JDSSV

Journal of Data Science, Statistics, and Visualisation

MMMMMM YYYY, Volume VV, Issue II. *doi: XX.XXXXX/jdssv.v000.i00*

Automatic Matching of Cartridge Case Impressions

Joseph Zemmels Iowa State University

Susan VanderPlas University of Nebraska - Lincoln

Heike Hofmann Iowa State University

Abstract

Forensic examinations attempt to solve the binary classification problem of whether two pieces of evidence originated from the same source. For example, a cartridge case found at a crime scene may be compared to a cartridge case fired from a suspect's firearm. Forensic examiners traditionally rely on high-powered comparison microscopes, case facts, and their own experience to arrive at a source conclusion. Recently, algorithms that provide an automatic and objective measure of similarity of the evidence have become more prevalent. We introduce the Automatic Cartridge Evidence Scoring (ACES) algorithm that encompasses preprocessing, feature extraction, and similarity scoring. Our primary contribution is a set of features used to classify whether two cartridge cases were fired from the same firearm. We use a train/test split on a data set of 510 cartridge case scans to fit and validate random forest and logistic regression models. We demonstrate that these models are more accurate than predominant algorithms on our data set.

Keywords: forensics, forensic statistics, pattern recognition, firearms and toolmarks, R, random forest, cross-validation, classification.

1. Introduction

A *cartridge case* is the part of firearm ammunition that houses the projectile and propulsive device. When a firearm is discharged and the projectile travels down the

barrel, the cartridge case moves in the opposite direction and slams against the back wall, the *breech face*, of the firearm. Markings on the breech face are "stamped" into the surface of the cartridge case leaving so-called *breech face impressions*.

In a traditional examination, forensic examiners use these impressions analogous to a fingerprint to determine whether two cartridge cases were fired from the same firearm. The top of Figure 1 illustrates this procedure (Xiao Hui Tai 2018; Zheng et al. 2014; Vorburger et al. 2015). First, two cartridge cases are collected - perhaps one is from a crime scene and the other is collected from a suspect's gun. An examiner places the two cartridge cases beneath a "comparison microscope" that merges the views of two compound microscopes into a single split view, similar to the side-by-side cartridge case image in Figure 1. The examiner assesses the degree of similarity between the markings on the cartridge cases and chooses one of four conclusions (AFTE Criteria for Identification Committee 1992):

1. **Identification:** cartridge cases were fired from the same firearm
2. **Elimination:** cartridge cases were not fired from the same firearm
3. **Inconclusive:** the evidence is insufficient to make an identification or elimination
4. **Unsuitable:** the evidence is unsuitable for examination

Critics of traditional forensic examinations cite a lack of "foundational validity" underlying the procedures used by firearm and toolmark examiners (National Research Council 2009; PCAST 2016). In particular, examiners rely largely on their subjective findings rather than on a well-defined procedure to measure similarity. PCAST (2016) pushed for "developing and testing image-analysis algorithms" to objectively measure the similarity between cartridge cases. An automatic comparison algorithm could supplement, inform, or dictate an examiner's conclusion (Swofford and Champod 2021).

We introduce a novel Automatic Cartridge Evidence Scoring (ACES) algorithm to objectively compare cartridge case evidence based on their breech face impressions. Our algorithm encompasses all stages of the comparison procedure after collecting a scan of the cartridge case surface including preprocessing, comparing, and scoring. Our ACES algorithm is available open-source as part of the scored R package.

In the following sections, we first review recently proposed algorithms to compare firearm evidence. We then discuss the data collection procedure to obtain 510 cartridge scans used in training and validating the ACES algorithm. To our knowledge, this is the largest published study of a cartridge case scoring algorithm to-date, with the next largest analyzing four different data sets totaling 195 cartridge cases (Chen et al. 2017). After describing the ACES algorithm, we present summary results from training and testing two binary classifier models: based on a random forest and logistic regression. We discuss the strengths and weaknesses of the two classifier models and compare the relative importance of the ACES features. We also argue that the ACES algorithm combines the classification rules of previously proposed cartridge case comparison algorithms while incorporating additional nuance. We conclude with our

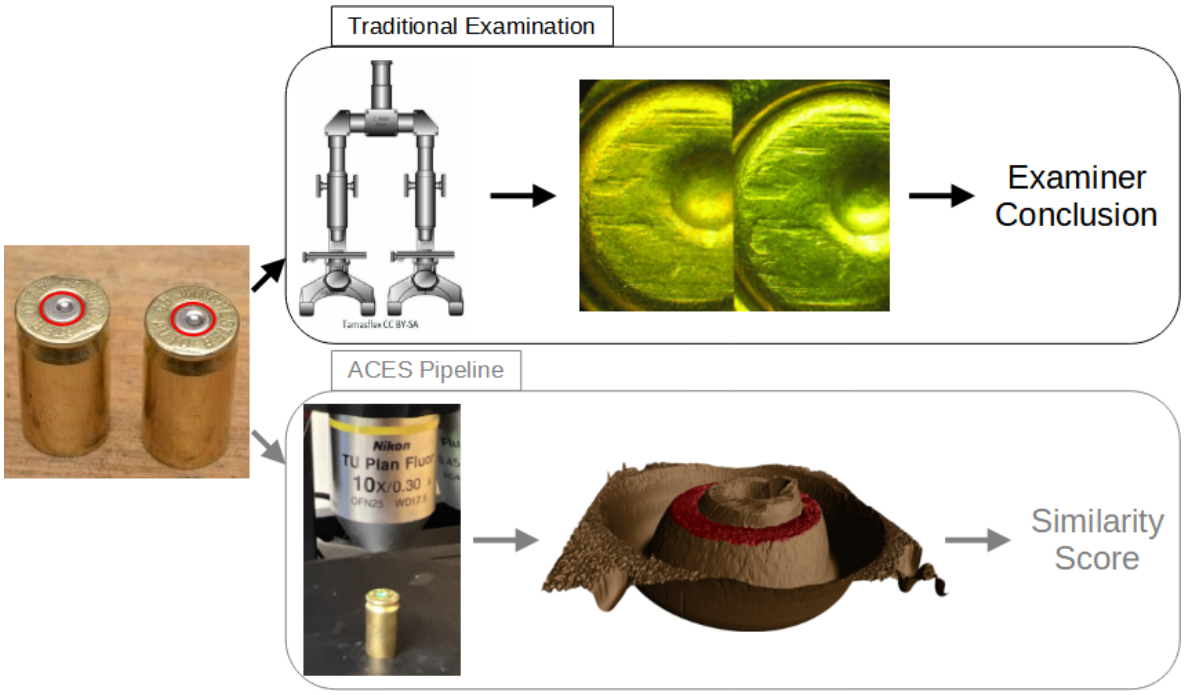

Figure 1: Comparison of the traditional examination vs. the currently proposed method for comparing cartridge cases. Both start with two fired cartridge cases. In traditional examination, an examiner uses a microscope to assess the "agreement" of markings on the two cartridge case surfaces. They decide whether or not the cartridge cases were fired from the same firearm, or if there is inconclusive evidence to decide. In the ACES algorithm, we take a topographical scan of the cartridge case surfaces and manually identify the regions containing distinguishable markings (shown in red). We pass these scans to the ACES algorithm, which processes and compares the two scans. The final result is a numerical measure of similarity of the two cartridge cases.

thoughts on how cartridge case comparison algorithms should be developed, validated, and shared going forward.

1.1. Previous Work

Many recent proposals for automatic cartridge case scoring algorithms borrow from image processing and computer vision techniques. For example, Vorburger et al. (2007) proposed using the cross-correlation function (CCF) to compare images or scans of cartridge case surfaces. The CCF measures the similarity between two matrices for all possible translations of one matrix against the other. Calculating the CCF while rotating one of the scans therefore allows for estimation of the optimal translation and rotation, together referred to as the *registration*, between the two scans; simply choose the rotation/translation at which the CCF is maximized. Hare et al. (2017) used the CCF, among other features, to compare scans of bullets. Tai and Eddy (2018) developed an open-source cartridge case comparison pipeline that compared cartridge case images using the CCF.

Song (2013) noted that two matching cartridge cases often share similar impressions in specific regions, so calculating the CCF between two full scans may not highlight their similarities. Instead, Song (2013) proposed partitioning one cartridge case scan into a grid of "cells" and calculating the CCF between each cell and the other scan. If two cartridge cases are truly matching, then the maximum CCF value between each cell and the other scan, particularly the cells containing distinguishable breech face impressions, should be relatively large. Furthermore, the cells should "agree" on the registration at which the CCF is maximized. Song (2013) outlined the "Congruent Matching Cells" algorithm to determine the number of cells that agree on a particular registration. A cell is classified as a Congruent Matching Cell (CMC) if its estimated registration is within some threshold of the median registration across all cells and its CCF value is above some threshold. A number of follow-up papers proposed alterations to the the original CMC method (Tong et al. 2015; Chen et al. 2017). Zemmels et al. (2022) introduced an open-source implementation of the CMC method in the cmcR R package. As an alternative to defining Congruent Matching Cells, Zhang et al. (2021) proposed using a clustering algorithm from Ester et al. (1996) to determine the number of cells in agreement on a specific registration.

Currently, there is no rigorous procedure for comparing different cartridge case comparison algorithms. This includes selecting optimal parameters for a specific algorithm. Zemmels et al. (2023) proposed an optimization criterion to select parameters for the CMC algorithm. Analogously, Hare et al. (2017) developed a validation procedure to select parameters for a bullet comparison algorithm. In this work, we introduce a novel cross-validation procedure to learn and test optimal parameters for the ACES algorithm.

2. Cartridge Case Data

We use 510 cartridge cases collected as part of a study by Baldwin et al. (2014). The authors of the original study fired 800 Remington 9mm pistol cartridge cases from each of 25 new Ruger SR9, 9mm Luger centerfire pistols.. They separated the collected car-

tridge cases into 15 sets of four to be sent to each of 218 forensic examiner participants. Each set of four consisted of three cartridge cases labeled as originating from the same firearm, the "known-match" cartridge cases. Participants performed an examination to determine whether a fourth "questioned" cartridge case shared a common source with the known-match triplet (or whether the evidence was inconclusive).

Across all 218 examiners, the true positive rate - proportion of correctly classified matching sets - was reported to be 99.6%. The reported true negative rate - the proportion of correctly classified non-matching sets - was 65.2%. The discrepancy between the true positive and true negative rates can be partially explained by the number of "inconclusive" decisions made by the examiners. Examiners reach an inconclusive decision when there is some agreement or disagreement in the characteristics between two cartridge cases, but not enough to make a match or non-match conclusion (AFTE Criteria for Identification Committee 1992). Roughly one out of five comparisons, 22.9%, were reported as inconclusive. The vast majority, 98.5%, of these inconclusives were truly non-matching comparisons, which justifies the true negative rate of 65.2%. There has recently been some debate about how to incorporate inconclusive decisions into accuracy/error rate estimation (Hofmann et al. 2021), so we do not report an overall accuracy here.

We scanned the 510 cartridge cases using the Cadre^{TM} 3D-TopMatch High Capacity Scanner. Briefly, this scanner collects images under various lighting conditions of a gel pad into which the base of a cartridge case is impressed. Proprietary software that accompanies this scanner combines these images into a 2D array called a *surface matrix*. The elements of a surface matrix represent the relative height values of the associated cartridge case surface. This surface matrix, along with metadata concerning parameters under which the scan was taken (dimension, resolution, author, etc.), are stored in the ISO standard XML 3D Surface Profile (`x3p`) file type (ISO 25178-72:2017 2017).

As discussed in the next section, our design differs from that used in Baldwin et al. (2014). Rather than basing error rates on the comparison of three known-match cartridge cases to one questioned cartridge case (3 to 1), we consider the classification error rate of pairwise comparisons (1 to 1). Further, we split the 510 cartridge cases by randomly selecting 10 of the 25 firearms for training and use the remaining 15 firearms for testing. This resulted in a training set of 210 cartridge cases, $\binom{210}{2} = 21,945$ pairwise comparisons, and a testing set of 300 cartridge cases, $\binom{300}{2} = 44,850$ pairwise comparisons.

[cite eventual DFSC data-in-brief or ISU datashare repo?]

3. Methods

We now discuss the methods behind the Automatic Cartridge Evidence Scoring (ACES) algorithm. We divide the methods into three stages:

1. **Preprocessing:** prepare cartridge case scans for comparison
2. **Comparing:** compare two cartridge cases and compute similarity features

3. **Scoring:** measure the similarity between the two cartridge cases using a trained classifier

The following sections detail each of these stages. Throughout, we treat "surface matrix" and "scan" synonymously.

The bottom of Figure 1 shows a summary of our procedure. After taking a topographical scan of the cartridge case surfaces, we manually annotate the breech face impression region (shown in red). ACES automatically preprocesses and compares the scans resulting in a similarity score, either a binary classification or class probability, derived from a classifier model.

3.1. Preprocessing

We first use the open-source FiX3P web application cite Talen Fisher to manually annotate the breech face impression region. An example of a manually-annotated cartridge case scan is shown in Figure 1. The FiX3P software includes functionality to "paint" the surface of a cartridge case using a computer cursor and save the painted regions to a *mask*. A mask is a 2D array of hexidecimal color values of the same dimension as its associated surface matrix. When initialized, every element of a mask is a shade of brown (#cd7f32) by default. Any elements painted over by the user will be replaced with the user's selected color value. In Figure 1, the breech face impression region was manually annotated using a shade of red (#ff0000).

We preprocess the raw scans by applying a sequence of functions available in the R packages x3ptools (Hofmann et al. 2022) and cmcR (Zemmels et al. 2022). Figure 2 shows the effect that each function has on the scan surface values. Gray pixels in each plot represent missing values in the surface matrix. The `x3p_delete` function removes values in the scan based on the associated mask. Next, the `preProcess_removeTrend` function subtracts a fitted conditional median plane from the surface values to "level-out" any global tilt in the scan. The `preProcess_gaussFilter()` function applies a bandpass Gaussian filter to remove small-scale noise and other large-scale structure, which better highlights the medium-scale breech face impressions. Finally, the `preProcess_erode()` function applies the morphological operation of erosion on the edge of the non-missing surface values (Haralick et al. 1987). This has the effect of shaving off values on the interior and exterior edge of the surface, which are often extreme "roll-off" values that unduly affect the comparing stage if not removed. The final result is a cartridge case surface matrix with emphasized breech face impressions.

Next, we compute a set of similarity features for two preprocessed cartridge case scans.

3.2. Comparing

In this section, we introduce a set of similarity features for two cartridge case scans. We calculate features at two scales: between two full scans and between individual cells. Analogous to how a forensic examiner uses a comparison microscope with different magnification levels, this allows us to assess the similarity between two scans at the macro and micro levels.

Notational Conventions

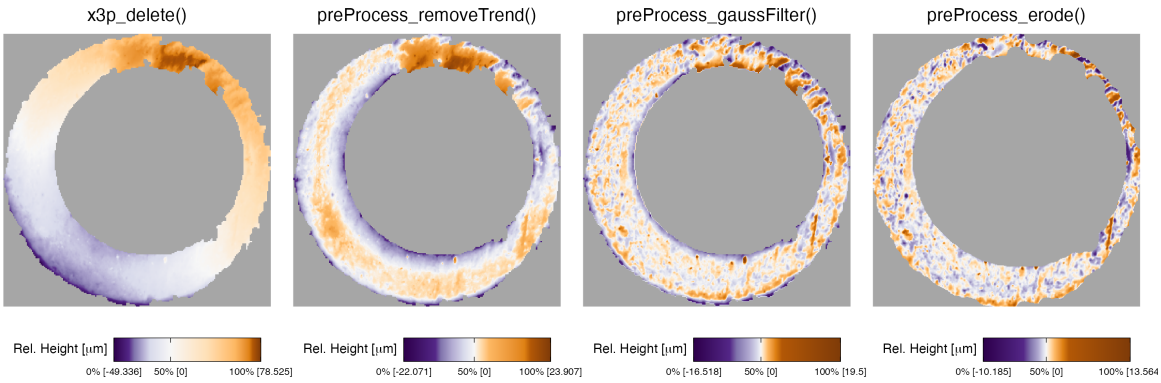

Figure 2: We apply a sequence of preprocessing functions to each scan. Each preprocessing step further emphasizes the breech face impressions in the scan.

