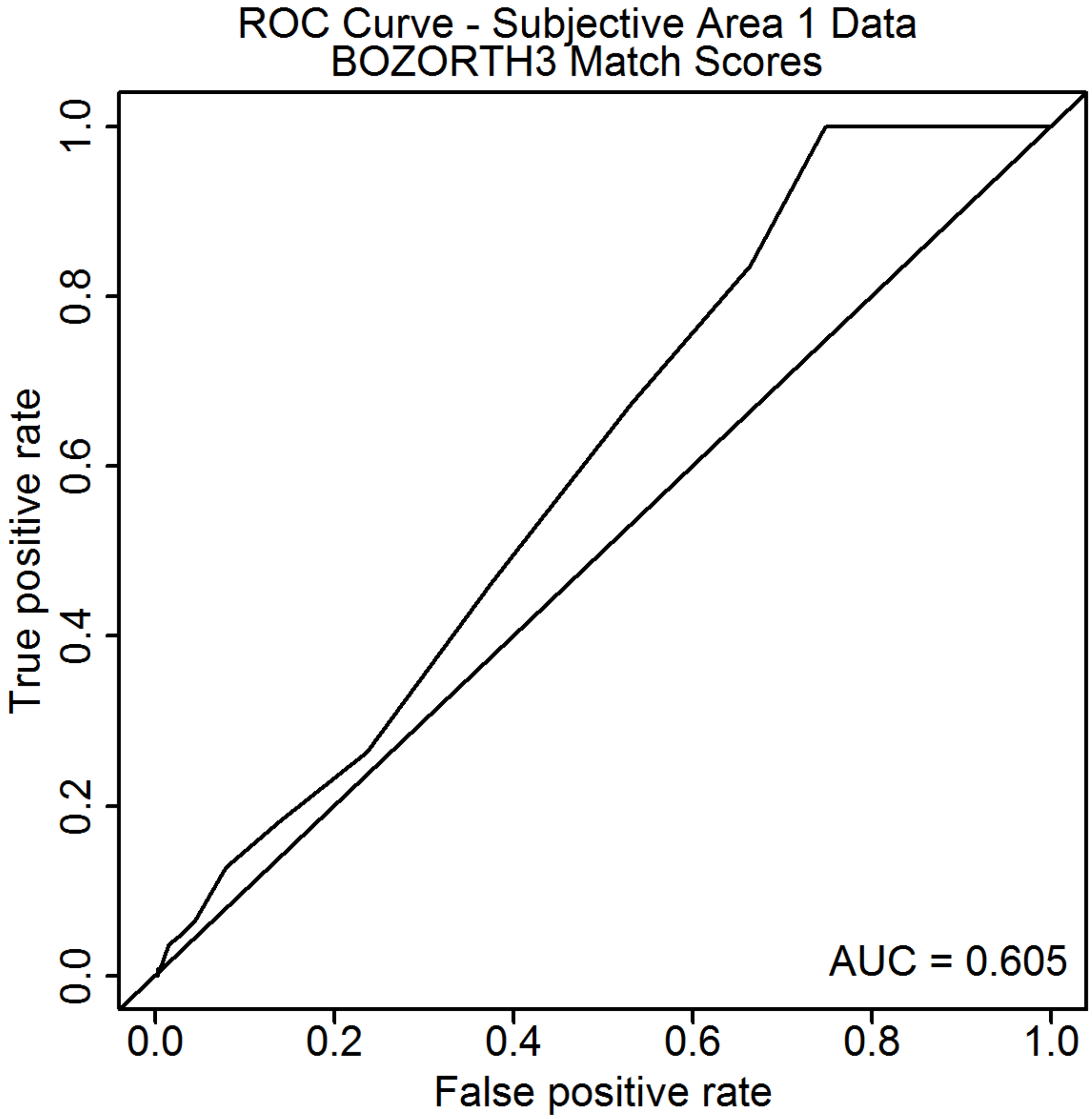


Figure E.13: This figure presents the ROC curve from the subjective Area 1 images for the match scores.

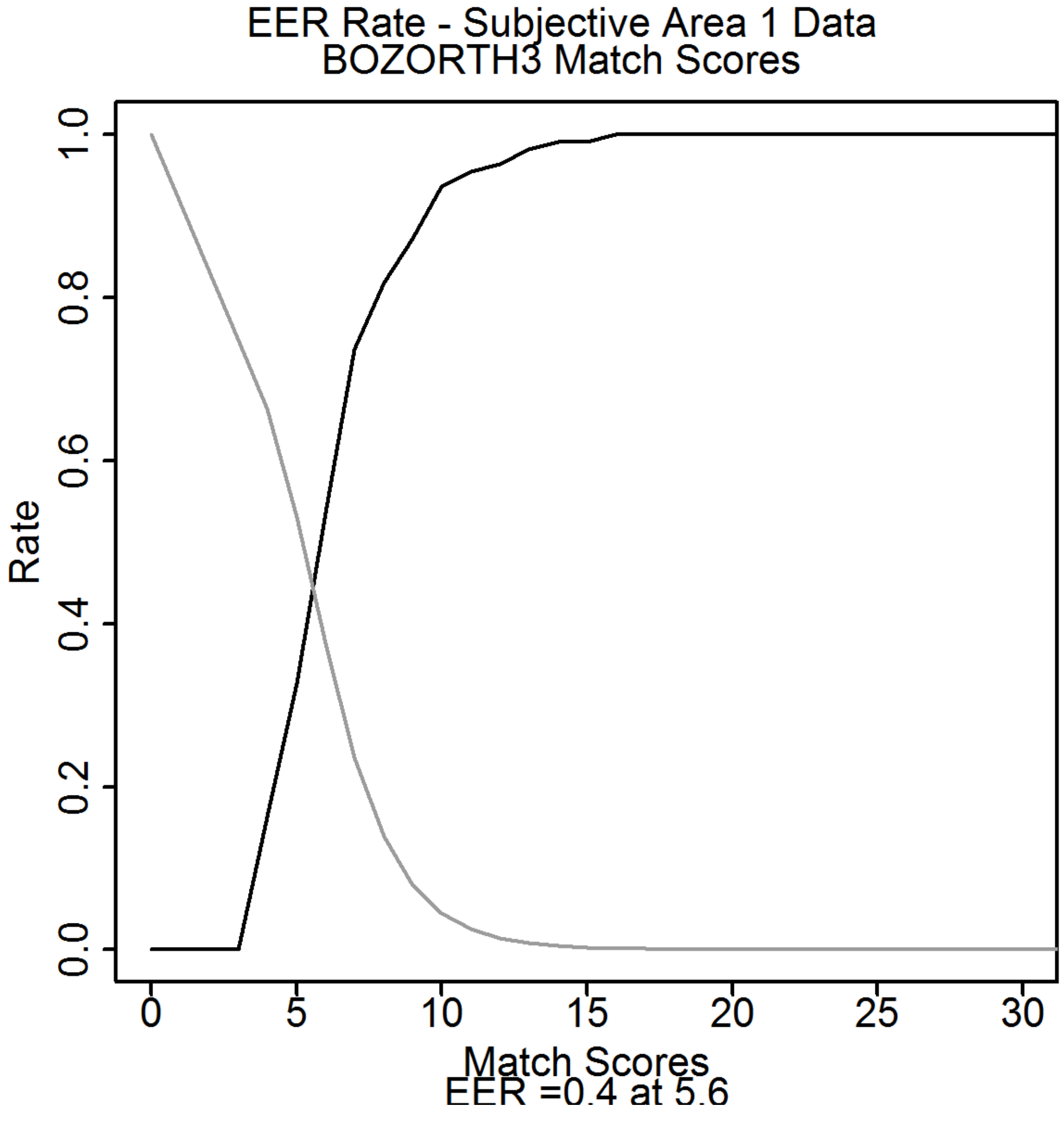


Figure E.14: This figure presents the EER plot from the subjective Area 1 images for the match scores.

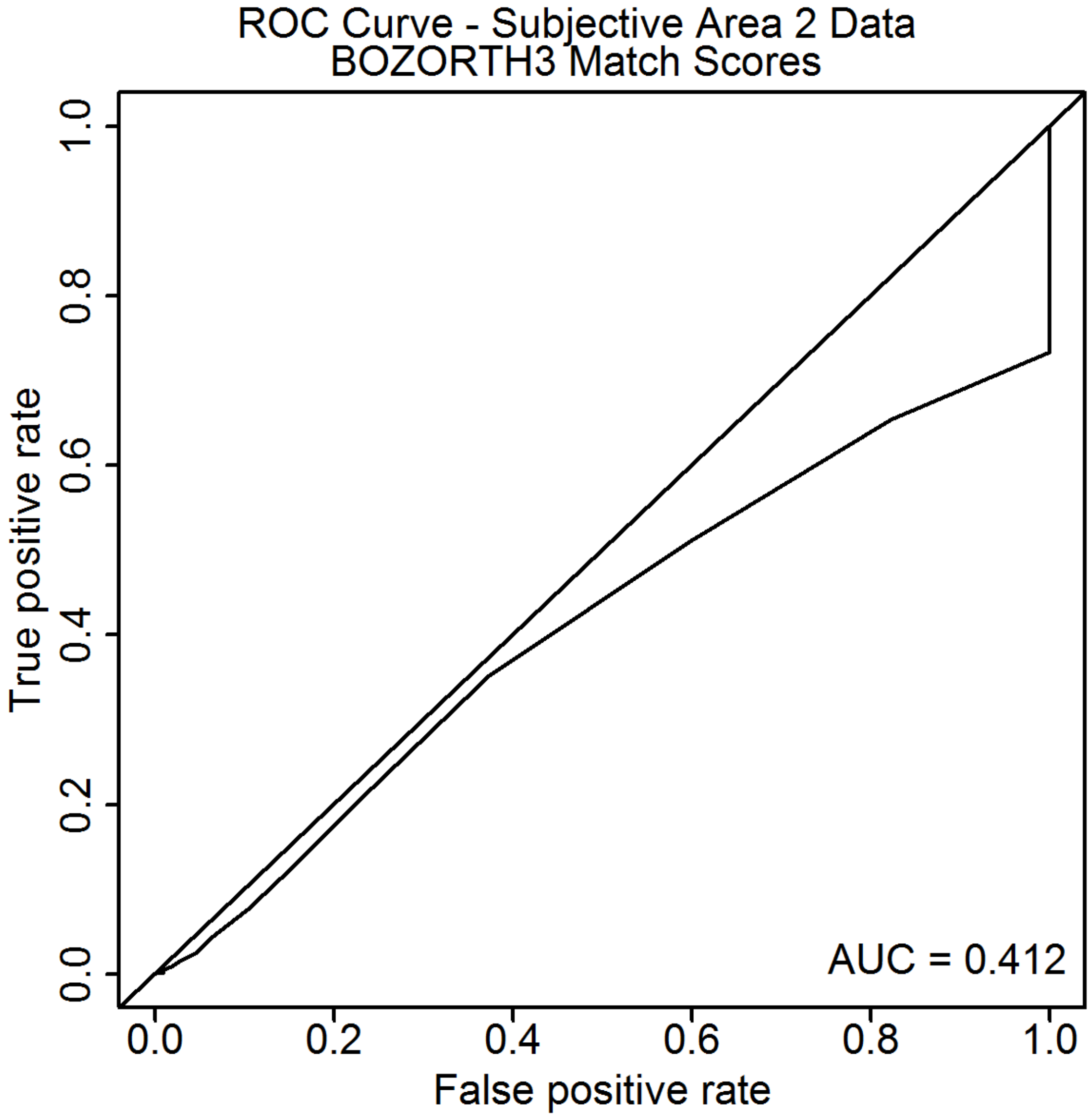


Figure E.15: This figure presents the ROC curve from the subjective Area 2 images for the match scores.

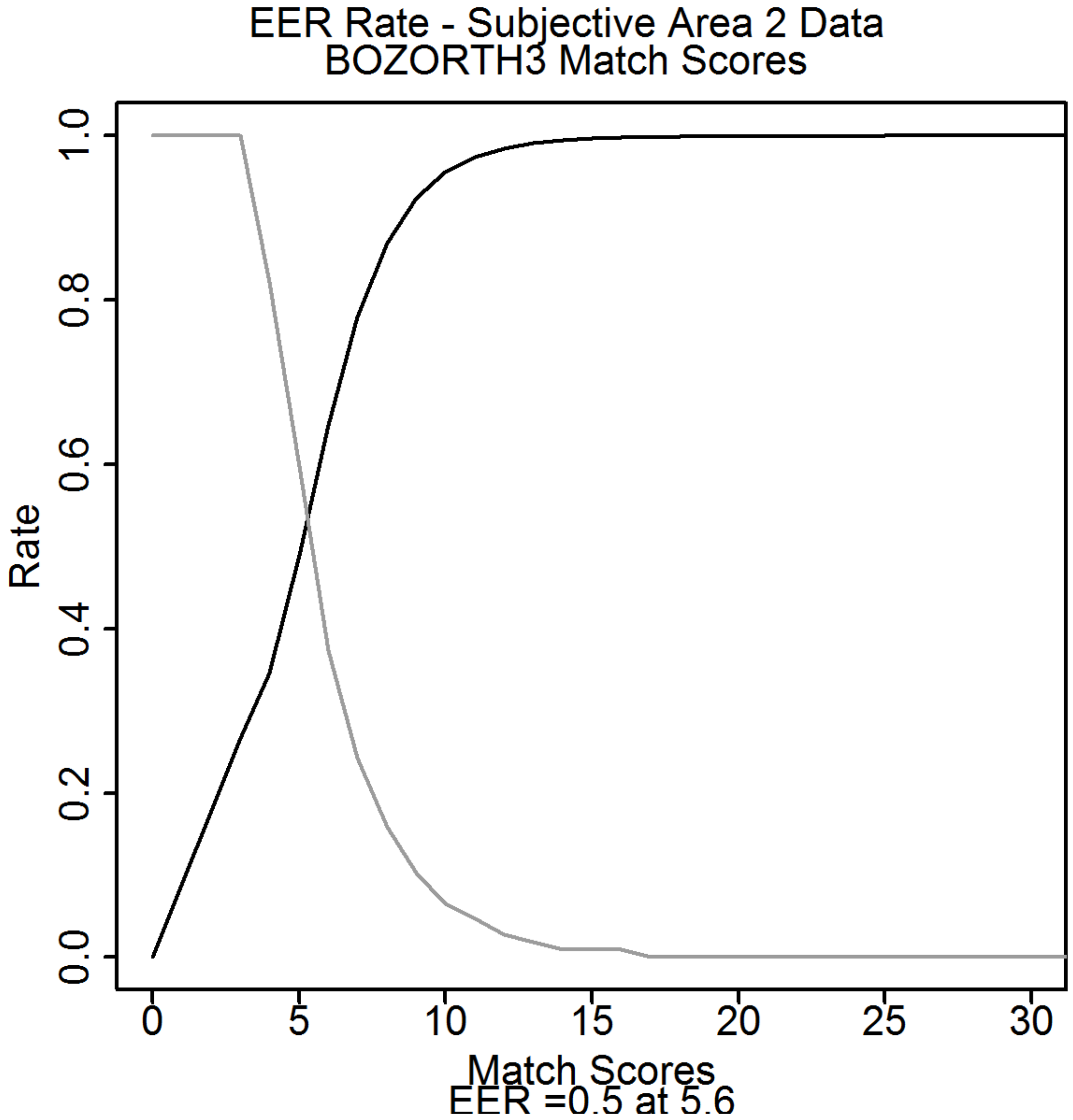


Figure E.16: This figure presents the EER plot from the subjective Area 2 images for the match scores.

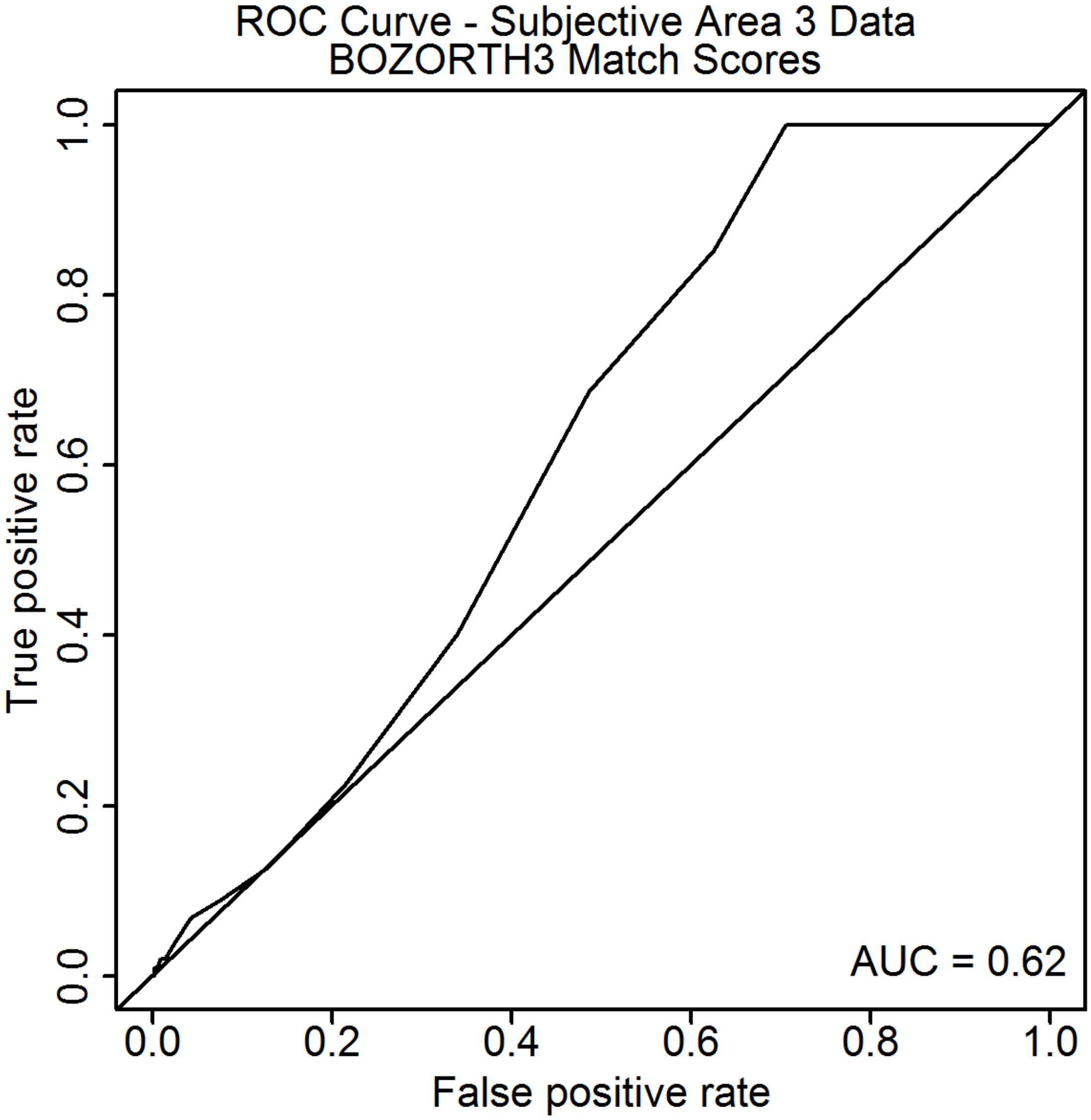


Figure E.17: This figure presents the ROC curve from the subjective Area 3 images for the match scores.

