

## ORIGINAL PAPER

## Criminalistics

# Automated detection of regions of interest in cartridge case images using deep learning

Marie-Eve Le Bouthillier MEng<sup>1,2</sup>  | Lynne Hrynkiw BASc<sup>1,2</sup>  | Alain Beauchamp PhD<sup>2</sup> |  
Luc Duong PhD<sup>1</sup>  | Sylvie Ratté PhD<sup>1</sup> 

<sup>1</sup>École de technologie supérieure, ÉTS,  
Montréal, Québec, Canada

<sup>2</sup>Ultra Electronics Forensic Technology,  
Inc., St-Laurent, Québec, Canada

**Correspondence**

Marie-Eve Le Bouthillier, École de  
technologie supérieure, ÉTS, 1100 Notre-  
Dame Street West, Montréal, Québec  
H3C 1K3, Canada.

Email: [marie-evele-bouthillier1@ens.etsmtl.ca](mailto:marie-evele-bouthillier1@ens.etsmtl.ca)

**Abstract**

This paper explores a deep-learning approach to evaluate the position of circular delimiters in cartridge case images. These delimiters define two regions of interest (ROI), corresponding to the breech face and the firing pin impressions, and are placed manually or by an image-processing algorithm. This positioning bears a significant impact on the performance of the image-matching algorithms for firearm identification, and an automated evaluation method would be beneficial to any computerized system. Our contribution consists in optimizing and training U-Net segmentation models from digital images of cartridge cases, intending to locate ROIs automatically. For the experiments, we used high-resolution 2D images from 1195 samples of cartridge cases fired by different 9MM firearms. Our results show that the segmentation models, trained on augmented data sets, exhibit a performance of 95.6% IoU (Intersection over Union) and 99.3% DC (Dice Coefficient) with a loss of 0.014 for the breech face images; and a performance of 95.9% IoU and 99.5% DC with a loss of 0.011 for the firing pin images. We observed that the natural shapes of predicted circles reduce the performance of segmentation models compared with perfect circles on ground truth masks suggesting that our method provide a more accurate segmentation of the real ROI shape. In practice, we believe that these results could be useful for firearms identification. In future work, the predictions may be used to evaluate the quality of delimiters on specimens in a database, or they could determine the region of interest on a cartridge case image.

**KEYWORDS**

automated firearms identification, ballistic, cartridge cases identification, CNN, deep learning, firearm analysis, firearm identification, segmentation, tool marks, U-NET

**Highlights**

- Deep Learning (U-Net) is promising for automated segmentation of ROI on cartridge case images.
- Improved ROI detection, with natural shape prediction for breech face and firing pin impressions.
- Segmentation method locates BF and FP impressions on 9 mm CC with a Dice Coefficient reaching 99%.

This is an open access article under the terms of the [Creative Commons Attribution-NonCommercial-NoDerivs](https://creativecommons.org/licenses/by-nc-nd/4.0/) License, which permits use and distribution in any medium, provided the original work is properly cited, the use is non-commercial and no modifications or adaptations are made.

© 2023 Ultra Electronics Forensic Technology Inc and The Authors. *Journal of Forensic Sciences* published by Wiley Periodicals LLC on behalf of American Academy of Forensic Sciences.

## 1 | INTRODUCTION

Bullets and cartridge cases discharged by a firearm provide ballistic samples that contain characteristic marks, which can supply information about the weapon used to fire the ammunition. Firearm identification is based on the principles of tool mark identification, a forensic discipline interested in establishing whether a specific tool has produced a mark on another object. In the context of ballistics, the firearm represents a tool that transfers marks onto bullets and cartridge cases [1–7]. To determine or eliminate the use of a particular firearm, the ballistic identification specialist relies on the observation, recognition, and matching of characteristic marks present on the specimens [8].

Since the 1980s, many researchers have turned to computers to improve the science of firearms identification. The increasing number of samples contained in databases, the time involved in visual evaluations by experts, and the inherent bias of human opinion all contribute to the automation of ballistic specimen acquisition and the matching process. In this regard, various computerized systems have emerged, supported by recent developments, namely the improvement of image acquisition equipment with 3D topography, the advance of image-processing technology, and research work on comparison algorithms introducing a quantitative and objective match between two samples [9]. For example, the Integrated Ballistic Identification System (IBIS), appointed by the Bureau of Alcohol, Tobacco, and Firearms (ATF), the Drugfire program (nowadays withdrawn), proposed by the Federal Bureau of Investigation (FBI) both developed in the early 1990s in America; and the National Integrated Ballistics Information Network (NIBIN), an evidence database established in 1999 [2, 10–12]. These systems automate the imaging of samples, identify regions of interest (ROI), and provide a numeric estimate of likely matches. They typically include an optical microscope, a signature analysis station, and a matching algorithm for an end-to-end ballistics evaluation [2, 4, 7, 13–15].

The firing process of a gun transfers several characteristic marks to the cartridge case, which vary according to the model of the firearm [2, 13]. Among these marks are the firing pin (FP) impression and the breech face (BF) impression, produced at different times during the firing process. The FP impression is partially created when the firing pin strikes the primer before the propellant ignites, thus, imprinting the negative topography of the firing pin surface on the case. This imprint's shape, position, dimensions, and depth are specific to the firearm model [3, 4]. The BF mark is created as a result of the sequence of events following the propellant ignition. The accumulation of hot gasses produced by the explosion projects the bullet through the barrel, recoiling the cartridge case in the opposite direction until it impacts the breech face of the firearm. At this instant (or upon collision), the negative topography of the breech face surface is imprinted on the head of the cartridge case. Specialists consider these impressions as reliable tool marks for identification purposes [4, 16]. Because the position of the cartridge case may vary between the instants of the FP strike and the BF collision, the relative alignment of the two marks may differ among exhibits. Therefore, they

are usually separated before conducting a computational ballistic analysis [1], and automating this procedure represents a valuable addition to any firearm identification system.

### 1.1 | Problem definition

Modern computerized systems typically use traditional image-processing methods to position circular delimiters on the cartridge case images to establish the ROI corresponding to the BF mark and the FP impression. This step is usually followed by human validation, where a system user or an expert verifies and repositions these delimiters as needed. This positioning bears a significant impact on the performance of the image-matching algorithms. Mainly because calculations to establish the level of similarity between the images in the database only consider the ROI area since it is the location of the microscopic marks specific to each firearm. As such, the ability to retrieve similar cartridge cases from the databases depends heavily on the position accuracy of the delimiters. Therefore, an automated method to evaluate the delimiters of images in a database or verify their positioning accuracy upon the image acquisition would be beneficial to any computerized system; whether an algorithm is responsible for a suboptimal position or a user repositioned them incorrectly.

### 1.2 | Background

Several studies involving ballistic cartridge case samples have proposed solutions for developing or improving matching algorithms to establish a quantitative similarity score between tool marks [1, 3, 4, 7, 8, 13, 14, 16–19]. But in many of these studies, the steps of detection and selection for the region of interest are not always well defined. Some studies even omit this step, and we assume that the ROI is selected manually by placing delimiting circles around them. However, this manual method is not ideal because rapid processing becomes unrealistic with increasing numbers of cartridge case samples. Second, manual segmentation by a human operator injects bias and variation into the samples [14, 15]. In other studies, the authors report using preprocessing steps to segment the images and separate the different types of marks. Still, they do not specify the methods to achieve this [1, 13, 17–19]. A few studies mention an automatic identification of the ROI using traditional image-processing methods, mainly edge detection, such as the Sobel operator or the Canny edge detector. Sometimes, image segmentation operations paired with thresholding techniques isolate the ROI from the background image and produce a mask (a binary image displaying black or white pixels) [2]. Tai et al. present a detailed description of an automated mark selection step using traditional image-processing methods: the primer region is initially identified with a combination of operations (Gaussian filter, histogram equalization, flood fill, dilation, and erosion). The FP impression is then removed with a similar combination of operations, with the addition of a Canny edge detector. This

combination retains the true shape of the firing pin mark since this shape yields an identification value [14]. A few studies also focused predominantly on the automated segmentation of ROI on cartridge cases images using traditional image-processing methods: a combination of methods including Connected Components Labeling, the Canny operator, and the Randomized Hough Transform; a second method using the image focus information for each pixel paired with a Hough transform [20], and thresholding technique using Surface Height Image information [15].

Machine learning is a scientific discipline using computer algorithms to identify congruences in data and use them in tasks such as classification or segmentation [7, 17]. Today, many fields are turning to machine-learning techniques to solve problems involving computer vision tasks. Some researchers use traditional image-processing methods to extract useful features from an image and a machine-learning algorithm to categorize or segment them. At the same time, others are interested in the opportunities arising from the rapid evolution of deep learning, which has repeatedly been shown to match and even surpass human performance on pattern recognition tasks. In addition, access to various open-source software options and the availability of a vast network filled with resources greatly facilitate using these modern techniques [21].