First, we introduce notation that will be used to define the features. Let A and B denote two surfaces matrices that we wish to compare. For simplicity, we assume that $A, B \in \mathbb{R}^{k \times k}$ for $k > 0$.[1] We use lowercase letters and subscripts to denote a particular value of a matrix: a_{ij} is the value in the i-th row and j-th column, starting from the top-left corner, of matrix A. We refer to the two known-match cartridge cases in Figure 3 as exemplar matrices A and B.

To accommodate structurally missing values, we adapt standard matrix algebra by extending the space of real values by an element encoding 'missingness' as follows: if an element of either matrix A or B is missing, then any element-wise operation including this element is also missing. Standard matrix algebra holds for non-missing elements. For example, the addition operator is then defined as:

$$A \oplus_{NA} B = (a_{ij} \oplus_{NA} b_{ij})_{1 \leq i,j \leq k} = \begin{cases} a_{ij} + b_{ij} & \text{if both } a_{ij} \text{ and } b_{ij} \text{ are numbers} \\ NA & \text{otherwise} \end{cases}$$

Other element-wise operations such as $\ominus_{NA}$ are defined similarly. For readability, we will use standard operator notation $+, -, >, <, I(\cdot), \ldots$ and assume the extended, element-wise operations as defined above. Note, that this definition of dealing with missing values is consistent with a setting of 'na.rm=FALSE' in terms of calculations in R R Core Team (2019).

Registration Estimation

A critical step in comparing A and B is to find a transformation of B such that it aligns best to A (or vice versa). In image processing, this is called *image registration.* Noting that A and B are essentially grayscale images with structurally missing values, we rely on a standard image registration technique (Brown 1992).

In our application, a registration is composed of a discrete translation by $(m, n) \in \mathbb{Z}^2$ and rotation by $\theta \in [-180°, 180°]$. To determine the optimal registration, we calculate

[1] This assumption of equally-sized, square matrices is easily enforced by padding the matrices with additional missing values. Due to the presence of (structurally) missing values around the breech face impression region, additional padding does not interfere with the structure of the scan.

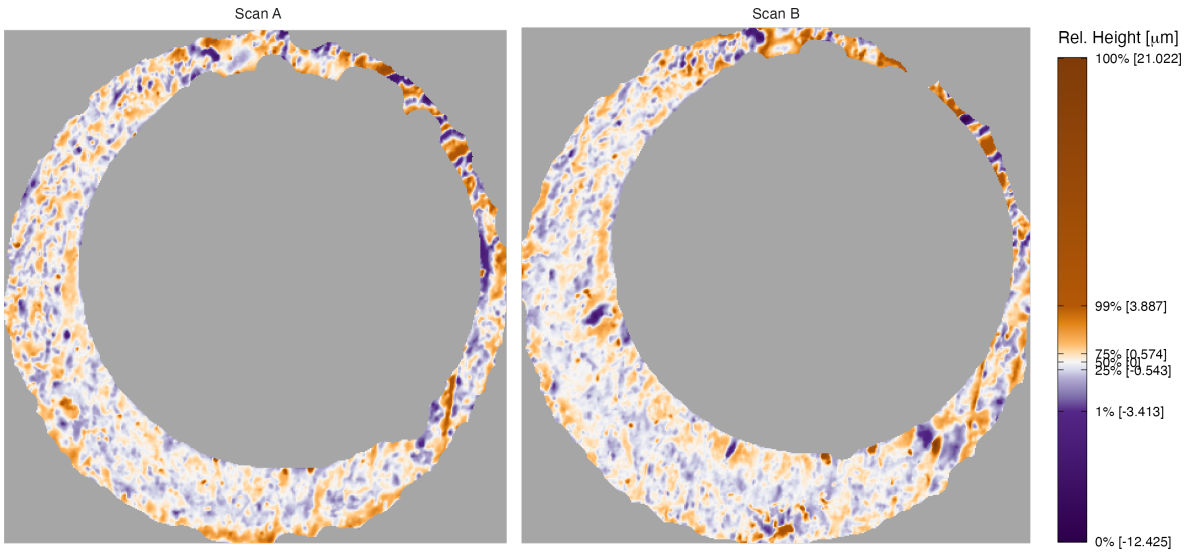

Figure 3: A matching pair of processed cartridge case scans. We measure the similarity between these cartridge cases using the distinguishable breech face impressions on their surfaces.

the *cross-correlation function* (CCF) between A and B, which measures the similarity between A and B for every possible translation of B, denoted $(A \star B)$. We estimate the registration by calculating the maximum CCF value across a range of rotations of matrix B. Let B_θ denote B rotated by an angle $\theta \in [-180°, 180°]$ and $b_{\theta_{mn}}$ the m, n-th element of B_θ. Then the estimated registration (m^*, n^*, θ^*) is:

$$(m^*, n^*, \theta^*) = \arg\max_{m,n,\theta}(a \star b_\theta)_{mn}.$$

In practice we consider a discrete grid of rotations $\boldsymbol{\Theta} \subset [-180°, 180°]$. The registration procedure is outlined in algorithm 1. We refer to the matrix that is rotated as the "target." The result is the estimated registration of the target matrix to the "source" matrix.

Data: Source matrix A, target matrix B, and rotation grid $\boldsymbol{\Theta}$
Result: Estimated registration of B to A, (m^*, n^*, θ^*), and cross-correlation function maximum, $CCF_{\max}$
for $\theta \in \boldsymbol{\Theta}$ **do**
  Rotate B by θ to obtain B_θ;
  Calculate $CCF_{\max,\theta} = \max_{m,n}(a \star b_\theta)_{mn}$;
  Calculate translation $[m^*_\theta, n^*_\theta] = \arg\max_{m,n}(a \star b_\theta)_{mn}$
end
Calculate overall maximum correlation $CCF_{\max} = \max_\theta\{CCF_{\max,\theta} : \theta \in \boldsymbol{\Theta}\}$;
Calculate rotation $\theta^* = \arg\max_\theta\{CCF_{\max,\theta} : \theta \in \boldsymbol{\Theta}\}$;
return *Estimated rotation* θ^**, translation* $m^* = m^*_{\theta^*}$ *and* $n^* = n^*_{\theta^*}$*, and* $CCF_{\max}$

Algorithm 1: Image Registration Procedure

To accommodate missing values, we also compute the *pairwise-complete correlation*

using only the complete value pairs, meaning neither value is missing, between A and B.

Registration-Based Features

Full-Scan Registration We first estimate the registration between two full scans A and B using algorithm 1 with a rotation grid $\boldsymbol{\Theta} = \{-30°, -27°, ..., 27°, 30°\}$. This results in an estimated registration (m^*, n^*, θ^*) and similarity measure $CCF_{\max}$. We also perform algorithm 1 with the roles of A and B reversed, meaning the target scan A is aligned to source scan B.

To accommodate these two comparison directions, we introduce a new subscript $d = A, B$, referring to the source scan in algorithm 1. Consequently, we obtain two sets of sets of estimated registrations, $(m_d^*, n_d^*, \theta_d^*)$ and $CCF_{\max,d}$, for $d = A, B$.[2] For $d = A$, we then apply the registration transformation $(m_A^*, n_A^*, \theta_A^*)$ to B to obtain B^* and compute the pairwise-complete correlation, $cor_{\text{full},A}$, between A and B^*. We repeat this in the other comparison direction to obtain $cor_{\text{full},B}$ and average the two:

$$cor_{\text{full}} = \frac{1}{2}\left(cor_{A,\text{full}} + cor_{B,\text{full}}\right).$$

We assume that the **full-scan pairwise-complete correlation** is large for truly matching cartridge cases.

Cell-Based Registration We next perform a cell-based comparison procedure, which begins with selecting one of the matrices, say A, as the "source" matrix that is partitioned into a grid of cells. The left side of Figure 4 shows an example of such a cell grid overlaid on a scan. Each of these source cells will be compared to the "target" matrix, in this case B^*. Because A and B^* are already partially aligned from the full-scan registration procedure, we compare each source cell to B^* using a new rotation grid of $\boldsymbol{\Theta}'_A = \{\theta_A^* - 2°, \theta_A^* - 1°, \theta_A^*, \theta_A^* + 1°, \theta_A^* + 2°\}$.

We now extend the surface matrix notation introduced previously to accommodate cells. Let A_t denote the t-th cell of matrix A, $t = 1, ..., T_A$ where T_A is the total number of cells containing non-missing values in scan A (e.g., $T_A = 43$ in Figure 4) and let $(a_t)_{ij}$ denote the i, j-th element of A_t.

The cell-based comparison procedure is outlined in algorithm 2.

[2] In reality, the true aligning registrations in the two comparison directions are opposites of each other. However, because we compare discretely-indexed arrays using a nearest-neighbor interpolation scheme, the estimated registrations differ slightly.

Data: Source matrix A, target matrix B^*, grid size $R \times C$, and rotation grid $\boldsymbol{\Theta}'_A$
Result: Estimated translations and $CCF_{\max}$ values per cell, per rotation
Partition A into a grid of $R \times C$ cells;
Discard cells containing only missing values, leaving T_A remaining cells;
for $\theta \in \boldsymbol{\Theta}'_A$ **do**
  Rotate B^* by θ to obtain B^*_θ;
  for $t = 1, ..., T_A$ **do**
    Calculate $CCF_{\max,A,t,\theta} = \max_{m,n}(a_t \star b^*_\theta)_{mn}$;
    Calculate translation $[m^*_{A,t,\theta}, n^*_{A,t,\theta}] = \arg\max_{m,n}(a_t \star b^*_\theta)_{mn}$
  end
end
return $\boldsymbol{F}_A = \{(m^*_{A,t,\theta}, n^*_{A,t,\theta}, CCF_{\max,A,t,\theta}, \theta) : \theta \in \boldsymbol{\Theta}'_A, t = 1, ..., T_A\}$

Algorithm 2: Cell-Based Comparison Procedure

Rather than exclusively returning the registration that maximizes the overall CCF as in algorithm 1, algorithm 2 returns the set $\boldsymbol{F}_A$ of translations and CCF values for each of the T_A cells and each rotation in $\boldsymbol{\Theta}'_A$.

Figure 4 shows the estimated registrations of cells between two non-match cartridge cases. We magnify the surface values captured by cell pairs 5, 1 and 7, 7 and note the similarities in the surface values; for example, the dark purple region in the middle of the cell 7, 7 pair.

Just as with the whole-scan registration, we calculate the pairwise-complete correlation between each cell A_t and a matrix $B^*_{\theta,t}$ of the same size extracted from B^*_θ after translating by $[m^*_{A,\theta}, n^*_{A,\theta}]$. From this we obtain a set of pairwise-complete correlations for each cell and rotation: $\{cor_{A,t,\theta} : t = 1, ..., T_A, \theta \in \boldsymbol{\Theta}'_A\}$.

We repeat algorithm 2 and the pairwise-complete correlation calculation using B as the source scan and A^* as the target, resulting in cell-based registration set $\boldsymbol{F}_B$ and pairwise-complete correlations $\{cor_{B,t,\theta} : t = 1, ..., T_B, \theta \in \boldsymbol{\Theta}'_B\}$.

For $d = A, B$ and $t = 1, ..., T_d$, define the cell-wise maximum pairwise-complete correlation as:

$$cor_{d,t} = \max_\theta \{cor_{d,t,\theta} : \theta \in \boldsymbol{\Theta}'_d\}$$

We compute two features, the **average** and **standard deviation of the cell-based pairwise-complete correlations**, using the correlation data:

$$\overline{cor}_{\text{cell}} = \frac{1}{T_A + T_B} \sum_{d \in \{A,B\}} \sum_{t=1}^{T_d} cor_{d,t}$$

$$s_{cor} = \sqrt{\frac{1}{T_A + T_B - 1} \sum_{d \in \{A,B\}} \sum_{t=1}^{T_d} (cor_{d,t} - \overline{cor}_{\text{cell}})^2}$$

We expect $\overline{cor}_{\text{cell}}$ and s_{cor} to be large for truly matching cartridge case pairs relative to non-matching pairs.

For $d = A, B$ and $t = 1, ..., T_d$, define the per-cell estimated translations and rotation

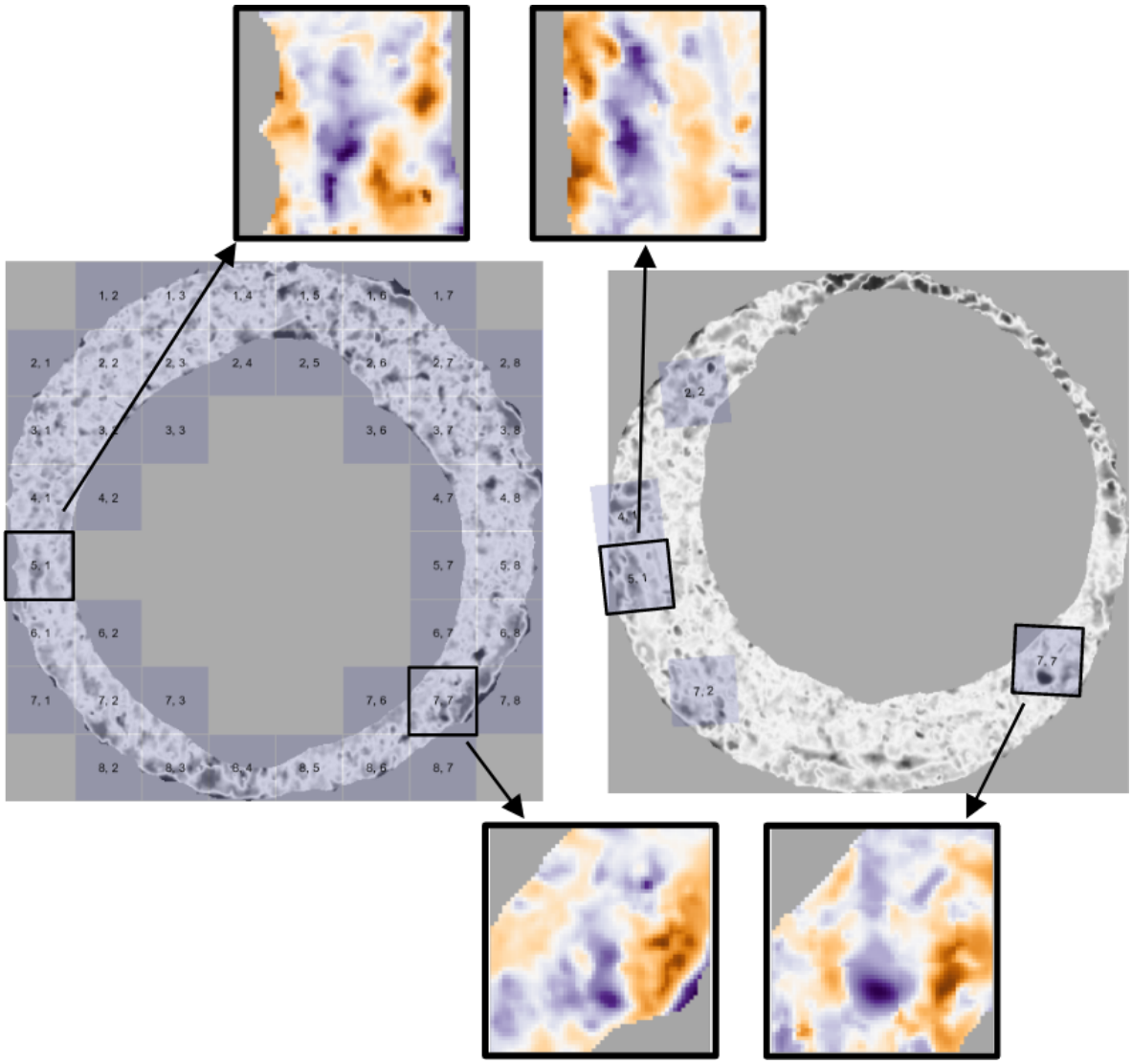

Figure 4: Estimated registrations of cells from a non-match pair of cartridge cases. A source scan (left) is separated into an 8×8 grid of cells. We exclude cells containing only missing values (visualized here as gray pixels). Each source cell is compared to a target scan (right) to estimate where it aligns best. We show a handful of cells at their estimated alignment in the target scan and magnify the surfaces captured by cell pairs 5, 1 and 7, 7. Although the cartridge case pair is non-matching, we note that there are similarities in the surface markings for these cell pairs.

as:

$$\theta^*_{d,t} = \arg\max_{\theta}\{CCF_{\max,d,t,\theta} : \theta \in \mathbf{\Theta}'_d\}$$
$$m^*_{d,t} = m^*_{d,t,\theta^*_{d,t}}$$
$$n^*_{d,t} = n^*_{d,t,\theta^*_{d,t}}$$

We compute the **standard deviation of the cell-based estimated registrations** using the estimated translations and rotations:

$$s_{\theta^*} = \sqrt{\frac{1}{T_A + T_B - 1}\sum_{d\in\{A,B\}}\sum_{t=1}^{T_d}(\theta^*_{d,t} - \bar{\theta}^*)^2}$$
$$s_{m^*} = \sqrt{\frac{1}{T_A + T_B - 1}\sum_{d\in\{A,B\}}\sum_{t=1}^{T_d}(m^*_{d,t} - \bar{m}^*)^2}$$
$$s_{n^*} = \sqrt{\frac{1}{T_A + T_B - 1}\sum_{d\in\{A,B\}}\sum_{t=1}^{T_d}(n^*_{d,t} - \bar{n}^*)^2}$$

where

$$\bar{m}^* = \frac{1}{T_A + T_B}\sum_{d\in\{A,B\}}\sum_{t=1}^{T_d} m^*_{d,t}$$
$$\bar{n}^* = \frac{1}{T_A + T_B}\sum_{d\in\{A,B\}}\sum_{t=1}^{T_d} n^*_{d,t}$$
$$\bar{\theta}^* = \frac{1}{T_A + T_B}\sum_{d\in\{A,B\}}\sum_{t=1}^{T_d} \theta^*_{d,t}.$$

We expect $s_{\theta^*}, s_{m^*}, s_{n^*}$ to be small for truly matching cartridge case pairs relative to non-matching pairs.