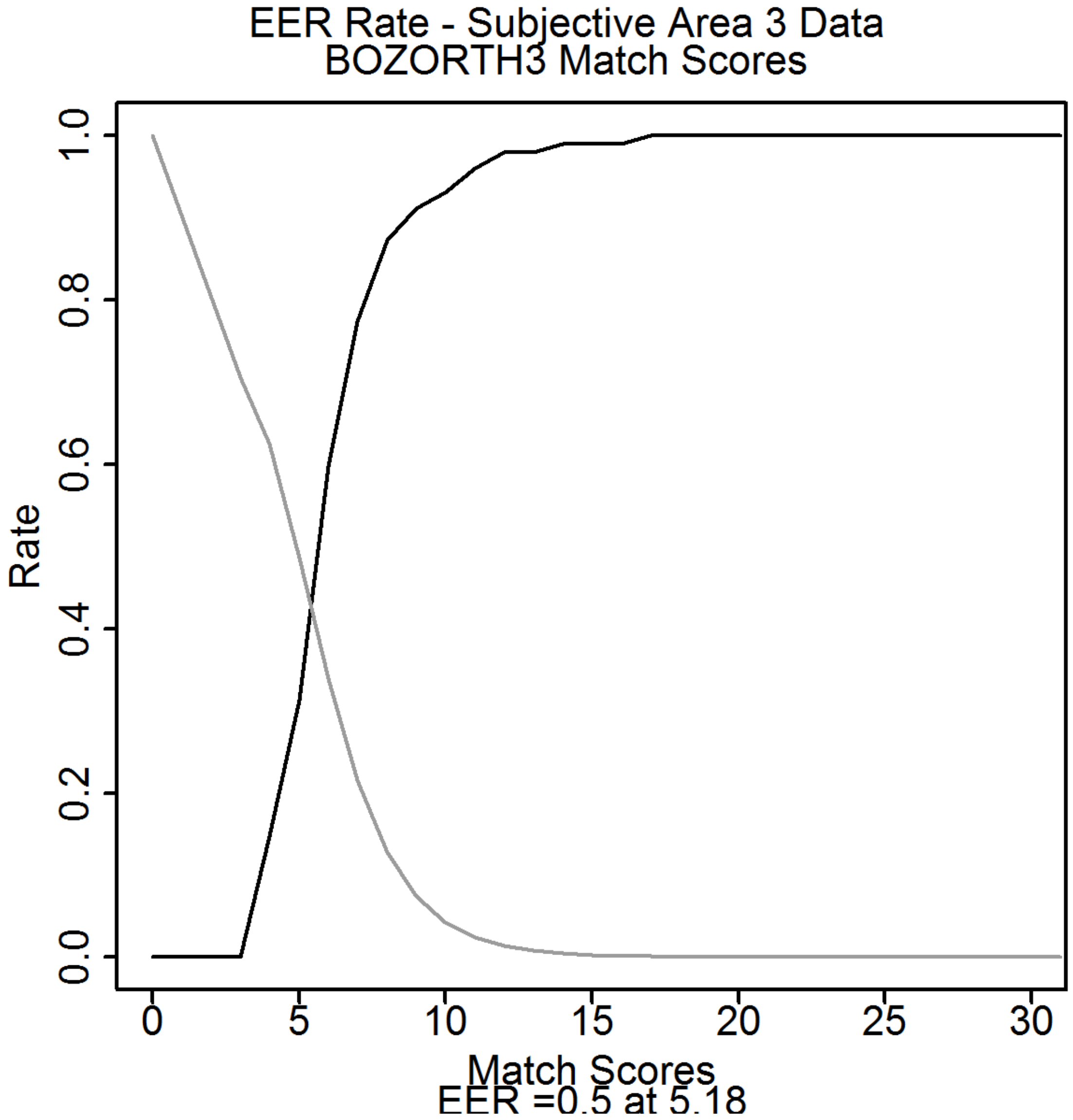


Figure E.18: This figure presents the EER plot from the subjective Area 3 images for the match scores.

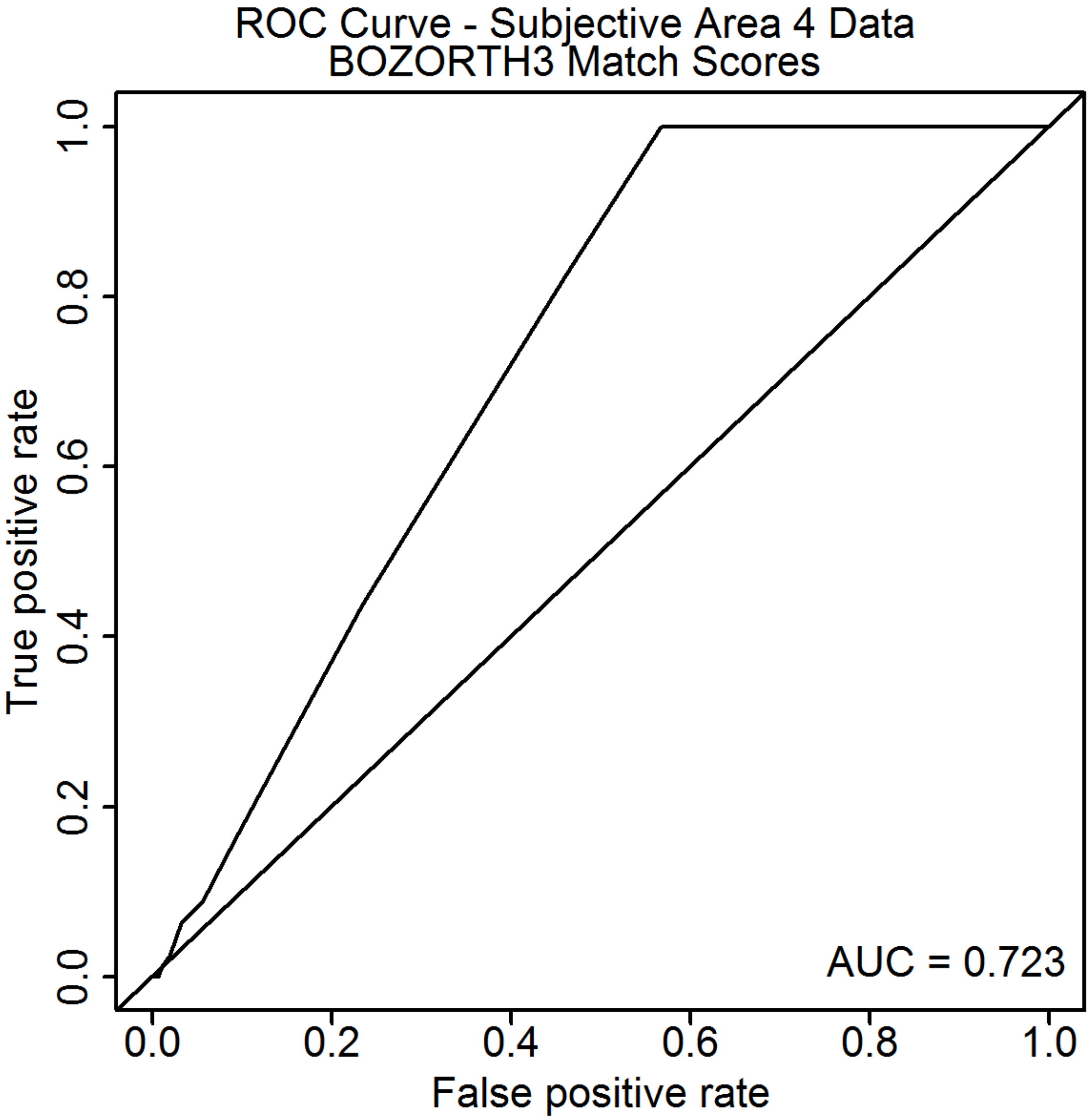


Figure E.19: This figure presents the ROC curve from the subjective Area 4 images for the match scores.

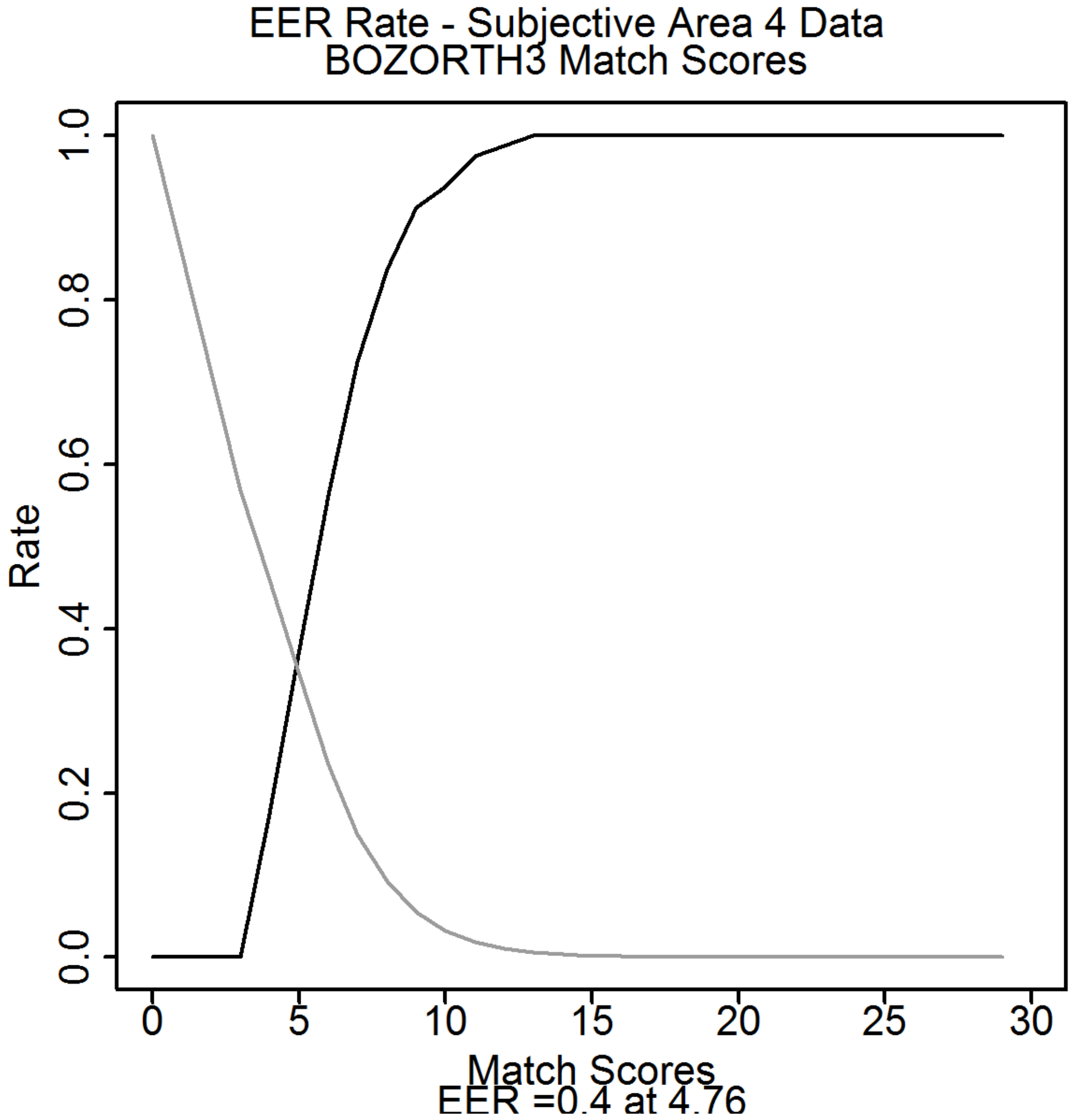


Figure E.20: This figure presents the EER plot from the subjective Area 4 images for the match scores.

E.2.2 Objective Method

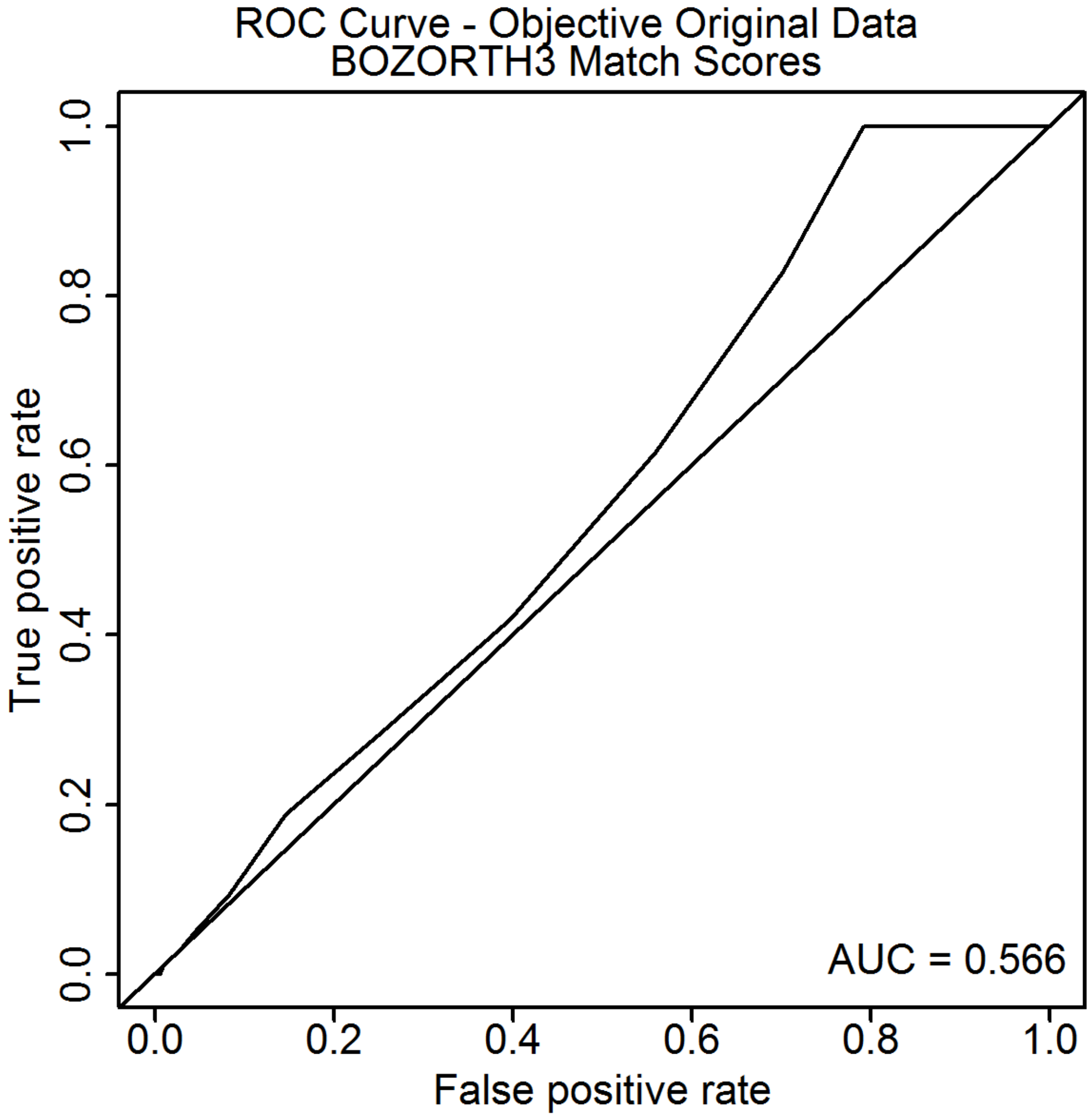


Figure E.21: This figure presents the ROC curve from the objective original images for the match scores.

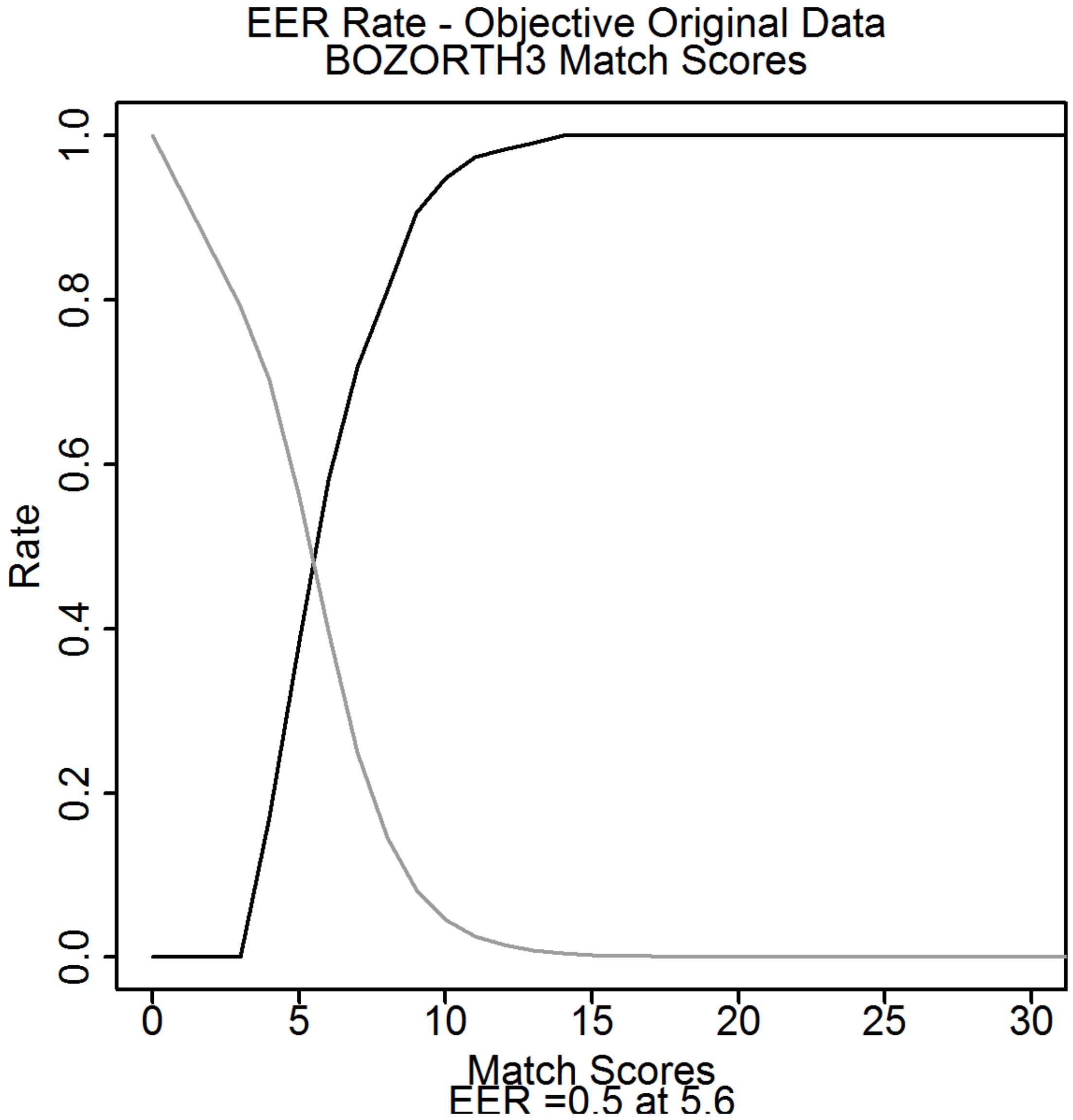


Figure E.22: This figure presents the EER plot from the objective original images for the match scores.

