

**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   Susan VanderPlas   Heike Hofmann**  
Iowa State University   University of Nebraska - Lincoln   Iowa State University

---

### Abstract

Forensic examinations attempt to solve the binary classification problem of whether two pieces of evidence originated from the same source. A cartridge case found at a crime scene may be compared to a cartridge case fired from a suspect's firearm. Historically, forensic examiners relied 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 a cartridge case comparison algorithm that encompasses preprocessing, feature extraction, and similarity scoring. We use a train/test split on a data set of 500 cartridge case scans to fit and validate a random forest model. We demonstrate that this random forest model yields improved accuracy compared to predominant algorithms. Finally, we use the random forest model to calculate score-based likelihood ratios that estimate the probative value of the evidence.

---

*Keywords:* forensics, forensic statistics, pattern recognition, firearms and toolmarks, R.

---

## 1. Introduction

Introduce the problem here. Explain what a cartridge case is. Explain breech face impressions.

The “ground-truth” of a forensic comparison is a binary classification problem. Briefly reference how comparisons are done by examiners currently. Keep focus on firearm and toolmark evidence.

Critics of traditional firearm and toolmark comparisons cite a lack of foundational validity (NAS 2009, PCAST 2016). Discuss what PCAST means by foundational validity and how firearm and toolmark evidence falls short according to NAS & PCAST. Recent studies of examiner proficiency estimate error rates to be low - between % and % according to [Baldwin]. Nonetheless, [NAS] and [PCAST] pushed for the development of “objective image processing algorithms to....[quote PCAST here...].” An automatic comparison algorithm could be used as part of an examination to supplement or inform an examiner’s opinion [cite Swofford taxonomy paper here].

## 2. Previous Work

Discuss current state of affairs for algorithmic F&T comparisons.

Cite Hare et al. as a parallel paper to this one applied to bullet data.

Cite Xiao Hui’s project.

Cite CMC method as predominant method. Broadly summarize cell-based comparison procedure and CMC method logic. Also reference Zhang et al. (2020) DBSCAN paper here.

Discuss limitations of current cartridge case comparison algorithms. Currently, there is no rigorous procedure for comparing different algorithms. This includes selecting optimal parameters for a specific algorithm. In this work, we introduce a novel validation procedure to learn and validate optimal parameters using a cross-validation procedure.

We introduce a novel set of features to measure the similarity between two cartridge cases. using these features, we train and test a random forest model. We show that this random forest model improves upon the error rate of predominant automatic comparison algorithms. Additionally, we demonstrate how the random forest model can be used to calculate score-based likelihood ratios.

## 3. Cartridge Case Data

Discuss Baldwin study here. Point out that it was the only appropriately-designed study according to PCAST. Types of cartridge cases, firearms. Design of the experiment (known and questioned samples).

Details of scanning procedure using Cadre 3D-TopMatch High Capacity Scanner. Describe x3p file format and surface matrices.

## 4. Methods

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

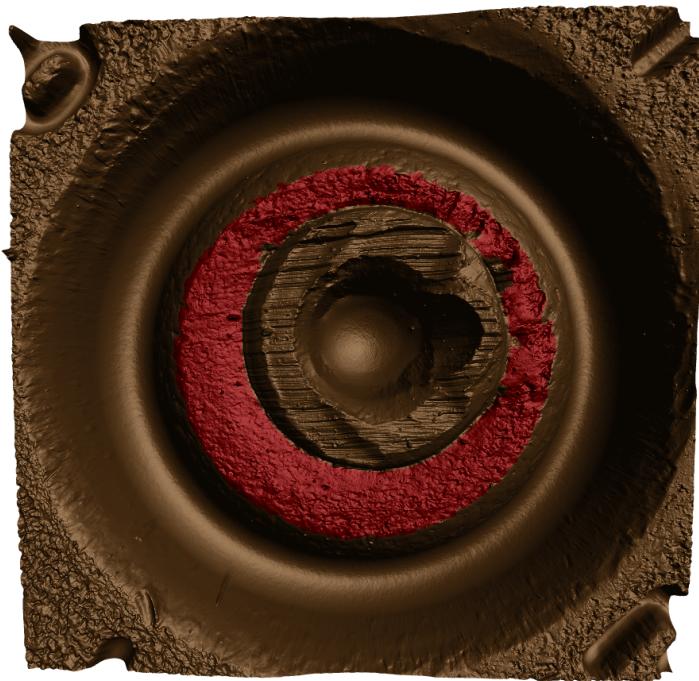
1. **Preprocessing:** prepare cartridge case scans for comparison

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 steps.

#### 4.1. Preprocessing

We first use the open-source FiX3P web application 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 hexadecimal 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 that are 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).



[Figure 1:](#) A cartridge case surface is manually annotated in red using the FiX3P software. The annotated region of interest contains breech face impressions.

Once read into an R environment, we use sequence of functions available in the [x3ptools] and [cmcR] packages to preprocess the raw scans. [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 [cite erosion reference]. 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.

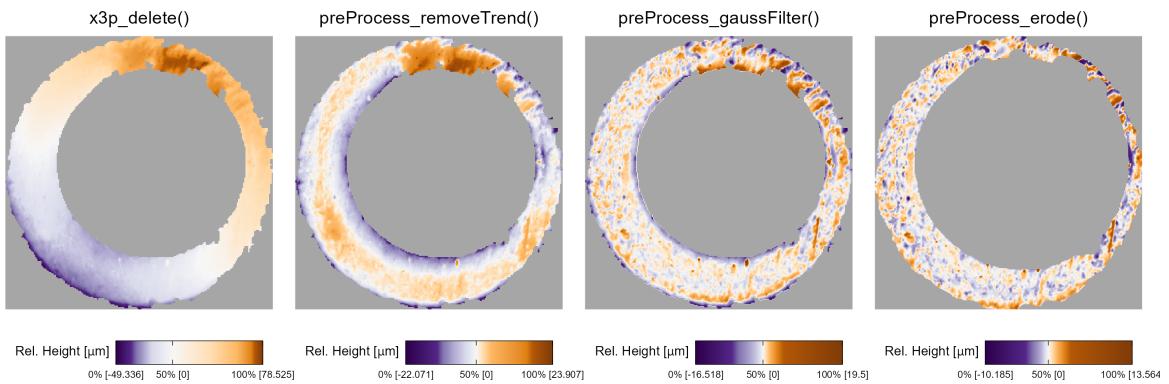


Figure 2: We apply a sequence of preprocessing functions to each scan. Each preprocessing step further emphasizes the breech face impressions in the scan.

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

## 4.2. Comparing

In this section, we introduce a set of similarity features for two cartridge case scans. We calculate features at two scales: between two whole scans and between individual cells similar to the CMC method [cite]. 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*

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  $k > 0$ .<sup>1</sup> 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  $A$ . Throughout this section, we will use 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 as follows: if an element of either matrix  $A$  or  $B$  is missing, then any element-wise operation including this element is also missing, otherwise standard matrix algebra holds. For

<sup>1</sup>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.

example, the addition operator is 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  $+, -, >, <, \dots$  and assume the extended operations as defined above.

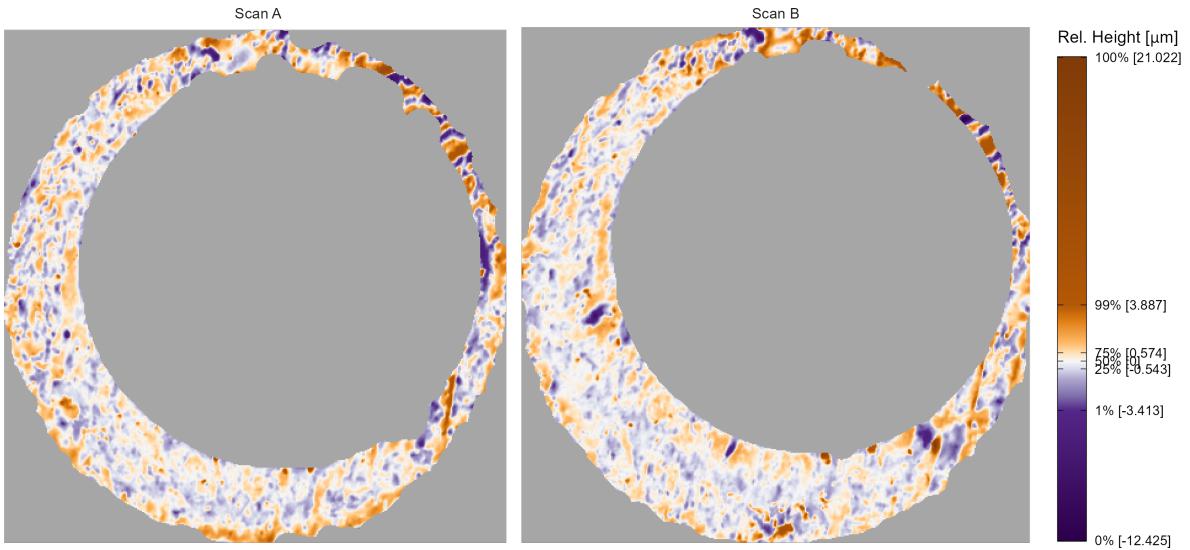


Figure 3: A matching pair of processed cartridge case scans. We measure the similarity between these cartridge cases using the distinguishable breech face impressions on their surfaces.