From the full-scan and cell-based registration procedures, we obtain six features summarized in Table 1.

cor_{full}	Full-scan pairwise-complete correlation
$\overline{cor}_{\text{cell}}$	Average cell-based pairwise-complete correlation
s_{cor}	Standard deviation of the cell-based pairwise-complete correlations
s_{m^*}	Standard deviation of the cell-based vertical translations (in microns)
s_{n^*}	Standard deviation of the cell-based horizontal translations (in microns)
s_{θ^*}	Standard deviation of the cell-based rotations (degrees)

Table 1: Six similarity features based on registering full scans and cells.

Density-Based Features

We wish to identify when multiple cells agree on, or cluster around, a particular registration value. However, pursuant with the notion that only certain regions of matching cartridge cases contain distinctive markings, it is unreasonable to assume and empirically rare that *all* cells agree on a single registration. In fact, it is common for many cells to disagree on a registration. For example, the left scatterplot in Figure 5 shows the per-cell estimated translations $[m^*_{A,t,\theta}, n^*_{A,t,\theta}]$ when scan A is used as source and B^* as target rotated by $\theta = 3°$. The right scatterplot shows the per-cell estimated translations with the roles of A and B^* reversed for $\theta = -3°$. We see distinctive clusters, the black points, in both plots among many noisy, gray points. The task is to isolate the clusters amongst such noise.

We use the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm proposed by Ester et al. (1996) to identify clusters. Compared to other clustering algorithms such as k-means (MacQueen 1967), DBSCAN does not require a pre-defined number of expected clusters. Instead, the algorithm forms clusters if the number of points within an $\epsilon > 0$ distance of a point exceeds some pre-defined threshold, $minPts > 1$. If a point does not belong to a cluster, then DBSCAN labels that point as "noise." In Figure 5, we use DBSCAN with $\epsilon = 5$ and $minPts = 5$ to identify clusters of size 14 and 13, respectively, visualized as black points. These cluster sizes suggest that the scans match. Additionally, the mean cluster centers are approximately opposites of each other: $(\hat{m}_A, \hat{n}_A, \hat{\theta}_A) \approx (16.9, -16.7, 3°)$ when A is used as source compared to $(\hat{m}_B, \hat{n}_B, \hat{\theta}_B) \approx (-16.2, 16.8, -3°)$ when B^* is used as source. This provides further evidence of a match.

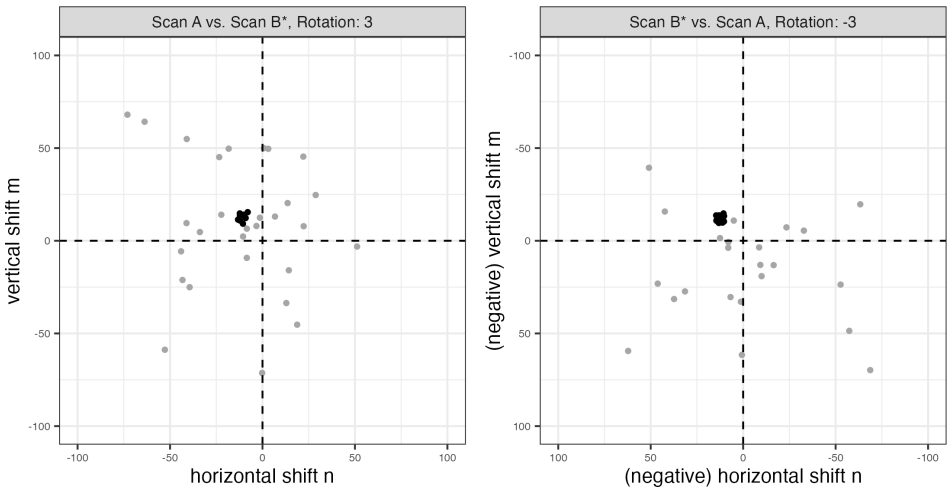

Figure 5: Cluster assignments based on the Density Based Spatial Clustering with Applications to Noise (DBSCAN) algorithm for estimated translations in two comparison directions. Using scan A as source results in a cluster of size 14 (left) compared to 13 when scan B^* is used as source (right). Noting the reversed axes in the right plot, we see that the clusters are located approximately opposite of each other. Points are jittered for visibility.

To calculate the density-based features, we first use a 2D kernel density estimator (Venables and Ripley 2002) to identify the rotation $\hat{\theta}_d$ at which the per-cell translations achieve the highest density. Next, we compute clusters using the DBSCAN algorithm

amongst the estimated translations $\{(m^*_{d,t,\hat{\theta}_d}, n^*_{d,t,\hat{\theta}_d}) : t = 1, ..., T_d\}$ like those shown in Figure 5.[3] Let $\boldsymbol{C}_d$ denote the set of cells in the DBSCAN cluster. We treat the mean cluster centers as the estimated translations $[\hat{m}_d, \hat{n}_d]$.

We calculate four features from the density-based clustering procedure: **average DBSCAN cluster size** C, the **DBSCAN cluster indicator** C_0, and the **root sum of squares of the density-estimated registrations** $(\Delta_\theta, \Delta_{\text{trans}})$ defined as:

$$\begin{aligned} C &= \frac{1}{2}\left(|\boldsymbol{C}_A| + |\boldsymbol{C}_B|\right) \\ C_0 &= I(|\boldsymbol{C}_A| > 0 \text{ and } |\boldsymbol{C}_B| > 0) \\ \Delta_\theta &= |\hat{\theta}_A + \hat{\theta}_B| \\ \Delta_{\text{trans}} &= \sqrt{(\hat{m}_A + \hat{m}_B)^2 + (\hat{n}_A + \hat{n}_B)^2} \end{aligned}$$

where $|\boldsymbol{C}_d|$ denotes the cardinality of $\boldsymbol{C}_d$ and $I(\cdot)$ is the identity function equal to 1 if the predicate argument "$\cdot$" evaluates to TRUE and 0 otherwise. We use both C and C_0 because of potential missingness in the values of C if no cluster is identified. Missing C values are imputed using the median non-missing value when fitting classifiers, so the missingness information is retained in C_0.

For truly matching cartridge case pairs, we expect C to be large, C_0 to be 1, and $\Delta_\theta, \Delta_{\text{trans}}$ to be small relative to non-matching pairs. We obtain four density-based features summarized in Table 2.

C	Average DBSCAN cluster size
C_0	DBSCAN cluster indicator
Δ_θ	Absolute sum of the density-estimated rotations (degrees)
Δ_{trans}	Root sum of squares of the density-estimated translations (in microns)

Table 2: Four similarity features based on the density-based clustering procedure.

Visual Diagnostic Features

The final set of features we calculate are based on visual diagnostic tools described in [visual diagnostics paper]. These numerical features quantify the qualitative observations one can make from the diagnostics.

To create the visual diagnostics, we perform element-wise matrix operations. For a matrix $X \in \mathbb{R}^{k\times k}$ and Boolean-valued condition matrix $cond : \mathbb{R}^{k\times k} \to \{TRUE, FALSE\}^{k\times k}$, we define an element-wise filter operation $\mathcal{F} : \mathbb{R}^{k\times k} \to \mathbb{R}^{k\times k}$ as:

$$\mathcal{F}_{cond}(X) = (f_{ij})_{1\le i,j\le k} = \begin{cases} x_{ij} & \text{if } cond \text{ is } TRUE \text{ for element } i, j \\ NA & \text{otherwise} \end{cases}$$

[3] If more than one cluster is identified, we binarize the points based on whether they were assigned to any cluster or if they are a noise point and proceed as if there is only one cluster. We assume that two or more clusters form only because of the course rotation grid considered. Were a finer grid used, the points would coalesce into a single cluster around the true translation value. This assumption has empirical support through our experimentation.

Of particular interest in our application is the (absolute) difference between surface matrices. For example, $\mathcal{F}_{|A-B^*|>\tau}(A)$ contains elements of matrix A where the pair of scans A and B^* deviate by at least $\tau > 0$. Surface values in A and B^* that are "close," meaning within τ distance, to each other are replaced with NA in this filtered matrix.

First, we calculate the correlation $cor_{d,\text{full,diff}}$ between the filtered matrices $\mathcal{F}_{|A-B^*|>\tau}(A)$ and $\mathcal{F}_{|A-B^*|>\tau}(B^*)$ for $d = A$ and $\mathcal{F}_{|A^*-B|>\tau}(A^*)$ and $\mathcal{F}_{|A^*-B|>\tau}(B)$ for $d = B$. We use the average **full-scan differences correlation** as a feature:

$$cor_{\text{full,diff}} = \frac{1}{2}\left(cor_{A,\text{full,diff}} + cor_{B,\text{full,diff}}\right).$$

We assume that $cor_{\text{full,diff}}$ will be large for matching cartridge case pairs relative to non-matching pairs. Said another way, we assume that regions of matching cartridge cases that are different will still follow similar trends. This can occur due to variability in the amount of contact between a cartridge case and breech face across multiple fires of a single firearm. We calculate the correlation by vectorizing the two filtered surface matrices and treating missing values by case-wise deletion.

As before, we extend our notation to accommodate cell comparisons $t = 1, ..., T_d$ for $d = A, B$ using subscripts: $cor_{d,t,\text{diff}}$. For example, $cor_{A,t,\text{diff}}$ is the correlation between cell filtered surface matrices $\mathcal{F}_{|A_t-B^*_{t,\theta^*_t}|>\tau}(A_t)$ and $\mathcal{F}_{|A_t-B^*_{t,\theta^*_t}|>\tau}(B^*_{t,\theta^*_t})$ where B^*_{t,θ^*_t} is the matrix extracted from B^* that maximizes the CCF with A_t. We calculate the **average cell-based differences correlation** across all cells and both directions:

$$\overline{cor}_{\text{cell,diff}} = \frac{1}{T_A + T_B} \sum_{d\in\{A,B\}} \sum_{t=1}^{T_d} cor_{d,t,\text{diff}}$$

Next, we consider features based on the elements of the Boolean *cond* matrix. Consider Figure 6 that shows the filtered element-wise average $\mathcal{F}_{|A-B^*|\leq\tau}\left(\frac{1}{2}(A + B^*)\right)$ on the left and the complementary *cond* matrix $|A - B^*| > \tau$ visualized in black-and-white in the middle with filtered elements whose *cond* value is $TRUE$ shown in white.

We first calculate the ratio between such a *cond* matrix and its complement. For $d = A$, we consider the *cond* matrices $|A - B^*| \leq \tau$ and $|A - B^*| > \tau$. The ratio is given by

$$r_d = \frac{\mathbf{1}^T I(|A - B^*| \leq \tau)\mathbf{1}}{\mathbf{1}^T I(|A - B^*| > \tau)\mathbf{1}}$$

where $\mathbf{1} \in \mathbb{R}^k$ is a column vector of ones and $I(\cdot)$ is the element-wise, matrix-valued indicator function. We consider the average **full-scan similarities vs. differences ratio** across the two comparison directions:

$$r_{\text{full}} = \frac{1}{2}(r_A + r_B).$$

We expect r_{full} to be large for matching pairs compared to non-matching pairs. That is, truly matching pairs will have more similarities than differences.

We also calculate features based on the ratio for cell comparisons $t = 1, ..., T_d$, $d = A, B$. Let $r_{d,t}$ denote the ratio for cell comparison t in direction d. We consider the **average**

and **standard deviation of the cell-based similarities vs. differences ratio**:

$$\bar{r}_{\text{cell}} = \frac{1}{T_A + T_B} \sum_{d \in \{A,B\}} \sum_{t=1}^{T_d} r_{d,t}$$

$$s_{\text{cell},r} = \sqrt{\frac{1}{T_A + T_B - 1} \sum_{d \in \{A,B\}} \sum_{t=1}^{T_d} (r_{d,t} - \bar{r}_{\text{cell}})^2}.$$

We expect $\bar{r}_{\text{cell}}$ and $s_{\text{cell},r}$ to be large for matching cartridge case pairs relative to non-match pairs.

Another aspect of the *cond* matrix we consider is the size of the individual filtered regions. For two matching cartridge cases, we expect that there are few differences compared to similarities *and* that the different regions are relatively small. We use a connected components labeling algorithm detailed in Hesselink et al. (2001) to identify individual "neighborhoods" of filtered elements (Barthelme 2023). More precisely, the algorithm returns a set of sets $\boldsymbol{S}_d = \{S_{d,1}, S_{d,2}, ..., S_{d,L_d}\}$ where each $S_{d,l}$ is a set of indices of the *cond* matrix that have a value of $TRUE$ and are connected by a chained-together sequence of 4 (Rook's) neighborhoods. The right side of Figure 6 shows each $S_{d,l}$ distinguished by different fill colors, $l = 1, ..., L_d$.

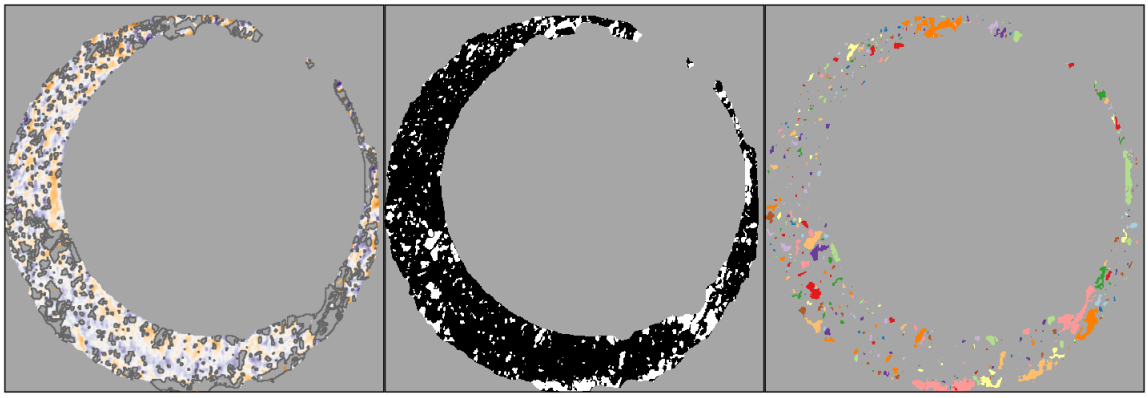

Figure 6: (Left) After aligning two scans, we filter regions that are "different" from each other, meaning the absolute difference between surface values is larger than some threshold. (Middle) We binarize the scan into "filtered" or "non-filtered" regions - shown in white and black, respectively. (Right) Using a connected components labeling algorithm, we identify connected "neighborhoods" of filtered elements. We assume that these neighborhoods will be small, on average, if comparing truly matching cartridge cases.