Nevertheless, to this day, few works have used deep learning-based methods to delimit ROI in images of cartridge cases. However, other domains have often used machine learning or deep learning to assist in the discovery of different ROI in images with characteristics resembling those of cartridge case images, for instance, high-resolution grayscale photographs and 3D topographic images. Specifically, medical imaging offers many studies involving monochrome or 3D images with reliefs (magnetic resonance, X-rays, computed tomography, etc.). Many studies in this field used deep-learning architectures specialized in classification, object detection, or segmentation to identify, locate, or recognize ROI in images. For example, convolutional neural networks (CNN) have succeeded in classifying liver tumors in magnetic resonance image (MRI) scans [22]; faster R-CNN, YOLOV3, and SSD networks have detected brain lesions in MRI [23]; a Hierarchical Convolutional Neural Network (HCNN) was used to segment iris in images of the eye for biometric identification purposes [24]; and the FCN, FCN-DenseNet, U-Net, and LinkNet architectures have been utilized to segment salivary glands in ultrasound images [25].

In other areas, the U-Net, SegNet, Dilated Net, PSPNet, DeepLab v3+, and Dilated Residual Net architectures have been tested to detect solar panels in high-resolution satellite images [26]. The U-Net architecture has successfully segmented areas corresponding to landslides, farmland, and clouds in remote-sensing images [27]. And an R-CNN based on the ResNet architecture successfully segmented topographic regions from images of a moon replica site [28].

This study aims to determine if a modern deep-learning approach could evaluate the positioning of circular delimiters in cartridge case images and eventually help identify images with suboptimal circular delimiters in ballistic sample databases. This solution could also be implemented in image acquisition software to perform segmentation

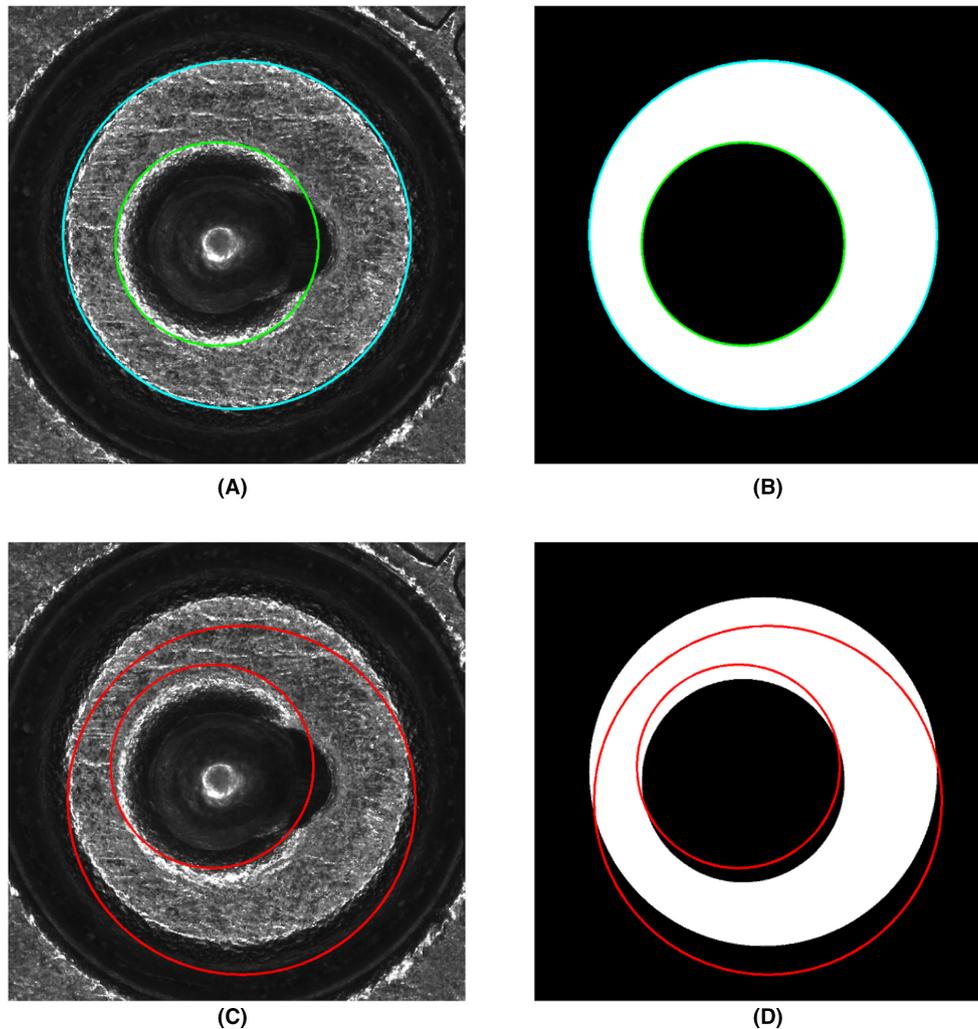
or validation during the acquisition of a cartridge case image. This study examines two types of cartridge case ROI. The breech face (BF) impression is represented by an image with two delimiting circles (Figure 1), and the firing pin (FP) impression is represented by an image with a single delimiting circle (Figure 2).

## 2 | MATERIALS AND METHODS

### 2.1 | Deep learning

Many segmentation architectures, such as the fully convolutional network FCN, are based on the convolutional neural network CNN and are divided into two parts: an encoder and a decoder. The function of the encoder is to downsample the input image to obtain a high-level contextual feature map. This part typically includes several convolutions and pooling operations. The decoder's task is to upsample this low-resolution feature map by carrying out transposed convolutions to predict a high-resolution dense label map. To train a segmentation model, the data set must include the images to be segmented, plus binary masks matching these images and representing the ground truth, that is, the masks manually segmented by an expert [26, 27, 29, 30]. The U-Net architecture entails an enhanced version of the FCN, incorporating skip connections to link the encoder's lower layers to the decoder's upper layers. These shortcuts improve the performance of the dense prediction, using a concatenation of the features extracted in the encoder and the ones obtained in the decoder to transfer the features already learned by the model, allowing the model to use the global information extracted from the early layers of the network when evaluating the pixels prediction [29,31]. Several layers comprise the U-Net architecture, divided into two sections: the encoder and the decoder. The encoder performs the downsampling through a series of layers, made of two convolutional operations, followed by a max-pooling operation with a ReLU activation function, where the number of filters is doubled between every layer. The decoder carries out the upsampling using a succession of transposed convolutional layers. At the network output, a  $1 \times 1$  convolutional layer maps each feature vector to the target class (0 or 1 in the case of a binary mask, contingent on the pixel being part of the ROI) [25, 31]. The R-CNN mask, another popular option, divides the segmentation problem into two subtasks: detection and localization of a ROI, followed by feature segmentation and mask creation [32].

To evaluate the quality of the results in segmentation tasks, several authors used the Dice Coefficient (DC), also called the F1 measure or Dice Metric (DM) [26, 29, 32–35]. The DC metric is computed between the predicted binary masks at the output of the network and the ground truth. DC denotes the ratio between the intersection of two regions and the union of these regions, namely the overlap between two images. It is defined by Equations 1 and 2, where A represents the network prediction, and B represents the ground truth. TP, FP, and FN represent the true positive, false positive, and false negative values [26].



**FIGURE 1** Images of the breech face mark captured with a ring light source. Circular delimiters correctly positioned: image (A), ground truth mask (B). Circular delimiters badly positioned: image (C), ground truth mask (D).

$$DC(A, B) = \frac{2|A \cap B|}{|A| + |B|} \quad (1)$$

$$DC = \frac{2TP}{2TP + FP + FN} \quad (2)$$

The Intersection over Union (IoU), also known as Jaccard's index, is another popular metric to evaluate the results of a segmentation task [26, 36]. Like DC, the IoU metric represents the overlap rate between the predictions and the ground truth [27]. This metric is defined by Equations 3 and 4.