### 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, we rely on a standard image registration technique [cite Brown, 1992].

In our application, a registration is composed of a discrete translation by  $(m, n) \in \mathbb{Z}^2$  and rotation by  $\theta \in [-180^\circ, 180^\circ]$ . 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 [cite a nearest-neighbor reference].

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.

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

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

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 [cite 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 [cite Tukey].

Using the CCF as an objective function, we estimate the registration by calculating the maximum CCF value across a range of rotations of matrix  $B$ . Let  $B_\theta$  denote  $B$  rotated by an angle  $\theta \in [-180^\circ, 180^\circ]$  and  $b_{\theta,mn}$  the  $m, n$ -th element of  $B_\theta$ . Then the estimated registration  $(m^*, n^*, \theta^*)$  is:

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

In practice we consider a discrete grid of rotations  $\Theta \subset [-180^\circ, 180^\circ]$ . 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  $\Theta$

**Result:** Estimated registration of  $B$  to  $A$ ,  $(m^*, n^*, \theta^*)$ , and cross-correlation function maximum,  $CCF_{\max}$

**for**  $\theta \in \Theta$  **do**

- | Rotate  $B$  by  $\theta$  to obtain  $B_\theta$ ;
- | 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  $CCF_{\max} = \max_\theta \{CCF_{\max,\theta} : \theta \in \Theta\}$ ;

Calculate rotation  $\theta^* = \arg \max_\theta \{CCF_{\max,\theta} : \theta \in \Theta\}$ ;

**return** Estimated rotation  $\theta^*$ , translation  $m^* = m_{\theta^*}^*$  and  $n^* = n_{\theta^*}^*$ , and  $CCF_{\max}$

**Algorithm 1:** Image Registration Procedure

## Handling Missingness

[Not sure what to do with this section. It needs to be mentioned, but in more or less detail?]

The registration estimation procedure outline above, namely the Fast Fourier Transform algorithm, does not permit missing values in  $A$  or  $B$ . It is common for cartridge case scans to contain many missing values - the gray regions in [preprocessing Figure] represent structural values in the scan. Thus, when calculating the CCF we impute these missing values with the average non-missing value in the scan.

We wish to measure the similarity between  $A$  and  $B$  while taking this missingness into account; to measure the similarity between the non-missing intersection of the aligned scans. We 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  $\Theta = \{-30^\circ, -27^\circ, \dots, 27^\circ, 30^\circ\}$ . This results in an estimated registration  $(m^*, n^*, \theta^*)$  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 obtain  $A^*$ .

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  $d = A, B$ .<sup>2</sup> 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,  $cor_{full,A}$ , between  $A$  and  $B^*$ . We repeat this in the other comparison direction to obtain  $cor_{full,B}$  and average the two:

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

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

**Cell-Based Registration** 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 matrix  $A$  partitioned into a grid of  $8 \times 8$  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

---

<sup>2</sup>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.

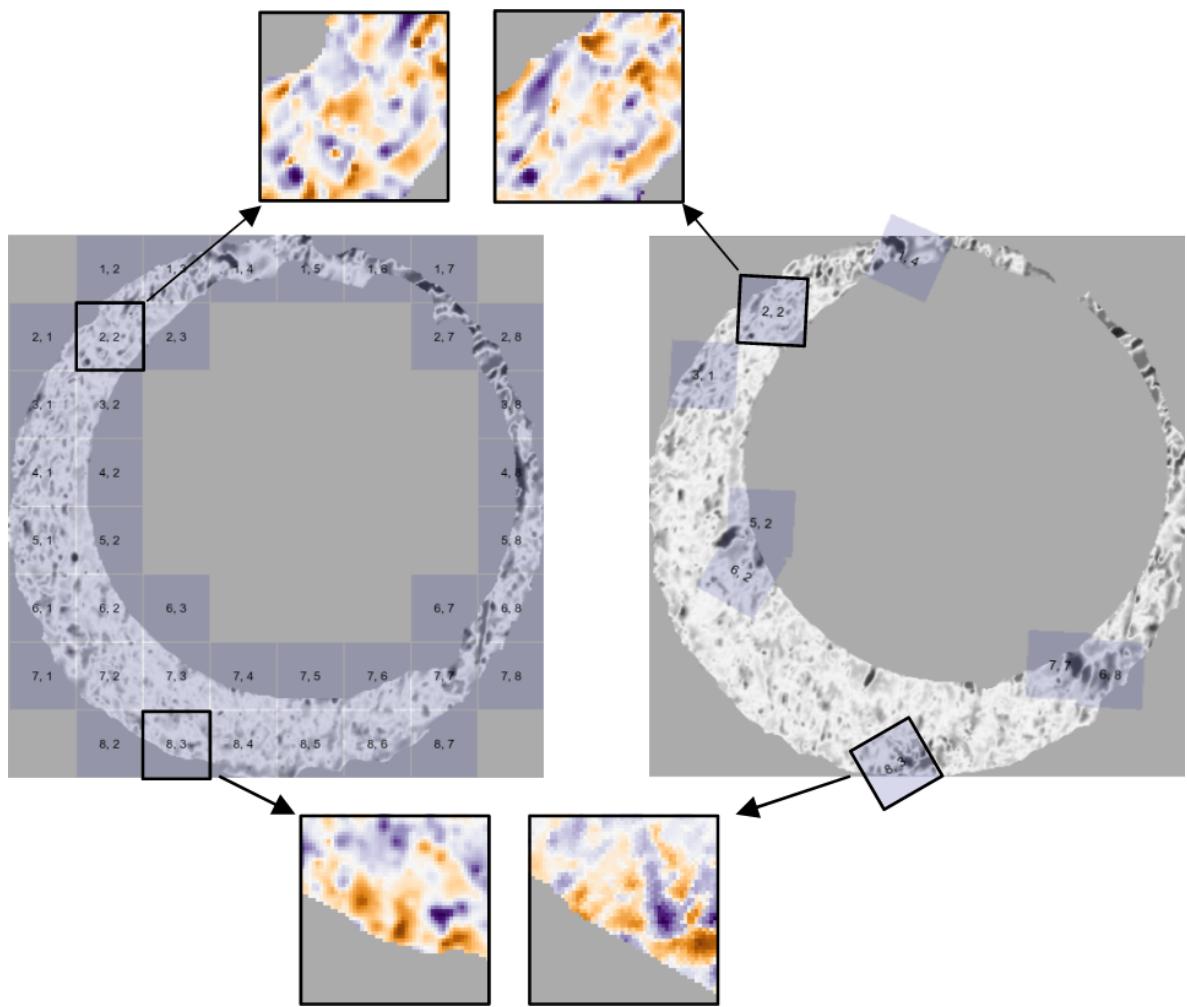


Figure 4: A source scan separated into a grid of  $8 \times 8$  cells. Each source cell is compared to a target scan to estimate where it aligns best. We exclude cells containing only missing values (visualized here as gray pixels).

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  $\Theta$ , we compare each source cell to  $B^*$  using a new rotation grid of  $\Theta'_A = \{\theta_A^* - 2^\circ, \theta_A^* - 1^\circ, \theta_A^*, \theta_A^* + 1^\circ, \theta_A^* + 2^\circ\}$ .

We now extend the surface matrix notation introduced previously to accommodate cells. Let  $A_t$  denote the  $t$ th cell of matrix  $A$ ,  $t = 1, \dots, T_A$  where  $T_A$  is the total number of cells containing non-missing values (e.g.,  $T_A = 38$  in Figure 4) in scan  $A$  and let  $(a_t)_{ij}$  denote the  $i, j$ -th element of  $A_t$ .