We calculate the following features using the full-scan labeled neighborhoods:

$$\overline{|S|}_{\text{full}} = \frac{1}{L_A + L_B} \sum_{d \in \{A,B\}} \sum_{l=1}^{L_d} |S_{d,l}|$$

$$s_{\text{full},|S|} = \sqrt{\frac{1}{L_A + L_B - 1} \sum_{d \in \{A,B\}} \sum_{l=1}^{L_d} (|S_{d,l}| - \overline{|S|}_{\text{full}})^2}$$

where $|S_{d,l}|$ is the cardinality of $S_{d,l}$. We assume that the **average** and **standard deviation of the full-scan neighborhood sizes** will be small for matching cartridge case pairs relative to non-matching pairs. That is to say, we assume that the the regions of A and B that are different will all be small, on average, and vary little in size. This assumption is appropriate assuming that the breech face leaves consistent markings on fired cartridge cases.

Again, we extend our notation to accommodate individual cells. Let $\boldsymbol{S}_{d,t} = \{S_{d,t,1}, ..., S_{d,t,L_{d,t}}\}$ denote the set of labeled neighborhoods for a cell $t = 1, ..., T_d$, $d = A, B$. We calculate the per-cell average and standard deviation of the labeled neighborhood cell size:

$$\overline{|S|}_{d,t} = \frac{1}{L_{d,t}} \sum_{l=1}^{L} |S_{d,t,l}|$$

$$s_{d,t,|S|} = \sqrt{\frac{1}{L_{d,t} - 1} \sum_{l=1}^{L_{d,t}} (|S_{d,t,l}| - \overline{|S|}_{\text{cell},d,t})^2}.$$

We assume that the cell-based $\overline{|S|}_{d,t}$ and $s_{d,t,|S|}$ will be small, on average, for truly matching cartridge cases. Consequently, we use the sample average of these as features:

$$\overline{|S|}_{\text{cell}} = \frac{1}{T_A + T_B} \sum_{d \in \{A,B\}} \sum_{t=1}^{T_d} \overline{|S|}_{d,t}$$

$$\bar{s}_{\text{cell},|S|} = \frac{1}{T_A + T_B} \sum_{d \in \{A,B\}} \sum_{t=1}^{T_d} s_{d,t,|S|}$$

We assume that the **average cell-wise neighborhood size** and the **average standard deviation of the cell-wise neighborhood sizes** will be small for matching cartridge case pairs relative to non-match pairs.

Table 3 summarizes the nine features based on visual diagnostics. This concludes our explanation of the ACES feature set. Next, we use the 19 ACES features to train and test classifier models.

$cor_{\text{full,diff}}$	Full-scan differences correlation
$\overline{cor}_{\text{cell,diff}}$	Average cell-wise differences correlation
r_{full}	Full-scan similarities vs. differences ratio
$\bar{r}_{\text{cell}}$	Average cell-based similarities vs. differences ratio
$s_{\text{cell},r}$	Standard deviation of the cell-based similarities vs. differences ratio
$\overline{\|S\|}_{\text{full}}$	Average full-scan neighborhood size (in microns)
$s_{\text{full},\|S\|}$	Standard deviation of the full-scan neighborhood sizes (in microns)
$\overline{\|S\|}_{\text{cell}}$	Average cell-wise neighborhood sizes (in microns)
$\bar{s}_{\text{cell},\|S\|}$	Average standard deviation of the cell-wise neighborhood sizes (in microns)

Table 3: Nine similarity features calculated based on visual diagnostics.

3.3. Scoring

We use a data set of 510 cartridge cases fired from 25 firearms. We randomly split the data into 10 firearms for training and 15 firearms for testing. This resulted in a training data set of 210 cartridge cases, $\binom{210}{2} = 21,945$ pairwise comparisons, and a testing set of 300 cartridge cases, $\binom{300}{2} = 44,850$ pairwise comparisons. Because we consider every pairwise comparison between these scans, there is a relatively large class imbalance between matches and non-matches in these data sets. Specifically, non-matching comparisons make up 19,756 of the 21,945 (90.0%) training comparisons and 41,769 of the 44,850 (93.1%) testing comparisons.

We use 10-fold cross-validation repeated thrice (Kuhn 2022) to train two binary classifiers based on a logistic regression and a random forest (Breiman 2001; Liaw and Wiener 2002). These models predict the probability that a pair of cartridge cases match. Then, the model classifies the pair as a match or non-match depending on whether the match probability exceeds a set threshold. On top of the tunable parameters of each model (e.g., the DBSCAN parameters ϵ and $minPts$), we treat this threshold as a parameter to be optimized.

Models trained to maximize accuracy on imbalanced data often exhibit a "preference" for classifying new observations as the majority class (Fernández et al. 2018), which in our case are non-matches. An optimization criterion commonly used for imbalanced data is to select the model that maximizes the area under the Receiver Operating Characteristic (ROC) curve, which measures the performance of a model under different threshold values (James et al. 2013). The model that maximizes this area, commonly abbreviated AUC, is one that performs best under a variety of threshold values relative to the other models - this consistency is a desired trait. Using the ROC curve, we choose the match probability threshold that balances the true negative and true positive (equivalently, the false positive and false negative) rates on the training data.

Once we have a trained model, we use it to predict the match probability and classify a new cartridge case pair. However, rather than referring to the number returned by the trained model as a "probability," which implicitly assumes a homogeneous source population between the training and test cartridge cases, we simply call the number a "score" where larger values correspond with more similar cartridge cases. We compute this score for the pairwise comparisons in the test data as a means of comparing the generalizability of the various models. The following section details the results of this cross-validation training/testing procedure.

4. Results

4.1. ROC Curves

First, we consider results from the training procedure. Figure 7 shows the resulting ROC curves for four classifier models trained on the training data set. We consider training the logistic regression (LR) and random forest (RF) models under two feature sets: a subset of the full ACES feature set consisting of the Cluster Indicator feature C_0 and the six registration-based features summarized in Table 1 vs. all 19 ACES features.

We consider the "C_0 + Registration" subset of features to represent the features used in Congruent Matching Cells methods (Song 2013; Zhang et al. 2021).

The ROC curves allow us to visually compare the behavior of these four classifiers under various score thresholds where curves closer to the top-left corner are preferred. Broadly speaking, the four models perform comparably as evidenced by the similar curves on the right side of Figure 7. The left side shows a zoomed-in version of the top left corner of plot, which makes it easier to compare the different curves. Visually, we see that the choice of feature group has a larger impact on the outcome classification behavior than the choice between the logistic regression or random forest models.

To numerically compare the four models, we compute the area under the ROC curve (AUC) as well as the score threshold (Thresh.) that balances the false negative and false positive rates (the equal error rate or EER). The AUC for the All ACES logistic regression and random forest classifiers are higher than the AUC of the two classifiers trained on the C_0 + Registration feature set. Each model has a different score threshold that yields the equal error rate, which we visualize as points along the four ROC curves in Figure 7. We use these thresholds to compute both the training and test classification results summarized below. We see that the All ACES, logistic regression model has the lowest equal error rate out of the four models with the All ACES, random forest model a close second.

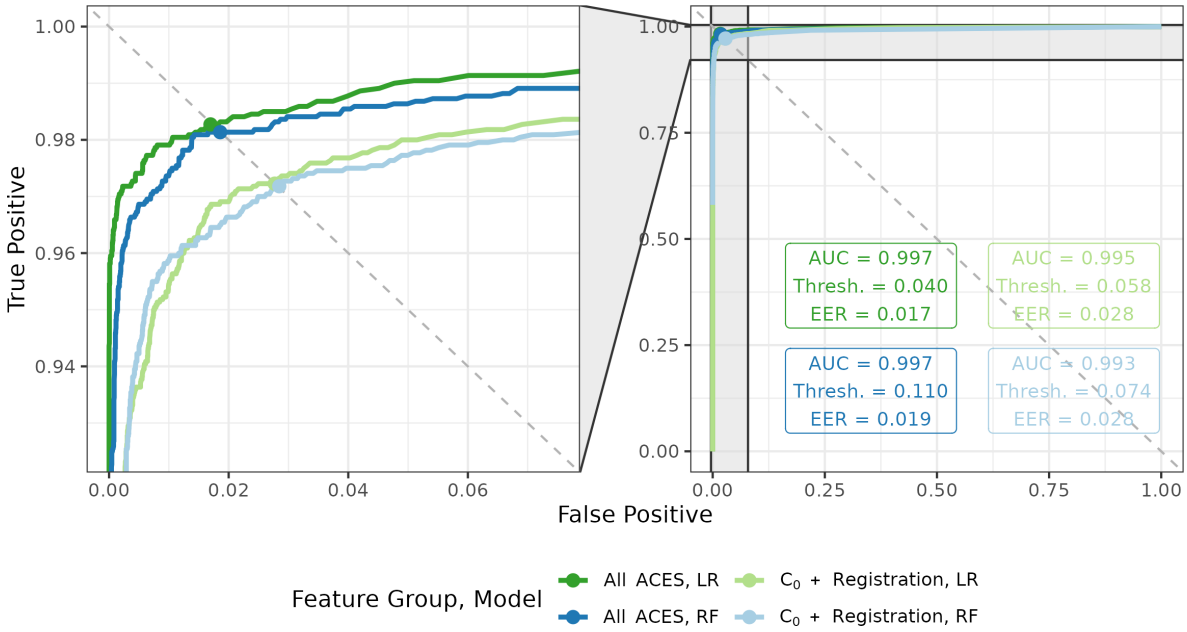

Figure 7: ROC curves for logistic regression (LR) and random forest (RF) models trained using two feature sets - all 19 ACES features vs. a subset of seven ACES features. On the left, we zoom into the top-left corner of the ROC curve plot to better distinguish between the four curves. We see that the models trained on the full ACES feature set have higher area under the curve (AUC) and lower equal error rate (EER) values than on the subset. We also show the score classification cutoffs (Thresh.) used for each of the four models to achieve the equal error rate values.

4.2. Optimized Model Comparison

Figure 8 summarizes the training and testing accuracy, true negative and true positive rates for five binary classifiers. We distinguish between the training and testing results using gray and black points/line segments, respectively, which allows us to assess the generalizability of the various models. The conclusions drawn from Figure 8 are intended to primarily be qualitative and comparative across models. Table 6 and Table 7 in the Appendix provide a numerical summary of these results.

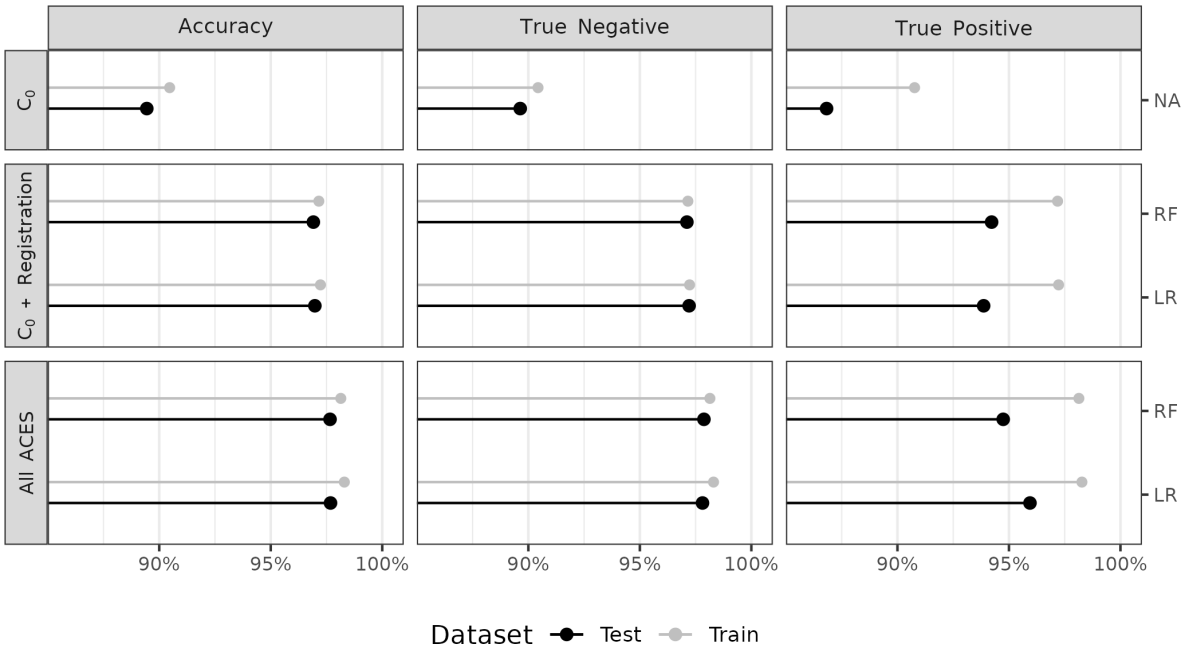

Figure 8: We summarize classification accuracy, true negative, and true positive rates for both the training and testing results, represented as gray and black points/lines respectively, for five binary classifier models. Our primary interest is the test data results, but visualizing the training data results allows us to assess the generalizability of the models after training. In the first row, we consider a classifier based on a single feature, the Cluster Indicator feature C_0, as a baseline. The remaining rows show results from training/testing classifiers based on a random forest (RF) and logistic regression (LR) under various feature sets and optimization critieria. The second row shows results based on a subset of seven features from the ACES feature set while the third row shows results using all 19 ACES features.

We first compare the training and testing results across the five models and three columns in Figure 8. In general, the true negative rates based on the test data are slightly lower than those of the training data indicating that the models' ability to distinguish between non-matching comparisons generalizes well to the testing data. In contrast, the true positive rates tend to be lower for the test data compared to the training data across the various models, which indicates a potential difference between the training and testing data. As we discuss below, there is a single firearm among the 15 test firearms that contributes the majority of false negative (misclassified match) test classifications. Despite lower true positive rates, the overall accuracy between the training and testing sets are comparable due to the large class imbalance between

matching and non-matching comparisons in both.

In the first row, we consider a baseline classifier based solely on the Cluster Indicator feature C_0. Namely, if the DBSCAN algorithm finds clusters in the cell-based translations from both directions of a cartridge case comparison, then that pair is classified as a match. This is analogous to the classification rule used in Zhang et al. (2021). We optimized this C_0-based classifier by choosing the DBSCAN parameters ϵ and $minPts$ that resulted in the most balanced training true negative and true positive rates, resulting in $\epsilon = 15$ and $minPts = 8$. The optimized C_0-based classifier performs considerably worse across the three measures compared to the other models with test accuracy 89.44%, true negative rate 89.64%, and true positive rate 86.82%.

The second row of Figure 8 summarizes results from training the two classifier models on a subset of the full ACES feature set consisting of the Cluster Indicator feature C_0 and the six registration-based features summarized in Table 1. We consider this subset of features to represent the features used in Congruent Matching Cells methods (Song 2013; Zhang et al. 2021). In general, we see that the logistic regression (LR) and random forest (RF) models perform comparable to each other in accuracy, true negative, and true positive rates. Despite the fact that the models in the second and third rows were selected based on balancing the training true negative and true positive rates, we note that these rates for the test data are not as well-balanced; namely, the true negative rates still tend to be larger than the true positive rates. Below, we explore this discrepancy by analyzing the contribution of various test firearms towards the true positive rates.

The third row of Figure 8 summarizes the classification results based on using all 19 ACES features. If we compare the "C_0 + Registration"-trained models in the second vs. the "All ACES"-trained models in the third row, we see that the addition of the other ACES features leads to improved test true negative and true positive rates (and consequently overall accuracy) with the most noticeable gains observed in the true positive rate. Across all five models, the All ACES-trained logistic regression model has the largest overall test accuracy and true positive rates of 97.68% and 95.94%, respectively. The All ACES-trained random forest model has the largest overall true negative rate of 97.87%, although the All ACES, logistic regression model is a close second at 97.81% (see Table 7 for more details).