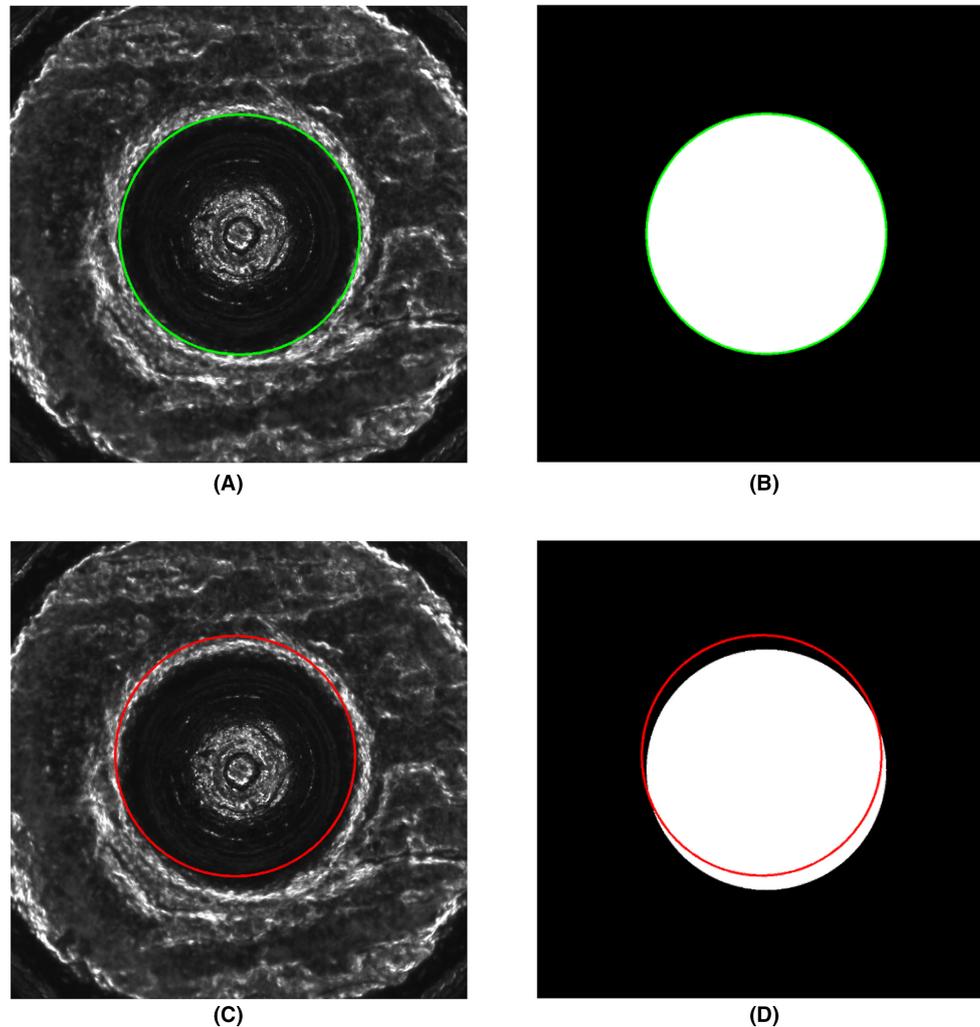
$$IoU = \frac{|A \cap B|}{|A \cup B|} \quad (3)$$

$$IoU = \frac{TP}{TP + FP + FN} \quad (4)$$

Both DC and IoU metric presents a number between 0 and 1. The closer this value is to 1, the higher the overlap and similarity between the two images and the more accurate the segmentation

predicted by the model. A value equal to 1 means a perfectly overlapping segmentation, whereas a value equal to 0 means the absence of overlap. The DC and the IoU metrics are positively correlated, which connotes that if one metric evaluates a model as superior to another, the second metric will yield the same evaluation. In general practice, the Dice Coefficient provides a measure of average performance, whereas the IoU provides a measure closer to the worst outcome [26, 34, 36].

Typically, neural networks work with an optimization process using an objective function to minimize or maximize a value. When this value is minimized, the objective function is called the loss function. The principle of the optimization process is to provide a set of weights to the model to make the predictions, then calculate an error on these predictions and adjust the weights to minimize the error on the next iteration. The more the predictions deviate from the ground truth, the higher the value of the loss function. Many basic architectures use stochastic gradient descent as an optimization algorithm, but other options could be better suited: for example, RMSProp, Adagrad, Adadelata, Adam, Adamax, Nadam. The Cross-Entropy loss function is commonly used to compute the error



**FIGURE 2** Images of the firing pin, captured with a ring light source. Circular delimiters correctly positioned: image (A), ground truth mask (B). Circular delimiters badly positioned: image (C), ground truth mask (D).

between the output and the ground truth in deep-learning models, including models for segmentation tasks [25, 29, 34, 37].

Alternatively, a few studies have used the Dice Loss (DL), defined by Equation 5. The Dice Coefficient, previously defined by Equation 1, determines this loss value. The Dice Loss also allows for the optimization of the DC value during a model's training [26]. Similarly, Equation 6 defines the Jaccard Loss (JL).

$$DL = 1 - DC \quad (5)$$

$$JL = 1 - IoU \quad (6)$$

Thus, for this project, we chose the U-Net segmentation architecture since it was shown to outperform FCN, is known to provide suitable results from relatively few images and offers a simple implementing solution. We also chose to use the standard metrics for evaluating the results of segmentation tasks: the Dice Coefficient (Equation 1) and the Intersection over Union (Equation 3). These metrics are well recognized and have been proven in many similar studies.

## 2.2 | Data

The data set is provided by the research department at Ultra Electronics Forensic Technology [11]. It consists of 1195 samples from cartridge cases fired by different 9MM guns. For each case, the data set contains images from the two regions of interest under study (breach face and firing pin), geometric information on the positioning of the delimiting circles, and a binary mask representing these circles. The images were obtained using a BrassTRAX acquisition station, a commercial system that captures high-resolution 2D images and 3D topographic information of the cartridge cases. For this project, we utilized 2D images illuminated from a ring light source. Initially positioned automatically by an image-processing algorithm, the circular delimiters were checked by an expert and repositioned as needed. The resulting masks were used as the ground truth and were corroborated by a visual inspection prior to the training. Figure S1 shows examples of the BF and FP images and displays the full headstamp, the frontal print of the cartridge case.

## 2.3 | Preprocessing

Similar studies use image resizing, image normalization, and data augmentation preprocessing techniques. Resizing involves changing the resolution of the original image and is mainly used to reduce computational complexity and training time, to accommodate the input requirements of a deep-learning architecture, or to mitigate hardware limitations [27, 38, 39]. In some cases, resizing has been used to homogenize the data resolution, especially when images come from different sources [34]. Data augmentation increases the number of available images by applying slight distortions to the original images. During training, this technique improves the likelihood of the network learning distinctive and useful features while reducing the risks of overfitting [22, 34, 40]. Some of the most popular modifications applied to images to carry out augmentation include rotations, shifts, scale transformations, horizontal and vertical flips [26, 34, 39, 41, 42], noise, blur masks [25, 38], and brightness variations [22]. For example, one study exploited the symmetry property of the lungs by applying vertical flips to radiographic images [38].

We randomly separated the original data and their masks into three sets for this project. The training and validation sets (80% and 10% of the data) are used when training the models, and the test set (10% of the data) is used to evaluate the models after completing the training sessions. The test set is important because it allows us to evaluate the models with new samples unseen during the training. The data split is applied before the data augmentation to prevent allocating similar variations of an image to different sets.

Next, we used augmentation techniques to increase the training and validation sets three times (i.e., adding two randomly modified images for each sample) and the test set six times (five modified images per sample). Figure 3 shows a schematic illustrating the preprocessing steps with the data augmentation procedure. We chose to apply randomly selected transformations on the images and the corresponding masks. The possible transformations are a rotation of 36° to 180°, a translation of 15 to 75 pixels in x or y, a horizontal or vertical flip, and variable salt and pepper noise. An edge effect can appear when we apply a rotation or translation to an image (see Figure S2). Gaussian or salt and pepper noise (with random grain size and gray tone) replaces the black areas to prevent the model from identifying these always-black regions as defining features of the images. This step does not apply to the image masks since their black background makes the edge effects indistinguishable. Figure S3 shows an image modified by three random transformations. As a last preprocessing step, we resized all images and their masks to a 256 × 256 resolution, as required for the U-Net input, and normalized the images by dividing each pixel by 255 so that all values lie in the range [0–1].

## 2.4 | Methodology

We then followed a typical machine-learning methodology divided into three main parts: optimizing the hyperparameters for the

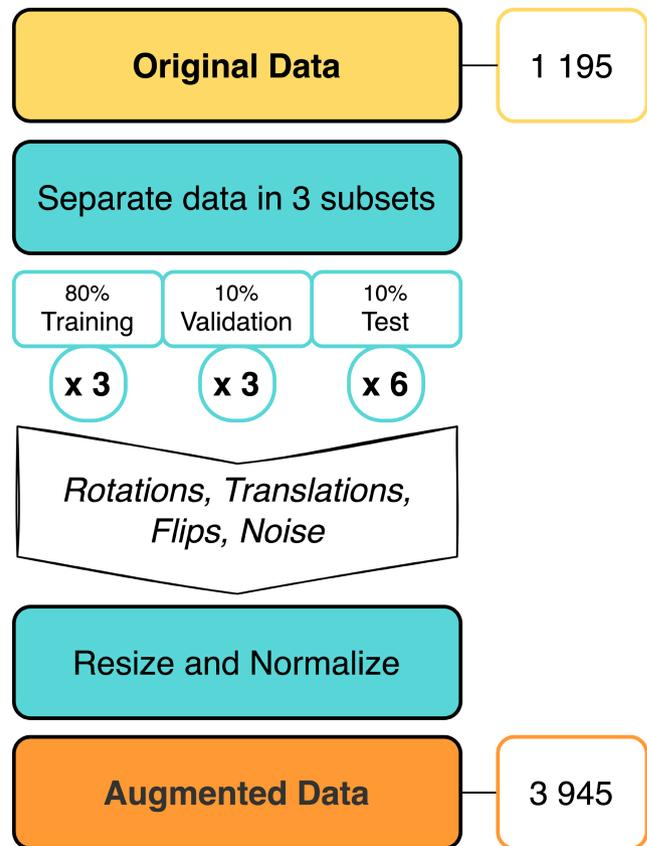


FIGURE 3 Data augmentation schema.

models, training the different models, and evaluating them to compare the results. This section presents the methodology details, and Figure 4 illustrates the general process of the model.