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

**Data:** Source matrix  $A$ , target matrix  $B^*$ , cell grid size  $R \times C$ , and rotation grid

$$\Theta'_A$$

**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**  $\theta \in \Theta'_A$  **do**

| Rotate  $B^*$  by  $\theta$  to obtain  $B_\theta^*$ ;

| **for**  $t = 1, \dots, T_A$  **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**  $\mathbf{F}_A = \{(m_{A,t,\theta}^*, n_{A,t,\theta}^*, CCF_{\max,A,t,\theta}, \theta) : \theta \in \Theta'_A, t = 1, \dots, T_A\}$

**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  $\mathbf{F}_A$  of translations and CCF values for each cell and each rotation considered. If two cartridge cases are truly matching, then we assume that multiple cells will “agree” on a particular translation value at the true rotation.<sup>3</sup> This agreement phenomenon is illustrated in Figure 5 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 [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.

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  $B_\theta^*$  after translating by  $[m_{A,\theta}^*, n_{A,\theta}^*]$ . From this we obtain a set of pairwise-complete correlations for each rotation:  $\{cor_{A,t,\theta} : \theta \in \Theta'_A\}$ . This whole procedure is repeated using  $B$  as the source scan and  $A^*$  as the target, resulting in registration set  $\mathbf{F}_B$  and pairwise-complete correlations  $\{cor_{B,t,\theta} : \theta \in \Theta'_B\}$ .

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

$$CCF_{\max,d,t} = \max_{\theta} \{CCF_{\max,d,t,\theta} : \theta \in \Theta'_d\}$$

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

---

<sup>3</sup>And that cells will not come to such an agreement for a non-matching pair of cartridge cases

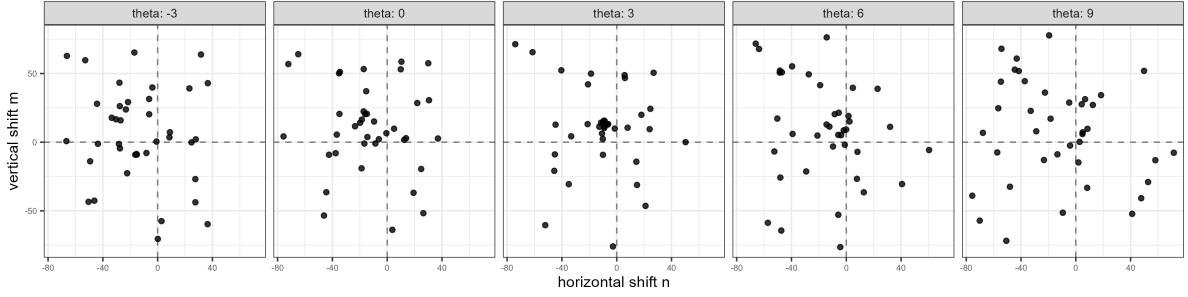


Figure 5: A scatterplot where points represent the cell-wise estimated translations faceted by rotation for a matching pair of cartridge cases. As evidenced by the tight cluster in the middle facet, it appears that multiple cells agree on a translation of  $[\hat{m}, \hat{n}] \approx [17, -16]$  after rotating by  $3^\circ$ . Points are jittered for visibility.

We compute the **average** and **standard deviation of the cell-based pairwise-complete correlations** features 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 the  $\overline{cor}_{\text{cell}}$  to be large and the  $s_{cor}$  small for truly matching cartridge case pairs.

For  $d = A, B$  and  $t = 1, \dots, T_d$ , define the per-cell estimated translations and rotation as:

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

$$m_{d,t}^* = m_{\theta_{d,t}^*, d, t}^*$$

$$n_{d,t}^* = n_{\theta_{d,t}^*, d, t}^*$$

We compute the **standard deviation of the cell-based estimated registration** using the estimated cell 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

$$\begin{aligned}\bar{m}^* &= \frac{1}{T_A + T_B} \sum_{d \in \{A, B\}} \sum_{t=1}^{T_d} m_{d,t}^* \\ \bar{n}^* &= \frac{1}{T_A + T_B} \sum_{d \in \{A, B\}} \sum_{t=1}^{T_d} n_{d,t}^* \\ \bar{\theta}^* &= \frac{1}{T_A + T_B} \sum_{d \in \{A, B\}} \sum_{t=1}^{T_d} \theta_{d,t}^*.\end{aligned}$$

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

Figure 6 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 exclusion from the final feature set. [More to say here?]

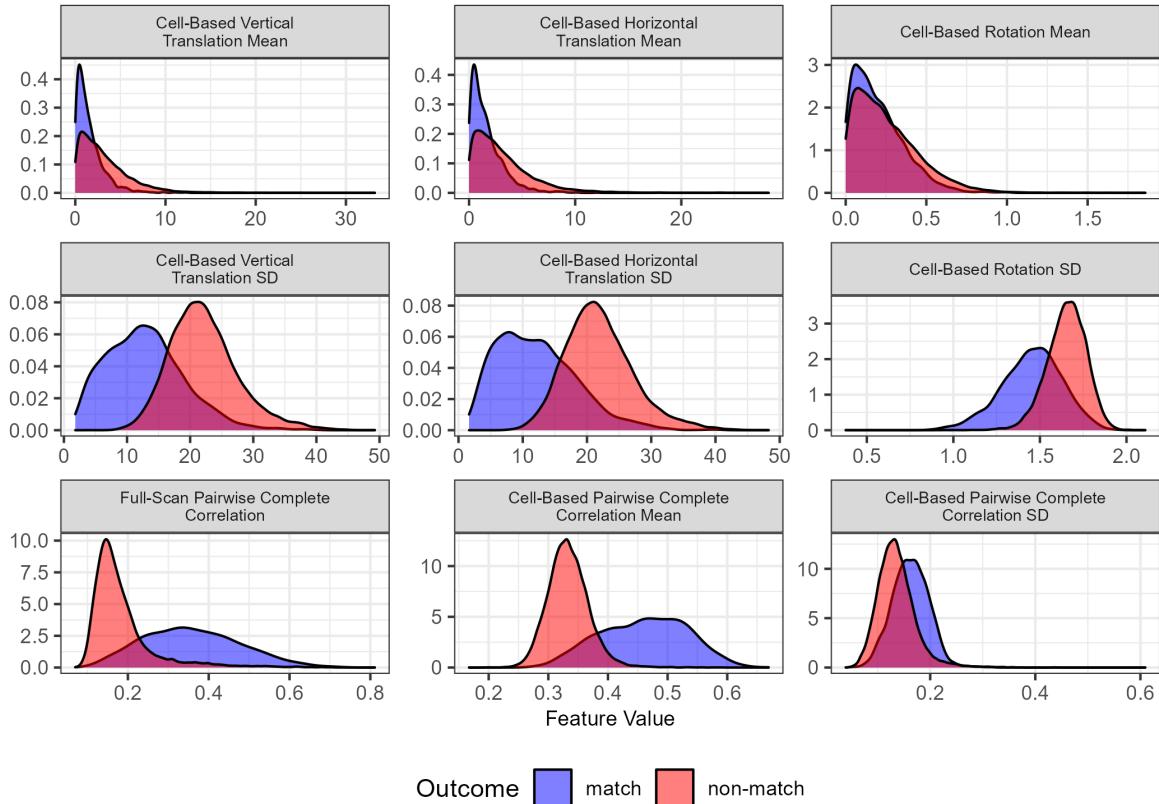


Figure 6: Density plots of the Registration-Based features for 21,945 cartridge case pairs. The standard deviation of the cell-based registrations distinguish between match and non-match pairs better than the mean values.

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

Notation	Feature Description
$cor_{full}$	<b>Full-scan pairwise-complete correlation</b> after aligning the scans.
$cor_{cell}$	<b>Average cell-based pairwise-complete correlation</b> after aligning source cells to the target matrix using the cross-correlation function
$s_{cor}$	<b>Standard deviation of the cell-based pairwise-complete correlation</b> after aligning source cells to the target matrix using the cross-correlation function
$s_m^*$	<b>Standard deviation of the cell-based vertical translations</b>
$s_n^*$	<b>Standard deviation of the cell-based horizontal translations</b>
$s_{\theta^*}$	<b>Standard deviation of the cell-based rotations</b>

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

### Density-Based Features

As discussed in the last section, we wish to identify when multiple cells agree on a particular registration. [Zhang et al. (2020)] proposed using the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm to identify clusters of points based on their density.