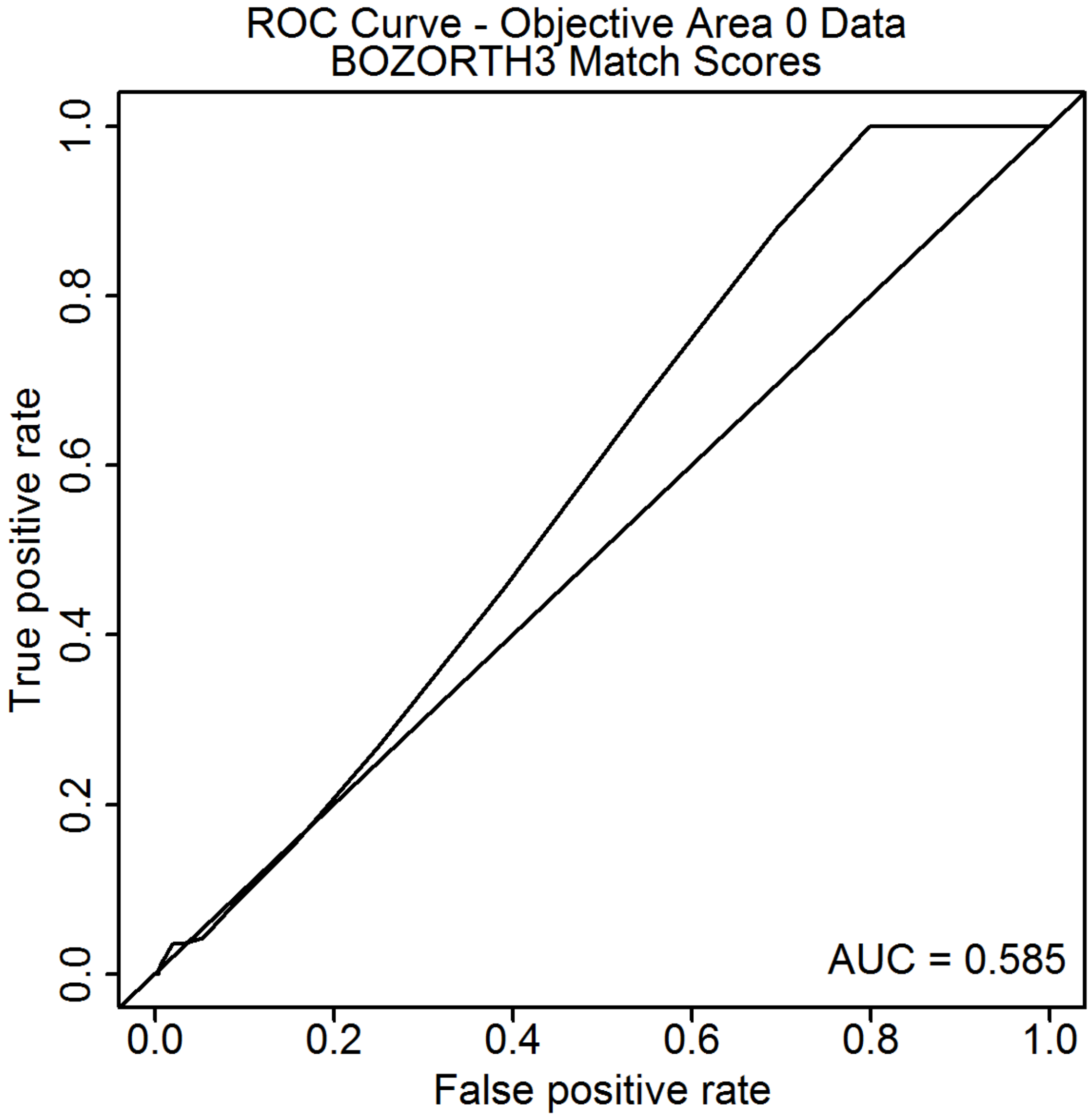


Figure E.23: This figure presents the ROC curve from the objective Area 0 images for the match scores.

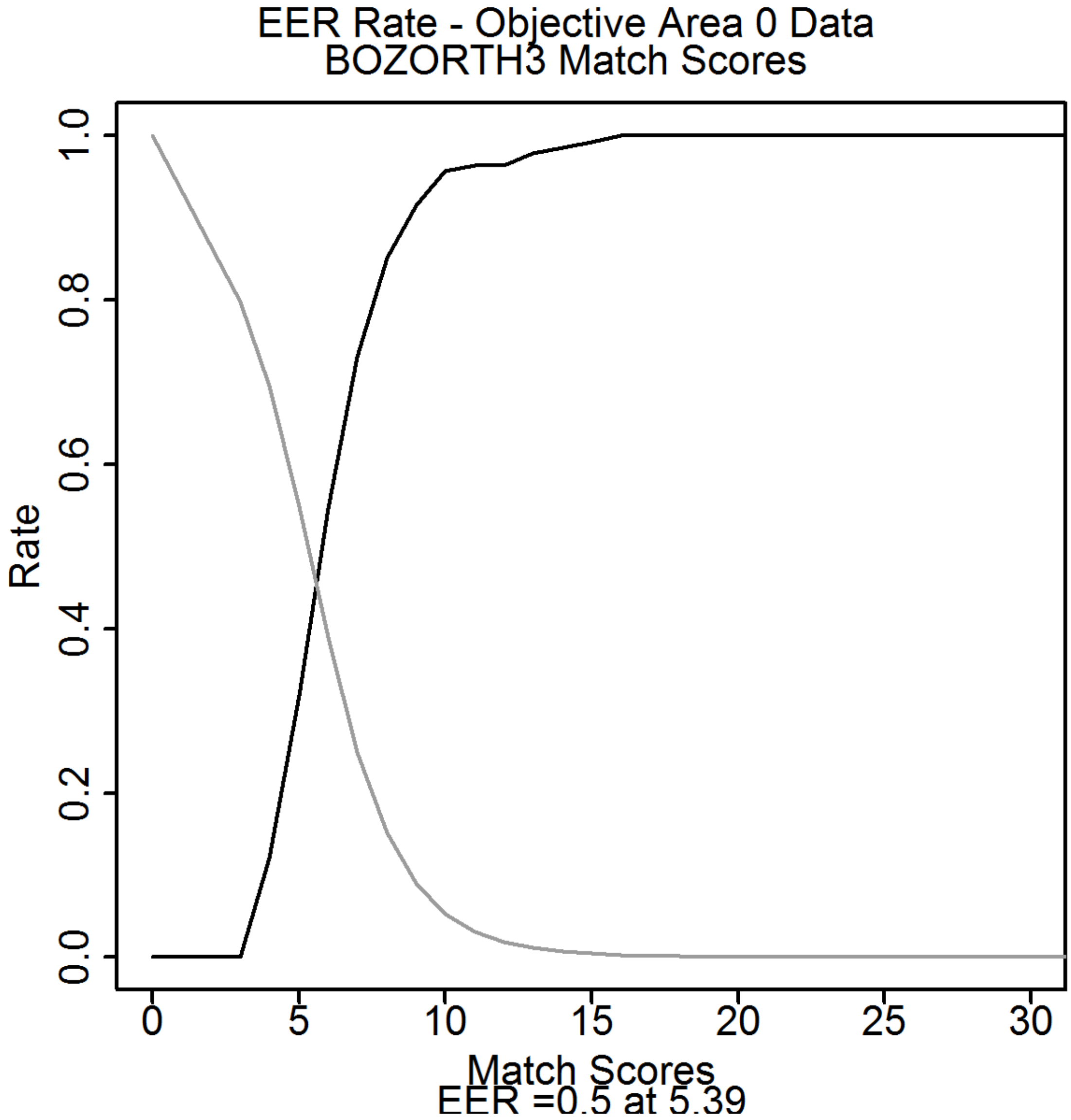


Figure E.24: This figure presents the EER plot from the objective Area 0 images for the match scores.

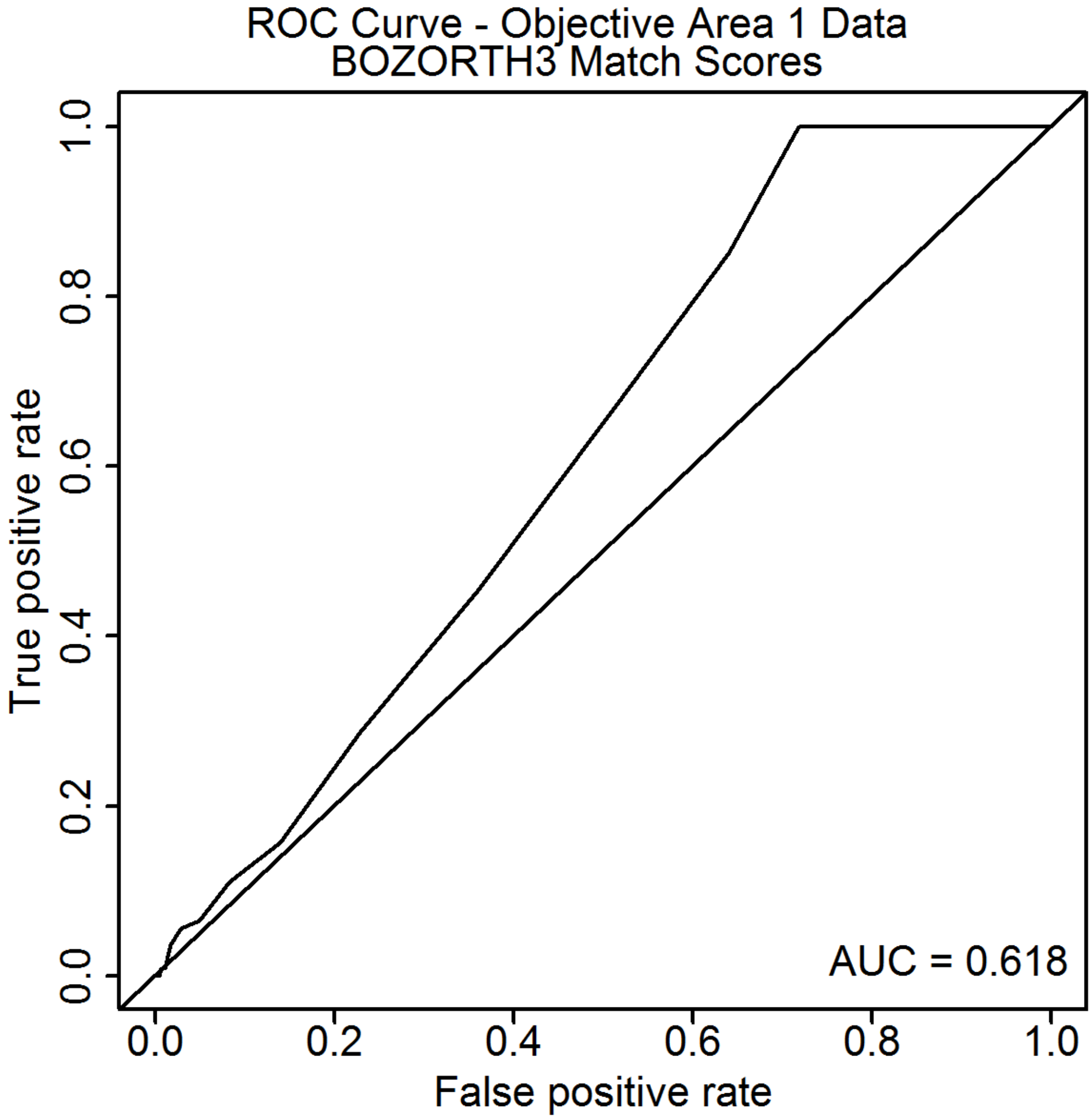


Figure E.25: This figure presents the ROC curve from the objective Area 1 images for the match scores.

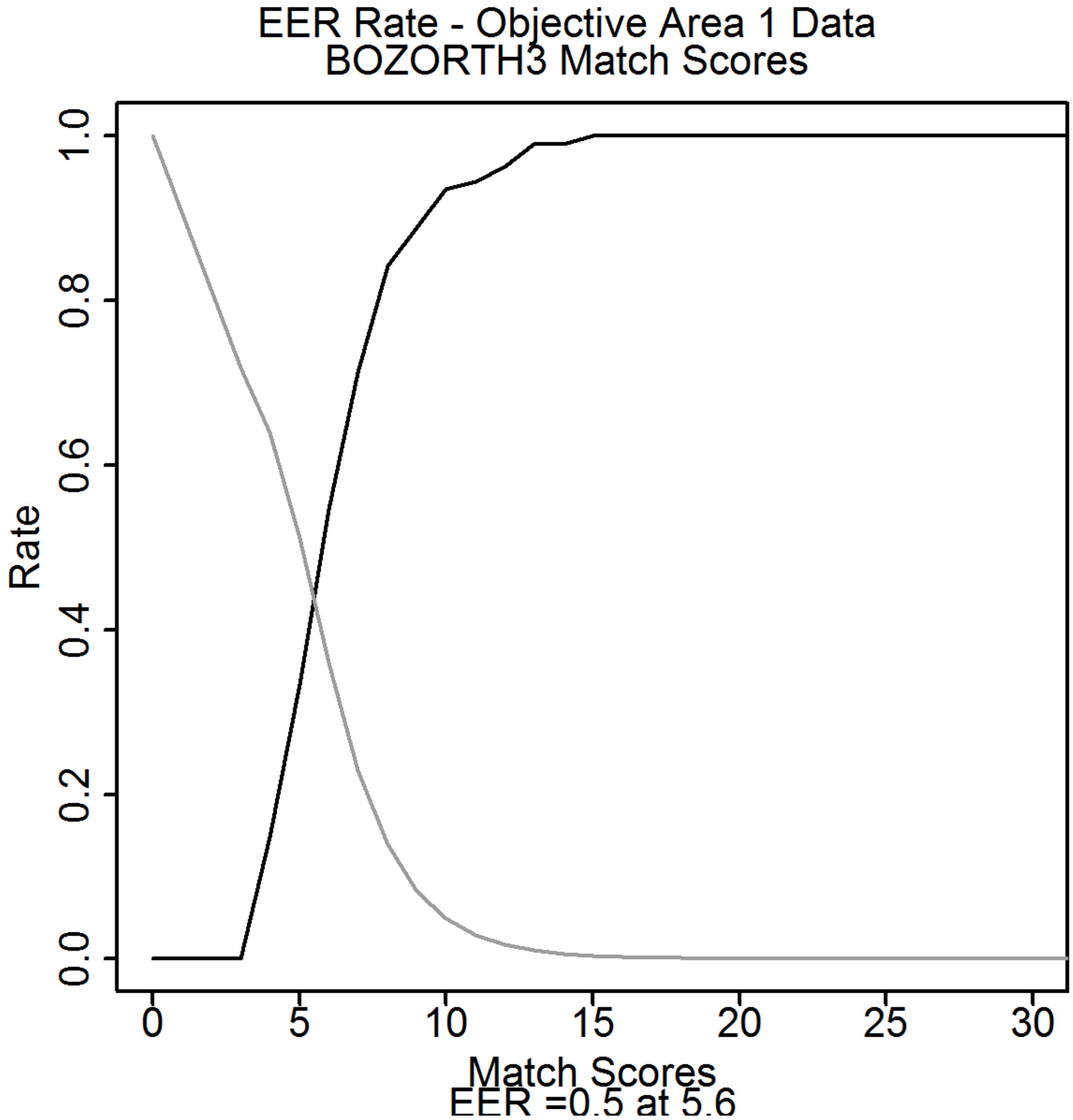


Figure E.26: This figure presents the EER plot from the objective Area 1 images for the match scores.

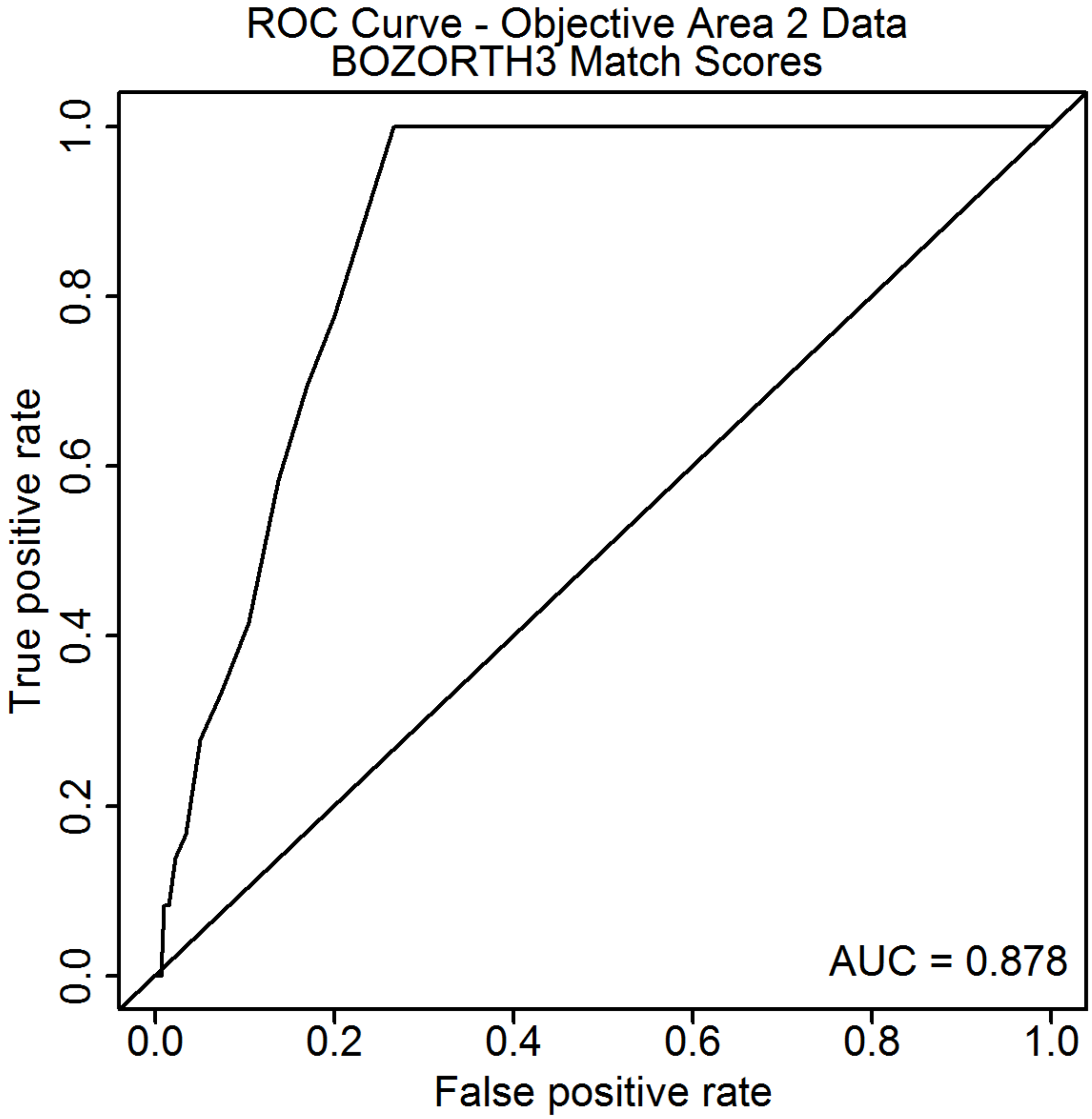


Figure E.27: This figure presents the ROC curve from the objective Area 2 images for the match scores.