The hyperparameter optimization stage involves a grid search, that is, the training of several U-Net models using different hyperparameter configurations to determine the ideal hyperparameter combination. We based our choice on the learning curves and the metrics obtained during the evaluation of the models. The learning curves are created from different values collected during training. They contain information about the evolution of the performance (the metric) and the optimization of a model (the loss). For the selection of the best hyperparameters, we compared the convergence and the visual fluidity of the learning curves (training and validation), and the values of the evaluation metrics and loss. For these optimization pieces of training, we used images of the breech face mark, photographed with a ring light (Figure S1A). The default learning rate value was used for all the optimizers tested. Some of the optimizers feature an adaptive learning rate, where the value of the learning rate is modified internally during the training process. For the non-adaptive optimizers, we used a parameter of the callback function that acts in a similar way, by reducing the learning rate when the validation loss has stopped improving. Table 1 shows the values considered for the three hyperparameters evaluated during these training iterations: batch size, loss function, and optimizer; whereas Table 2

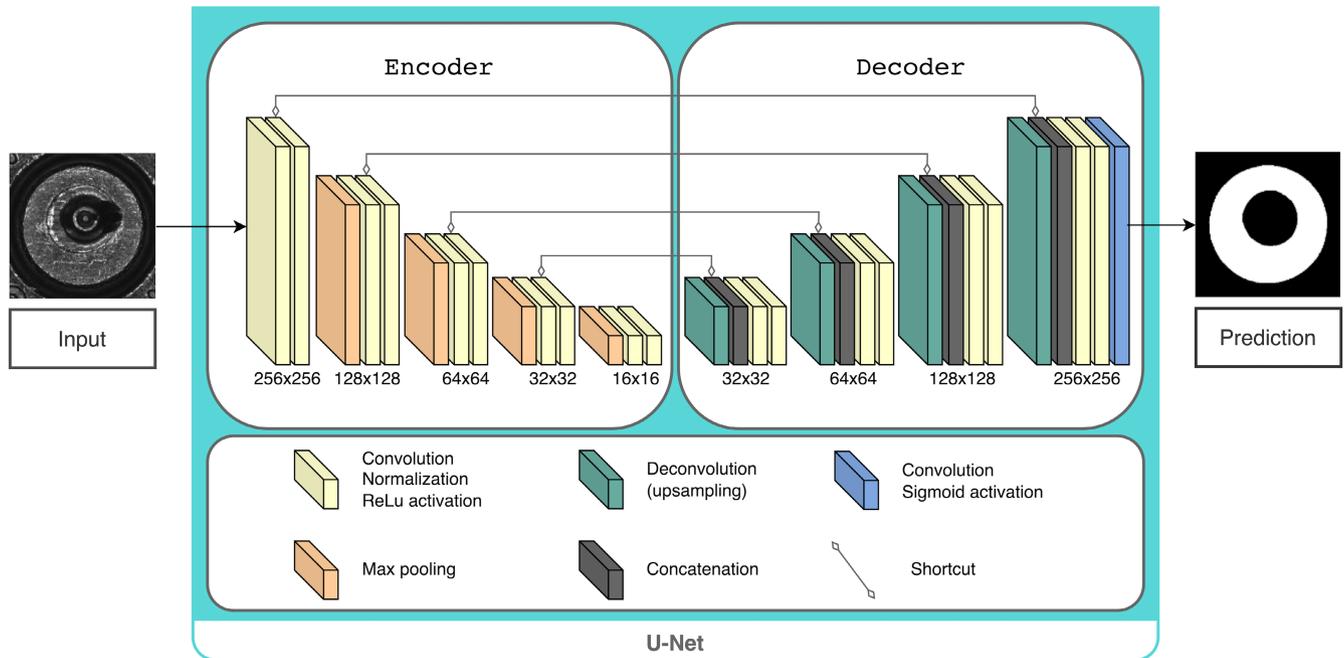


FIGURE 4 General process.

TABLE 1 Hyperparameters for segmentation models training.

Hyperparameter	Values considered
Batch size	8, 16, 32, 64
Loss function	Binary cross-entropy, Dice Loss (DL), Jaccard Loss (JL)
Optimizer	SGD, RMSprop, Adagrad, Adadelata, Adamax, Nadam, Adam

TABLE 2 Selected hyperparameters for segmentation.

Hyperparameter	Values
Batch size	8
Loss function	Jaccard Loss
Optimizer	Adam

presents the optimal combination determined at the end: a batch size of 8, a Jaccard Loss function, and the Adam optimizer.

Once identified, we trained four segmentation models from different image sources using this optimal hyperparameter combination. We trained one model from the original (nonaugmented) data set and one model from the augmented data set for each type of ROI considered in this project: the breech face and the firing pin. Table 3 shows the four models trained. We analyzed the results for each of these models, first comparing the training curves. The training learning curve (from the training data) presents a snapshot of how well the model learned. The validation learning curve (from the validation data) presents an overview of the model's generalization ability. Once the training was completed, we examined the metrics obtained from the model evaluations (from the test data) to gain information on the performance of the models with unseen data. The values

TABLE 3 Segmentation models.

Model	Ensemble	ROI	Images
BF-O	Original data	Breech face	
BF-A	Augmented data		
FP-O	Original data	Firing pin	
FP-A	Augmented data		

considered are the same as those used for the training curves: IoU, DC, and JL. For each model, we conducted two evaluations: a first evaluation with a nonaugmented test set and a second evaluation with an augmented test set, to simulate cases where the images are not ideal (e.g., decentered or noisy images). This double evaluation allows us to get a better idea about the performance of the models and to be able to assess the improvements obtained when we train the models with augmented data.

## 2.5 | Implementation

The experimentations were performed on a 2019 MacBook Pro (see Table 5 for specifications). U-Net was implemented in Python and Keras, following the original architecture, without using pretrained weights [31]. All models include an early stop option for the training via a callbacks parameter. This early stopping allows the configuration of certain conditions to trigger the end of the training. For this

TABLE 4 Models evaluation.

	BF-O		BF-A		FP-O		FP-A	
	Test set		Test set		Test set		Test set	
	Original	Augmented	Original	Augmented	Original	Augmented	Original	Augmented
DC	99.25%	97.78%	99.32%	99.29%	99.45%	97.99%	99.45%	99.40%
IoU	95.15%	87.00%	95.62%	95.40%	95.85%	86.54%	95.89%	95.57%
Loss	0.0159	0.0453	0.0143	0.0150	0.0114	0.0406	0.0113	0.0124

project, training stops when the validation loss has not changed for 10 epochs. If this condition is never reached, the training continues for the predetermined number of epochs (iterations).

### 3 | RESULTS

This section presents the results obtained during segmentation experiments with the U-Net architecture.

#### 3.1 | Segmentation models for breach face markings

##### 3.1.1 | Model BF-O: Breach face trained with original data

This training spanned 32 epochs (iterations) using the early stop feature. The average training time is 278 seconds per epoch (~5 min), for a total training time of about two and a half hours. With a batch size of 8, each epoch comprises 120 steps, representing an average training time of 2.32 seconds per step. Figure S4 shows two graphs containing the training and validation learning curves for performance with IoU and DC values and for optimization with the values of the Jaccard Loss.

Once training was completed, we used the two test sets (original and augmented) to evaluate the models on unseen data. Table 4 shows the results obtained during the evaluations. For the BF-O model, we can observe a decrease in the value of the metrics (more pronounced for the IoU metric) and a slight increase in the value of the loss when we evaluate using the augmented test set compared with using the original set. This implies that the augmented test data contain variations the model did not learn during training. These results were to be expected, as we trained this model on the original, that is, nonaugmented data.

##### 3.1.2 | Model BF-A: Breach face trained with augmented data

This training lasted 27 epochs, using the early stop feature. The average training time is 989 s per epoch (~16 min), for a total training time of approximately seven and a half hours. With a batch size of 8, each epoch features 359 steps, representing an average training

TABLE 5 Hardware configuration.

MacBook Pro	
Processor	2.3GHz 8-Core Intel Core i9
Memory	16GB 2667MHz DDR4
Graphics	AMD Radeon Pro 5500M 4GB Intel UHD Graphics 6,301,536 MB

time of 2.75 s per step. Figure S5 shows the performance and optimization training curves, and Table 4 presents the results obtained from the two evaluations of the BF-A model. The original and augmented test sets give similar metric and loss value results. These results indicate that the model successfully learned the variations of the augmented data.