Figure 7 depicts an illustration of the DBSCAN algorithm [cite Wikimedia commons]. The algorithm has two parameters: a neighborhood radius  $\epsilon$  and a minimum point threshold  $Minpts$ . In Figure 7,  $Minpts = 4$  and  $\epsilon$  is arbitrary and represented by the radius of the circles drawn around each point - each circle represents the  $\epsilon$ -neighborhood for its center point. First, the algorithm identifies cluster “core” points that contain at least  $Minpts$  points within an  $\epsilon$  distance.<sup>4</sup> These points form the beginning of a cluster and are shown in red in Figure 7. The yellow points  $B$  and  $C$  are within the  $\epsilon$ -neighborhood of a core point, but are not themselves core points. They are also included in the cluster making the overall cluster size 8. Finally, the blue point labelled  $N$  is not in any core point’s  $\epsilon$ -neighborhood and is thus classified as a “noise point.” Unlike other clustering algorithms, the DBSCAN algorithm does not require a specified number of expected clusters as a parameter; any points not belonging to a cluster are “noise.”

Figure 8 shows an example of DBSCAN cluster assignments for the known-match pair  $A$  and  $B$  shown in Figure 3. The left scatterplot shows the per-cell estimated translations  $[m_{d,t,\theta}^*, n_{d,t,\theta}^*]$  for  $\theta = 3^\circ$  when scan  $A$  is used as source and  $B^*$  as target, resulting a cluster of size 14. The right scatterplot shows the per-cell estimated translations with the roles of  $A$  and  $B^*$  reversed: now  $B^*$  is partitioned into a grid of source cells that are compared to  $A$ , resulting in a cluster of size 13.

Because  $A$  and  $B$  are truly matching, we expect the estimated registrations in these two comparison directions to be opposites. Indeed, the mean cluster centers in Figure 8 are  $(\hat{m}_A, \hat{n}_A, \hat{\theta}_A) \approx (16.9, -16.7, 3^\circ)$  when  $A$  is used as source compared to  $(\hat{m}_B, \hat{n}_B, \hat{\theta}_B) \approx$

<sup>4</sup>Euclidean distance, in our application

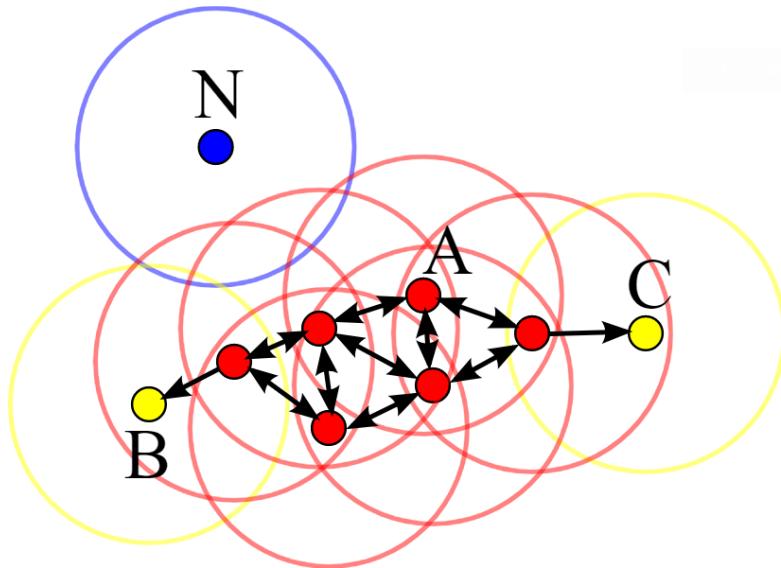


Figure 7: Illustration of the DBSCAN algorithm [cite]. Red and yellow points are part of the same cluster with the former forming the "core" of the cluster. The blue point is not part of a cluster and is classified as a "noise point." Figure by Chire - Own work, CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=17045963>

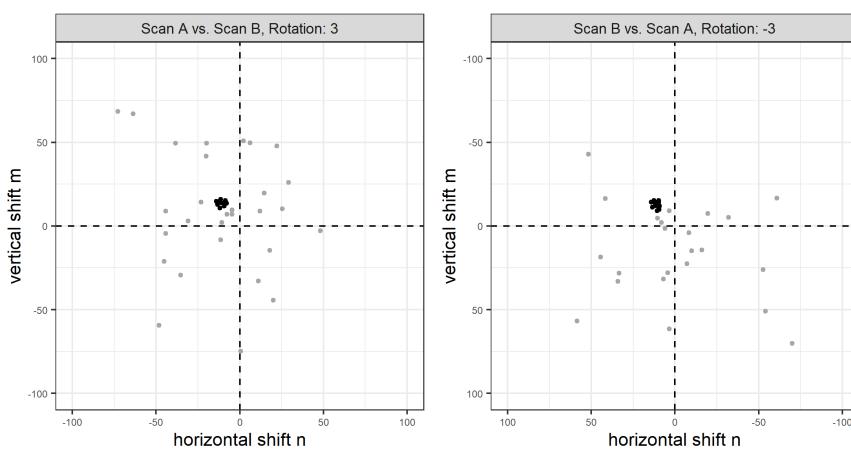


Figure 8: Cluster assignments based on the Density Based Spatial Clustering with Applications to Noise (DBSCAN) algorithm for estimated translations in two comparison directions. Using scan *A* as source results in a cluster of size 14 (left) compared to 13 when scan *B* is used as source (right). Noting the reversed axes in the right plot, we see that the clusters are located approximately opposite of each other. Points are jittered for visibility.

$(-16.2, 16.8, -3^\circ)$  when  $B^*$  is used as source.

We calculate numerical features based on the DBSCAN cluster assignments. We first use a 2D kernel density estimator [cite kde2d from MASS(?)] 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  $\{(m_{d,t,\hat{\theta}_d}^*, n_{d,t,\hat{\theta}_d}^*) : t = 1, \dots, T_d\}$  like those shown in Figure 8.<sup>5</sup> Let  $\mathcal{C}_d$  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 consider features related to whether a DBSCAN cluster is identified in both comparison directions and, if such clusters are identified, the average size of the clusters. We also compare the density-estimated rotations and translations across the two comparison directions. These are summarized in the **average DBSCAN cluster size**, the **DBSCAN cluster indicator**, and the **root squared sum of the density-estimated registrations**:

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

where  $|\mathcal{C}_d|$  denotes the cardinality of  $\mathcal{C}_d$  and  $I(\cdot)$  is the identify function equals 1 if the predicate argument “.” 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  $C_0$ .

Figure 9 shows the distributions of the density-based features  $C$ ,  $\Delta_\theta$ , and  $\Delta_{\text{trans}}$ . 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 outcome. 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  $\Delta_{\text{trans}}$  value is 23.9. As expected,  $C$  tends to be relatively large for matching comparisons while  $\Delta_\theta$  and  $\Delta_{\text{trans}}$  tends to be small.

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

### Visual Diagnostic Features

The final set of features we calculate are based on visual diagnostic tools described in [Zemmels et al. (2023)]. These numerical features quantify the qualitative observations one can make from the diagnostics.

---

<sup>5</sup>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 coarse 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.