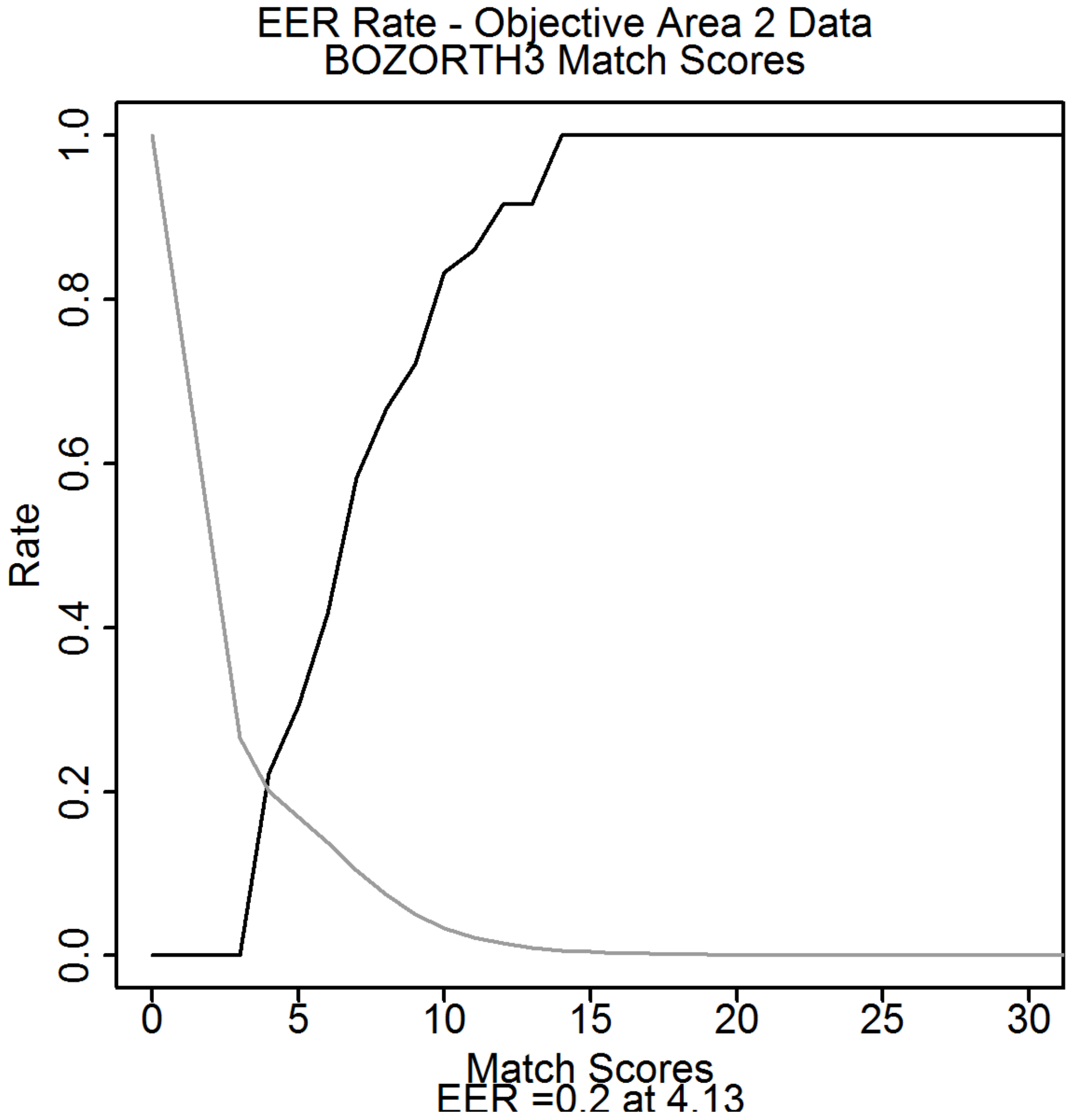


Figure E.28: This figure presents the EER plot from the objective Area 2 images for the match scores.

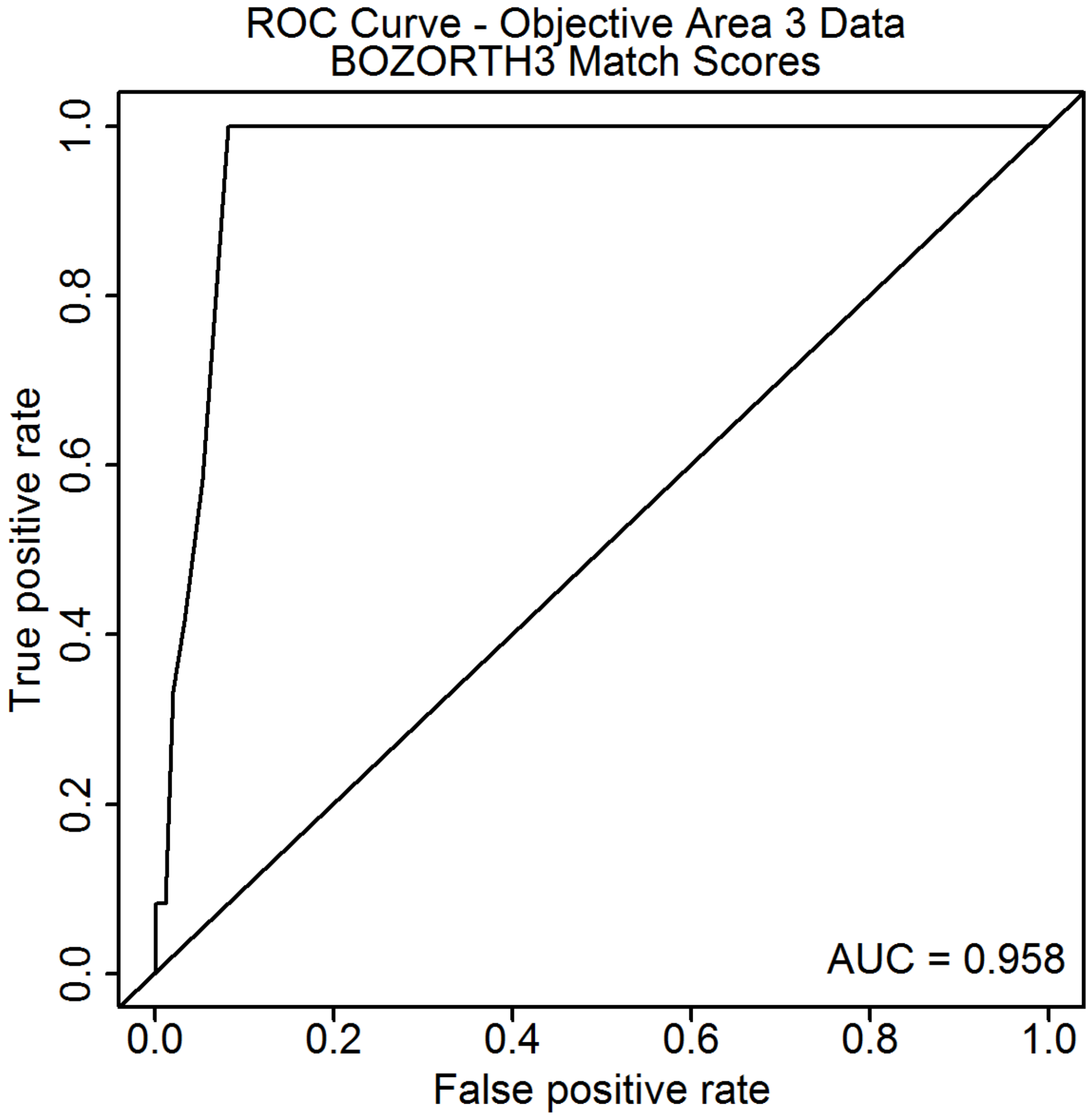


Figure E.29: This figure presents the ROC curve from the objective Area 3 images for the match scores.

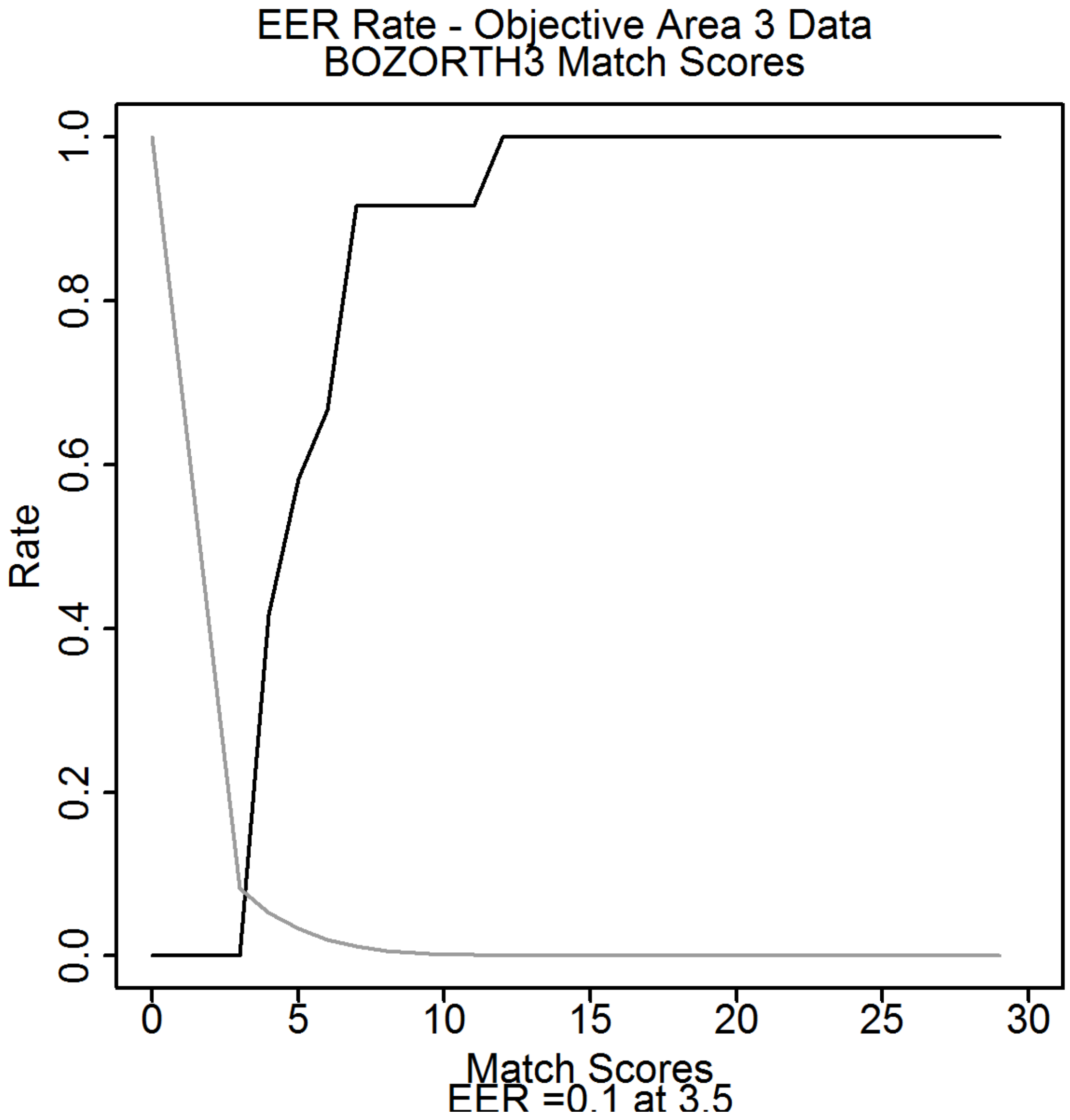


Figure E.30: This figure presents the EER plot from the objective Area 3 images for the match scores.

E.3 ROC Curves: Rank

E.3.1 Subjective Method

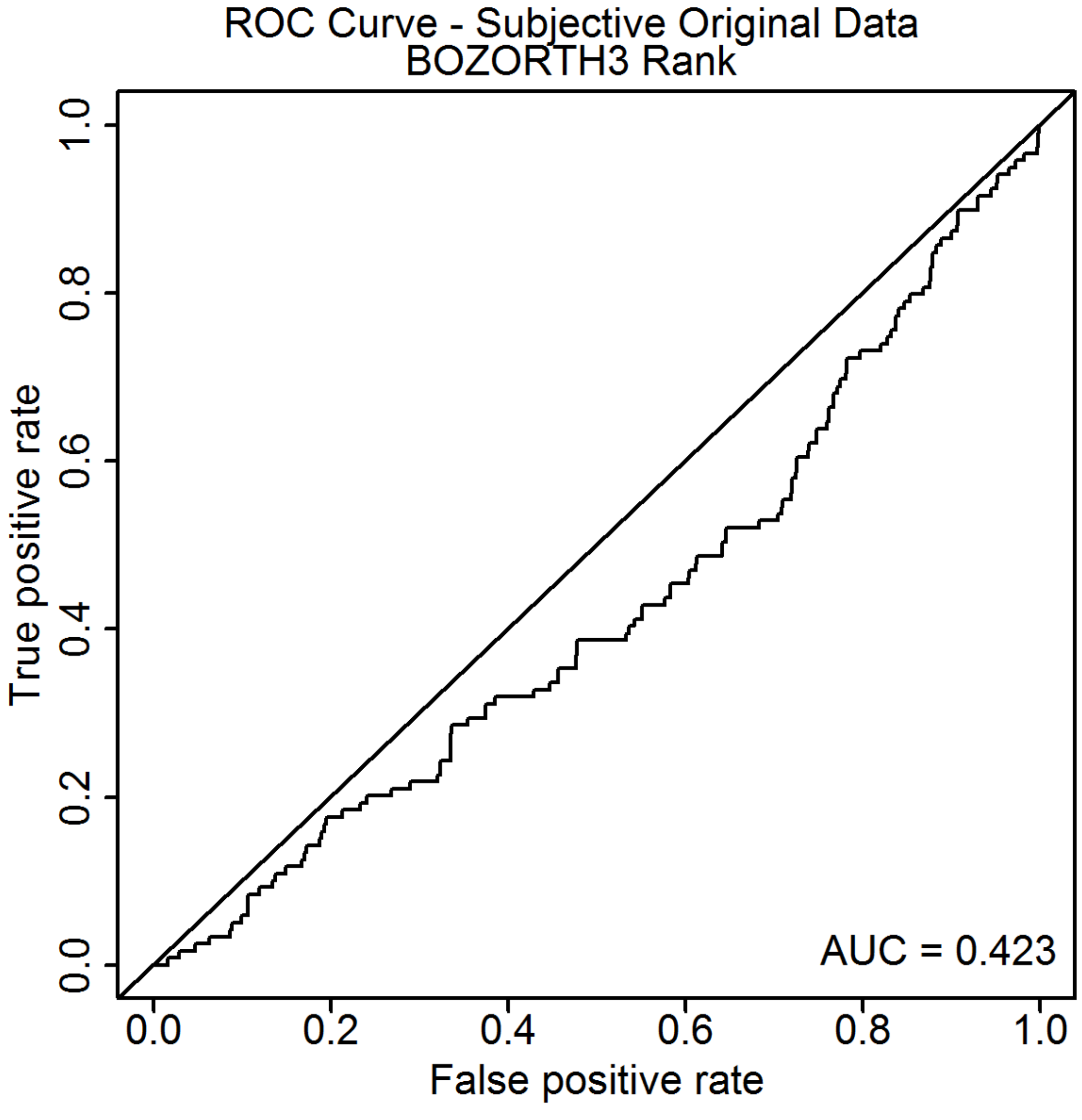


Figure E.31: This figure presents the ROC curve from the subjective original images for the ranks.

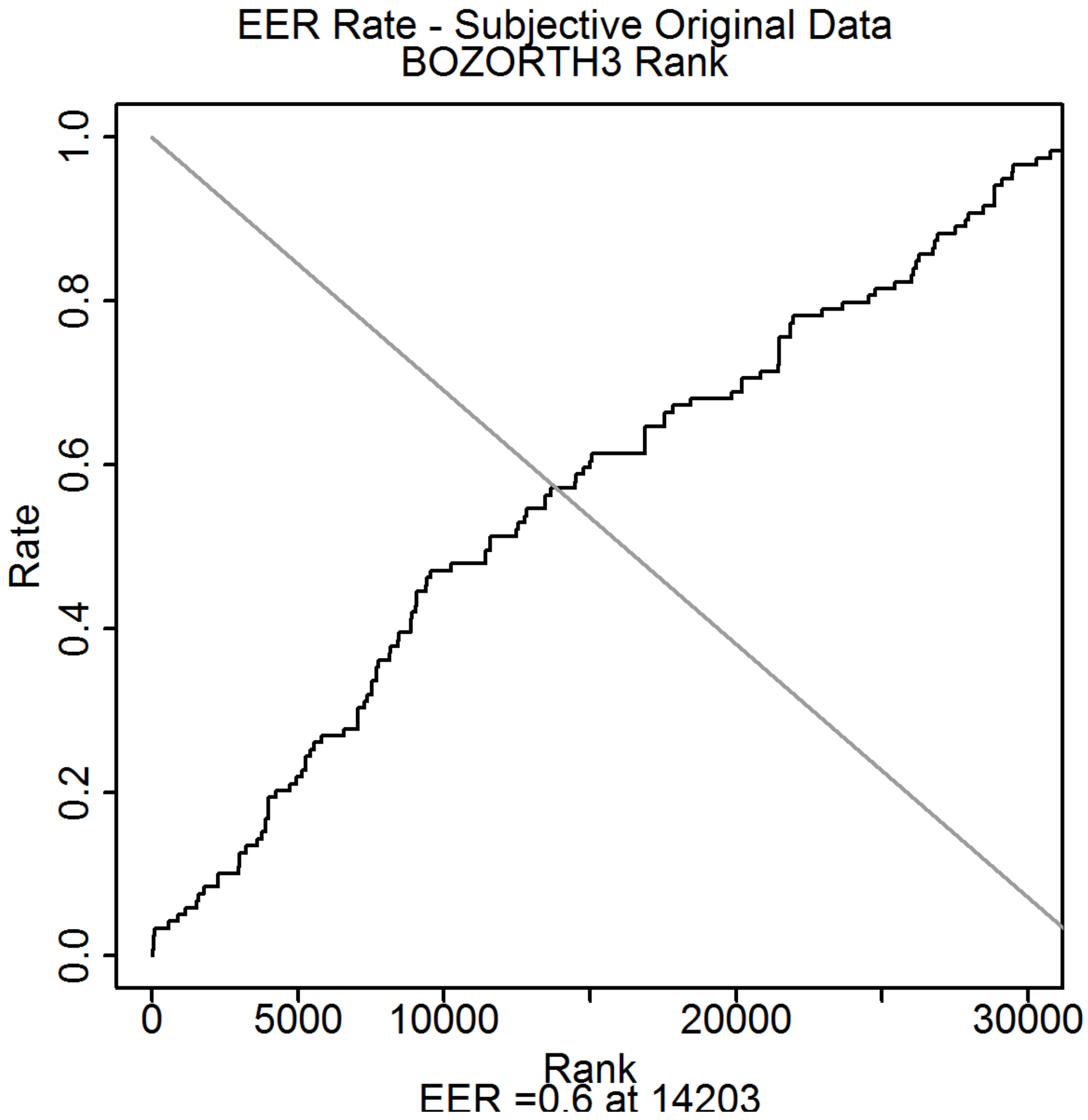


Figure E.32: This figure presents the EER plot from the subjective original images for the ranks.