##### 3.1.3 | Comparison of the BF-O and BF-A models

The comparison graphs in Figure 5 show the metrics and the loss of training curves for the two models trained from the breach face images. The green curve represents the BF-O model trained from the original (nonaugmented) data, and the orange curve represents the BF-A model trained from the augmented data. We can see that the two models show similar behavior: the shape of the two training curves is similar in all the comparison plots. The evaluation of both models also delivers similar performances. Still, we observe that the metrics are more stable with the BF-A model when shown the different data sets, indicating the effectiveness of the data augmentation strategy. We also notice a slight performance increase when the models are trained from the augmented data set (99.25% vs. 99.32% DC with the original test set evaluation).

#### 3.2 | Segmentation models for firing pin impressions

##### 3.2.1 | Model FP-O: Firing pins, trained with original data

This training spanned 43 epochs, using the early stop feature. The average training time is 298 s per epoch (~5 min), for a total training time of approximately three and a half hours. With a batch size of 8, each epoch comprises 120 steps, representing an average training

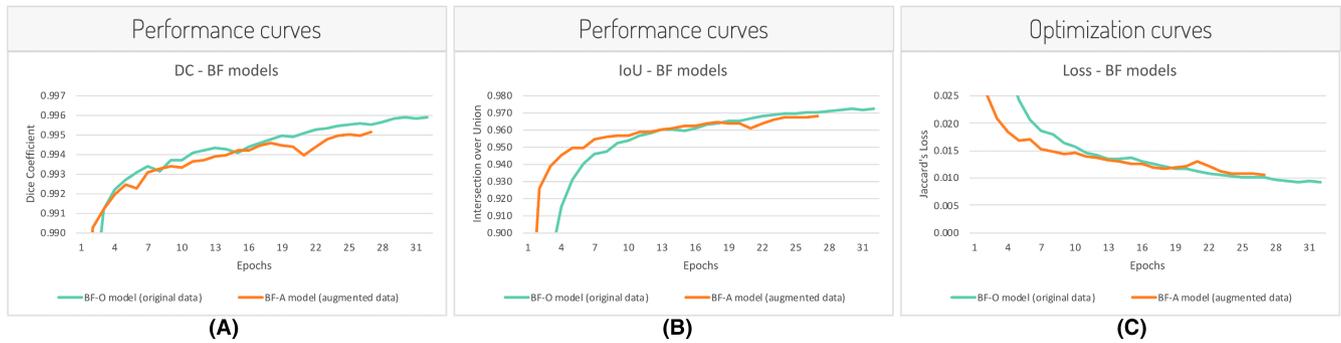


FIGURE 5 Comparative training plots of BF segmentation models: IoU (A), DC (B), and loss (C).

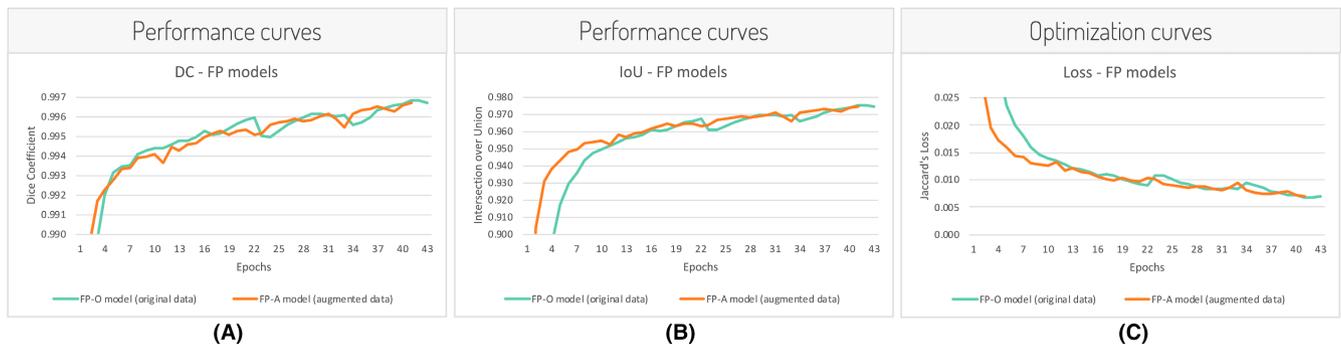


FIGURE 6 Comparative training plots of FP segmentation models: IoU (A), DC (B), and loss (C).

time of 2.49s per step. Figure S6 shows the performance and optimization training curves, and Table 4 presents the results obtained during the two evaluations of the FP-O model. We can observe the same behavior as with the BF-O model: a decrease in the value of the metrics and an increase in the value of the loss when the evaluation uses an augmented test set compared to the original test set.

### 3.2.2 | Model FP-A: Firing pin, trained with augmented data

This training took place over 41 epochs, using the early stop feature. The average training time is 1094s per epoch (~18min), for a total training time of approximately twelve and a half hours. With a batch size of 8, each epoch features 358 steps, representing an average training time of 3.06s per step. Figure S7 shows the performance and optimization training curves, and Table 4 gives the results obtained during the two evaluations of the FP-A model. Similar to the BF-A model, we observe comparable metrics and loss values for the original and augmented test sets.

### 3.2.3 | Comparison of the FP-O and FP-A models

The comparison graphs in Figure 6 show the training curves for the metrics and loss of the two firing pin segmentation models. We can

observe that the training of the FP models behaves similarly to the BF models, with both augmented and nonaugmented data. The same applies to the evaluation performances: increased stability with the FP-A model, trained with augmented data.

## 4 | DISCUSSION

When evaluated with the original test set, the BF-A model obtained an IoU value of 95.62% with a loss of 0.0143, signifying a substantial overlap between the ground truth masks and the masks predicted by the model. Figure 7 shows an example of segmentation with a high overlap rate for a BF image and a variation of the same image.

Most results predicted by the BF-A model display this high overlap rate. Figure S8 presents some instances of these successful predictions, chosen from unusual images where segmentation could have been problematic. For example, a very thin area of interest (a), a circle cut on two edges due to translations (b), a challenging inner circular delimiter (c), and a stretched inner circular delimiter (d).

In a few cases, the segmentation appears correct. Still, a protrusion (Figure 8A-E) or a cavity (Figure 8E,F) emerges on the outer circular delimiter. Placing a perfect circle to rectify the misshapen prediction would likely improve the overall results.

We can hypothesize that visible marks on the cartridge case are causing the prediction error for the image in Figure 8D. But in most cases, it is difficult to determine why the network predicted these

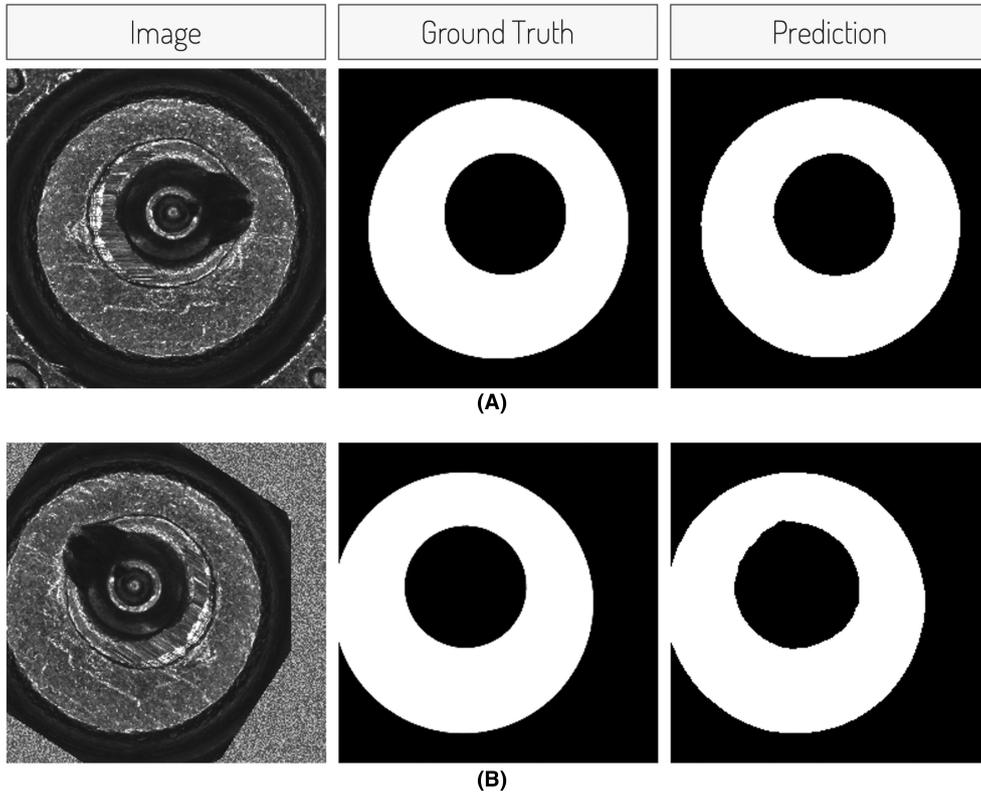


FIGURE 7 BF images, segmentation results with high overlap: original image (A) and transformed image (B).