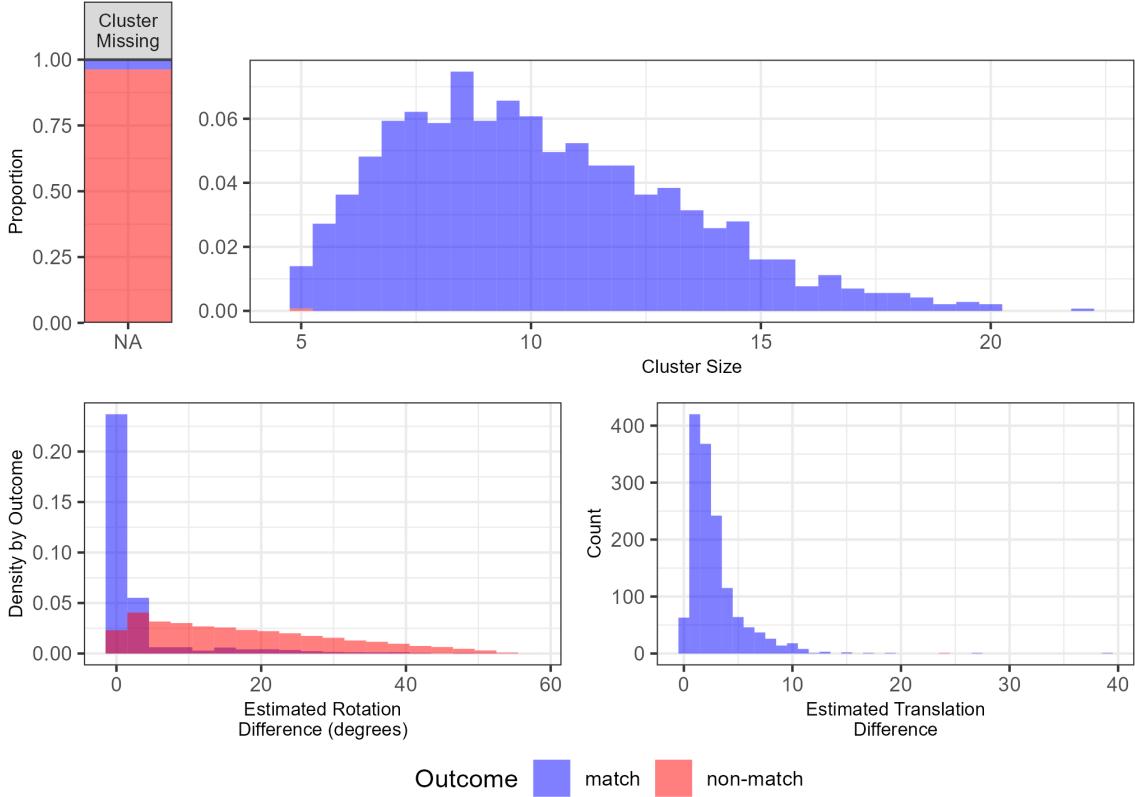


Figure 9: Distributions of the density-based features for 21,945 cartridge case pairs. The Cluster Size and Estimated Translation Difference features may be missing (NA) if no DBSCAN cluster is identified, which commonly occurs for non-matching cartridge case pairs as evidenced by the stacked bar chart in the top left. This explains the near absence of non-matching comparisons from Cluster Size and Estimated Translation Difference plots. Whether a cluster is identified for a particular comparison strongly predicts whether it is a match or a non-match, which justifies the inclusion of the cluster indicator feature  $C_0$ .

Notation	Feature Description
$C$	<b>Average DBSCAN cluster size</b> across both comparison directions
$C_0$	<b>DBSCAN cluster indicator</b> of whether DBSCAN clusters exist in both comparison directions
$\Delta_\theta$	<b>Absolute sum of the density-estimated rotations</b> between both comparison directions
$\Delta_{\text{trans}}$	<b>Root sum of squares of the cluster-estimated translations</b> between both comparison directions

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

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

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

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  $\tau \in \mathbb{R}$ . Surface values in  $A$  and  $B^*$  that are “close,” meaning within  $\tau$  distance, to each other are replaced with  $NA$  in this filtered matrix.

The Complementary Comparison Plot visualizes the similarities and differences between two scans. [Figure 10](#) shows a Complementary Comparison plot between scan  $A$  and  $B^*$  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  $\mathcal{F}_{|A^*-B|<\tau}\left(\frac{1}{2}(A^* + B)\right)$ ,  $\mathcal{F}_{|A^*-B|>\tau}(A^*)$  and  $\mathcal{F}_{|A^*-B|>\tau}(B)$ .<sup>6</sup>

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. We now describe each feature and their associated assumptions.

We first consider the correlation  $cor_{d,\text{full,filt}}$  between the filtered matrices  $\mathcal{F}_{|A-B^*|>\tau}(A)$  and  $\mathcal{F}_{|A-B^*|>\tau}(B^*)$  when  $d = A$  and  $\mathcal{F}_{|A^*-B|>\tau}(A^*)$  and  $\mathcal{F}_{|A^*-B|>\tau}(B)$  when  $d = B$ . The average of these is used as a feature:

$$\overline{cor}_{\text{full,filt}} = \frac{1}{2} (cor_{A,\text{full,filt}} + cor_{B,\text{full,filt}}).$$

We assume that **average filtered full-scan pairwise-complete correlation** will be larger for truly matching  $A$  and  $B$  than non-matching. Said another way, we assume that even surface regions of  $A$  and  $B$  that are different will follow similar trends, which can occur due to variability in the amount of contact between a cartridge case and breech face across multiple fires of a single firearm. The correlation is calculated by vectorizing the two filtered surface matrices and treating missing values by case-wise deletion.

As before, we extend this notation to accommodate cell comparisons  $t = 1, \dots, T_d$  for  $d = A, B$  using subscripts:  $cor_{d,t,\text{filt}}$ . For example,  $cor_{A,t,\text{filt}}$  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  $B_{t,\theta_t^*}$  is the

---

<sup>6</sup>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.

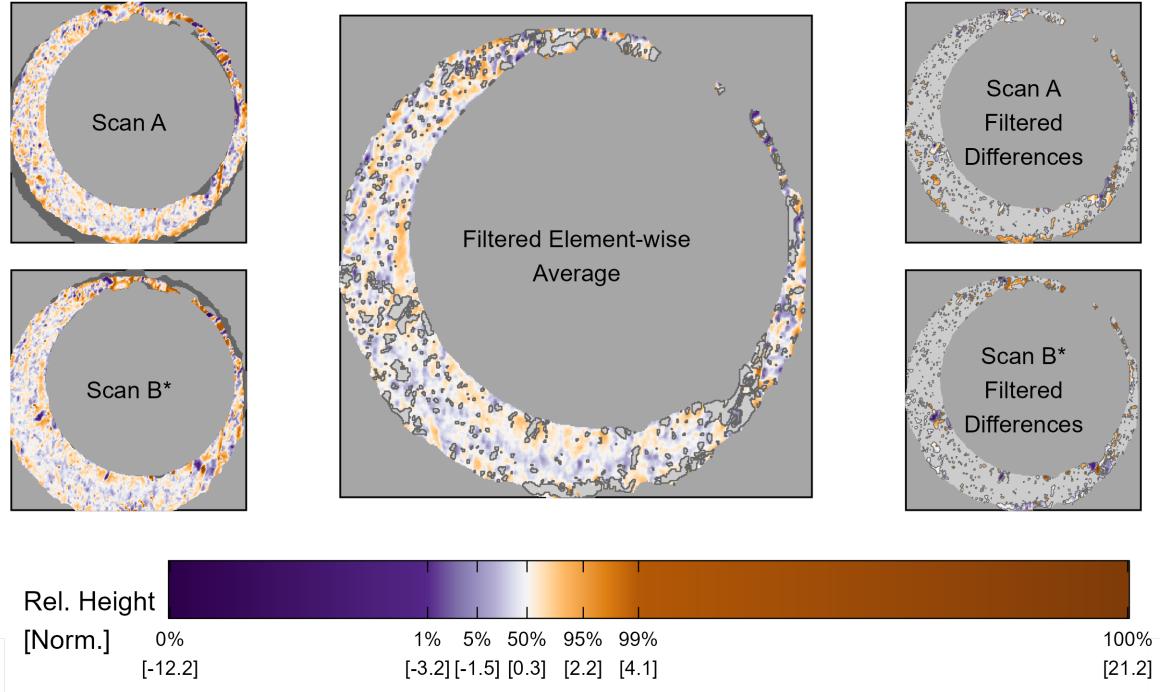


Figure 10: Full scan comparison plot.

matrix extracted from  $B^*$  that maximizes the CCF with  $A_t$ . We calculate the sample mean of the filtered correlation values across all cells and both directions:

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

Next, we consider features based on the elements of the Boolean *cond* matrix. Consider Figure 11 that shows the filtered element-wise average  $\mathcal{F}_{|A-B^*|<\tau} \left( \frac{1}{2}(A + B^*) \right)$  on the left and the associated *cond* matrix  $|A - B^*| < \tau$  visualized in black-and-white in the middle where filtered elements are shown in white. We use a connected components labeling algorithm detailed in [Haralick and Shapiro (1992)] to identify individual neighborhoods of filtered elements. More precisely, the algorithm returns a set of sets  $S_d = \{S_{d,1}, S_{d,2}, \dots, S_{d,L_d}\}$  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 11 shows each  $S_{d,l}$  distinguished by different fill colors,  $l = 1, \dots, L_d$ .