4.3. Similarity Score Investigation

While it's useful to consider the accuracy, true negative, and true positive rates to compare various models, forensic examiners would likely not use the binary classification returned by a model in casework. Instead, they would consider the match probability predicted by the model as a similarity score and incorporate it into their decision-making process. As such, we also consider the distribution of the predicted similarity scores for matching and non-matching comparisons. Figure 9 shows a dot plot of the predicted similarity scores for the 41,769 non-match and 3,181 match comparisons in the test set. Specifically, these probabilities are predicted by the logistic regression model selected to maximize the AUC based on the full ACES feature set. As we expect, few non-match comparisons have large similarity scores, which justifies the low false positive rate observed in Figure 8. However, there is a surprising number of matching

comparisons that also have a low match probability.

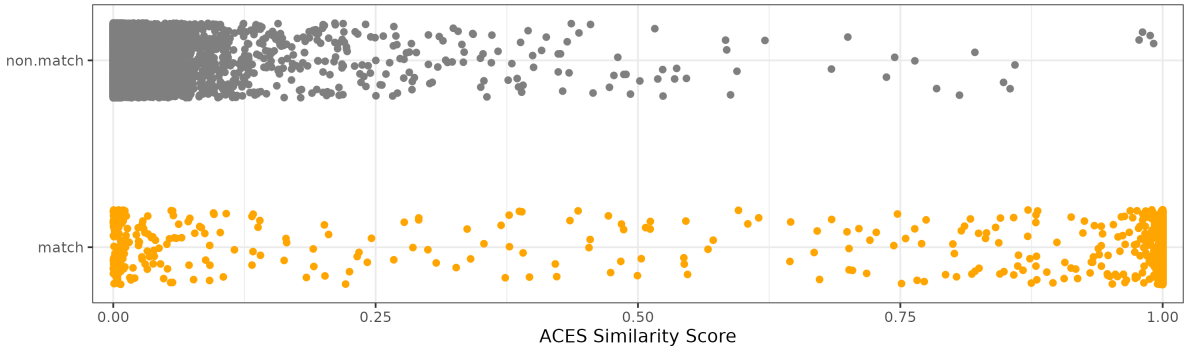

Figure 9: A dot plot of the predicted similarity scores for the non-match and match comparisons in the test set based on a logistic regression model. As we expect, the non-match comparisons tend to have a low match probability. However, we see that there are many matching comparisons that also have a low match probability.

To explain the matching comparisons with low similarity scores, we visualize in Figure 10 the predicted similarity scores for matching test comparisons distinguished by the 15 test firearm ID. We see that the firearm T has far more matching comparisons with low similarity scores compared to the other 14 test firearms. This is further underscored by the right side of the Figure 10, which shows the ratio of misclassifications to total comparisons for every pair of test firearms based on the same logistic regression model used in Figure 9. The main diagonal shows the false negative misclassifications while the off-diagonal shows the false positives. We use blank tiles for comparisons where 0 misclassifications occurred. We see that the false negative rate for firearm T of 27.1% is far greater than that of other firearm pairs. The 95 false negative firearm T comparisons comprise 76% of all 125 false negative test comparisons and about 3% of all 3,181 matching test comparisons. In sum, the model performs distinctly worse at identifying matching comparisons from firearm T compared to the other firearms, which partially explains the lower test true positive rates noted in Figure 8. Upon visual inspection of the scans from firearm T, we noted a lack of consistent markings on their surfaces, which isn't the case for scans from other test firearms.

4.4. Feature Importance

Finally, we consider the relative importance of the 19 ACES features by fitting 10 replicate random forests using the full ACES feature set with fixed random seeds. For each replicate, we measure a variable's importance using the Gini Index, which measures the probability of making a misclassification for a given model (Hastie et al. 2001). A larger decrease in the Gini Index corresponds with higher importance. Figure 11 shows the distribution of the mean Gini Index decrease for the 19 ACES features. Noting the log scale on which these points are plotted, we see that the most important features consist of a combination of density-based features C_0, C, and Δ_{trans} and registration-based correlation features $\overline{\text{cor}}_{\text{cell}}$ and cor_{full}. In general, the visual diagnostic features tend to have lower importance scores compared the registration and density-based features. We discuss the sensitivity of these importance scores to various algorithm parameter choices in the next section.

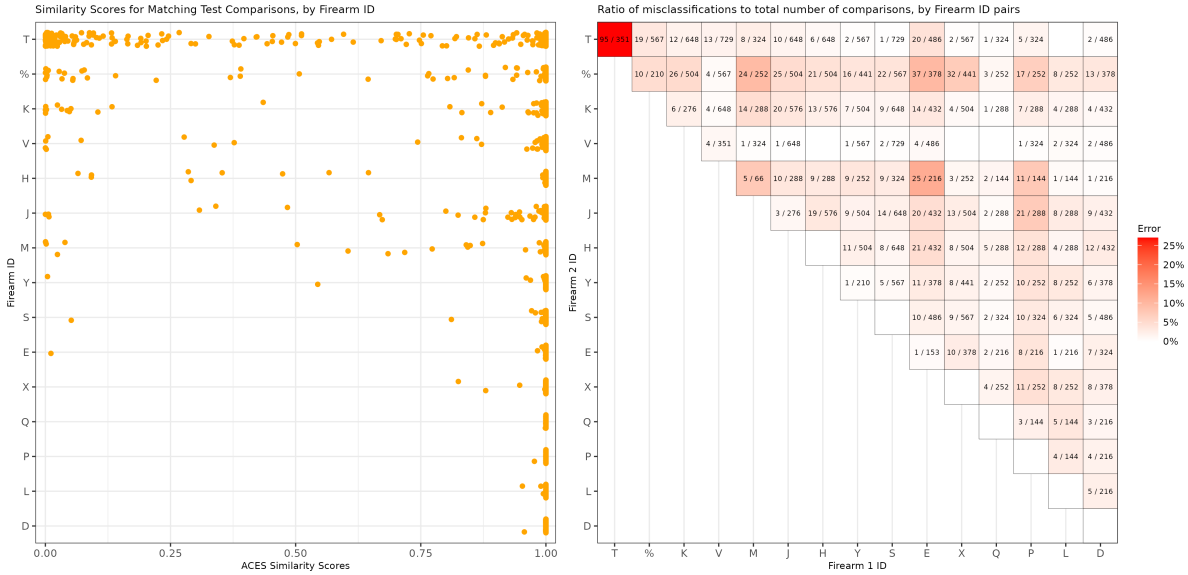

Figure 10: (Left) A dot plot of the predicted similarity scores for the match comparisons in the test set based on a logistic regression model, separated by firearm. We see that firearm T has more matching comparisons with low similarity scores than the other test firearms. (Right) Misclassifications divided by total number of pairwise comparisons for each pair of test firearms based on the same logistic regression model. We do not show comparisons with 0 misclassifications. We note that the proportion of misclassified matching comparisons from firearm T of 27.1% is much higher than that of other comparisons.

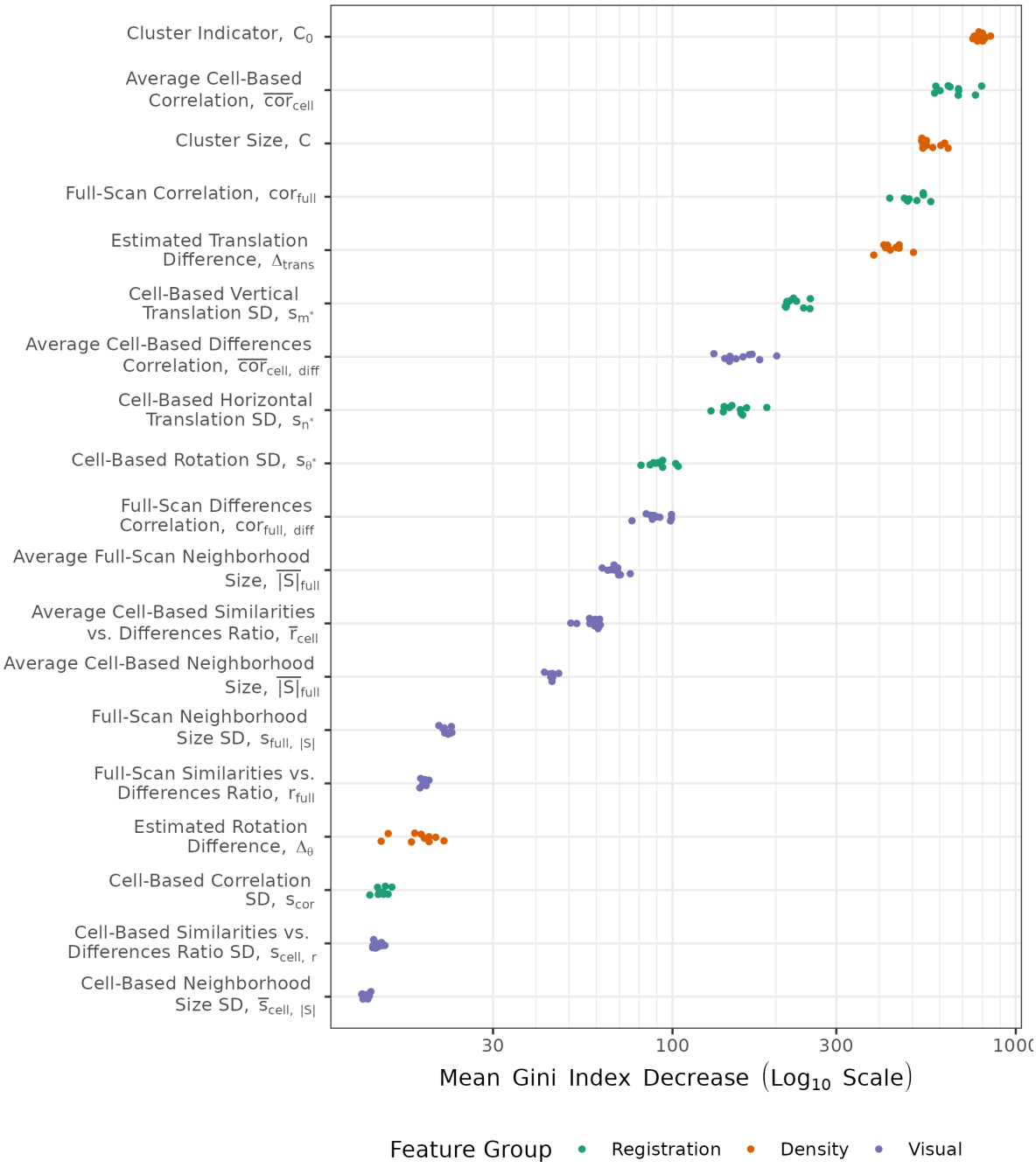

Figure 11: Variable importance measures from fitting a random forest to the training data set, repeated 10 times under various random seeds. The top features consist of density-based features C and Δ_{trans} and registration-based features $\overline{\text{cor}}_{\text{cell}}$ and cor_{full}. We plot points on a log scale and vertically jitter them for visibility.

5. Discussion

5.1. Comparison to CMC Methodology

We use a C_0-based classifier as a baseline because it is analogous to the classification rule proposed in Zhang et al. (2021). Similarly, the cell-based registration features are based on the the same cell-based comparison procedure used in Song (2013) and summarized in algorithm 2. Together, we consider C_0 and the registration-based features a fusion of previously proposed cartridge case similarity scoring algorithms. This is why we fit separate classifiers based on these features for the training and testing results shown in Figure 10. Table 4 summarizes the similarities between the ACES algorithm and the algorithms proposed in Zhang et al. (2021) and Song (2013). Another key difference between ACES and both of the previous algorithms is the training/testing procedure used to optimize and validate model parameters.

Original Paper	Similarities to ACES	Original Use	ACES Use
Song (2013)	Use algorithm 2 to estimate cell-based registrations	Call cells Congruent Matching Cells if their registrations are close to a reference value. Classify a cartridge case pair as a match if the CMC count is at least 6.	Compute six summative features based on full-scan and cell registrations. Use features in a classifier model.
Zhang et al. (2021)	Use DBSCAN algorithm to identify cells that reach a consensus registration	Classify a cartridge case pair as a match if a DBSCAN cluster is identified.	Compute four numerical features based on DBSCAN clusters across both comparison directions. Use features in a classifier model.

Table 4: Comparison of the ACES algorithm to previous work. Although ACES shares similarities to previously-proposed algorithms, it includes additional nuance by computing features across both comparison directions and using these features in a classifier model.

Table 5 shows the test classification error rates of the Congruent Matching Cells (CMC) algorithm proposed in Song (2013), the C_0-based classifier like the one proposed in Zhang et al. (2021), and the All ACES logistic regression model. We obtained the CMC results by applying the implementation available in the cmcR R package (Zemmels et al. 2022) on the same test data set used in the Results section. We selected CMC parameters based on which maximized the AUC across various CMC count threshold on the training data, which resulted in translation thresholds $T_x = T_y = 25$ pixels, rotation

threshold $T_\theta = 6°$, correlation threshold $T_{CCF} = 0.45$. The CMC results in Table 5 summarize the classification error rates based on selecting a CMC count threshold that achieved the most balanced false positive and false negative error rates on the training data, which was $T_{CMC} = 3$. The C_0-based error rates are the same as those shown in the first row of Figure 8.

We see that the only C_0-based classifier has the highest overall error rate of 9.5%, although this consists of far more balanced false positive and false negative error rates compared to the CMC method. Despite selecting CMC model parameters to maximize the AUC and balancing the false positive and false negative error rates, the CMC method misclassifies truly matching cartridge case at a much higher rate than truly non-matching pairs. In our experimentation, we noticed that many truly matching cartridge case pairs were assigned low CMC count similarity scores, which justifies the error rate imbalance in Table 5. The logistic regression model trained on the full ACES data set achieves the lowest overall, false positive, and false negative error rates across the three classification methods. We note that there were other combinations of parameters for the CMC and C_0-based classifiers resulting in higher overall accuracy. However, the improved accuracy were due to a strong tendency to classify pairs as non-matches (the majority class in the training data), so we didn't feel it appropriate to include in the final results.

Classification Method	Error (%)	False Positive (%)	False Negative (%)
CMC method	3.9	2.3	25.8
Only C_0 feature	9.5	9.6	9.2
ACES LR	2.3	2.2	3.8

Table 5: Testing classification error, false positive, and false negative rates for four types of classifier models. The CMC method results are derived from the implementation available in Zemmels et al. (2022). The "Only C_0 feature" classifier is analogous to the classification rule used in Zhang et al. (2021). The last row shows results from the Logistic Regression classifier trained on the all 19 ACES features.

Both the registration and density-based features aim to measure similarities between two cartridge case surfaces. These features embody the notion assumed in the CMC methodology that matching cartridge cases should have similar markings, so their cell-based correlations should be large and estimated registrations should agree. However, Figure 4 demonstrates that even non-matching cartridge case pairs may share similar markings. We are bound to find similarities if that is all we look for, so it is worthwhile to also consider dissimilarities. The visual diagnostic features accomplish this by partitioning scans into similar and different regions. The similarities vs. differences ratio and labeled neighborhood size features measure how extreme the differences are between two scans while the differences correlation features determine whether there are similarities among the different regions.

This direct comparison of the surface values aligns with the Theory of Identification which says that an examination should involve the comparison of the "relative height or depth, width, curvature and spatial relationship" of cartridge case impressions (AFTE Criteria for Identification Committee 1992). Comparison algorithms like ACES will inevitably be used to augment the opinion of a forensic examiner, who may need to

present algorithmic results to judges or juries as part of their expert testimony. As such, it is important that forensic examiners are able to interpret and explain the results of a comparison algorithm. The visual diagnostic features are useful for explaining the behavior of the algorithm in a manner that aligns with more traditional identification theory.