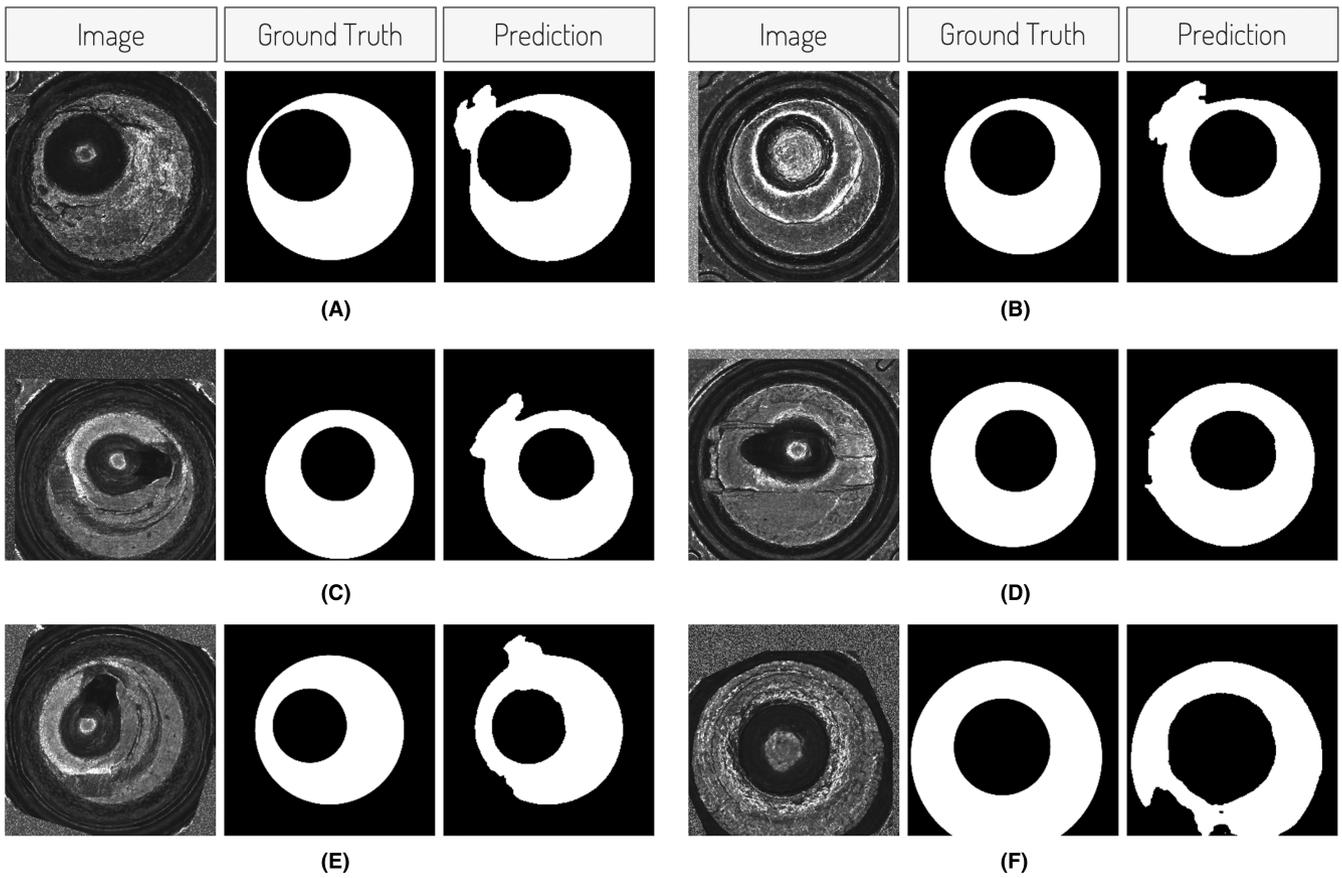


FIGURE 8 BF images and segmentation results show different protrusions and cavities.

protrusions or cavities by looking at the images. We also noticed that some prediction errors vary with the transformations applied to the images during the augmentation step. Figure S9 provides some examples of error variations depending on the transformation. The images in the left column show predictions with a significant error, while those in the right column show a variation of the same image with a lessened error. We speculate that since directional transformations in x and y are applied with the convolution and clustering operations of the neural network during training, a rotation of the image could affect the impact of these transformations on certain features.

Finally, an interesting observation is that when an image presents a segmentation result with a high error rate, most variations of this image from the augmentation step also show some segmentation difficulties. This suggests that, in such cases, the base image presents segmentation difficulties for the network. It could also suggest that the segmentation method provide a more accurate segmentation of the real ROI shape than the one provided by the

ground truth mask. Figure S10 depicts four variations of the same image with different error rates.

When evaluated with the original test set, the FP-A model achieved an IoU value of 95.89% with a loss value of 0.0113. Figure 9 and Figure S11 illustrate segmentation with high overlap rates for different firing pin images.

Generally, we notice smaller prediction errors on the firing pin images' circular delimiter than the outer delimiter in the breech face images. We also observe the same pattern described earlier. When an image provokes a prediction error, most variations of that image show similar errors. As with the BF images, the prediction error of the FP images varies with the transformations but with subtler dissimilarities. Figure 10 presents an image with segmentation errors and three variations of the same image. Figure S12 shows some examples of firing pin image segmentation, selected from cases with the highest error rates. Sometimes, the reason for the error seems obvious. For example, the unusual double center may explain the challenge experienced by the network during segmentation (a); the

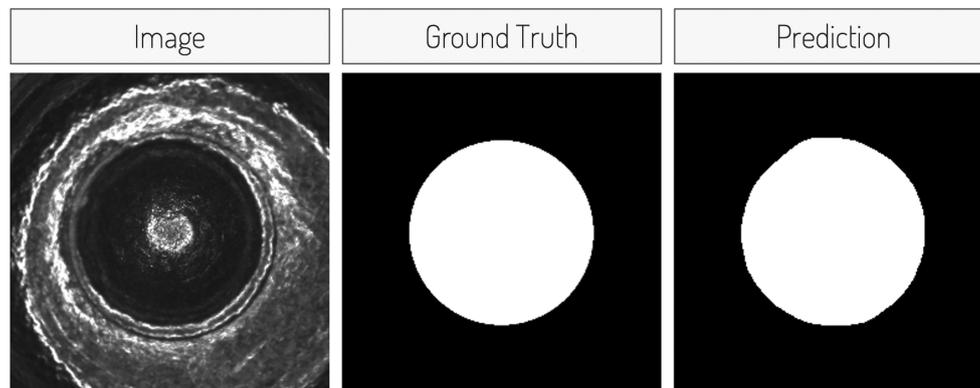


FIGURE 9 FP images, segmentation results with high overlap.

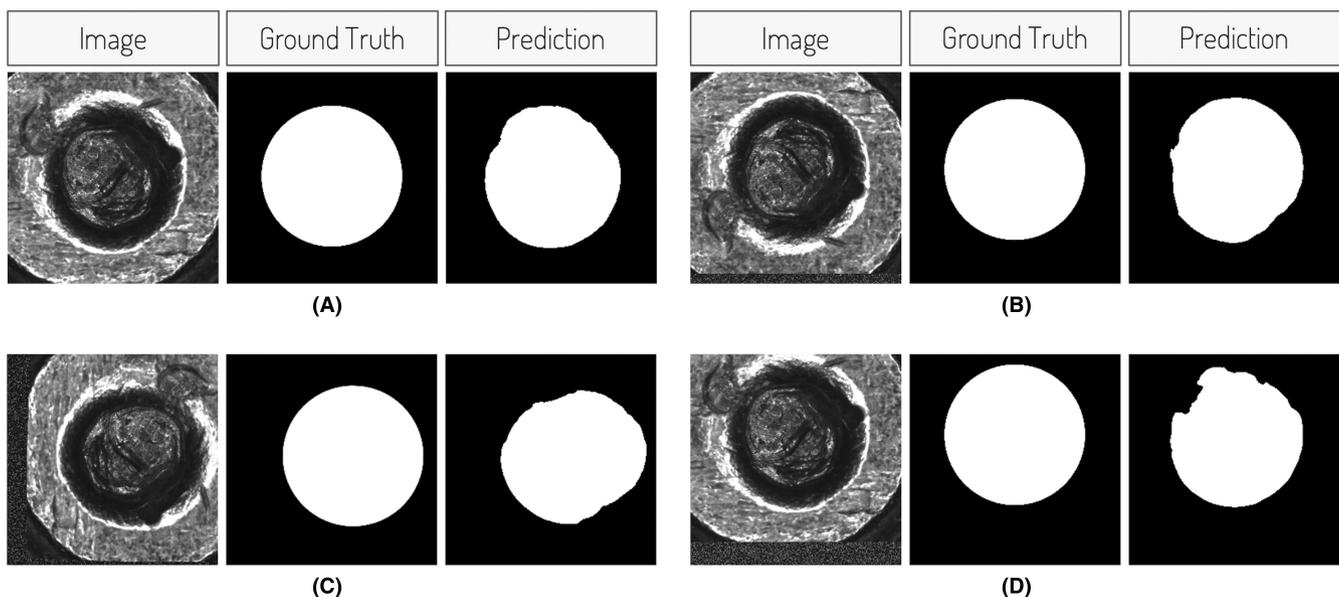


FIGURE 10 FP images, segmentation results with high error rates show different variations of an image.

many circles in the image may be difficult to differentiate (b); blurred and dark images might diminish details distinction (c–f).