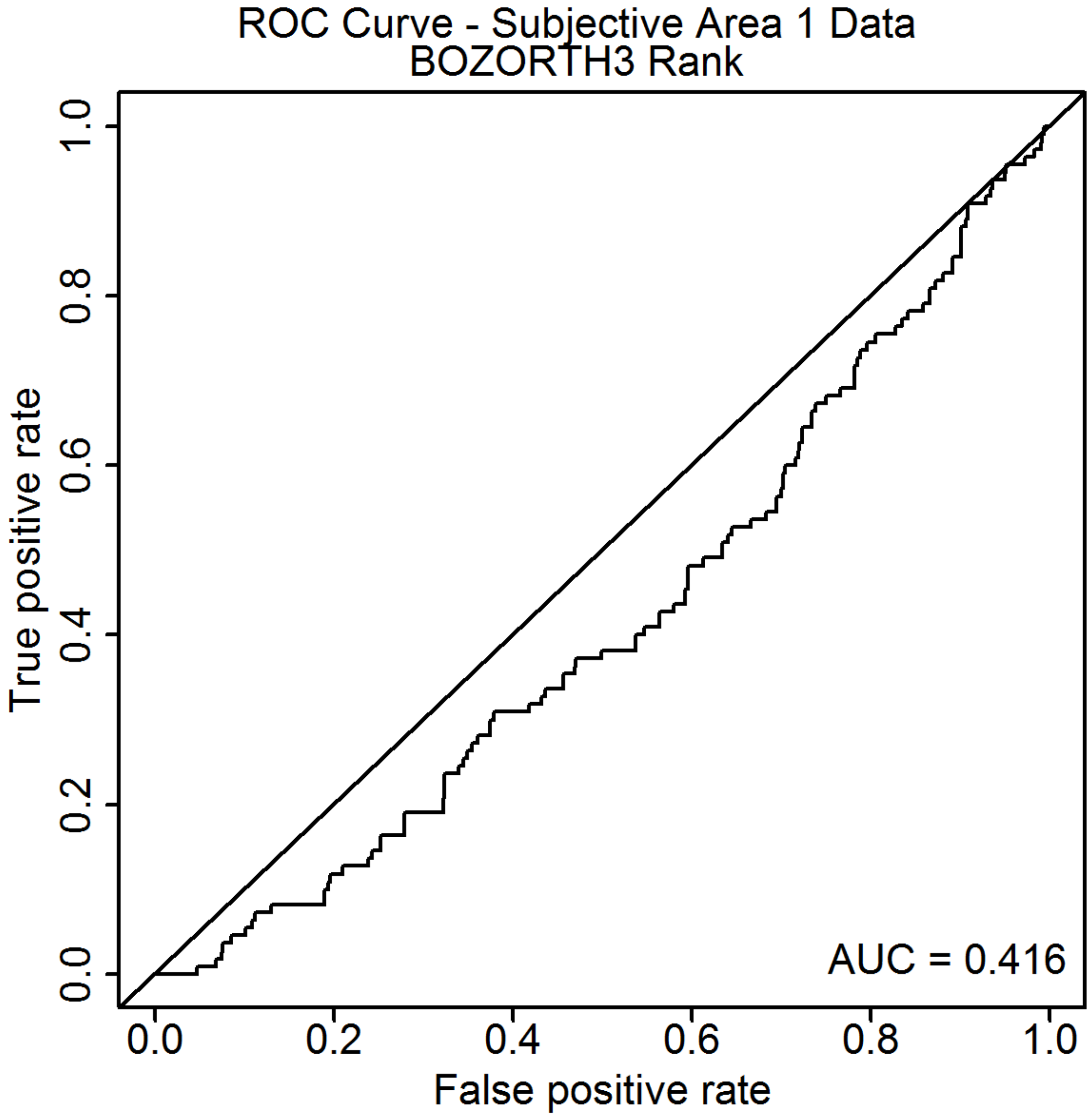


Figure E.33: This figure presents the ROC curve from the subjective Area 1 images for the ranks.

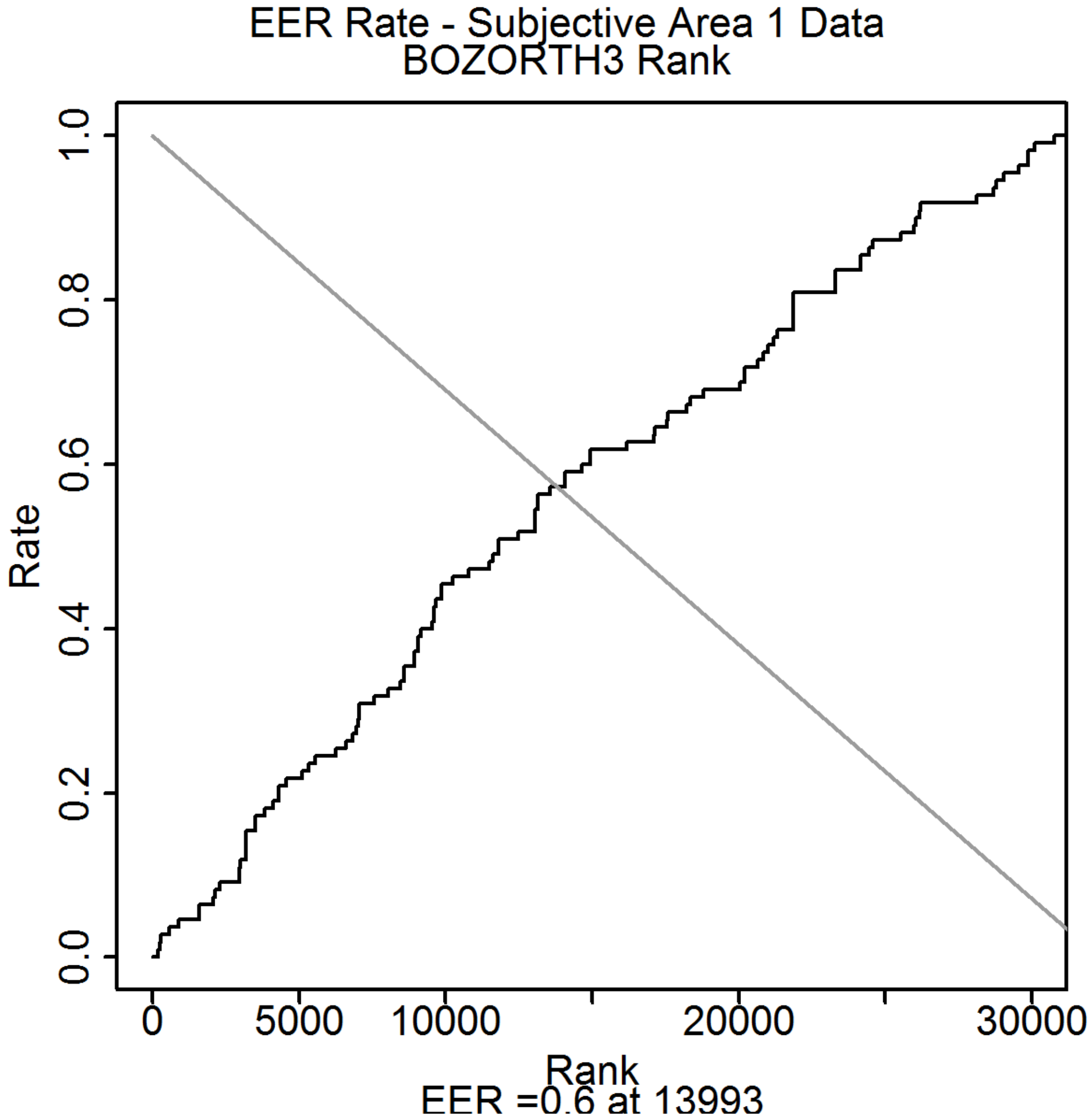


Figure E.34: This figure presents the EER plot from the subjective Area 1 images for the ranks.

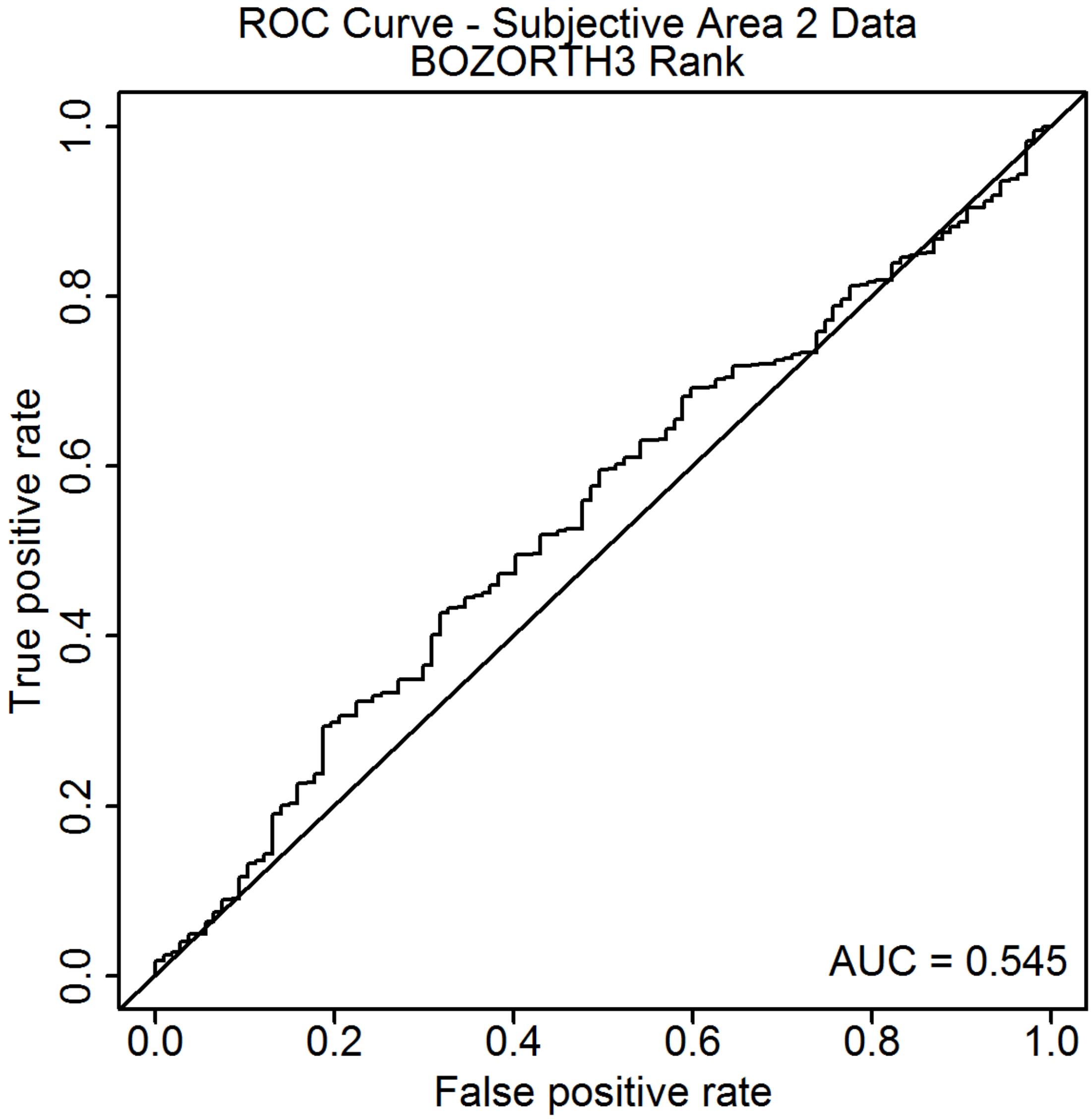


Figure E.35: This figure presents the ROC curve from the subjective Area 2 images for the ranks.

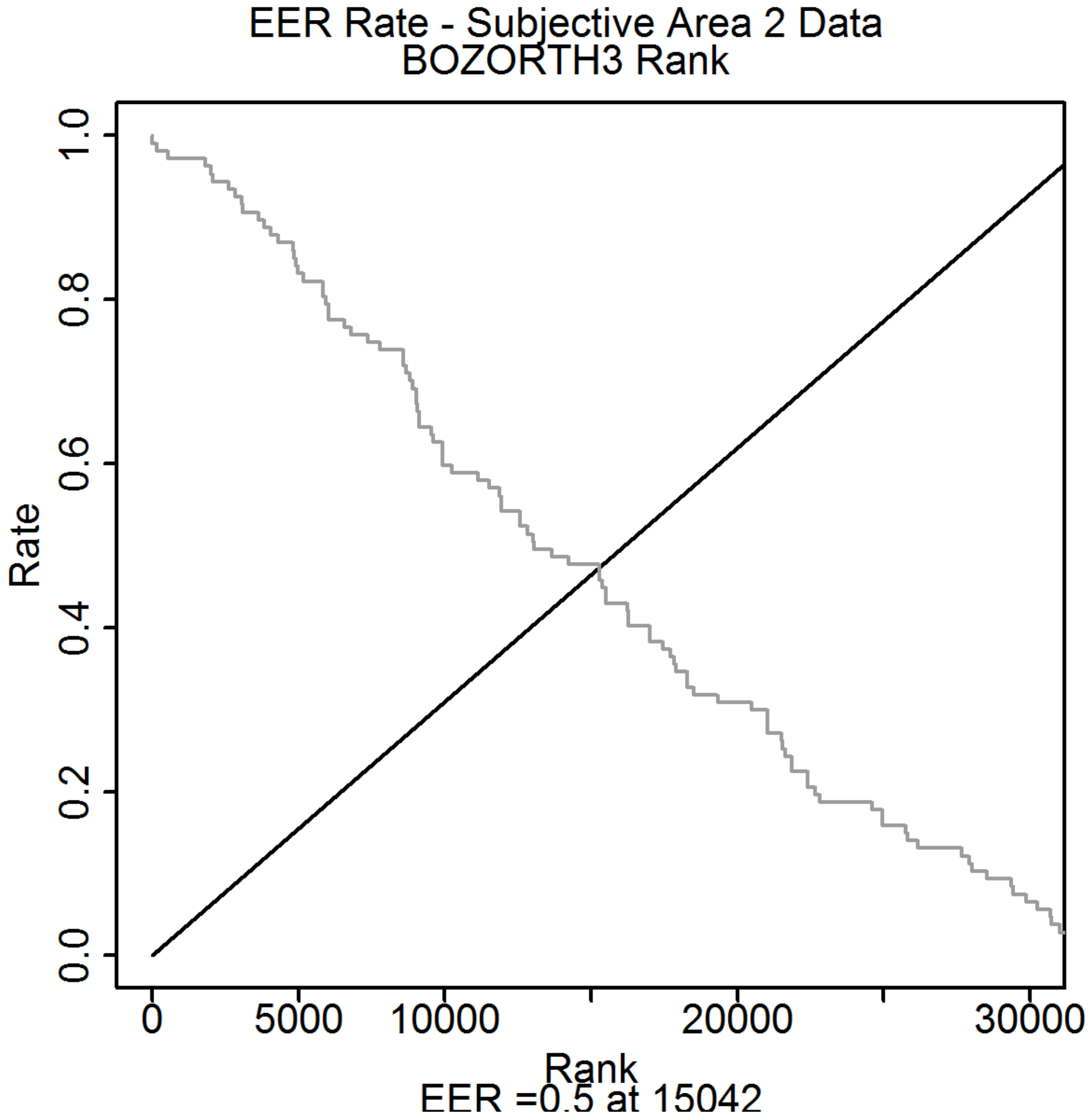


Figure E.36: This figure presents the EER plot from the subjective Area 2 images for the ranks.

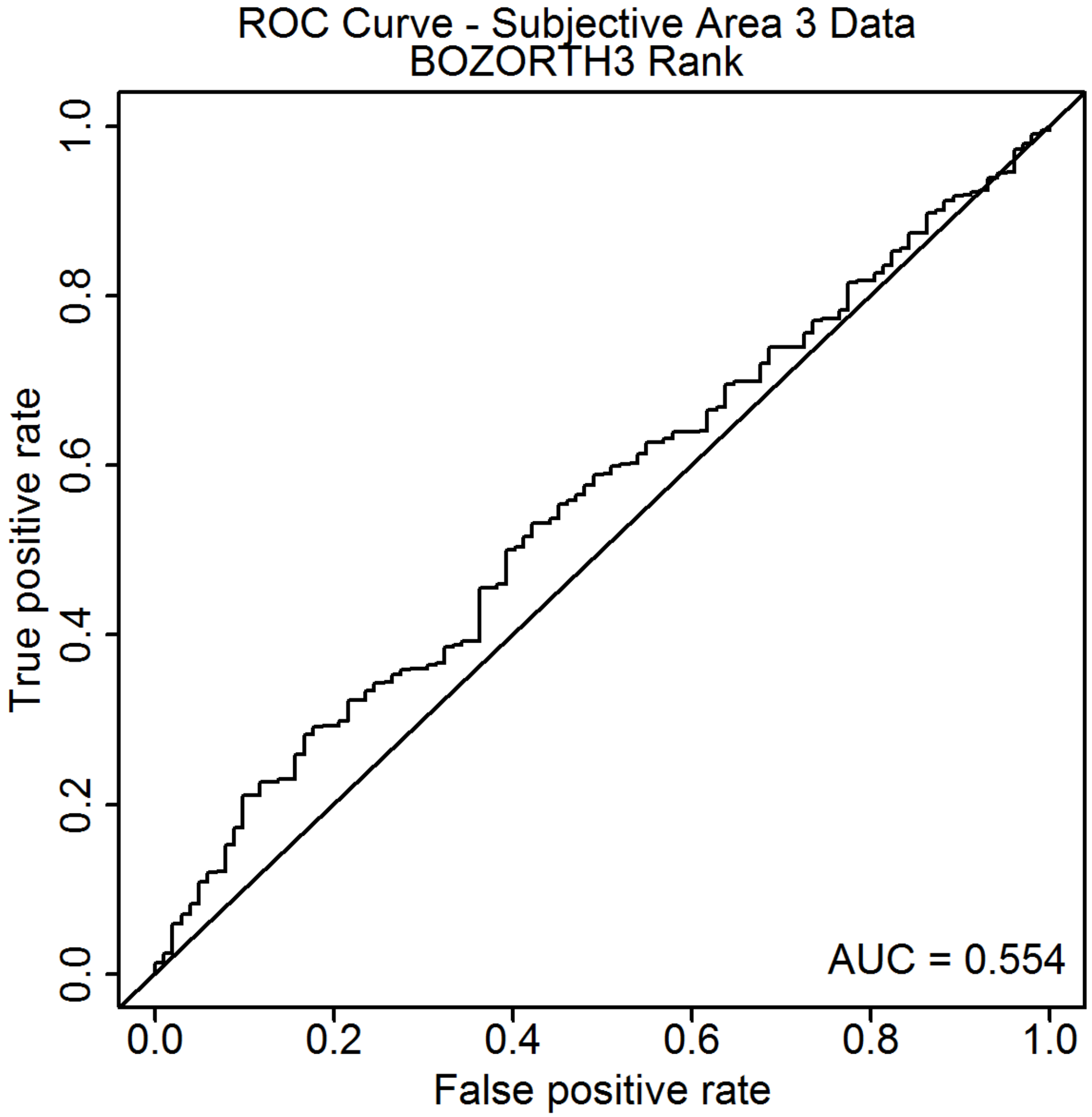


Figure E.37: This figure presents the ROC curve from the subjective Area 3 images for the ranks.