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

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

where  $|S_{d,l}|$  is the size of the set  $S_{d,l}$ . We assume that the **average** and **standard deviation of the filtered full-scan neighborhood sizes** will be small for truly

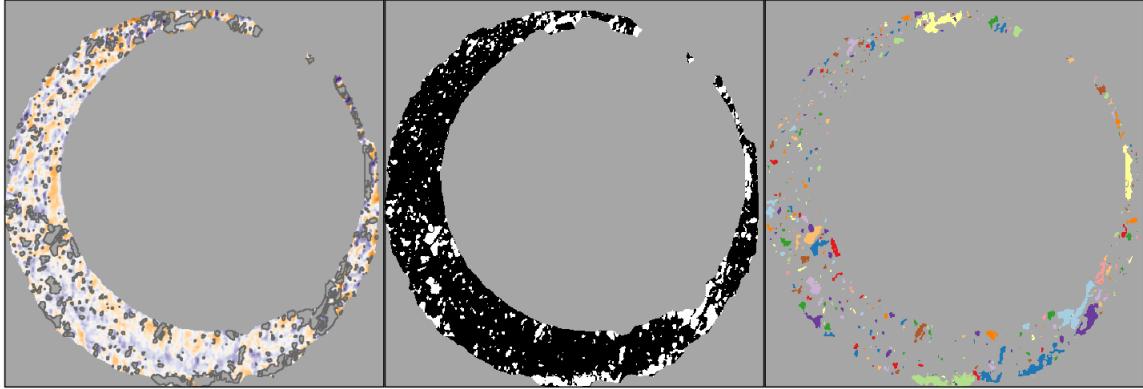


Figure 11: (Left) After aligning two scans, we filter regions that are "different" from each other, meaning the absolute difference between surface values is larger than some threshold. (Middle) We binarize the scan into "filtered" or "non-filtered" regions - shown in white and black, respectively. (Right) 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.

matching cartridge cases. That is to say, we assume that the the surface 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 the notation to accommodate individual cells. Let  $\mathcal{S}_{d,t} = \{S_{d,t,1}, \dots, S_{d,t,L_{d,t}}\}$  denote the set of labeled neighborhoods for a cell  $t = 1, \dots, T_d$ ,  $d = A, B$ . We calculate the per-cell average and standard deviation of the labeled neighborhood cell size:

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

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

We assume that the cell-based  $\overline{|S|}_{d,t}$  and  $s_{d,t,|S|}^2$  will be small, on average, for truly matching cartridge cases. Consequently, we use the sample average of these as features:

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

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

Again, we assume that the **average cell-wise neighborhood size** and the **average standard deviation of the cell-wise neighborhood sizes** will be small for truly matching cartridge cases.

Figure 12 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.

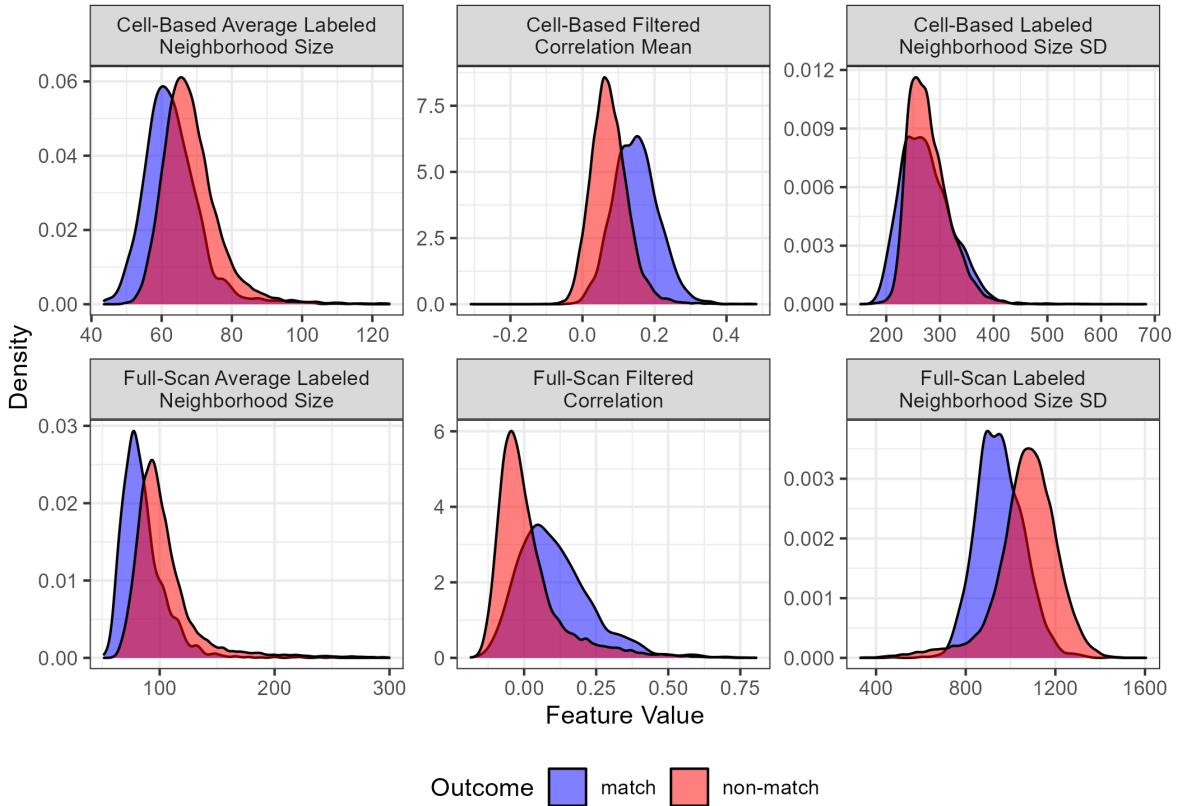


Figure 12: Distributions of the visual diagnostic-based features for 21,945 cartridge case pairs. Matching comparisons tend to have smaller neighborhood sizes on average and higher correlation values than non-matches indicating their utility in a classifier.

Table 3 summarizes the 7 features calculated based on the visual diagnostics.

### 4.3. Scoring

We randomly split the cartridge case data set into 10 barrels for training and 15 barrels for testing. Multiple cartridge cases were fired from each barrel, so 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.

We perform 10-fold cross-validation to train binary classifiers. We compare the results of three classifiers: based on a logistic regression, a Classification and Regression Tree (CART) model, and a random forest ([R Core Team 2019](#); [Therneau and Atkinson 2022](#); [Liaw and Wiener 2002](#); [Kuhn 2022](#)).

[Write full logistic regression model here]

We consider the pros and cons of each of these models. The logistic regression and CART models are more interpretable than a random forest yet, as we will see, a random

Notation	Feature Description
$\bar{cor}_{full,filt}$	<b>Average filtered full-scan correlation</b> across both comparison directions
$\bar{cor}_{cell,filt}$	<b>Average filtered cell-wise correlation</b> across all cells in both comparison directions
$\bar{ S }_{full}$	<b>Average filtered full-scan neighborhood size</b> across both comparison directions
$s_{full, S }$	<b>Standard deviation of the filtered full-scan neighborhood sizes</b> across both comparison directions
$\bar{ S }_{cell}$	<b>Average filtered cell-wise neighborhood sizes</b> across all cells in both comparison directions
$\bar{s}_{cell, S }$	<b>Average standard deviation of the cell-wise neighborhood sizes</b> across all cells in both comparison direction

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

forest tends to be more accurate. The following section detail the results of this cross-validation procedure.

## 5. Results

### 5.1. Training Results

Figure 13 shows the cross-validation estimated accuracies for the three trained models. We consider the performance of the three models under different subsets of the ACES feature set, which provides insight into the importance of the various feature groups. We see that the random forest trained on the full ACES data set results in the highest overall accuracy of 98.9%. For each feature group, the the random forest yields the highest accuracy followed by the logistic regression and CART models. We see that the removing the cluster-based features has a notable impact on the accuracy of the logistic regression and CART models, while the random forest is more robust.

Figure 15 shows the distribution of a variable importance measure for each feature across fittings of a random forest model using 10 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 [cite Gini Index resource]. [More exposition on Gini Index?] Noting the log scale on which these points are plotted, we see that the “no cluster indicator” variable is considered by far the most important variable across the random forest fittings. The “cell-based pairwise-complete correlation” and “cluster size” are also important, although it’s less clear which is more important due to their distributional overlap. Overall, the results indicate that the cluster-based aggregation and cell-based registration features are considered most important by the random forest models.

