Forensic Science International 375 (2025) 112554

Contents lists available at ScienceDirect

Forensic Science International

journal homepage: www.elsevier.com/locate/forsciint

Automated segmentation of the breech and firing pin faces of fired cartridge case images

Muthu Rama Krishnan Mookiah [b], Roberto Puch-Solis [a],*, Santo Farhan [a], Busayo Ajala [a], Niamh Nic Daeid [a]

[a] Leverhulme Research Centre for Forensic Science, School of Science and Engineering, University of Dundee, Nethergate, Dundee, Scotland, UK
[b] Computing, School of Science and Engineering, University of Dundee, Nethergate, Dundee, Scotland, UK

ARTICLE INFO

Keywords:
Segmentation
Firearm identification
Cartridge case identification
Deep learning
UNet
Dense UNet
Deeplabv3+
CNN

ABSTRACT

Firearm identification plays a crucial role in criminal justice globally. The capability to link firearms to specific crimes is invaluable for investigations and court cases. Each firearm leaves distinctive markings on bullets and cartridge cases, creating a “mechanical fingerprint” that can be used for the comparison of bullets and cartridge cases and underpins this area of forensic science. Cartridge cases fired from the same firearm exhibit similar markings on their bases. These traces can be used for investigation purposes as a means to potentially provide a link between more than one scene where cartridge cases have been recovered, or to provide a potential evidential link between a firearm and a cartridge case. These applications involve comparing the markings on the base of two or more cartridge cases, consisting of the headstamp, breech face and firing pin areas. The headstamp area usually contains information about the manufacturer and the calibre. Once this is considered, the remaining task is to compare the breech and firing pin areas of the two cartridges. Currently, some automated methods exist for this comparison, all of which involve the removal of the headstamp area to minimize bias. Some semi-automated methods for headstamp removal are available, and recently, an automated deep learning method that can be applied to 256 × 256 pixel resolution images has been introduced. In this article, we also propose a deep learning method addressing a more computationally demanding task of removing the head stamp area in higher-resolution images, 512 × 512 and 2592 × 1944 pixels, which will permit the automated extraction of finer features at a higher resolution. We also (a) introduce a post-processing method that improves the performance of our method, (b) provide the labelled data that we have produced so it can be used, together with the NIST database of cartridge case images, as a benchmark for future research, and (c) provide the estimated weights and models of the convolutional neural networks that can either be used directly or as initial values for further research. This article contributes to the emerging body of research on deep learning applications in forensic science.

1. Introduction

Firearm identification is essential to criminal justice systems worldwide [1]. Each firearm imparts markings on discharged bullets and cartridge cases, creating a “mechanical fingerprint” of the firearm, which forms the basis for the comparison of these evidence types and testing the competing propositions that bullets were fired from the same or from different firearms. This article focuses on cartridge cases.

A cartridge case, Fig. 1, is ejected from a firearm immediately after a bullet is fired. Cartridge cases fired from the same firearm would be expected to contain similar distinctive markings on their bases [2]. These markings are used to group cartridge cases recovered from a crime scene, or to potentially link the recovered traces to a firearm. These applications involve comparing the markings on the base of two or more cartridge cases.

A fired cartridge case base is a circular region consisting of three areas, namely (i) the Head Stamp Area (HSA), the Breech Face Area (BFA), and the Firing Pin Area (FPA), Fig. 2. The HSA contains factual information consisting of text that specifies the manufacturer and the calibre of the bullet, and the ejector’s mark. The BFA is the area where the firearm’s breech face impacts the cartridge case, and the FPA is the area where the firing pin hits the cartridge case, see [3,4] for a detailed description.

Once the information of the HSA has been taken into consideration, the focus lies on the comparison of the BFA and FPA of two or more

* Corresponding author.
E-mail address: rpuchsolis001@dundee.ac.uk (R. Puch-Solis).

https://doi.org/10.1016/j.forsciint.2025.112554
Received 20 January 2025; Received in revised form 4 June 2025; Accepted 30 June 2025
Available online 12 July 2025
0379-0738/© 2025 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

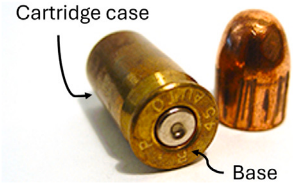


Fig. 1. A cartridge case of a fired bullet (Robert M. Thompson, National Institute of Standards and Technology).

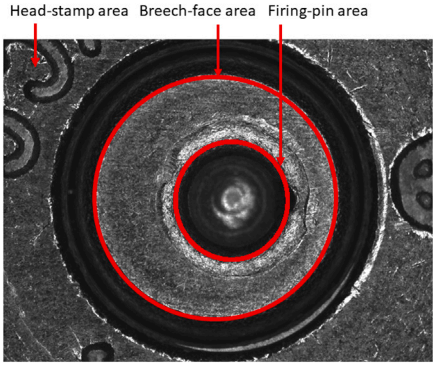


Fig. 2. Areas of the base of a fired cartridge case.

fired cartridge cases to examine these areas for similarities and/or differences. Firearm identification current practice relies mostly on visual comparison of these two areas by an expert using a comparison microscope [5–7]. There are some automated methods for cartridge case comparison [4,8], which may improve the speed and improves the objectivity of the process. These methods require automated extraction of features from the FPA and BFA. The automated removal of the HSA from the cartridge case image has become crucial so these algorithms extract features only from the regions of interest (ROIs).