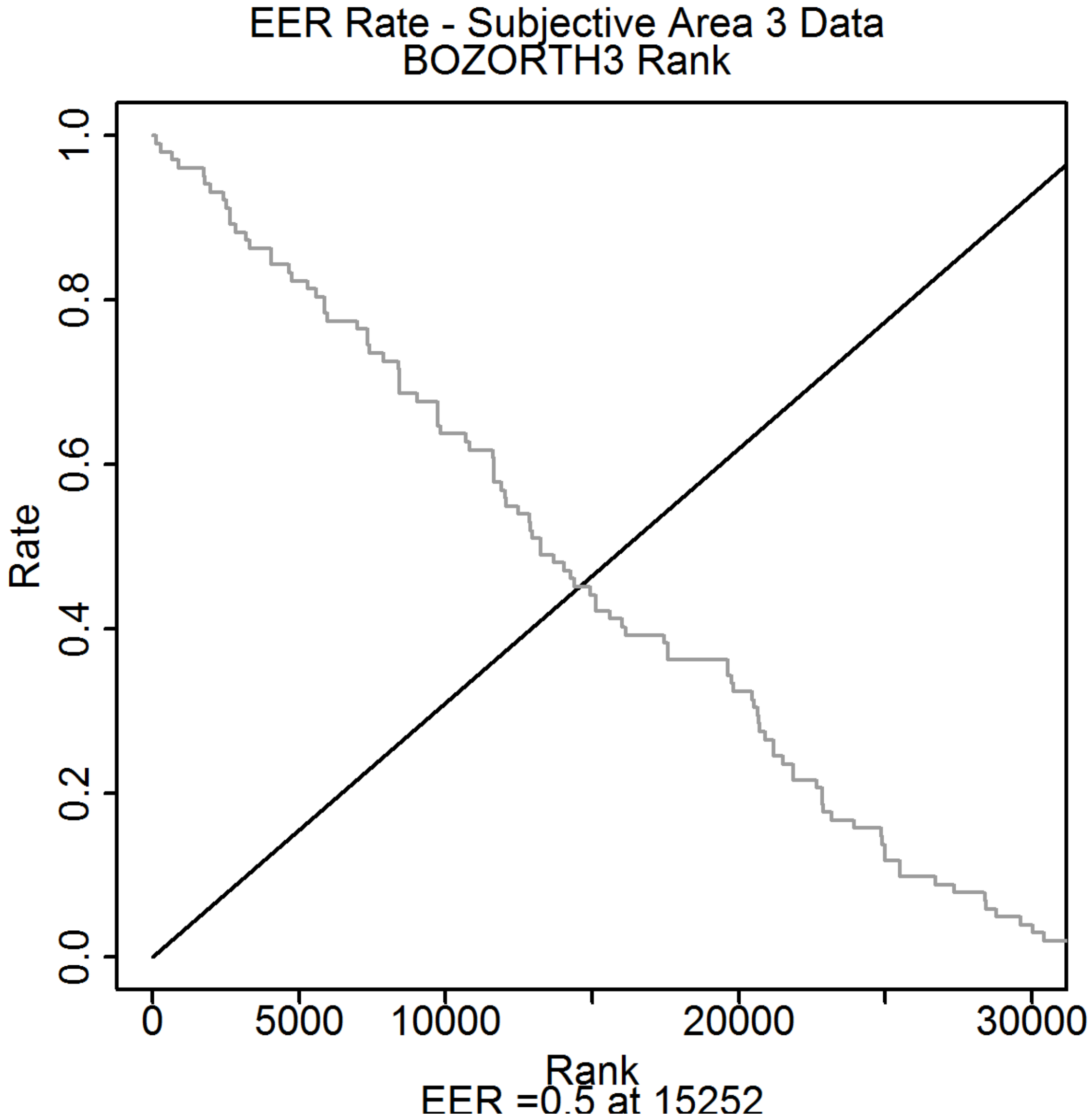


Figure E.38: This figure presents the EER plot from the subjective Area 3 images for the ranks.

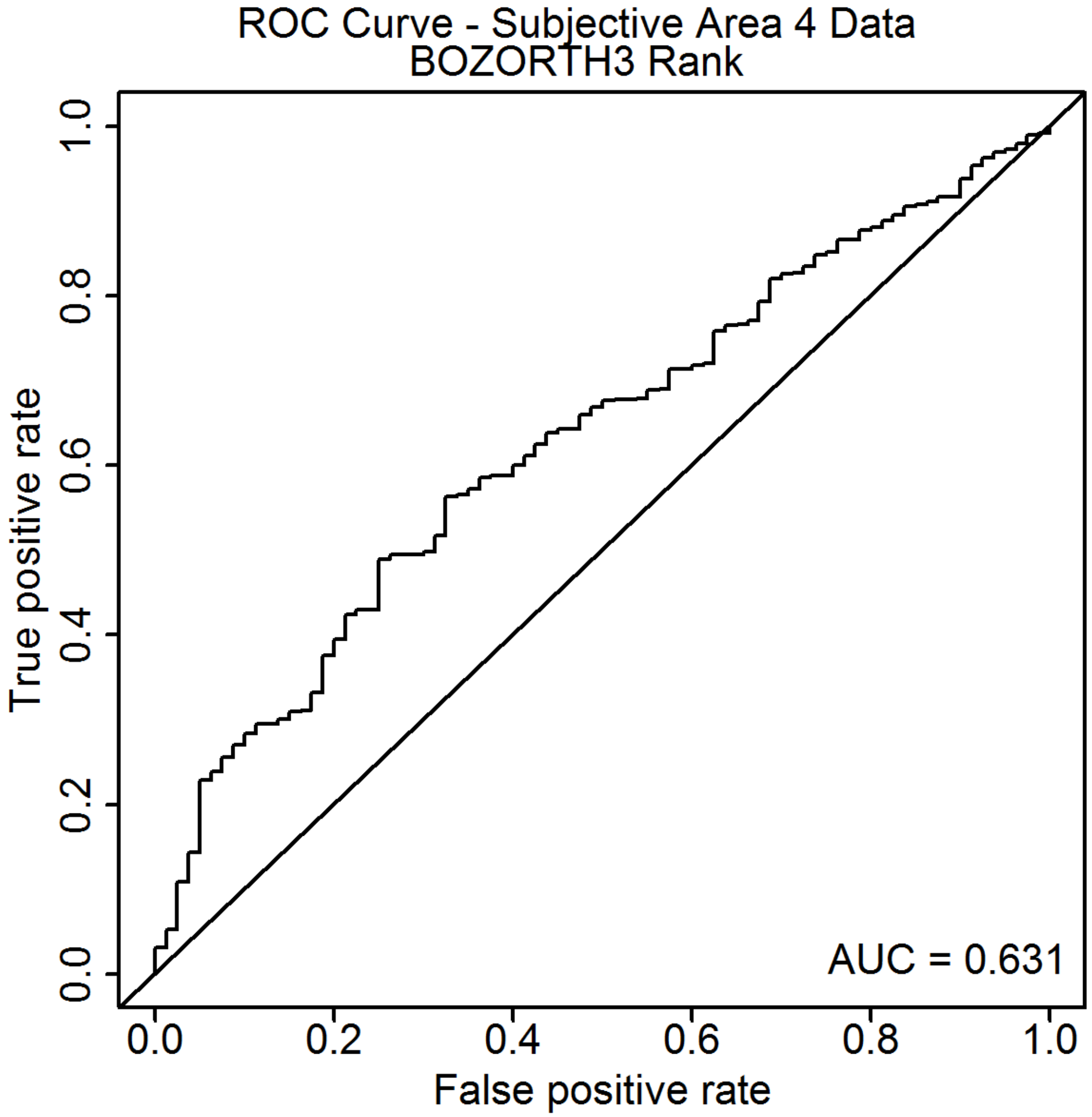


Figure E.39: This figure presents the ROC curve from the subjective Area 4 images for the ranks.

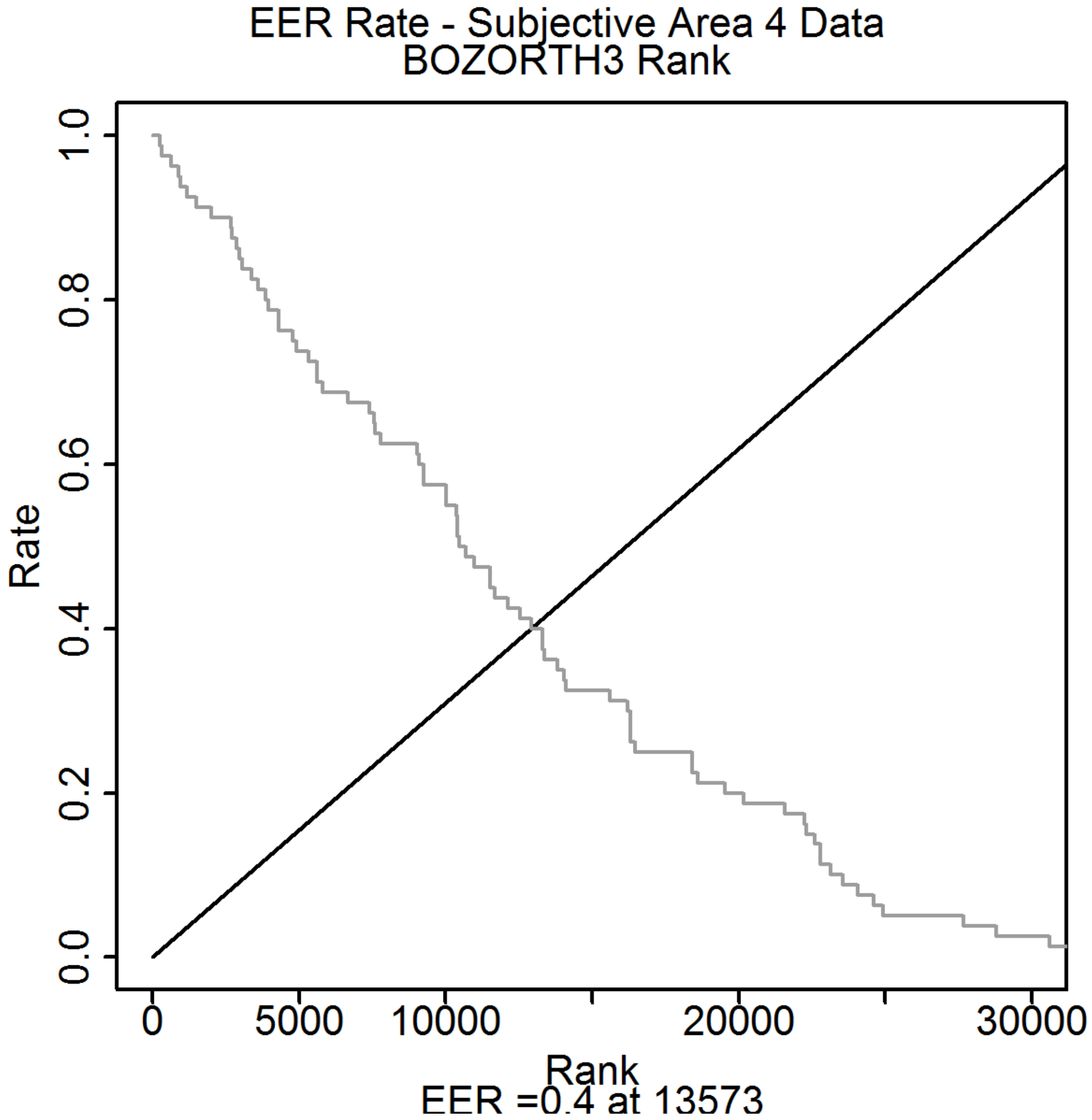


Figure E.40: This figure presents the EER plot from the subjective Area 4 images for the ranks.

E.3.2 Objective Method

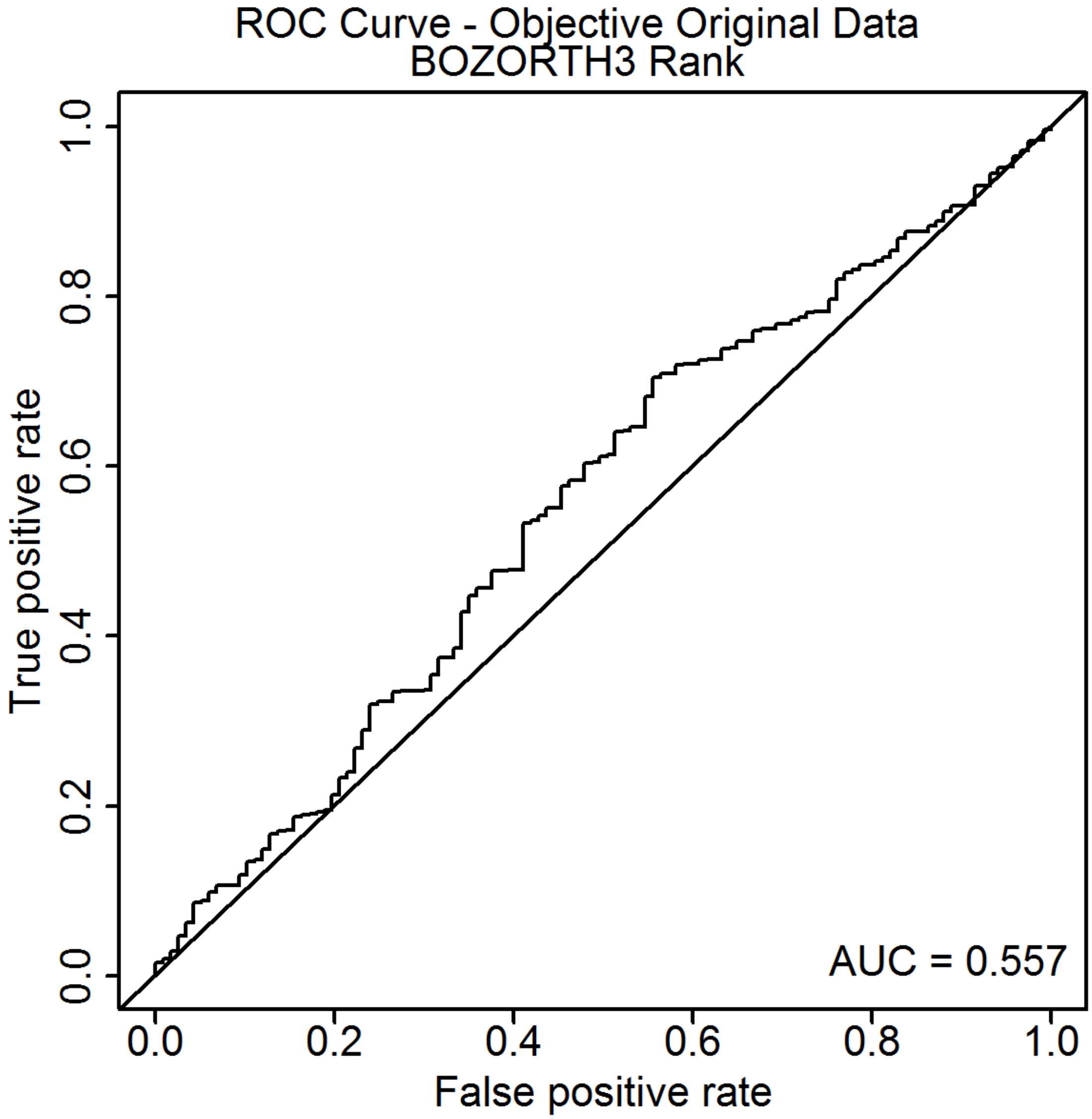


Figure E.41: This figure presents the ROC curve from the objective original images for the ranks.

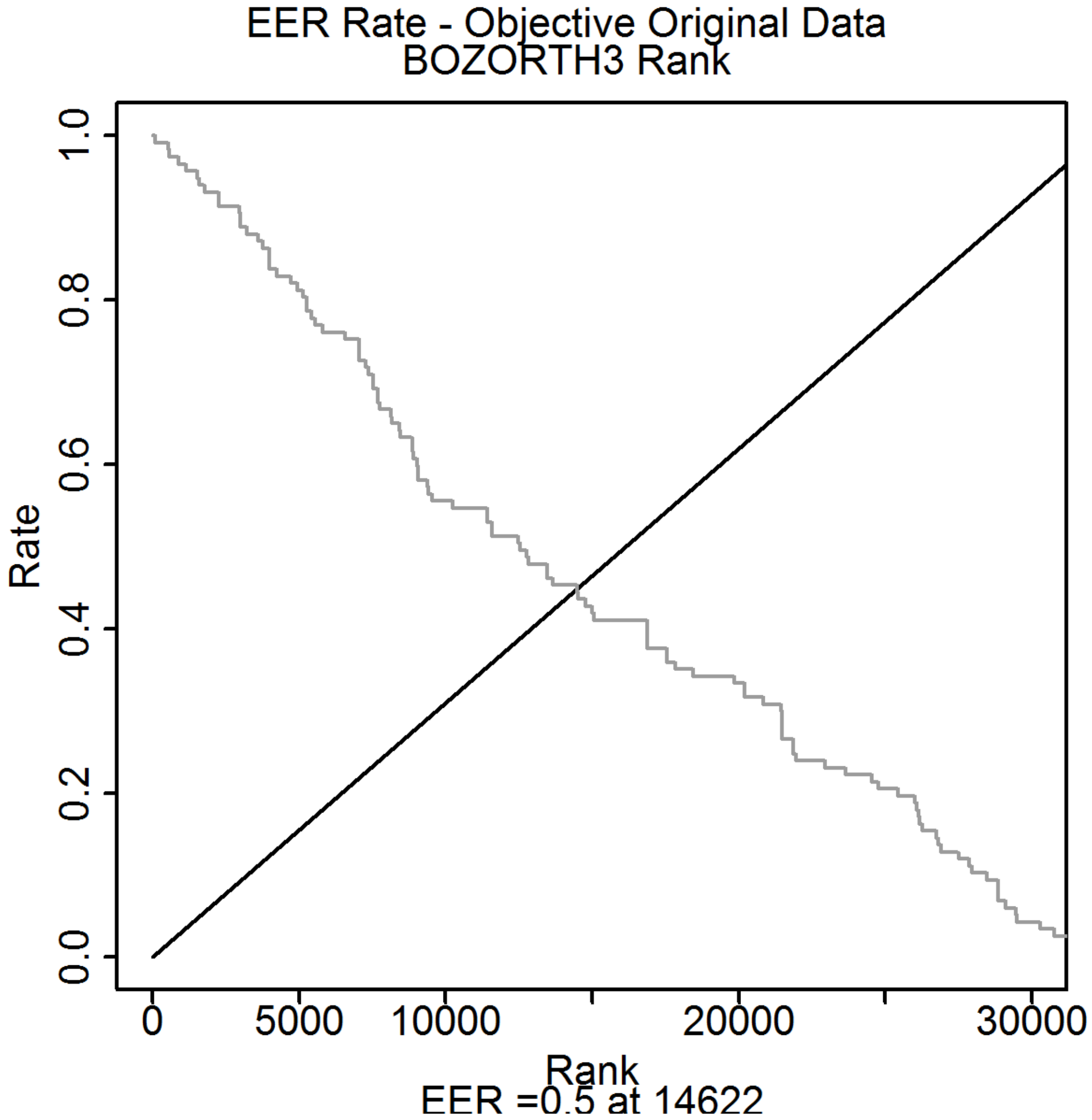


Figure E.42: This figure presents the EER plot from the objective original images for the ranks.