5.2. Sensitivity to Parameter Choice

When selecting the optimal models presented in Figure 8, we performed a good deal of searching across various parameter choices. For example, given the relative importance of the density-based features illustrated in Figure 11, we were interested in assessing the sensitivity of the various classifier models to DBSCAN parameter choice. Figure 12 shows a heat map of AUC values for the four combinations of feature group and classifier model shown in Figure 8 across a grid of DBSCAN parameter values $\epsilon \in \{3, ..., 15\}$ and $minPts \in \{3, ..., 10\}$. Darker tiles correspond with higher AUC values, which in turn are associated with more preferred models. We draw a black square around the specific $(\epsilon, minPts)$ values resulting in the maximum AUC for each of the models (which we also show in Figure 7). Interestingly, we see that all four models achieve optimum AUC for smaller values of ϵ and $minPts$. Larger values of ϵ will naturally lead to larger clusters as the ϵ-neighborhood around each point grows. It makes sense then for ϵ to remain small so as to avoid the formation of false positive clusters. Conversely, larger values of $minPts$ will naturally lead to fewer clusters, albeit of larger size. The fact that the optimal ϵ and $minPts$ are both relatively small suggests that matching comparisons may not have many cells that "agree" on a registration, but the cells that do agree form a strong consensus (i.e., form tight clusters).

We also note the variability in the AUC values across the DBSCAN parameter space. Specifically, we see that the "C_0 + Registration" models achieve the highest AUCs along a set of values in the bottom-left corner - where $\epsilon \approx minPts$ for $\epsilon, minPts < 10$. In our experimentation, we noticed that these $\epsilon, minPts$ values are also where the cluster indicator C_0 feature has highly-ranked importance (as is the case in Figure 13). Both the AUC values and the variable importance of C_0 decrease as either ϵ or $minPts$ increase, which indicates that the C_0 + Registration models rely heavily on C_0 to distinguish between matching and non-matching comparisons. Comparatively, we see that the AUC values for the "All ACES" models are more consistently high across the parameter space, indicating a robustness to parameter choice.

To better understand the behavior of the AUC values in Figure 12, consider Figure 13 showing the Mean Gini Decrease for each of the 19 ACES features across the grid of $\epsilon \in \{3, ..., 15\}$ and $minPts \in \{3, ..., 10\}$ values. We see that the density Cluster Indicator C_0, Cluster Size C, and Translation Difference Δ_{trans} have high variable importance for some combinations of ϵ and $minPts$, but are generally less consistent than the registration cell-based and full scan correlations $\overline{cor}_{\text{cell}}$ and cor_{full}. This implies that the density features can be highly informative under optimal conditions, yet quickly lose importance under sub-optimal conditions. In our experimentation, we noticed that other ACES features "take up the mantle" when C_0 or C have low importance, which explains the relative stability in AUC values observed in the "All ACES"-trained models shown in Figure 12.

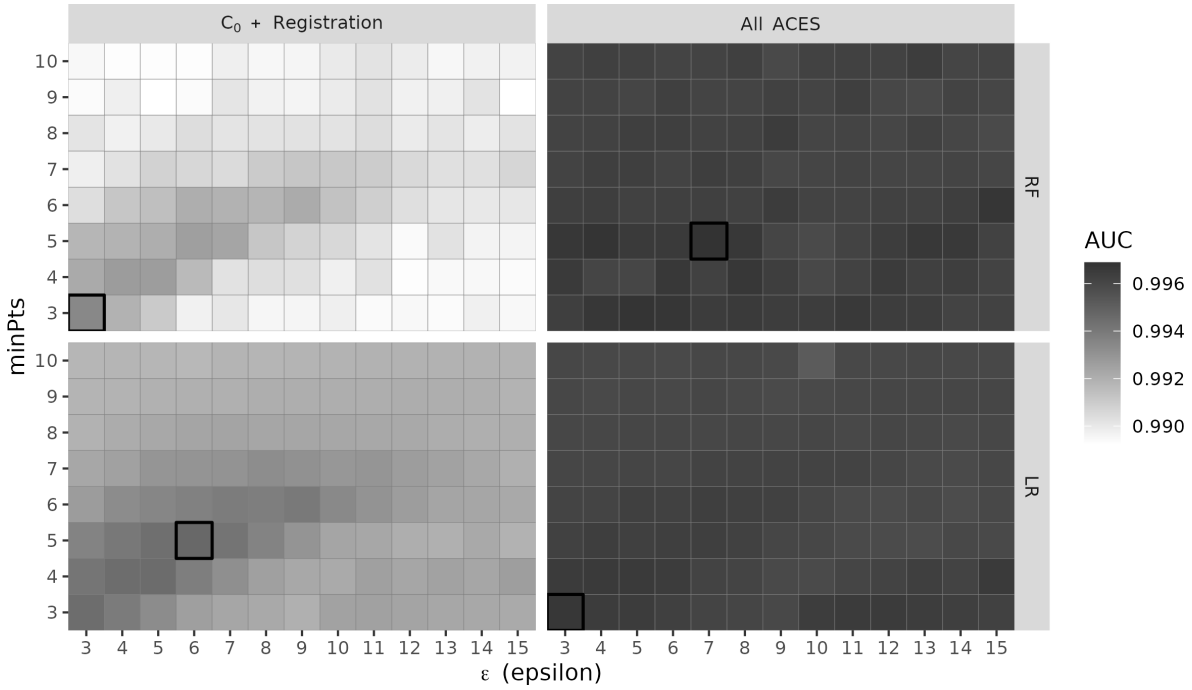

Figure 12: A heat map of AUC values associated with four classifier models across a grid of values for the two DBSCAN parameter ϵ and $minPts$. Darker tiles correspond with higher AUC. The four models are a combination of two feature groups (C_0 + Registration vs. All ACES) and two models (Random Forest and Logistic Regression). The "All ACES"-trained models have higher and more consistent AUCs compared to the "C_0 + Registration"-trained models.

The relationship between C_0 and C (first two plots in the first row) is noteworthy by the sharp boundary defined by the line $minPts = \epsilon$. Above this line, when $minPts > \epsilon$, we see that C_0 is considered more important than C. In other words, as the minimum size required to be classified as a cluster ($minPts$) and the neighborhood radius (ϵ) both become stricter, there will naturally be fewer clusters formed. In these instances, knowing whether a cluster is formed is more informative than the size of that cluster. However, the importance of C_0 below the line, when $minPts < \epsilon$, is seemingly replaced by C rising in importance. That is, when $minPts$ and ϵ are less strict, then more clusters will form and the actual size of the cluster becomes more informative than the fact that it exists.

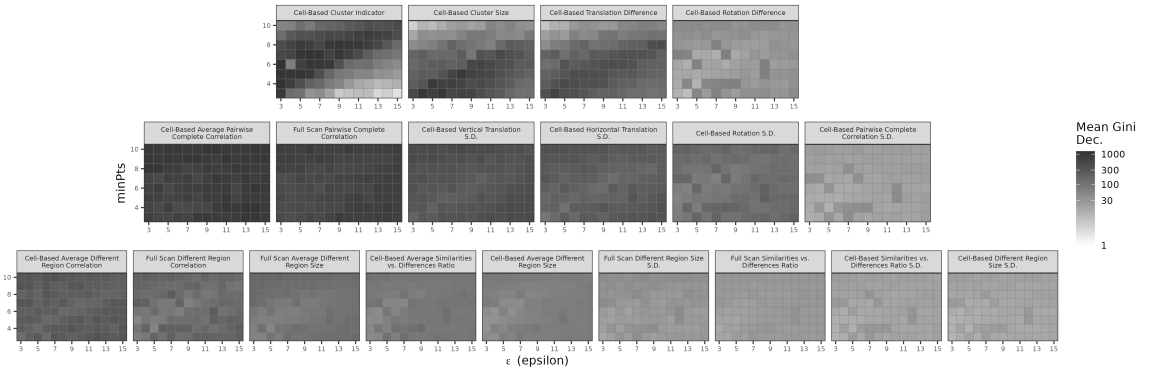

Figure 13: A heat map of variable importance measures for the 19 features in the ACES data set across a grid of values for the two DBSCAN parameter ϵ and $minPts$. Darker tiles correspond with higher importance. As in Figure 11, we visualze the importance measures on a log-transformed color scale to more clearly see variability among smaller values. We see that features like the Cell-Based Cluster Indicator C_0 and Cell-Based Cluster Size C have inconsistent importance measures across the difference values of ϵ and $minPts$ while the cell-based and full scan correlation features $\overline{cor}_{\text{cell}}$ and cor_{full} have more consistent importance.

The ϵ and $minPts$ values are not the only parameters that can be tuned. We also computed the ACES features using two different cell grids: 4×4 and 8×8. Each cell in the 4×4 grid captures a larger portion of the cartridge case's surface compared to the 8×8 grid, which would presumably be useful for the visual diagnostic-based features. However, the 8×8 grid has the benefit of having more cells with which to measure consensus using the registration and density-based features. Our experimentation showed that the 4×4 cell grid features resulted in categorically better classification results compared to the 8×8 features. Throughout this paper, we present the 4×4 results.

One would expect that having more cells would make it easier to measure the consensus. However, even the density-based features, such as the cluster size feature C, had better separation between matching and non-matching comparisons when the 4×4 cell grid was used compared to the 8×8 grid. Further, performing classification using a combination of the 4×4 and 8×8 features actually led to *lower* overall test accuracy compared to just using the 4×4 grid. We chalk this outcome up to the specific cell-based registration procedure we used. Recall from algorithm 1 that we first perform a pre-registration using the full scans and a rotation grid $\boldsymbol{\Theta}$. Using the full scan-estimated rotation θ_d^*, we

then perform the cell-based comparison procedure algorithm 2 using a limited rotation grid $\mathbf{\Theta}'_d = \{\theta^*_d - 2°, \theta^*_d - 1°, ..., \theta^*_d + 2°\}$.

To save on computational time, rather than comparing each source cell to the full target scan, we compare it to a slightly larger region that is located in the same position in the target scan (1.1× the side length of the source cell). Assuming the full scan registration was successful, a source cell isn't allowed to "move" very far in this region to register. This in contrast to, for example, the original CMC methodology proposed in Song (2013) where each source cell is compared to a region in the target scan that is much larger, 2 or 3 times the side length. The consequence of our implementation is that cells, even for non-matching comparisons, tend to have registration values close to the origin (i.e., no movement), and are therefore more likely to form DBSCAN clusters compared to if we used larger target regions. In this case, a higher number of cells actually leads to a higher chance of forming "false positive" clusters for a non-match comparison, which is exactly what we observe from the 8×8 comparisons. The formation of false positive clusters is far rarer in the 4×4 cell grid case. We hypothesize that the 8×8 cell grid results could improve if a different target region size were used. However, our implementation makes sense pragmatically, since the CCF computation grows exponentially with the size of the region cell or target region, and conceptually, since the full scan registration should result in a rough alignment of two matching scans before applying the cell-based comparison with a finer rotation grid.

5.3. Model Selection Considerations

Our intention in fitting the logistic regression and random forest classification models using different feature sets is to explore each model's strengths and weaknesses. A critical step in putting the ACES algorithm into practice will be to settle on a single model. Pragmatically, it seems reasonable to choose the model with the highest estimated accuracy on available test data. However, we noted that models trained by this optimization criterion on imbalanced data tend to over-classify the majority class. This is the case for the CMC method results we summarized in Table 5, but is also true for ACES statistical models trained to maximize overall accuracy. For example, if we were to shift the similarity score classification threshold for the All ACES logistic regression model to maximize the overall accuracy on the training data, the resulting score threshold is 0.54 with test accuracy, true negative, and true positive rates of 99.4%, 99.9%, and 92.4%. Given the large true negative rate, we might favor this model from an ethical perspective since misclassifying a truly non-matching cartridge case pair may incriminate an innocent individual. However, the true positive rate is considerably lower than the "balanced" results summarized in Figure 8. Further exploration of different optimization criteria is warranted.

Another aspect to consider when choosing a model is interpretability and explainability. If an algorithm is applied in forensic casework, then evidentiary conclusions derived from the algorithm's output will inevitably be presented to a non-expert judge or jury. More interpretable models are easier to understand, and therefore should be preferred. The classification behavior of the logistic regression and classification tree models are arguably easier to explain than the random forest model. For example, the logistic regression model parameters can be understood in terms of the estimated increase in

odds of a match. Paired with its comparable performance to the random forest, we propose the logistic regression model with all 19 ACES features as the preferred model that balances interpretability with accuracy.

6. Conclusion

In this paper, we introduced the Automatic Cartridge Evidence Scoring (ACES) algorithm to measure the similarity between two fired cartridge cases based on their breech face impressions. In particular, we defined a set of 19 similarity features and used these features to train and test classifier models. We validated our algorithm on a set of 510 cartridge cases - the largest validation study of a cartridge case similarity scoring algorithm to-date. Compared to predominant algorithms like the CMC algorithm described in Song (2013), the ACES logistic regression model achieves higher test accuracy rates while having more balanced true positive and true negative rates. We propose a logistic regression classifier trained on the ACES feature set as a new benchmark to which future scoring methods are compared.

Before the ACES algorithm can be put into practice, we must devise new stress-tests, using new ammunition and firearm combinations, to assess its robustness. There is also an opportunity to optimize additional parameters, such as the number of cells used in algorithm 2 or parameters used in pre-processing, to measure their effects on final results. A variety of factors, such as make/model and wear of the evidence or the algorithm parameters used, affect the discriminative power of the 19 features defined in this paper. We view the current version of the ACES algorithm as more a foundation for future improvements than a final solution. We expect the ACES feature set to evolve over time; for discriminatory features to replace less informative features. Given the gravity of the application, we stress interpretability as a guiding principle for future feature engineering and model selection. A misunderstood feature or result may lead a lay judge or juror to an incorrect conclusion. Additionally, we urge future researchers to use a train/test procedure similar to the one outlined in this paper to validate proposed methods.

We developed the scored R package as an open-source companion to this paper. The code and data used in this paper are available at `https://github.com/jzemmels/jdssvSubmission`.

Computational Details

If necessary or useful, information about certain computational details such as version numbers, operating systems, or compilers could be included in an unnumbered section. Also, auxiliary packages (say, for visualizations, maps, tables, ...) that are not cited in the main text can be credited here.

The results in this paper were obtained using R~4.2.2.R Core Team (2019) R itself and all packages used are available from the Comprehensive R Archive Network (CRAN) at https://CRAN.R-project.org/.

We generated this report using the RStudio graphical user interface (RStudio Team 2020) a number of R packages including knitr (Xie 2014), rmarkdown (Xie et al. 2020), the tidyverse suite (Wickham et al. 2019).

Acknowledgments

This work was partially funded by the Center for Statistics and Applications in Forensic Evidence (CSAFE) through Cooperative Agreement 70NANB20H019 between NIST and Iowa State University, which includes activities carried out at Carnegie Mellon University, Duke University, University of California Irvine, University of Virginia, West Virginia University, University of Pennsylvania, Swarthmore College and University of Nebraska, Lincoln.

References

AFTE Criteria for Identification Committee (1992). Theory of identification, range striae comparison reports and modified glossary definitions. *AFTE Journal*, 24(3):336–340.

Baldwin, D. P., Bajic, S. J., Morris, M., and Zamzow, D. (2014). A Study of False-Positive and False-Negative Error Rates in Cartridge Case Comparisons. Technical report, Ames Lab IA, Performing, Fort Belvoir, VA, DOI: 10.21236/ADA611807.

Barthelme, S. (2023). *imager: Image Processing Library Based on 'CImg'*, https://CRAN.R-project.org/package=imager. R package version 0.42.18.

Breiman, L. (2001). Random Forests. *Machine Learning*, 45(1):5–32, DOI: 10.1023/a:1010933404324, http://dx.doi.org/10.1023/A:1010933404324.

Brigham, E. O. (1988). *The Fast Fourier Transform and Its Applications*. Prentice-Hall, Inc., USA, ISBN: 0133075052.

Brown, L. G. (1992). A survey of image registration techniques. *ACM Computing Surveys*, 24(4):325–376, DOI: 10.1145/146370.146374, https://doi.org/10.1145/146370.146374.

Chen, Z., Song, J., Chu, W., Soons, J. A., and Zhao, X. (2017). A convergence algorithm for correlation of breech face images based on the congruent matching cells (CMC)

method. *Forensic Science International*, 280:213–223, ISSN: 03790738, https://doi.org/10.1016/j.forsciint.2017.08.033.