Ultimately, the segmentation architecture provides a detailed answer to the original problem: automated validation of a circle delimiting a ROI. Rather than providing a simple “yes or no” answer (correctly or incorrectly positioned), the segmentation prediction provides a percentage error. We could then modify the acceptance threshold for this error to fit different applications without requiring the training of new models. Furthermore, we could use the predicted masks directly to automate a ROI selection.

## 5 | CONCLUSION

This study was to investigate using a deep-learning method to evaluate the positioning of circular delimiters in images of cartridge cases. Following the literature review, we selected a segmentation approach that could detect the entire region of interest; and metrics that supply a value of overlap between the prediction and another mask, the ground truth in this case. We considered images with two types of ROI: the breech face mark and the firing pin impression. The BF images have two circular delimiters, forming a doughnut-shaped ROI. The FP images have a single circular delimiter, forming a circle-shaped ROI. The initial data set includes several image formats for each region of interest, with a binary mask representing the ground truth and a file listing the coordinates of the circular delimiters. We selected 960×960 resolution images photographed with a ring light for our experiments.

We chose the U-Net deep-learning architecture for segmentation since it stands out in similar studies, with excellent performances on semantic segmentation tasks. We then chose to use commonly used metrics for evaluating the models: the Intersection over Union (IoU) and the Dice Coefficient (DC), and training curves of performance and optimization to compare the robustness of the models. To prepare the data for training, we split the set into three random subsets: 80% for a training set, 10% for a validation set, and 10% for a test set. To complete the data preparation, we created augmented training and validation sets by increasing the data by a factor of three and an augmented test set, increased by a factor of six. To this effect, we used random transformations of rotation, translations, flips, and salt and pepper noise.

Following a typical machine-learning methodology, we then started experiments with the U-Net segmentation architecture. We began by optimizing the hyperparameters of the models and selected a batch size 8, Jaccard's Loss function, and Adam's optimizer train the final models. Next, we trained two final models for both regions of interest: breech face mark and firing pin impression (BF and FP). We trained the first model (BF-O, FP-O) with the original (nonaugmented) data set and the second model (BF-A, FP-A) with the augmented data set. We presented model training curves and model evaluation results. The BF-A model achieved a performance of 95.6% IoU and 99.3% DC with a loss of 0.014. The FP-A model achieved a performance of 95.9% IoU and 99.5% DC with a loss of 0.011. We could improve these performances by perfectly fitting

circles over the natural shapes of the predicted masks, thus correcting various prediction errors.

Finally, we presented images of segmentation predictions and discussed some observable errors. We propose that with more work, the segmentation predictions could be used to detect misplaced delimiters in databases, using a threshold to control the rate of acceptable errors. We conclude that using a deep-learning segmentation method shows promising results applicable to the domain of firearm identification.

### 5.1 | Future work

More work must be done to use the segmentation predictions in a tool to detect misplaced circles in a database. We also believe that the method could be used as a stand-alone ROI selector. Further work is therefore needed to assess how this method performs compared with other traditional methods currently employed in automated ROI detection. Another idea consists of creating a new mask with perfect circles positioned over the predicted ROI, meaning drawing perfect circles on top of the segmented shapes, which rather follow the natural outline of the ROI on the cartridge case. This could be achieved with contour detection on the predicted masks, followed by coordinate extraction for the points belonging to these contours. An algorithm could then draw a perfect circle passing through these points. The main interest of this idea is that perfect circles could explain some overlapping errors in the metrics; the ground truth masks feature perfect circles, while the predicted masks have irregularities depending on the natural formation of the ROI. They could also correct some segmentation errors in the results, such as those in the predictions in [Figure 8](#) or [Figure S12](#). Future work could also include additional metrics on the data set, such as intra-annotation and extra-annotation information. Finally, some additional work involving the use of other images to optimize and train the models, for instance, the 3D topographic images or a solution leveraging combinations of 2D and 3D images. Images from different calibers of firearms might also constitute interesting possibilities and could add more flexibility to the tool.

### CONFLICT OF INTEREST STATEMENT

The authors have no conflict of interest to declare.

### ORCID

Marie-Eve Le Bouthillier  <https://orcid.org/0000-0001-9044-3080>

Lynne Hryniuk  <https://orcid.org/0009-0008-4975-7318>

Luc Duong  <https://orcid.org/0000-0003-3118-5389>

Sylvie Ratté  <https://orcid.org/0000-0002-6021-4129>

### REFERENCES

1. Riva F, Champod C. Automatic comparison and evaluation of impressions left by a firearm on fired cartridge cases. *J Forensic Sci.* 2014;59(3):637–47. <https://doi.org/10.1111/1556-4029.12382>