### 5.2. Testing Results

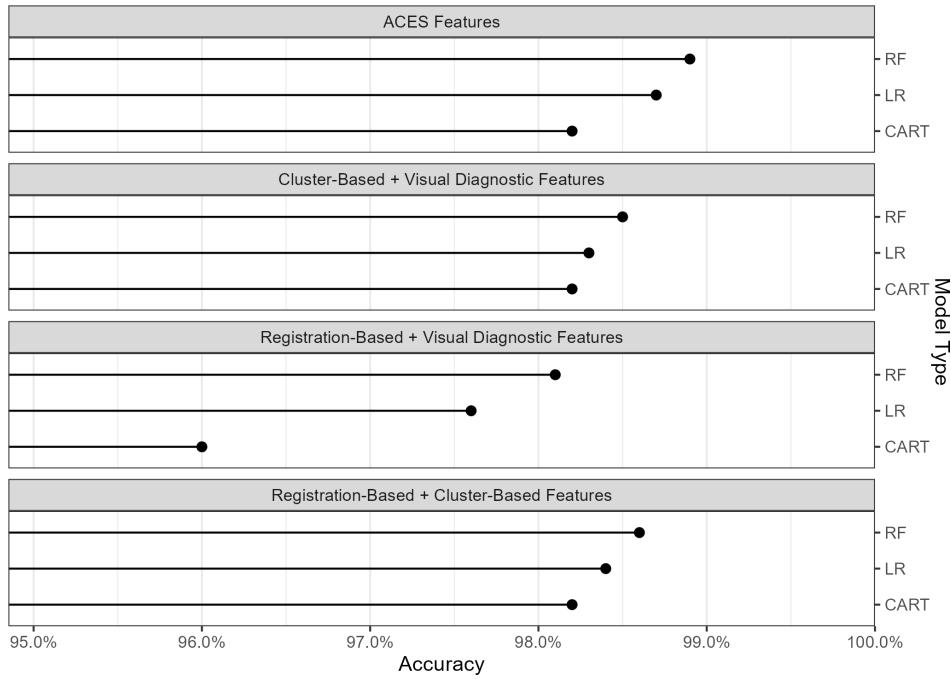


Figure 13: Training classification accuracy for random forest (RF), logistic regression (LR), and classification and regression tree (CART) models based on various subsets of the training data set features. These accuracies are estimated based on 10-fold cross validation repeated thrice. In general, the Classification and Regression Tree (CART) model performs poorest while the Random Forest performs best. Removing the cluster-based features has the largest impact on the accuracies.

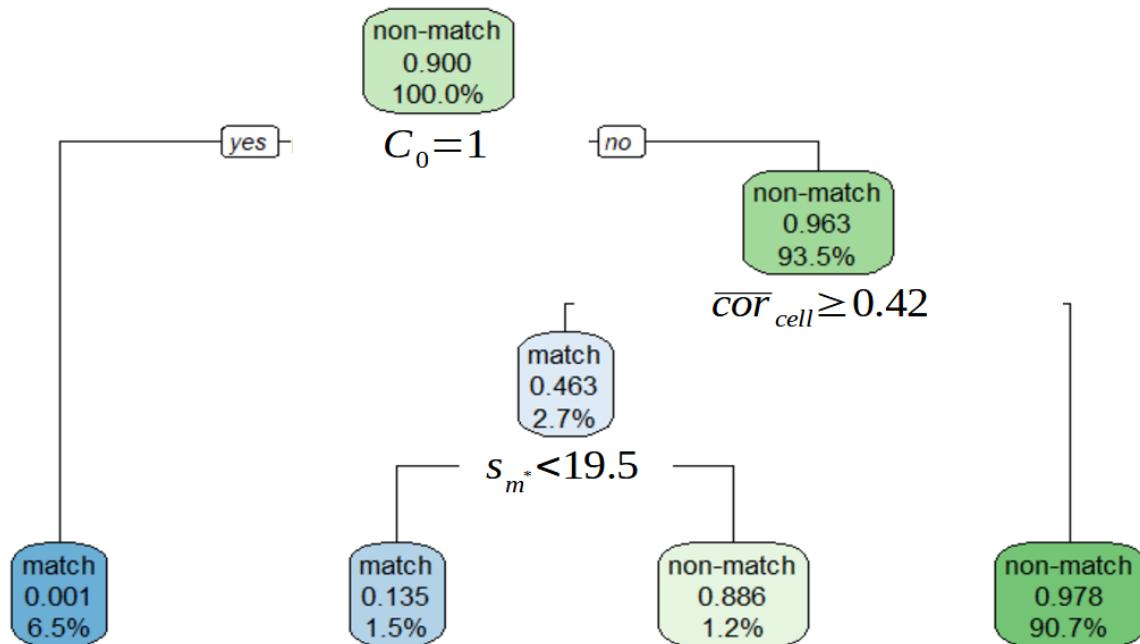


Figure 14: Trained Classification and Regression Tree model.

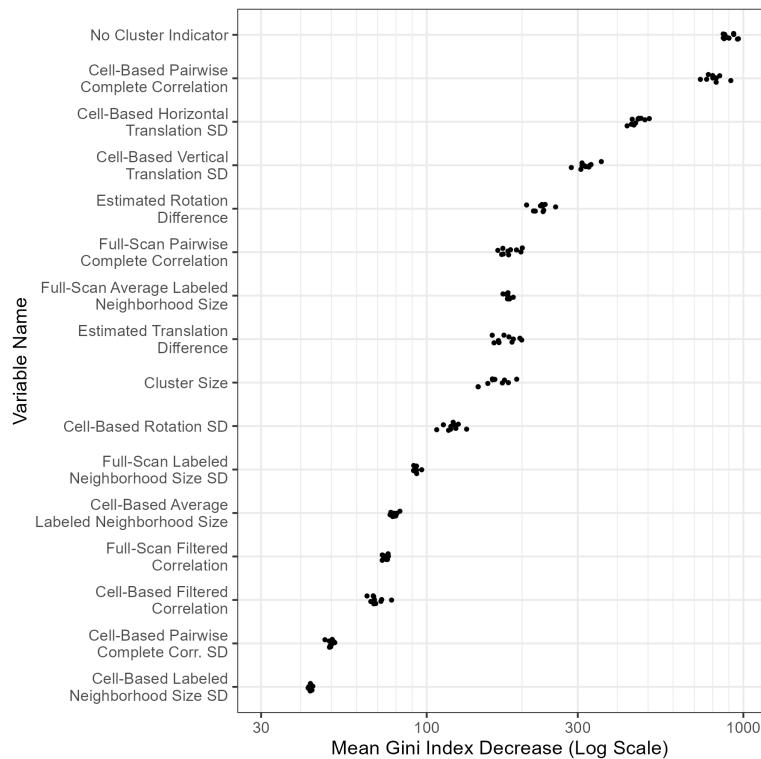


Figure 15: Variable importance measures from fitting a random forest to the training data set, repeated 10 times under various random seeds. Points are plotted on a log scale and vertically jittered for visibility. The No Cluster Indicator feature is by far the most important feature, as measured by the mean decrease in the Gini index.