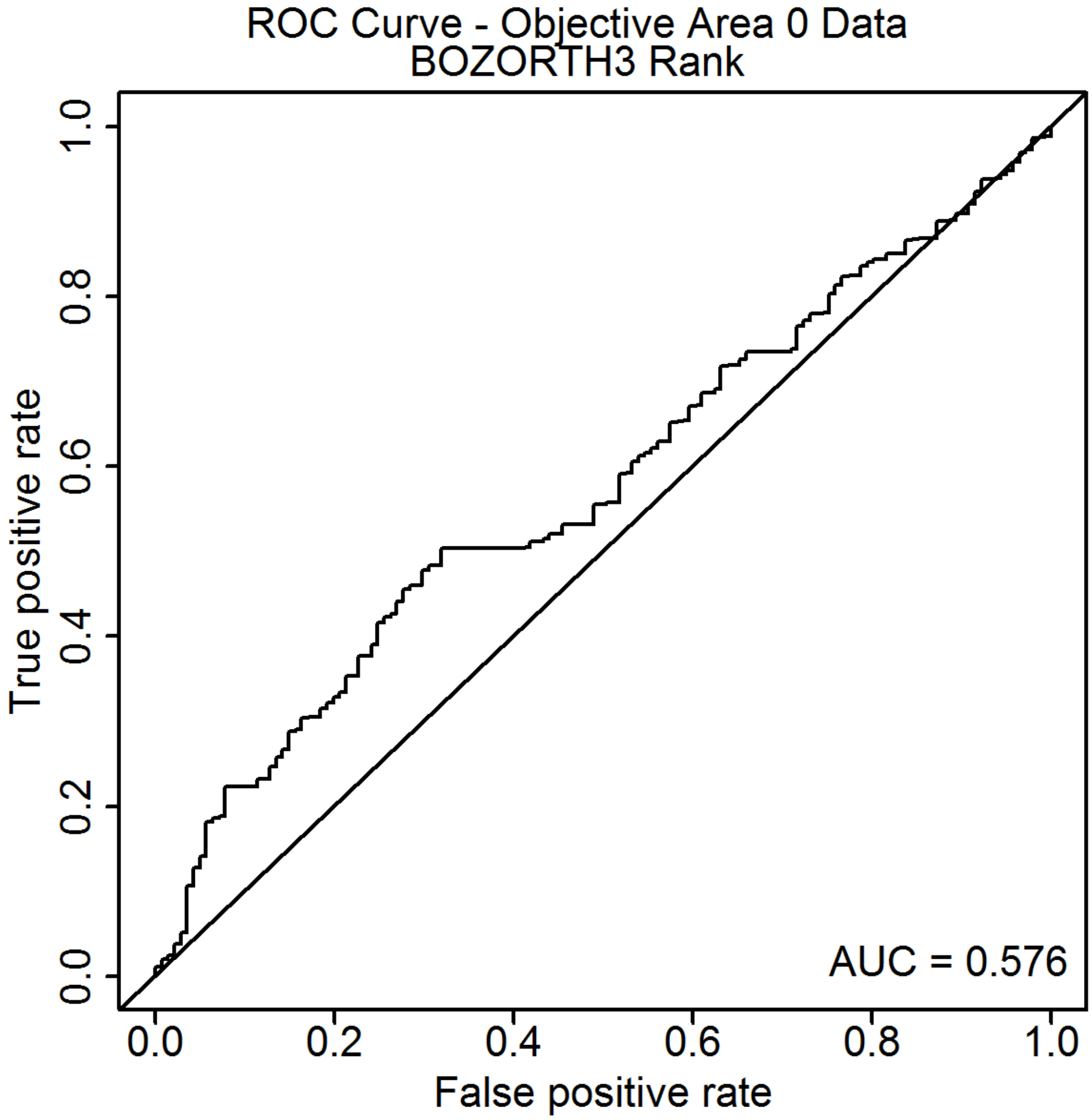


Figure E.43: This figure presents the ROC curve from the objective Area 0 images for the ranks.

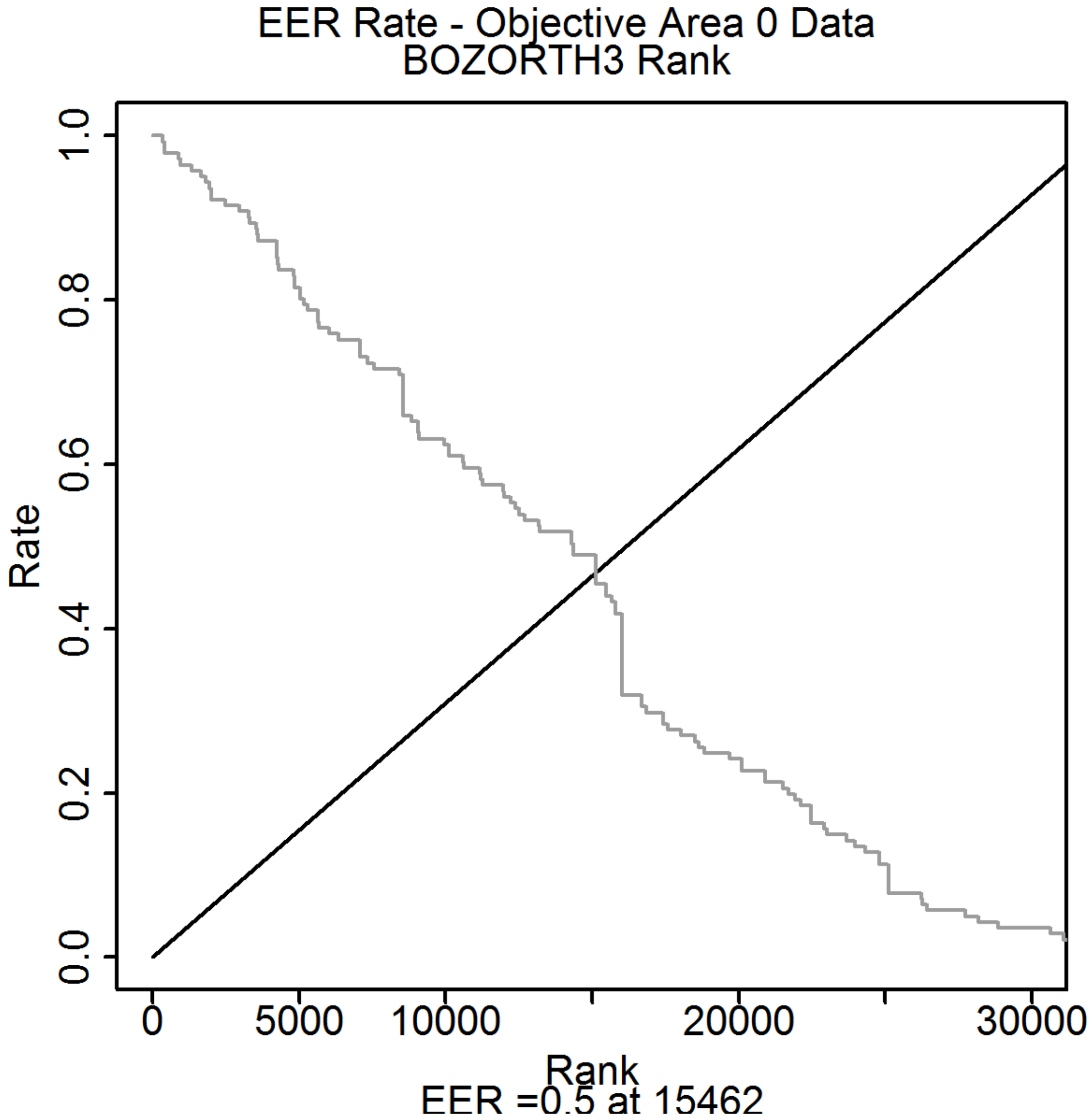


Figure E.44: This figure presents the EER plot from the objective Area 0 images for the ranks.

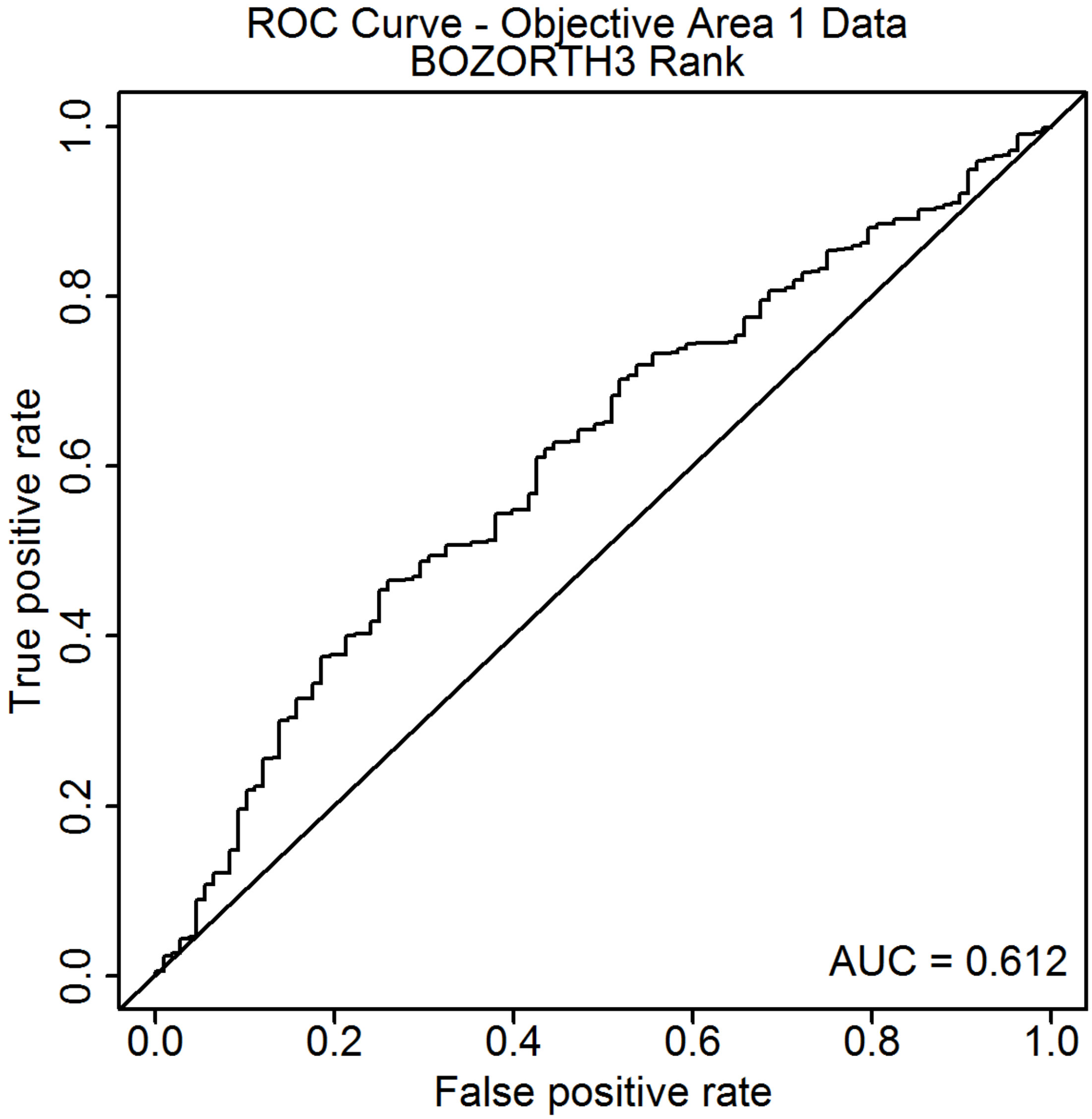


Figure E.45: This figure presents the ROC curve from the objective Area 1 images for the ranks.

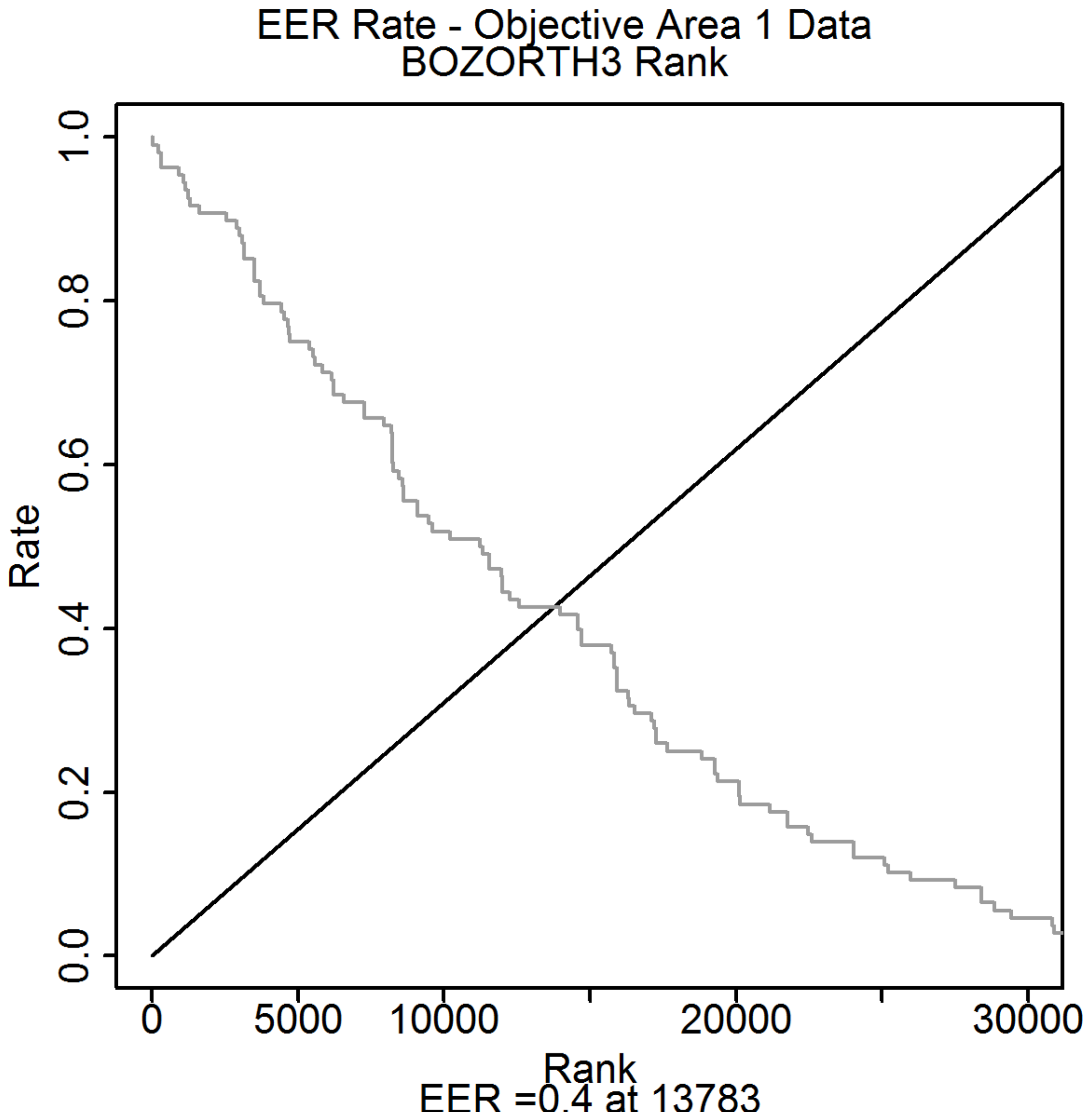


Figure E.46: This figure presents the EER plot from the objective Area 1 images for the ranks.

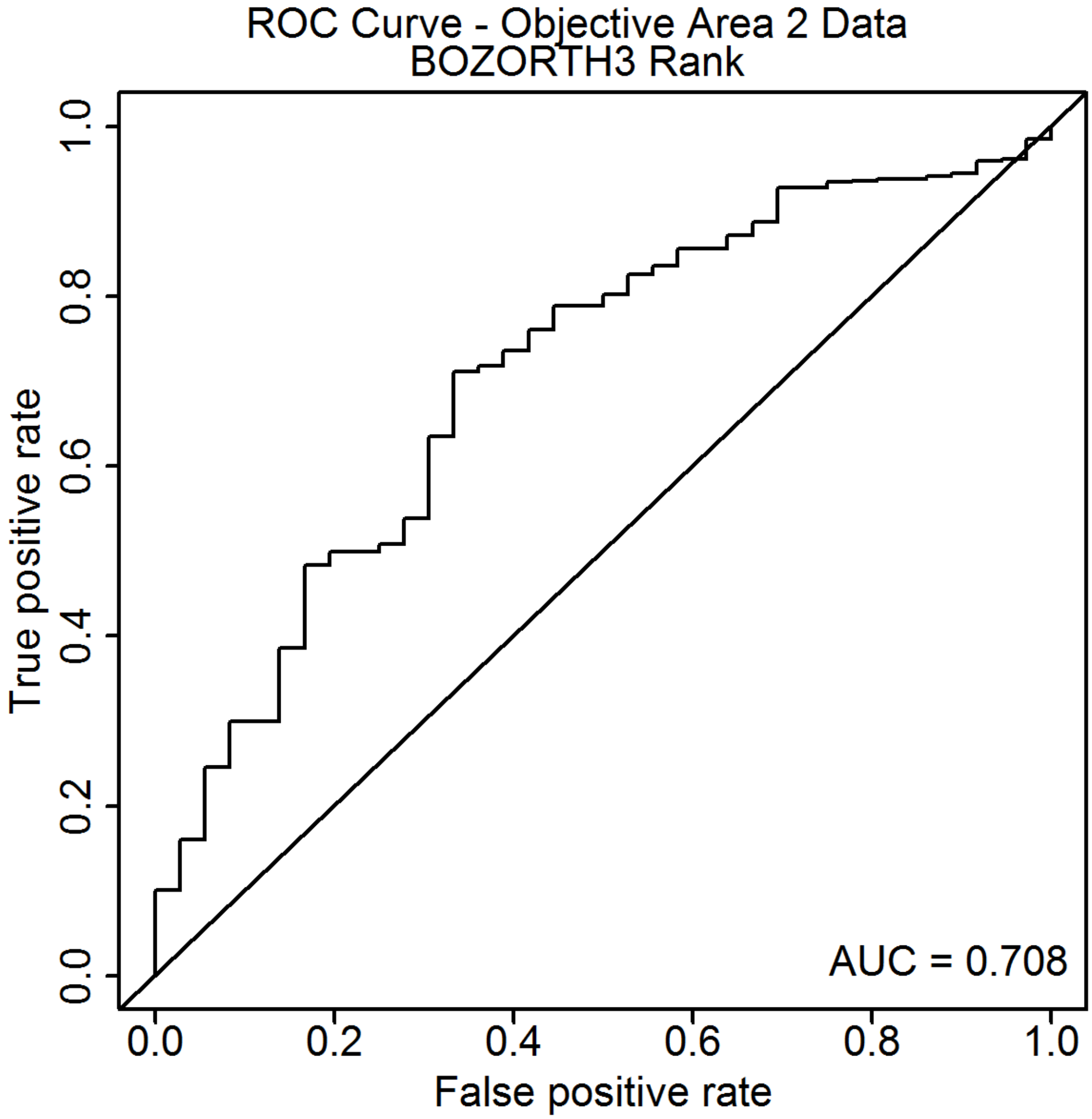


Figure E.47: This figure presents the ROC curve from the objective Area 2 images for the ranks.