Cooley, J. and Tukey, J. (1965). An algorithm for the machine calculation of complex fourier series. *Mathematics of Computation*, 19(90):297–301.

Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, page 226–231. AAAI Press.

Fernández, A., García, S., Galar, M., Prati, R. C., Krawczyk, B., and Herrera, F. (2018). *Learning from Imbalanced Data Sets*. Springer International Publishing, DOI: 10.1007/978-3-319-98074-4, https://doi.org/10.1007/978-3-319-98074-4.

Haralick, R. M., Sternberg, S. R., and Zhuang, X. (1987). Image analysis using mathematical morphology. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9(4):532–550, DOI: 10.1109/tpami.1987.4767941. https://doi.org/10.1109/tpami.1987.4767941.

Hare, E., Hofmann, H., and Carriquiry, A. (2017). Automatic matching of bullet land impressions. *The Annals of Applied Statistics*, 11(4):2332–2356, ISSN: 19326157, http://www.jstor.org/stable/26362188.

Hastie, T., Tibshirani, R., and Friedman, J. (2001). *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA.

Hesselink, W. H., Meijster, A., and Bron, C. (2001). Concurrent determination of connected components. *Science of Computer Programming*, 41(2):173–194, DOI: 10.1016/s0167-6423(01)00007-7, https://doi.org/10.1016/s0167-6423(01)00007-7.

Hofmann, H., Carriquiry, A., and Vanderplas, S. (2021). Treatment of inconclusives in the AFTE range of conclusions. *Law, Probability and Risk*, 19(3-4):317–364, ISSN: 1470-8396, DOI: 10.1093/lpr/mgab002, https://doi.org/10.1093/lpr/mgab002.

Hofmann, H., Vanderplas, S., Krishnan, G., and Hare, E. (2022). *x3ptools: Tools for Working with 3D Surface Measurements*, https://github.com/heike/x3ptools. R package version 0.0.3.9000.

ISO 25178-72:2017 (2017). Geometrical product specifications (GPS) — Surface texture: Areal — Part 72: XML file format x3p. Standard, International Organization for Standardization, Geneva, CH, https://www.iso.org/standard/62310.html.

James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). *An Introduction to Statistical Learning: with Applications in R*. Springer, https://faculty.marshall.usc.edu/gareth-james/ISL/.

Kuhn, M. (2022). *caret: Classification and Regression Training*, https://CRAN.R-project.org/package=caret. R package version 6.0-91.

Liaw, A. and Wiener, M. (2002). Classification and regression by randomforest. *R News*, 2(3):18–22, https://CRAN.R-project.org/doc/Rnews/.

MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. In Cam, L. M. L. and Neyman, J., editors, *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press.

National Research Council (2009). *Strengthening forensic science in the United States: a path forward*. The National Academies Press, Washington, D.C.

PCAST, P. (2016). Forensic science in criminal courts: Ensuring scientific validity of feature-comparison methods. Technical report, Executive Office of The President's Council of Advisors on Science and Technology, Washington DC.

R Core Team (2019). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, https://www.R-project.org/.

RStudio Team (2020). *RStudio: Integrated Development Environment for R*. RStudio, PBC., Boston, MA, http://www.rstudio.com/.

Song, J. (2013). Proposed "NIST Ballistics Identification System (NBIS)" Based on 3D Topography Measurements on Correlation Cells. *American Firearm and Tool Mark Examiners Journal*, 45(2):11, https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=910868.

Swofford, H. and Champod, C. (2021). Implementation of algorithms in pattern & impression evidence: A responsible and practical roadmap. *Forensic Science International: Synergy*, 3:100142, DOI: 10.1016/j.fsisyn.2021.100142, http://dx.doi.org/10.1016/j.fsisyn.2021.100142.

Tai, X. H. and Eddy, W. F. (2018). A Fully Automatic Method for Comparing Cartridge Case Images,. *Journal of Forensic Sciences*, 63(2):440–448, ISSN: 00221198, http://doi.wiley.com/10.1111/1556-4029.13577.

Tong, M., Song, J., and Chu, W. (2015). An Improved Algorithm of Congruent Matching Cells (CMC) Method for Firearm Evidence Identifications. *Journal of Research of the National Institute of Standards and Technology*, 120:102, ISSN: 2165-7254, https://doi.org/10.6028/jres.120.008.

Venables, W. N. and Ripley, B. D. (2002). *Modern Applied Statistics with S*. Springer, New York, fourth edition, http://www.stats.ox.ac.uk/pub/MASS4. ISBN 0-387-95457-0.

Vorburger, T. V., Song, J., and Petraco, N. (2015). Topography measurements and applications in ballistics and tool mark identifications. *Surface Topography: Metrology and Properties*, 4(1):013002, DOI: 10.1088/2051-672x/4/1/013002, https://doi.org/10.1088/2051-672x/4/1/013002.

Vorburger, T. V., Yen, J. H., Bachrach, B., Renegar, T. B., Filliben, J. J., Ma, L., Rhee, H. G., Zheng, A., Song, J. F., Riley, M., Foreman, C. D., and Ballou, S. M. (2007). Surface topography analysis for a feasibility assessment of a national ballistics imaging database. Technical Report NIST IR 7362, National Institute of Standards and Technology, Gaithersburg, MD, DOI: 10.6028/NIST.IR.7362, https://nvlpubs.nist.gov/nistpubs/Legacy/IR/nistir7362.pdf. Edition: 0.

Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L. D., François, R., Grolemun, G., Haye, A., Henr, L., Heste, J., Kuh, M., Pederse, T. L., Mille, E., Bach, S. M., Müll, K., , J. O., , D. R., , D. P. S., , V. S., , K. T., , D. V., , C. W., , K. W., and , H. Y. (2019). Welcome to the tidyverse. *Journal of Open Source Software*, 4(43):1686, DOI: 10.21105/joss.01686.

Xiao Hui Tai (2018). Comparing cartridge breechface marks: 2d versus 3d. https://forensicstats.org/blog/portfolio/comparing-cartridge-breechface-marks-2d-versus-3d/.

Xie, Y. (2014). knitr: A comprehensive tool for reproducible research in R. In Stodden, V., Leisch, F., and Peng, R. D., editors, *Implementing Reproducible Computational Research*. Chapman and Hall/CRC. ISBN 978-1466561595.

Xie, Y., Dervieux, C., and Riederer, E. (2020). *R Markdown Cookbook*. Chapman and Hall/CRC, Boca Raton, Florida, ISBN: 9780367563837, https://bookdown.org/yihui/rmarkdown-cookbook.

Zemmels, J., Hofmann, H., and VanderPlas, S. (2022). *cmcR: An Implementation of the 'Congruent Matching Cells' Method*. R package version 0.1.9.

Zemmels, J., VanderPlas, S., and Hofmann, H. (2023). A study in reproducibility: The congruent matching cells algorithm and cmcR package. *The R Journal*, 14(4):79–102, DOI: 10.32614/rj-2023-014, https://doi.org/10.32614/rj-2023-014.

Zhang, H., Zhu, J., Hong, R., Wang, H., Sun, F., and Malik, A. (2021). Convergence-improved congruent matching cells (CMC) method for firing pin impression comparison. *Journal of Forensic Sciences*, 66(2):571–582, ISSN: 1556-4029, DOI: 10.1111/1556-4029.14634, https://onlinelibrary.wiley.com/doi/abs/10.1111/1556-4029.14634.

Zheng, X., Soons, J., Vorburger, T. V., Song, J., Renegar, T., and Thompson, R. (2014). Applications of surface metrology in firearm identification. *Surface Topography: Metrology and Properties*, 2(1):014012, DOI: 10.1088/2051-672x/2/1/014012, https://doi.org/10.1088/2051-672x/2/1/014012.

A. Appendix

A.1. Registration Procedure Details

A registration is composed of a discrete translation by $(m, n) \in \mathbb{Z}^2$ and rotation by $\theta \in [-180°, 180°]$. Under this transformation, the index i, j maps to a new index i^*, j^* by:

$$\begin{pmatrix} j^* \\ i^* \end{pmatrix} = \begin{pmatrix} n \\ m \end{pmatrix} + \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix} \begin{pmatrix} j \\ i \end{pmatrix}.$$

The value b_{ij} now occupies the index i^*, j^*. In practice, we use *nearest-neighbor interpolation* meaning i^* and j^* are rounded to the nearest integer.

To determine the optimal registration, we calculate the *cross-correlation function* (CCF) between A and B, which measures the similarity between A and B for every possible translation of B. Denoted $(A \star B)$, the CCF between A and B is a 2D array of dimension $2k - 1 \times 2k - 1$ with the m, n-th element given by:

$$(a \star b)_{mn} = \sum_{i=1}^{k} \sum_{j=1}^{k} a_{mn} \cdot b_{i+m, j+n}$$

where $1 \leq m, n \leq 2k - 1$. The value $(a \star b)_{mn}$ quantifies the similarity between A and B after B is translated m elements horizontally and n elements vertically. The CCF is often normalized between -1 and 1 for interpretability.

The above definition of the CCF is computationally taxing, particularly for large matrices. The Cross-Correlation Theorem provides an equivalent expression for the CCF:

$$(A \star B) = \mathcal{F}^{-1}\left(\overline{\mathcal{F}(A)} \odot \mathcal{F}(B)\right)$$

where $\mathcal{F}$ and $\mathcal{F}^{-1}$ are the discrete Fourier and inverse discrete Fourier transforms, respectively, $\overline{\mathcal{F}(A)}$ is the complex conjugate, and $\odot$ is an element-wise (Hadamard) product (Brigham 1988). We trade the moving sum computation from the previous CCF expression for two forward Fourier transforms, an element-wise product, and an inverse Fourier transform. The Fast Fourier Transform (FFT) algorithm reduces the computational load considerably (Cooley and Tukey 1965).

We estimate the registration by calculating the maximum CCF value across a range of rotations of matrix B. Let B_θ denote B rotated by an angle $\theta \in [-180°, 180°]$ and $b_{\theta_{mn}}$ the m, n-th element of B_θ. Then the estimated registration (m^*, n^*, θ^*) is:

$$(m^*, n^*, \theta^*) = \arg\max_{m,n,\theta} (a \star b_\theta)_{mn}.$$

In practice we consider a discrete grid of rotations $\boldsymbol{\Theta} \subset [-180°, 180°]$. The registration procedure is outlined in algorithm 1. We refer to the matrix that is rotated as the "target." The result is the estimated registration of the target matrix to the "source" matrix.

It is common for cartridge case scans to contain many missing values. For example, the gray pixels in Figure 3 represent structural values in the scan. The Fast Fourier Transform algorithm used in algorithm 1 does not permit missing values in A or B. Thus, when calculating the CCF we impute these missing values with the average non-missing value in the scan. To measure the similarity between A and B while accounting for missingness, we calculate the correlation between the non-missing intersection of the aligned scans.

Cell-Based Registration Details

Following the full-scan registration, we next perform a cell-based registration procedure. Song (2013) points out that breech face impressions rarely appear uniformly on a cartridge case surface. Rather, distinguishing markings appear in specific, usually small, regions of a scan (the author refers to these as *valid correlation regions*). Calculating a correlation between two whole scans does not necessarily capture the similarity between these regions. Song (2013) proposes partitioning a scan into a rectangular grid of "cells" to isolate the valid correlation regions. Figure 4 shows an example of two non-match cartridge cases where the source matrix (left) is partitioned into an 8×8 grid of cells.

The cell-based comparison procedure begins with selecting one of the matrices, say A, as the "source" matrix to be partitioned into a grid of cells. Each of these source cells will be compared to the "target" matrix, in this case B^*. Because A and B^* are already partially aligned based on the course rotation grid $\boldsymbol{\Theta}$, we compare each source cell to B^* using a new rotation grid of $\boldsymbol{\Theta}'_A = \{\theta^*_A - 2°, \theta^*_A - 1°, \theta^*_A, \theta^*_A + 1°, \theta^*_A + 2°\}$.

If two cartridge cases are truly matching, then we assume that multiple cells will "agree" on a particular translation value at the true rotation. This agreement phenomenon is illustrated in Figure 14 where each point represents the translation that maximizes the CCF for a particular cell and rotation. The points appear randomly distributed for most of the rotation values except around $\theta = 3$ where a tight cluster of points forms around translation $[m, n] \approx [17, -16]$. This is evidence to suggest that a true registration exists for these two cartridge cases, implying that they match. The task is to determine when cells reach a registration consensus.

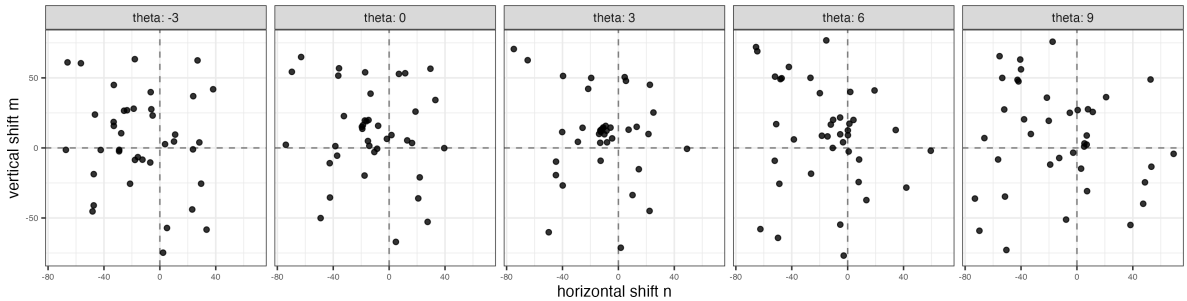

Figure 14: A scatterplot where points represent the cell-wise estimated translations faceted by rotation for a matching pair of cartridge cases. As evidenced by the tight cluster in the middle facet, it appears that multiple cells agree on a translation of $[m, n] \approx [17, -16]$ after rotating by 3°. Points are jittered for visibility.

Registration-Based Feature Distributions

Figure 15 shows density plots of the registration-based features for 21,945 cartridge case pairs. The first two rows show densities for the sample mean and standard deviation of the cell-based registrations, respectively. The third row shows densities for the pairwise-complete correlation features. The standard deviation of the cell-based registrations discriminate more between match vs. non-match pairs than the sample means, which justifies their inclusion in the final feature set. [More to say here?]

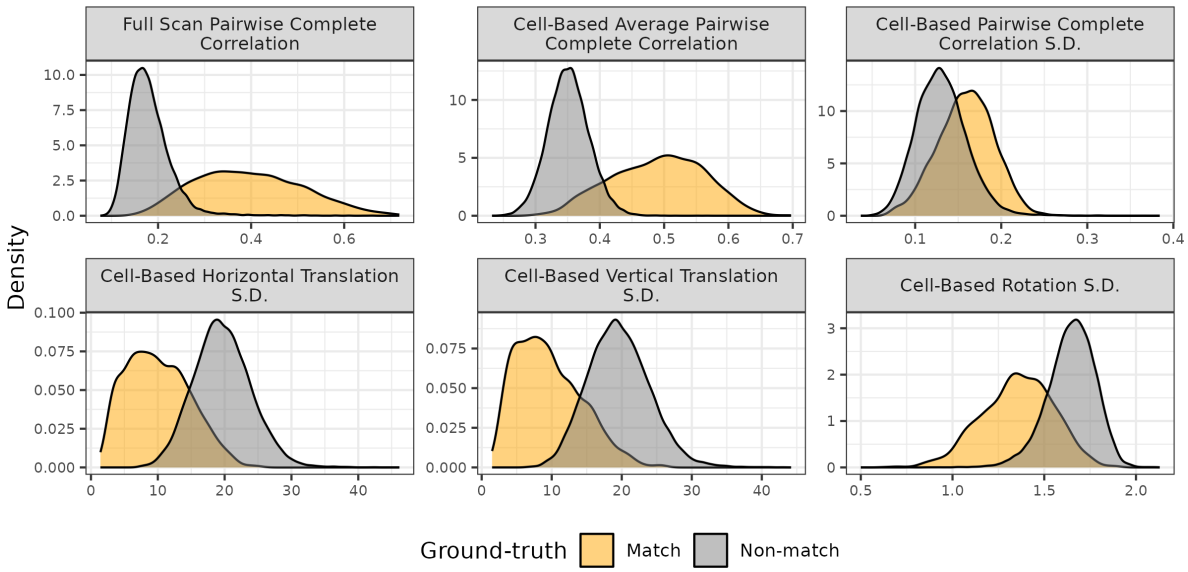

Figure 15: Density plots of the Registration-Based features for 21,945 cartridge case pairs. The standard deviation of the cell-based registrations distinguish between match and non-match pairs better than the mean values.