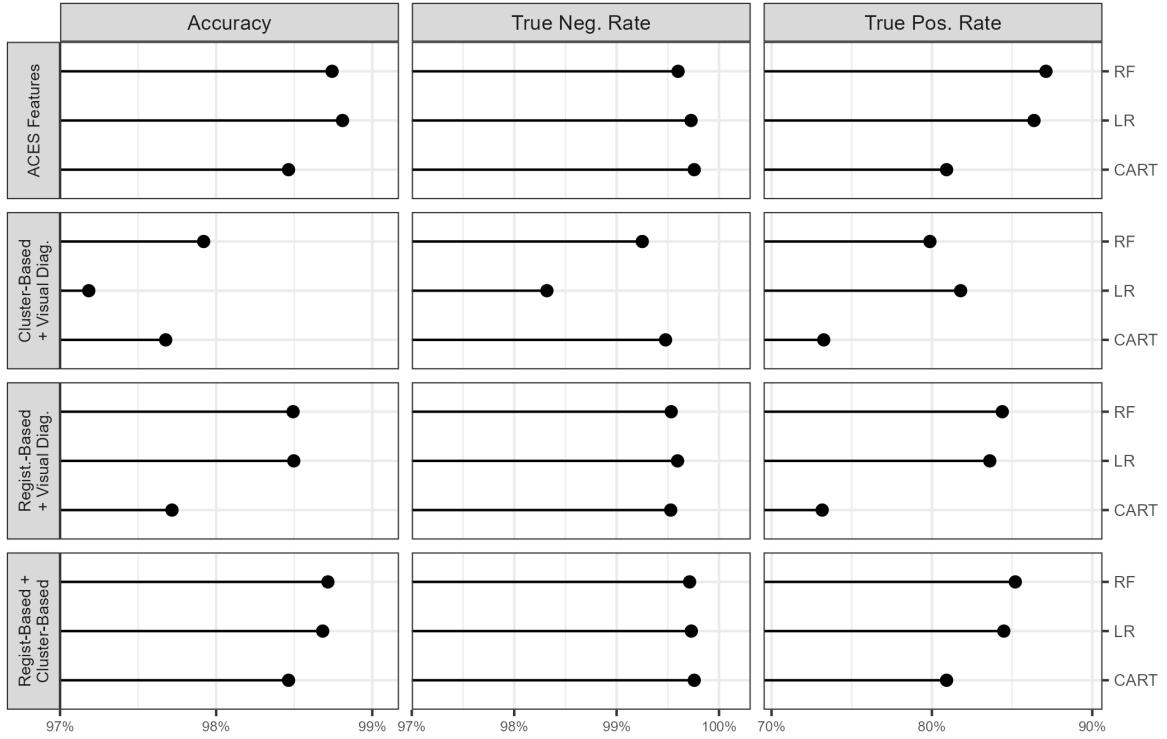


Figure 16: Testing classification accuracy, false negative rate, and false positive rate faceted by various subsets of the testing data set features. The Logistic Regression model performs about as well as the Random Forest model in classifying matches and non-matches amongst the test data set. This is primarily because the Logistic Regression model has a higher true negative rate while the Random Forest model has a higher true positive rate. The CART model lags behind in the three metrics.

For each of the 44,850 cartridge case pairs in the testing data set, we use each model to predict whether the pair is a match or non-match. An error occurs when this prediction does not match the ground-truth nature of the cartridge case pair. Figure 16 summarizes the error rates for each method. The Accuracy is the overall percentage of correct classifications. The True Negative rate is percentage of correctly-classified non-match pairs. Conversely, the True Positive rate is the percentage of correctly-classified matching pairs.

We see that the testing accuracy across models and feature groups is similar to that of the training results in Figure 13. The random forest is more robust to changes in feature group while the CART model performs uniformly worse than the other models. Interestingly, the logistic regression model performs slightly better than the random forest for some feature groups. Considering the true positive and negative rates, this can be explained by the specificity of the models: the logistic regression model classifies non-matches more effectively in these instances, although this difference is slight.

## 6. Discussion

### 6.1. Comparing Classifier Models

Our intention in fitting different classification models was to compare each model's strengths and weaknesses. As indicated by [Figure 16](#), the random forest have similar testing accuracies with the random forest being more robust to changes in feature group. In particular, the random forest model is better at identifying truly matching cartridge cases than the logistic regression classifier, yet worse at identifying non-matches.

Pragmatically, it seems reasonable to choose the model with the highest estimated accuracy. Ethically however, we might favor the model that makes the fewest false positive classifications since mis-classifying a truly non-matching cartridge case pair may incriminate an innocent individual. While the input of statisticians is important, this decision needs to be weighed by the wider forensic and legal communities.

While the random forest is generally more accurate, the CART and logistic regression models are more interpretable. For example, as seen in [figure], the CART model provides a set of simple, binary rules by which we can arrive at a classification. [More on CART? Perhaps compare a single decision tree to random forest?]

The estimated coefficients in the logistic regression model help us understand the effect that each feature has on the odds that a cartridge case pair matches. [Table] shows the multiplicative change in the odds that a cartridge case pair matches for a one unit increase in each feature.

Discuss benefits of three models here

- CART model is a clear set of binary “rules” that lead to a classification
- Logistic Regression provides estimate of how odds of match change with a one unit increase of each feature
- It also performs similar to the random forest
- Random Forest seems to be more robust to changes

### 6.2. Comparison to Previous Work

Our results corroborate the conclusions made in previous papers. [Zhang et al. (2020)] proposed a binary classifier using the DBSCAN algorithm: if a cluster is identified, then classify the cartridge case pair as a match and otherwise a non-match. This is analogous to defining a classification rule based solely on the ACES feature  $C_0$ . Given the importance of  $C_0$  indicated in [Figure 15](#), our results indicate that a classifier based on  $C_0$  would have a reasonably high accuracy. However, information about the size and location of clusters in a classifier adds important nuance as indicated by the ranking of  $\Delta_\theta$ ,  $\Delta_{\text{trans}}$ , and  $C$  in [Figure 15](#). For example, although one non-match pair in the training data was assigned a DBSCAN cluster, this pair's estimated translation difference,  $\Delta_{\text{trans}}$ , was relatively large. The ACES logistic regression and random forest models correctly classify this pair as a non-match.

Cells identified by a DBSCAN cluster could be considered an alternative definition of a “Congruent Matching Cell” (CMC) as defined in [Song (2013)]. While the intention behind the CMC method and the density-based ACES features is the same, that is to determine the number of cells that “agree” on a registration value, the manner in which agreement is measured differs. In the original CMC algorithm, a cell is called a CMC if its estimated registration is within some threshold of a reference value, typically the median registration across all cells. The registration thresholds are set manually in most CMC papers except for [Zemmels et al. (2022)] where they are selected based on an optimization criterion. As discussed in [Zemmels et al. (2022)], the CMC algorithm is quite sensitive to the choice of thresholds as well as to noisiness in the cell-based registrations. In the ACES algorithm, we measure agreement by the number of cells that are close to each other - a reference value is not required, but is a byproduct of the DBSCAN algorithm (e.g., treating cluster centroids as the estimated translations). Further, we simplify optimization by defining “closeness” through the single  $\epsilon$  parameter of the DBSCAN algorithm instead of three threshold parameters (horizontal/vertical translation and rotation) in the CMC algorithm.

[Song (2013)] measures the similarity between two cartridge cases using the total number of CMCs and proposes classifying a pair as matching if the CMC count exceeds five. Considering the DBSCAN cluster size  $C$  as analogous to the CMC count, a decision boundary for  $C$  equal to five seems reasonable; especially in light of the distribution of  $C$  shown in Figure 9. Similarly, reasonable matching registration cutoffs based on Figure 9 are  $3^\circ$  for rotation and 10 pixels for translation. These are similar to manually-selected thresholds used across various CMC papers [Tong et al. (2015), Chen et al. (2017)].

[Compare results to our best CMC method (and Baldwin?).]

The ACES algorithm simultaneously substantiates the classification rules used by previously proposed cartridge case comparison algorithms while also infusing their logic with additional nuance.

## 7. Conclusion

More experimentation is needed. It is reasonable to assume that the version of ACES discussed in this paper would be effective at classifying cartridge cases of the same brand, fired from the same make/model of firearm, and scanned using the same topographical scanner [cite TopMatch]. It remains to be seen whether the fitted models generalize to other types of ammunition or firearm.

The train/test procedure outlined in this manuscript should be adopted by any future researchers to validate proposed methods. [Discuss availability of data and code]

Nonetheless, this paper provides the largest study of automatic cartridge case comparison algorithms published to-date. Our results indicate that there exist effective, robust, and interpretable automatic classifiers for cartridge case evidence.

We expect the ACES feature set to evolve over time; for discriminatory features to replace less informative features. We stress interpretability as a guiding principle for future feature engineering. Ideally, forensic practitioners will eventually use such algorithms to supplement their expert opinion. We believe it paramount that practitioners understand and can explain, at least at a high level, to a jury of lay people the features

used for classification. [More on why this is important]

## 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~3.5.1. R itself and all packages used are available from the Comprehensive R Archive Network (CRAN) at <https://CRAN.R-project.org/>.

## Acknowledgments

All acknowledgments should be collected in this unnumbered section before the references. It may contain the usual information about funding and feedback from colleagues/reviewers/etc. Furthermore, information such as relative contributions of the authors may be added here (if any).

## References

- 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/>.
- R Core Team (2019). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, <https://www.R-project.org/>.
- Therneau, T. and Atkinson, B. (2022). *rpart: Recursive Partitioning and Regression Trees*, <https://CRAN.R-project.org/package=rpart>. R package version 4.1.16.

**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](mailto: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](mailto: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](mailto:heike@iastate.edu)  
URL: <https://github.com/heike>