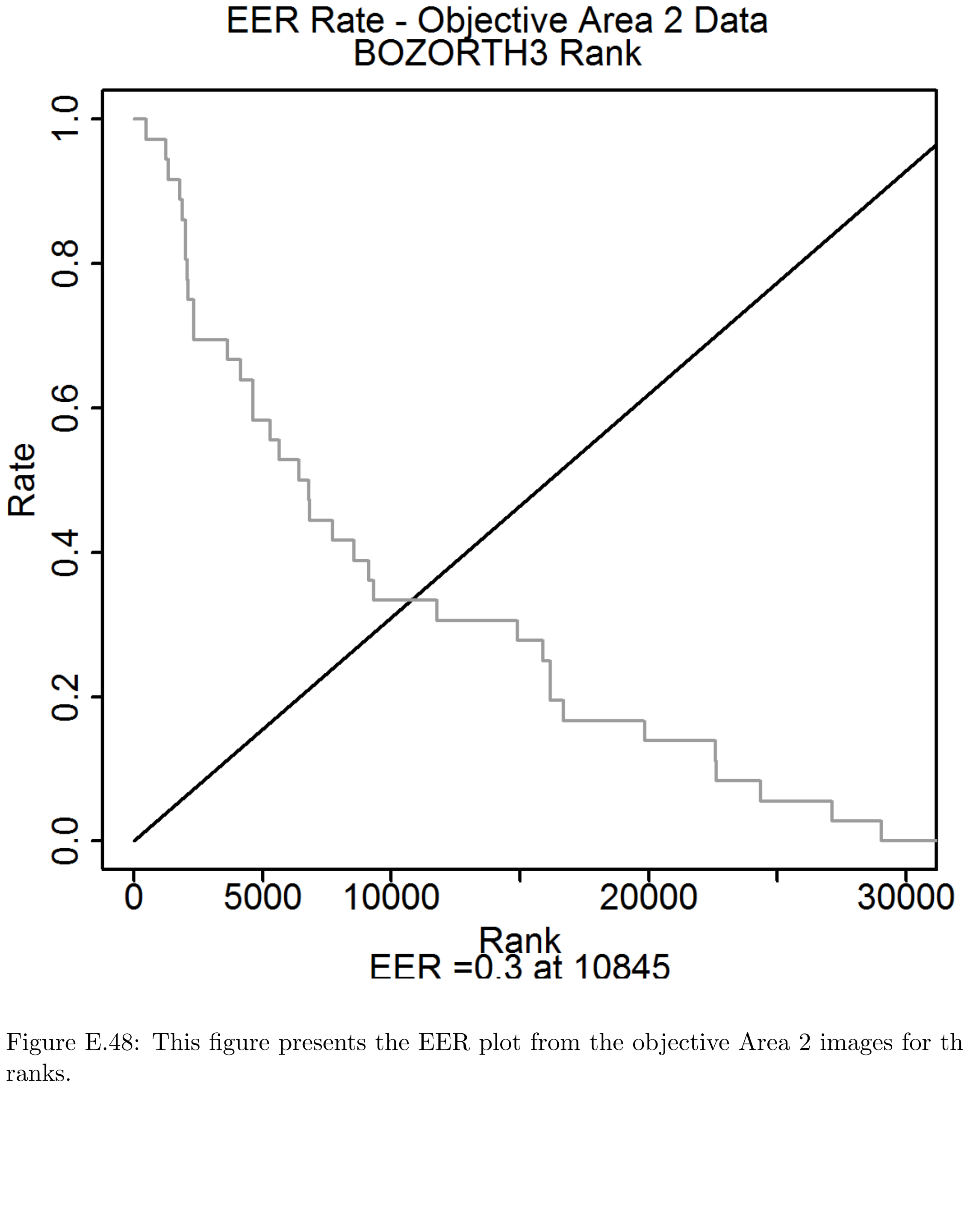


Figure E.48: This figure presents the EER plot from the objective Area 2 images for the ranks.

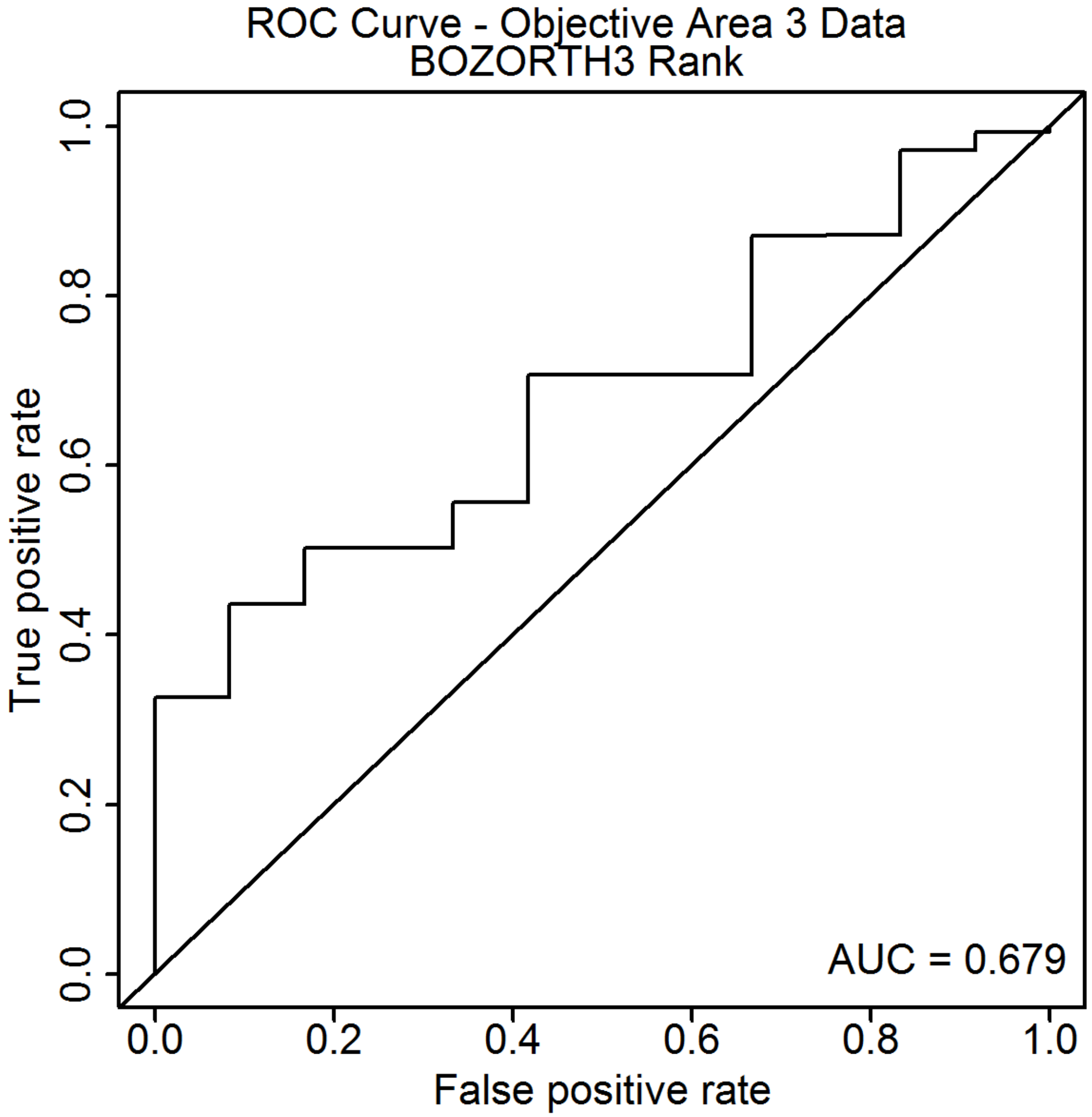


Figure E.49: This figure presents the ROC curve from the objective Area 3 images for the ranks.

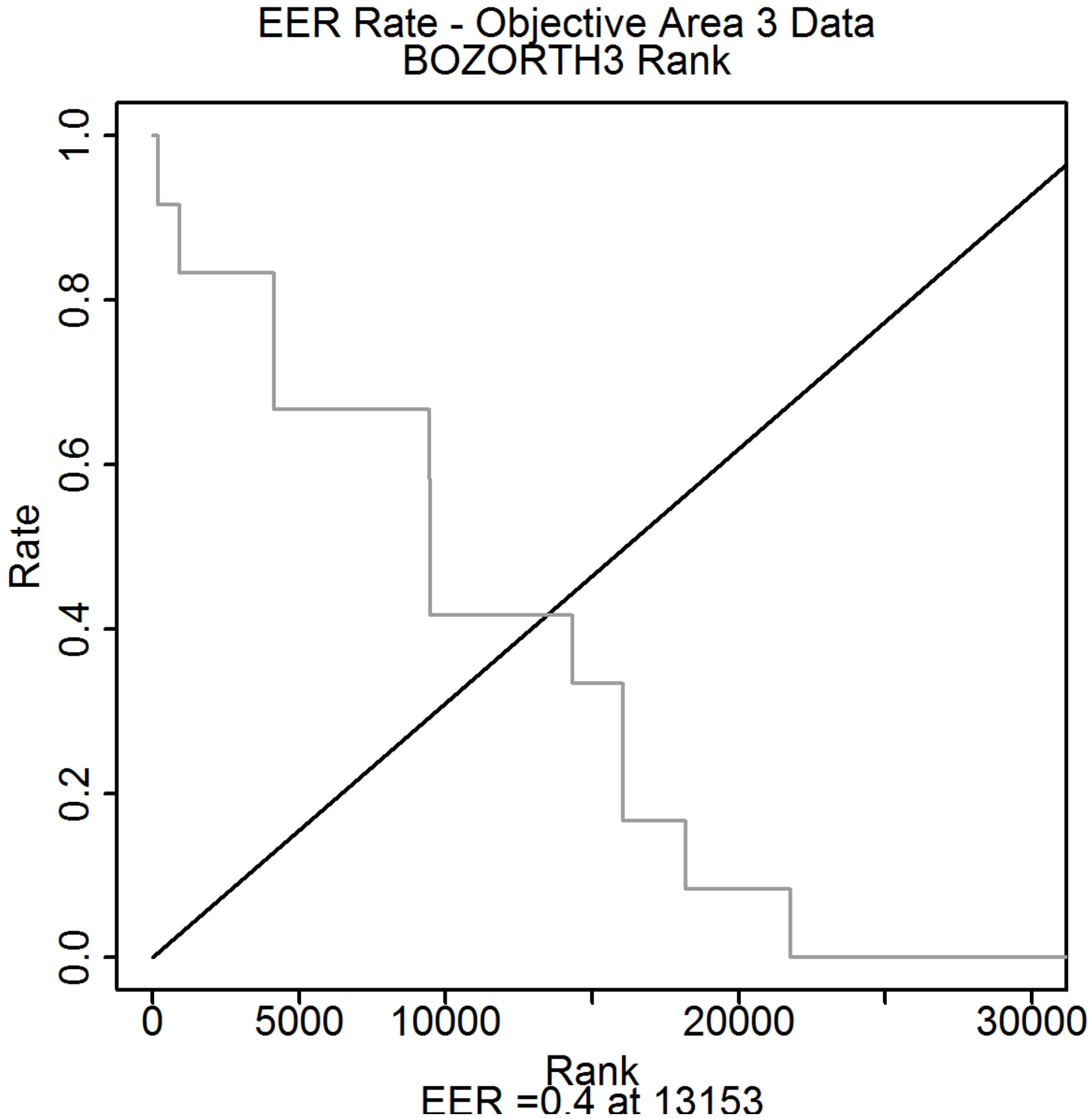


Figure E.50: This figure presents the EER plot from the objective Area 3 images for the ranks.

Bibliography

[1] E. Lim, X. Jiang, and W. Yau. Fingerprint quality and validity analysis. feb 2002.

[2] T.P. Chen, X. Jiang, and W. Y. Yau. Fingerprint image quality analysis. In *2004 International Conference on Image Processing, 2004.* Institute of Electrical & Electronics Engineers (IEEE), 2004. doi: 10.1109/icip.2004.1419725. URL `http://dx.doi.org/10.1109/ICIP.2004.1419725`.

[3] D.P. Pulsifer, S.A. Muhlberger, S.F. Williams, and R.C. Shaler. An objective fingerprint quality-grading system. *Forensic Science International*, 231:204–207, 2013.

[4] X. Jiang, W. Yau, and W. Ser. Detecting the fingerprint minutiae by adaptive tracing gray level ridge. *Pattern Recognition*, 34(5):999–1013, May 2001.

[5] C.I. Watson, M.D. Garri, E. Tabassi, C.L. Wilson, R.M. McCabe, S. Janet, and K. Ko. *User's Guide to Exported Controlled Distribution of NIST Biometric Image Software (NBIS-EC).* National Institute of Standards and Technology, sep 2004.

[6] C.I. Watson, M.D. Garri, E. Tabassi, C.L. Wilson, R.M. McCabe, S. Janet, and K. Ko. *User's Guide to NIST Biometric Image Software (NBIS).* National Institute Of Standards and Technology.

[7] B.J. Wing. *American National Standard for Information Systems- Data Format for the Interchange of Fingerprint, Facial and Other Biometric Information.* techreport, National Institute of Standards and Technology, U.S. Department of Commerce, 2012. NIST Special Publication 500-290.

[8] M. R. Hawthorne. *Fingerprints:Analysis and Understanding.* CRC Press 2008, 2008.

[9] C. Champod, C.J. Lennard, P. Margot, and M. Stoilovic. *Fingerprints and Other Ridge Skin Impressions, Second Edition.* CRC Press 2016, 2016.

[10] V. Galloway and D. Charlton. *Forensic Human Identification: An Introduction*, chapter Fingerprints, pages 56–72. CRC Press 2006, 2006.

[11] D. R. Ashbaugh. *Quantitative-Qualitative Friction Ridge Analysis: An Introduction to Basic and Advanced Ridgeology.* CRC Press 1999, 1999.

[12] Scientific Working Group on Friction Ridge Analysis, Study and Technology. Standards for examining friction ridge impressions and resulting conclusions, 2011.

[13] E. Kukula, S. Elliott, H. Kim, and C. San Martin. The impact of fingerprint force on image quality and the detection of minutiae. In *2007 IEEE International Conference on Electro/Information Technology*. Institute of Electrical & Electronics Engineers (IEEE), May 2007. doi: 10.1109/eit.2007.4374512. URL `http://dx.doi.org/10.1109/EIT.2007.4374512`.

[14] R.A. Hicklin, J. Buscaglia, and M.A. Roberts. Assessing the clarity of friction ridge impressions. *Forensic Science International*, 226:106–117, 2013.

[15] K. Moses. *Fingerprint Sourcebook*, chapter Automated Fingerprint Identification System (AFIS), pages 1–33. National Criminal Justice Reference Service, 2010.

[16] P. Komarinski. *Cold Case Homicides Practical Investigative Techniques*, chapter Automated Fingerprint Identification Systems. CRC Press 2006, 2006.

[17] F. Alonso-Fernandez, J. Fierrez, J. Ortega-Garcia, J. Gonzalez-Rodriguez, H. Fronthaler, K. Kollreider, and J. Bigun. A comparative study of fingerprint image-quality estimation methods. *IEEE Transactions on Information Forensics and Security*, 2 (4):734–743, dec 2007.

[18] The GIMP Team. GIMP GNU Image Manipulation Program. Online. URL `https://www.gimp.org/`.

[19] Wolfram Mathematica. Online. URL `https://www.wolfram.com/mathematica/`.

[20] Z. Yao, J.-M. Le Bars, C. Charrier, and C. Rosenberger. Fingerprint quality assessment with multiple segmentation. In *2015 International Conference on Cyberworlds (CW)*. Institute of Electrical & Electronics Engineers (IEEE), oct 2015. doi: 10.1109/cw.2015.18. URL `http://dx.doi.org/10.1109/CW.2015.18`.

[21] J. Wu, S.J Xie, D.-H. Sei, and W.D Lee. A new approach for classification of fingerprint image quality. In Zhang-D. Latombe J.-C. Wang, Y. and W. Kinsner, editors, *Cognitive Informatics, 2008. ICCI 2008. 7th IEEE International Conference.*, pages 375–383. IEEE, IEEE, aug 2008. doi: 10.1109/COGINF.2008.4639191. URL `http://dx.doi.org/10.1109/COGINF.2008.4639191`.

[22] Stephan Saalfeld. Enhance local contrast (CLAHE) (Fiji). Online, October 2015. URL `https://imagej.net/Enhance_Local_Contrast_(CLAHE)`.

[23] Federal Bureau of Investigation. Standard fingerprint form (fd-258). Online.

[24] Simon Barthelme. Package ‘imager’. Online, April 2017. URL `https://cran.r-project.org/web/packages/imager/imager.pdf`.

[25] S.H. Park, J.M. Goo, and C-H. Jo. Receiver operating characteristic (ROC) curve: Practical review for radiologist. *Korean Journal of Radiology*, 5(1):11–18, March 2004.

[26] A. Nagar, H. Choi, and A. K. Jain. Evidential value of automated latent fingerprint comparison: An empirical approach. *IEEE Transactions on Information Forensics and Security*, 7(6):1752–1765, dec 2012.

[27] J.C. Wu, A.F. Martin, and R.N. Kacker. Measures, uncertainties, and significance test in operational ROC analysis. *Journal of Research of the National Institute of Standards and Technology*, 116(1):517–537, January 2011.

[28] Ted Dunstone and Neil Yager. *Biometric System and Data Analysis: Design, Evaulation and Data Mining.* Springer, 2009.

[29] G.W. Corder and D.I. Foreman. *Nonparametric Statistics for Non-Statisticians: A Step-By-Step Approach.* John Wiley & Sons, Inc., 2009.

[30] W.J. Conover. *Practical Nonparametric Statistics.* John Wiley & Sons, Inc., 3rd edition, 1999.

[31] Brian DeCann and Arun Ross. Can a “poor” verification system be a “good” identification system? a preliminary study. *IEEE International Workshop on Information Forensics and Security (WIFS)*, December 2012.