A.2. DBSCAN Algorithm Details

Density-Based Feature Distributions

Figure 16 shows the distributions of the density-based features C, Δ_θ, and Δ_{trans}. The stacked bar chart in the top-left shows the proportion of comparisons where no DBSCAN cluster is identified by outcome (match or non-match). We see that the vast majority of comparisons for which no DBSCAN cluster is identified are non-match comparisons. indicating that C_0 is a good indicator of ground-truth. In fact, there is only one non-match comparison that resulted in a DBSCAN cluster. It's difficult to see in the plots, but the C value for this non-match pair is 5 and the Δ_{trans} value is 23.9. As expected, C tends to be relatively large for matching comparisons while Δ_θ and Δ_{trans} tends to be small.

A.3. Visual Diagnostic Details

The Complementary Comparison Plot visualizes the similarities and differences between two scans. Figure 17 shows a Complementary Comparison plot between scan A and B^*

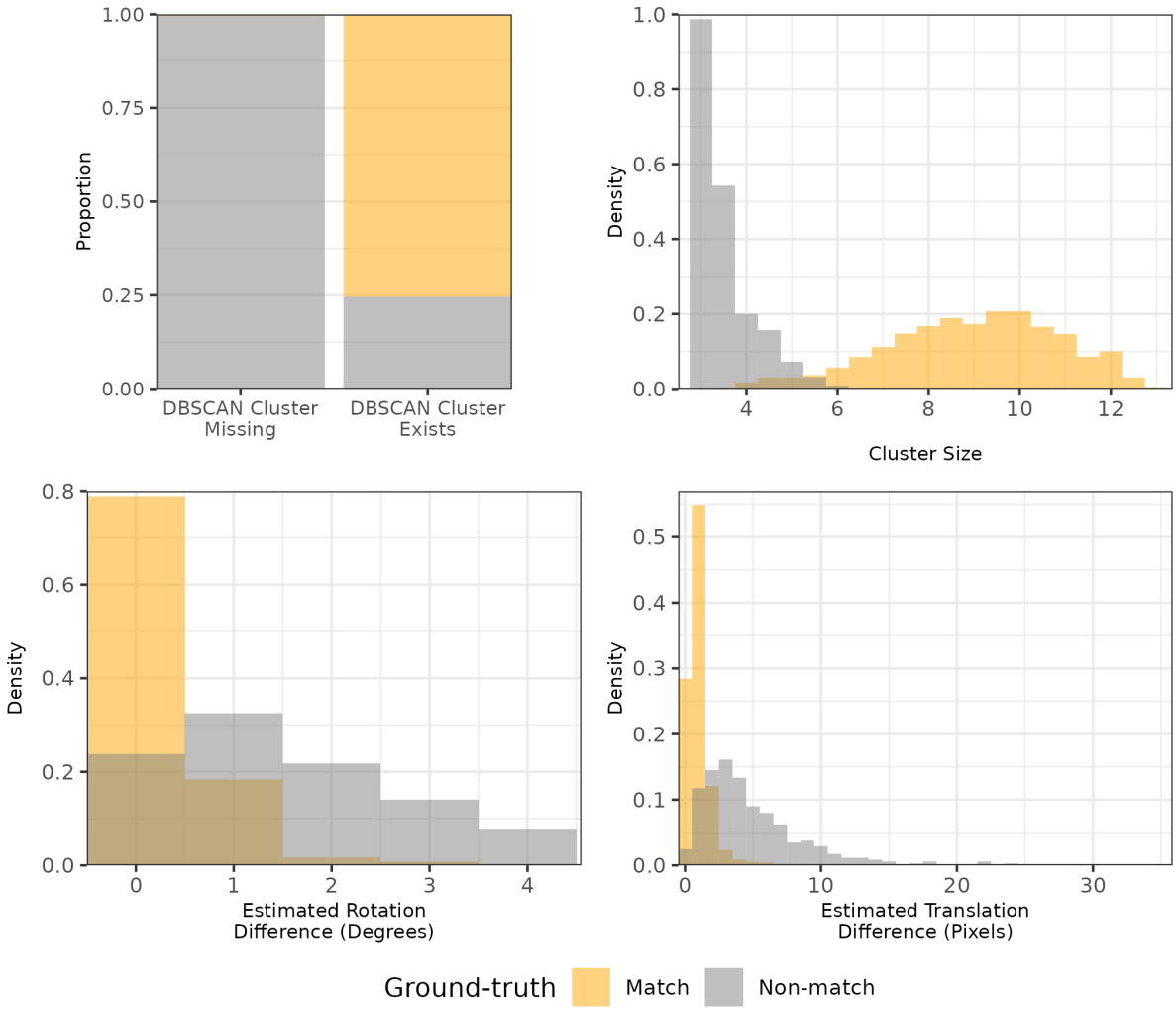

Figure 16: Distributions of the density-based features for 21,945 cartridge case pairs. The Cluster Size and Estimated Translation Difference features may be missing (`NA`) if no DBSCAN cluster is identified, which commonly occurs for non-matching cartridge case pairs as evidenced by the stacked bar chart in the top left.

defined previously. The left column shows Scans A and B^*. The middle column shows a filtered element-wise average between A and B^*; namely $\mathcal{F}_{|A-B^*|<\tau}\left(\frac{1}{2}(A+B^*)\right)$. This filtered element-wise average emphasizes similarities between A and B^*. The right column shows $\mathcal{F}_{|A-B^*|>\tau}(A)$ and $\mathcal{F}_{|A-B^*|>\tau}(B^*)$ on top and bottom, respectively. These plots emphasize the differences between the two scans. The complementary comparison plot is a powerful tool for assessing the estimated alignment and identifying similarities and differences between two surface matrices. We repeat this in the other comparison direction ($d = B$) to obtain filtered matrices $\mathcal{F}_{|A^*-B|\leq\tau}\left(\frac{1}{2}(A^*+B)\right)$, $\mathcal{F}_{|A^*-B|>\tau}(A^*)$ and $\mathcal{F}_{|A^*-B|>\tau}(B)$.[4]

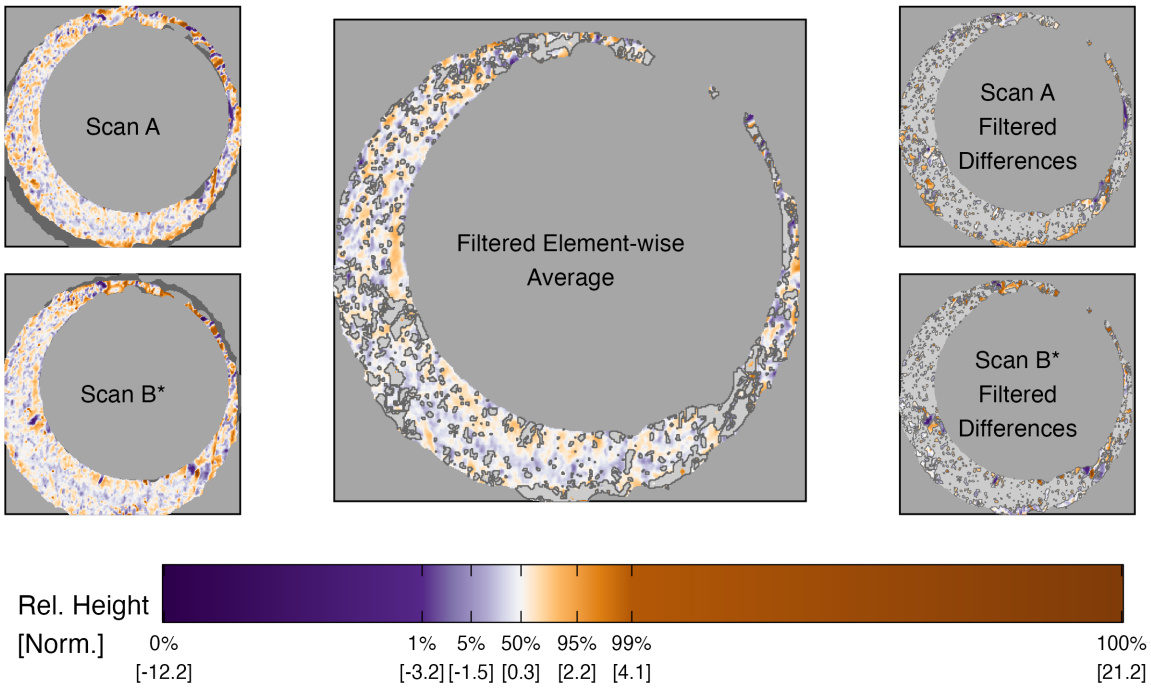

Figure 17: Full scan comparison plot.

We make a series of qualitative assumptions related to how a Complementary Comparison Plot will look for matching and non-matching cartridge case pairs. We develop a set of features that measure the degree to which these assumptions are met by a particular cartridge case pair.

Visual Diagnostic Feature Distributions

Figure 18 shows the distribution of the six visual diagnostic-based features. As expected, matching comparisons at the full-scan and cell-based levels tend to have smaller neighborhood sizes and higher correlation values on average.

A.4. Model-Specific Results

[4] As with the registration-based features, in reality these matrices should be equivalent across the two comparison directions. However, there are slight differences due to the discretely-indexed nature of the surface matrices.

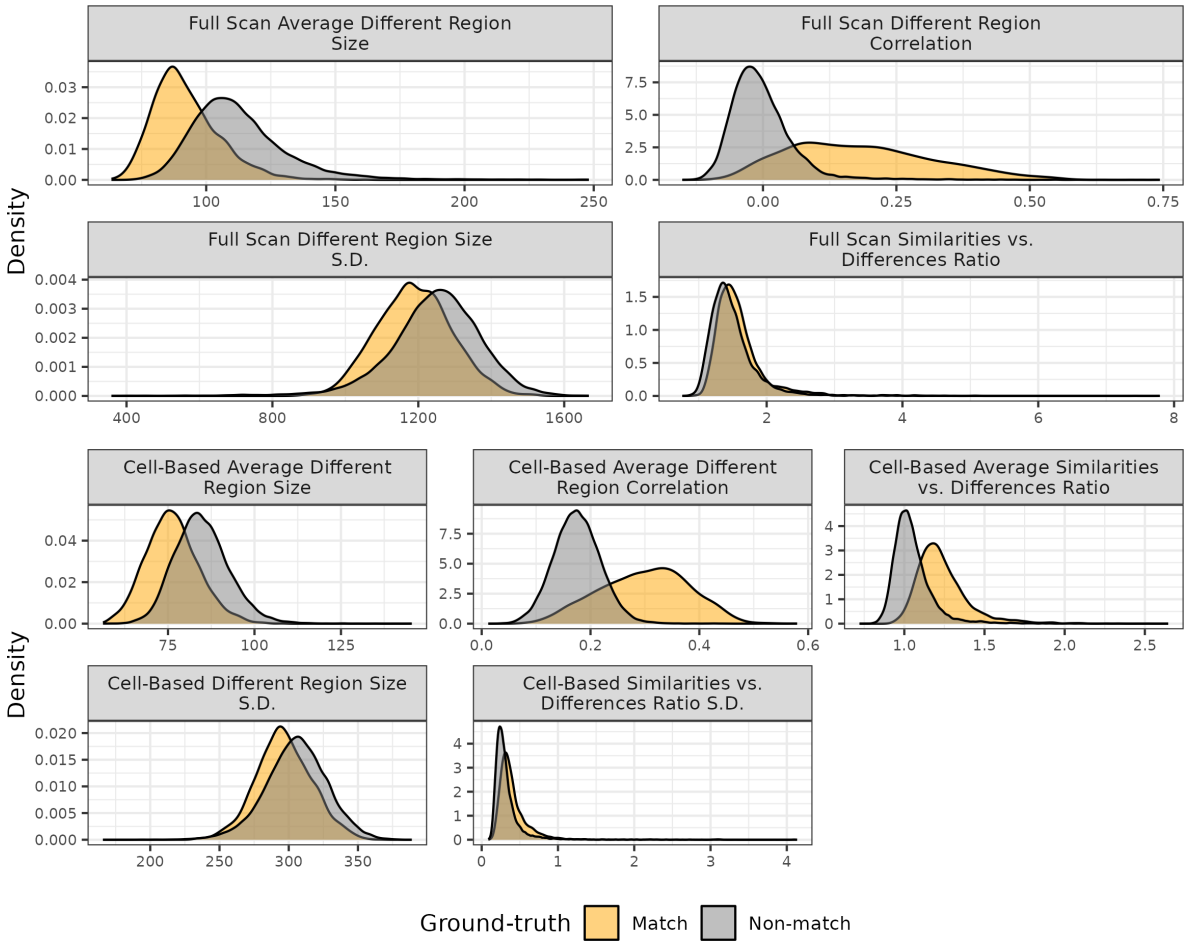

Figure 18: Distributions of the visual diagnostic-based features for 21,945 cartridge case pairs. Matching comparisons tend to have smaller neighborhood sizes on average and higher correlation values than non-matches indicating their utility in a classifier.

Table 6: Accuracy, True Positive, and True Negative rates based on the training data for the 13 binary classifier models. This table shows a numeric summary of the results shown in Figure 8. We bold the largest values in each column for emphasis.

Feature Set	Model	Accuracy	True Neg.	True Pos.
C_0	Baseline	90.47	90.43	90.77
C_0 + Registration	RF	97.16	97.15	97.18
C_0 + Registration	LR	97.23	97.23	97.23
All ACES	RF	98.14	98.14	98.14
All ACES	LR	**98.3**	**98.3**	**98.27**

Table 7: Accuracy, True Positive, and True Negative rates based on the test data for the 13 binary classifier models. This table shows a numeric summary of the results shown in Figure 8. We bold the largest values in each column for emphasis.

Feature Set	Model	Accuracy	True Neg.	True Pos.
C_0	Baseline	89.44	89.64	86.82
C_0 + Registration	RF	96.91	97.11	94.22
C_0 + Registration	LR	96.98	97.21	93.87
All ACES	RF	97.66	**97.87**	94.74
All ACES	LR	**97.68**	97.81	**95.94**

Table 6 summarizes the accuracy, true positive, and true negative rates based on the training data for the 13 binary classifier models. We see that the Logistic Regression (LR) and Random Forest (RF) models perform comparably, particularly having the exact same True Negative rate in the last two rows of the table. The CART model performs consistently worse compared to the other two models.

Table 7 summarizes the accuracy, true positive, and true negative rates based on the test data for the 13 binary classifier models. We see that the Logistic Regression (LR) model performs slightly better than the Random Forest (RF) model in most cases while the CART model consistently lags behind the other two. The true positive rates for the test data are noticeably lower to those for the training data summarized in Table 6, although the true negative rates are similar.

Affiliation:

Joseph Zemmels
Iowa State University
Center for Statistics and Applications in Forensic Evidence
Iowa State University
195 Durham Center
613 Morrill Road
Ames, IA 50011
E-mail: jzemmels@iastate.edu
URL: https::/jzemmels.github.io

Susan VanderPlas
University of Nebraska - Lincoln
Department of Statistics
University of Nebraska - Lincoln
343D Hardin Hall
3310 Holdrege St
Lincoln, NE 68588
E-mail: susan.vanderplas@unl.edu
URL: https://srvanderplas.netlify.app/

Heike Hofmann
Iowa State University
Center for Statistics and Applications in Forensic Evidence
Iowa State University
195 Durham Center
613 Morrill Road
Ames, IA 50011
E-mail: heike@iastate.edu
URL: https://github.com/heike

Journal of Data Science, Statistics, and Visualisation https://jdssv.org/
published by the International Association for Statistical Computing
http://iasc-isi.org/

MMMMMM YYYY, Volume VV, Issue II *Submitted:* yyyy-mm-dd
doi:XX.XXXXX/jdssv.v000.i00 *Accepted:* yyyy-mm-dd