2. Gerules G, Bhatia SK, Jackson DE. A survey of image processing techniques and statistics for ballistic specimens in forensic science. *Sci Justice*. 2013;53:236–50. <https://doi.org/10.1016/j.scijus.2012.07.002>
3. Kara I. Investigation of ballistic evidence through an automatic image analysis and identification system. *J Forensic Sci*. 2016;61(3):775–81. <https://doi.org/10.1111/1556-4029.13073>
4. Zheng X, Soons J, Vorburger TV, Song J, Renegar T, Thompson R. Applications of surface metrology in firearm identification. *Surf Topogr*. 2014;2:014012. <https://doi.org/10.1088/2051-672X/2/1/014012>
5. Morris KB, Law EF, Jefferys RL, Dearth EC, Fabyanic EB. An evaluation of the discriminating power of an integrated ballistics identification system® heritage™ system with the NIST standard cartridge case (standard reference material 2461). *Forensic Sci Int*. 2017;280:188–93. <https://doi.org/10.1016/j.forsciint.2017.09.004>
6. Pisantanaroj P, Tanpisuth P, Sinchavanwat P, Phasuk S, Phienphanich P, Jangtawee P, et al. Automated firearm classification from bullet markings using deep learning. *IEEE Access*. 2020;8:78236–51. <https://doi.org/10.1109/ACCESS.2020.2989673>
7. Kudonu M, AlShamsi MA, Philip S, Khokhar G, Hari PB, Singh N. Artificial intelligence: Future of firearm examination. In: *Proceedings of the 2022 Advances in Science and Engineering Technology International Conferences (ASET)*; 2022 Feb 21–24; Dubai, United Arab Emirates. Piscataway, NJ: Institute of Electrical and Electronics Engineers (IEEE); 2022. p. 1–5. <https://doi.org/10.1109/ASET53988.2022.9735105>.
8. Riva F, Hermsen R, Mattijssen E, Pieper P, Champod C. Objective evaluation of subclass characteristics on breech face marks. *J Forensic Sci*. 2017;62(2):417–22. <https://doi.org/10.1111/1556-4029.13274>
9. Vorburger TV, Song J, Petraco N. Topography measurements and applications in ballistics and tool mark identifications. *Surf Topogr*. 2015;4:013002. <https://doi.org/10.1088/2051-672X/4/1/013002>
10. Sibert RW. Drugfire: revolutionizing forensic firearms identification and providing the foundation for a national firearms identification network. *Crime Laboratory Digest*. 1994;21:63–7.
11. UEFT. Ultra Electronics Forensic Technology. n.d [Accessed 20 Jun 2022]. Available from: <https://www.ultra-forensictchnology.com/en>
12. NIBIN. National integrated ballistic information network. Tobacco, Firearms and Explosives: ATF Bureau of Alcohol; n.d. [Accessed 20 Jun 2022]. Available from <https://www.atf.gov/firearms/national-integrated-ballistic-information-network-nibin>
13. Song J, Chu W, Tong M, Soons J. 3D topography measurements on correlation cells—a new approach to forensic ballistics identifications. *Meas Sci Technol*. 2014;25:064005. <https://doi.org/10.1088/0957-0233/25/6/064005>
14. Tai XH, Eddy WF. A fully automatic method for comparing cartridge case images. *J Forensic Sci*. 2018;63(2):440–8. <https://doi.org/10.1111/1556-4029.13577>
15. Sakarya U, Topçu O, Leloğlu UM, Soysal M, Tunali E. Automated region segmentation on cartridge case base. *Forensic Sci Int*. 2012;222:277–87. <https://doi.org/10.1016/j.forsciint.2012.07.004>
16. Zhu J, Hong R, Robin AUZ, Zhang H. Deep-learning based method for breech face comparisons. *Proceedings of the 2022 6th international conference on machine learning and soft computing (ICMLSC)*; 2022 Jan 15–17; Haikou, China. New York, NJ: Association for Computing Machinery; 2022. p. 15–9. <https://doi.org/10.1145/3523150.3523153>
17. Gambino C, McLaughlin P, Kuo L, Kammerman F, Shenkin P, Diaczuk P, et al. Forensic surface metrology: tool mark evidence. *Scanning*. 2011;33:272–8. <https://doi.org/10.1002/sca.20251>
18. Riva F, Mattijssen EJAT, Hermsen R, Pieper P, Kerkhoff W, Champod C. Comparison and interpretation of impressed marks left by a firearm on cartridge cases – towards an operational implementation of a likelihood ratio based technique. *Forensic Sci Int*. 2020;313:110363. <https://doi.org/10.1016/j.forsciint.2020.110363>
19. Fischer R, Vielhauer C. Towards automated firearm identification based on high resolution 3D data: rotation-invariant features for multiple line-profile-measurement of firing pin shapes. In: Sitnik R, Puech W, editors. *Proceedings SPIE 9393 – IS&T/SPIE electronic imaging*; 2015 mar 17. San Francisco, CA. Bellingham, WA: Society of Photo-Optical Instrumentation Engineers; 2015. 93930Q. <https://doi.org/10.1117/12.2077567>
20. Brein C. Segmentation of cartridge cases based on illumination and focus series. In: Said A, Apostolopoulos JG, editors. *Proceedings SPIE 5685 – image and video communications and processing*; 2005 Jan 16–20; San Jose, CA. Bellingham, WA: Society of Photo-Optical Instrumentation Engineers; 2005. p. 228–38. <https://doi.org/10.1117/12.585763>
21. Byeon W, Domínguez-Rodrigo M, Arampatzis G, Baquedano E, Yravedra J, Maté-González MA, et al. Automated identification and deep classification of cut marks on bones and its paleoanthropological implications. *J Comput Sci*. 2019;32:36–43. <https://doi.org/10.1016/j.jocs.2019.02.005>
22. Trivizakis E, Manikis GC, Nikiforaki K, Drevelegas K, Constantinides M, Drevelegas A, et al. Extending 2-D convolutional neural networks to 3-D for advancing deep learning cancer classification with application to MRI liver tumor differentiation. *IEEE J Biomed Health Inform*. 2019;23:923–30. <https://doi.org/10.1109/JBHI.2018.2886276>
23. Zhang S, Xu S, Tan L, Wang H, Meng J. Stroke lesion detection and analysis in MRI images based on deep learning. *J Healthc Eng*. 2021;2021:5524769. <https://doi.org/10.1155/2021/5524769>
24. Gunasekaran E, Muthuraman V. Hierarchical convolutional neural network based iris segmentation and recognition system for biometric authentication. *Proceedings of the 2020 international conference on communication and signal processing (ICCSP)*; 2020 Jul 28–30. Chennai, India. Piscataway, NJ: Institute of Electrical and Electronics Engineers (IEEE); 2020. p. 448–52. <https://doi.org/10.1109/ICCSP48568.2020.9182251>
25. Vukicevic AM, Radovic M, Zabotti A, Milic V, Hocevar A, Callegher SZ, et al. Deep learning segmentation of primary Sjögren's syndrome affected salivary glands from ultrasonography images. *Comput Biol Med*. 2021;129:104154. <https://doi.org/10.1016/j.combiomed.2020.104154>
26. Mujtaba T, Wani MA. Automatic solar panel detection from high-resolution orthoimagery using deep learning segmentation networks. In: Wani MA, Khoshgoftaar TM, Palade V, editors. *Deep learning applications*. Volume 2. Singapore: Springer Nature; 2021. p. 101–22. [https://doi.org/10.1007/978-981-15-6759-9\\_5](https://doi.org/10.1007/978-981-15-6759-9_5)
27. Wang H-C. General deep learning segmentation process used in remote sensing images. *Int arch Photogramm remote Sens spatial Inform Sci*. 2020;XLIII-B2-2020:1289–96. <https://doi.org/10.5194/isprs-archives-XLIII-B2-2020-1289-2020>
28. Lee KB, Shin HS, Hong SC, Park JM. Identification of targeted regions on an analogue site of the moon by using deep learning segmentation algorithm. 2021 earth and space conference proceedings; 2021 April 19–23; held virtually. Reston, VA: American Society of Civil Engineers (ASCE); 2021. p. 1037–46. <https://doi.org/10.1061/9780784483374.096>
29. Sun L, Shao W, Zhang D, Liu M. Anatomical attention guided deep networks for ROI segmentation of brain MR images. *IEEE Trans Med Imaging*. 2020;39:2000–12. <https://doi.org/10.1109/TMI.2019.2962792>
30. Krohn J. *Deep learning illustrated: a visual, interactive guide to artificial intelligence*. Boston, MA: Addison-Wesley; 2019.

31. Ronneberger O, Fischer P, Brox T. U-net: convolutional networks for biomedical image segmentation. In: Navab N, Hornegger J, Wells WM, Frangi AF, editors. Proceedings of the 18th international medical image computing and computer-assisted intervention conference – MICCAI 2015; 2015 Oct 5–9; Munich, Germany. New York, NY: Springer International Publishing; 2015. p. 234–41. [/10.1007/978-3-319-24574-4\\_28](https://doi.org/10.1007/978-3-319-24574-4_28)
32. Luo G, An R, Wang K, Dong S, Zhang H. A deep learning network for right ventricle segmentation in short-axis MRI. Proceedings of the 2016 computing in cardiology conference; 2016 Sept 11–14. Vancouver, BC. Piscataway, NJ: Institute of Electrical and Electronics Engineers (IEEE); 2016. p. 485–8.
33. Offe M, Fraley D, Adamson P, Principi S, Wang AS, Jordan P, et al. Evaluation of deep learning segmentation for rapid, patient-specific CT organ dose estimation using an LBTE solver. In: Bosmans H, Chen G-H, editors. Proceedings SPIE 11312 – medical imaging 2020; 2020 mar 17–20. Houston, TX. Bellingham, WA: Society of Photo-Optical Instrumentation Engineers; 2020. <https://doi.org/10.1117/12.2550314>
34. Rostami MT, Raie AA. Deep learning network for fully automatic left ventricle segmentation. Proceedings of the 2018 4th Iranian conference on signal processing and intelligent systems (ICSPIS); 2018 Dec 25–27. Tehran, Iran. Piscataway, NJ: Institute of Electrical and Electronics Engineers (IEEE); 2018. p. 141–4. <https://doi.org/10.1109/ICSPIS.2018.8700528>
35. Chen Y, Chen J, Wei D, Li Y, Zheng Y. OctopusNet: a deep learning segmentation network for multi-modal medical images. In: Li Q, Leahy R, Dong B, Li X, editors. Proceedings of multiscale multimodal medical imaging workshop (MMMI 2019); 2019 Oct 13; Shenzhen, China. Cham, Switzerland: Springer International Publishing; 2020. p. 17–25. [https://doi.org/10.1007/978-3-030-37969-8\\_3](https://doi.org/10.1007/978-3-030-37969-8_3)
36. Faghani S, Khosravi B, Zhang K, Moassefi M, Jagtap JM, Nugen F, et al. Mitigating bias in radiology machine learning: 3. Performance Metrics Radiol Artif Intell. 2022;4:e220061. <https://doi.org/10.1148/ryai.220061>
37. Goodfellow I, Bengio Y, Courville A. Deep learning. Cambridge, MA: The MIT Press; 2016.
38. Yadav O, Passi K, Jain CK. Using deep learning to classify X-ray images of potential tuberculosis patients. Proceedings of the 2018 IEEE international conference on bioinformatics and biomedicine; 2018 Dec 3–6; Madrid, Spain. Piscataway, NJ: Institute of Electrical and Electronics Engineers (IEEE); 2018. p. 2368–75. <https://doi.org/10.1109/BIBM.2018.8621525>
39. Ashraf R, Afzal S, Rehman AU, Gul S, Baber J, Bakhtyar M, et al. Region-of-interest based transfer learning assisted framework for skin cancer detection. IEEE Access. 2020;8:147858–71. <https://doi.org/10.1109/ACCESS.2020.3014701>
40. Sladojevic S, Arsenovic M, Anderla A, Culibrk D, Stefanovic D. Deep neural networks based recognition of plant diseases by leaf image classification. Comput Intell Neurosci. 2016;2016:3289801. <https://doi.org/10.1155/2016/3289801>
41. Lan L, Ye C, Wang C, Zhou S. Deep convolutional neural networks for WCE abnormality detection: CNN architecture, region proposal and transfer learning. IEEE Access. 2019;7:30017–32. <https://doi.org/10.1109/ACCESS.2019.2901568>
42. Proenca H, Neves JC. Deep-PRWIS: periocular recognition without the iris and sclera using deep learning frameworks. IEEE Trans Inf Forensics Secur. 2018;13:888–96. <https://doi.org/10.1109/TIFS.2017.2771230>

#### SUPPORTING INFORMATION

Additional supporting information can be found online in the Supporting Information section at the end of this article.

**How to cite this article:** Le Bouthillier M-E, Hrynkiw L, Beauchamp A, Duong L, Ratté S. Automated detection of regions of interest in cartridge case images using deep learning. *J Forensic Sci.* 2023;00:1–14. <https://doi.org/10.1111/1556-4029.15319>