With a large dataset, deep learning algorithms could be trained to extract features only from the BFA and FPA. However, large datasets are difficult to obtain. The removal of the HSA will permit the application of machine learning and deep learning algorithms for classifying cartridge cases, which incorporate automated feature extraction only from the BFA and FPA. In machine learning, the removal of sections of an image is usually called semantic segmentation or simply segmentation.

There have been some image-processing methods for removing the HSA [9,10], see [11,12] for a detailed description. In recent years, deep learning has produced excellent image segmentation results in other areas [13,14] including in medical imaging and autonomous vehicles. Recently, deep learning has been applied to image segmentation of cartridge cases [12]. In this work, a UNet convolutional neural network (CNN) was trained with 1,195 images of (256 × 256 pixel resolution) cartridge cases fired by 9 mm calibre ammunition firearms.

This article addresses the more computationally demanding task of segmentation of cartridge case images at a higher resolution, which will permit the automated extraction of finer features at a higher resolution. We use NIST's publicly available dataset [15] consisting of 1,703 images of cartridge cases also fired by 9 mm calibre ammunition firearms. In contrast with [12], the input images used here are of higher resolution: 512 × 512 pixels which would allow finer feature extraction at a higher resolution. Three CNNs were trained: UNet [16], Dense UNet [17] and DeepLabv3+ [18]. A post-processing step was added to increase performance. DeepLabv3+ returned a better performance on the NIST dataset, surpassing both UNet and Dense UNet. The segmented masks, used to produce segmented images, were upsampled to their original resolution of 2592 × 1944 pixels. This would permit the use of the images for feature extraction at an even higher resolution. The dataset of masks for the ROIs is provided as a benchmark for future research performance comparisons. The estimated parameters for the three models and the models are also provided, allowing them to be readily applied or used as starting values for further training. The predicted segmentation masks for the images in the test dataset are provided in the supplementary material.

Table 1
2D images used from the NIST database.

Study	Firearm	Cartridge	Number
Cary Wong	Ruger P89	Winchester	182
De Kinder	Sig Sauer P226	Remington CCI Wolf Winchester Speer Federal	70
Thomas Fadul	Ruger P95PR15	Federal	80
Hampby	Hi-Point C9	Remington	60
Kong	Smith and Wesson 10	Flocchi	72
Laura Lighttone	Smith and Wesson 40VE	PMC	60
FBI: Colt	Colt VM	Remington	180
FBI: Glock	Glock VM	Remington	180
FBI: Ruger	Ruger VM	Remington	200
FBI: Smith & Wesson	Smith and Wesson VM	Remington	260
FBI: Sig Sauer	Sig Sauer VM	Remington	259
Todd Weller	Ruger P95DC	Winchester	100

2. Materials and methods

2.1. Data

2.1.1. NIST dataset

We use an open-access research database, namely the NIST ballistic toolmark research database [15]. It is the largest public dataset with bullet and cartridge case toolmark data. The NIST database contains both 2D and 3D images from different studies conducted by various groups in the firearm and toolmark community. In this study, we use 1,703 2D cartridge case images discharged from a variety of weapons, including models from Ruger, Sig Sauer, Hi-Point, Smith & Wesson, Colt, and Glock, tested with multiple ammunition manufacturers such as Remington, CCI, Wolf, Winchester, Speer, Federal, Fiocchi, and PMC, Table 1. Examples of these images are shown in Fig. 3.

2.1.2. Segmentations masks

Training a CNN for cartridge case segmentation requires labelling each pixel of an image to belong to one of the three areas of interest, Fig. 2. This can be achieved by creating three masks for each cartridge case image, one for each area. A mask is a black-and-white image of the same size as the cartridge case image where the white pixels correspond to the area of interest. The ROI boundaries are rugged and close to a circular shape. It would require a significant amount of time to label them. Instead, for practical reasons, circles that enclose the vast majority of the ROIs were used. However, predictions may follow rugged boundaries. One of the authors labelled the great majority of the images (1663) and another author labelled only 40 images. Therefore, there is a very small variability introduced by people creating ground-truth data. The authors of [12] also used circles for labelling ROIs.

Each ground truth mask of a FPA was generated by placing a circle in the boundary of the FPA and BFA on the cartridge case image. Two points were selected from the boundary and a minimum bounding circle was calculated. The FPA mask was constructed by setting pixels inside the circle to white and outside the circle to black, Fig. 4(c,d). A mask containing the BFA and FPA was created using the same method, Fig. 4(a,b). A mask for the BFA was created from the joint mask for the BFA and FPA by setting the pixels corresponding to the FPA to black. A

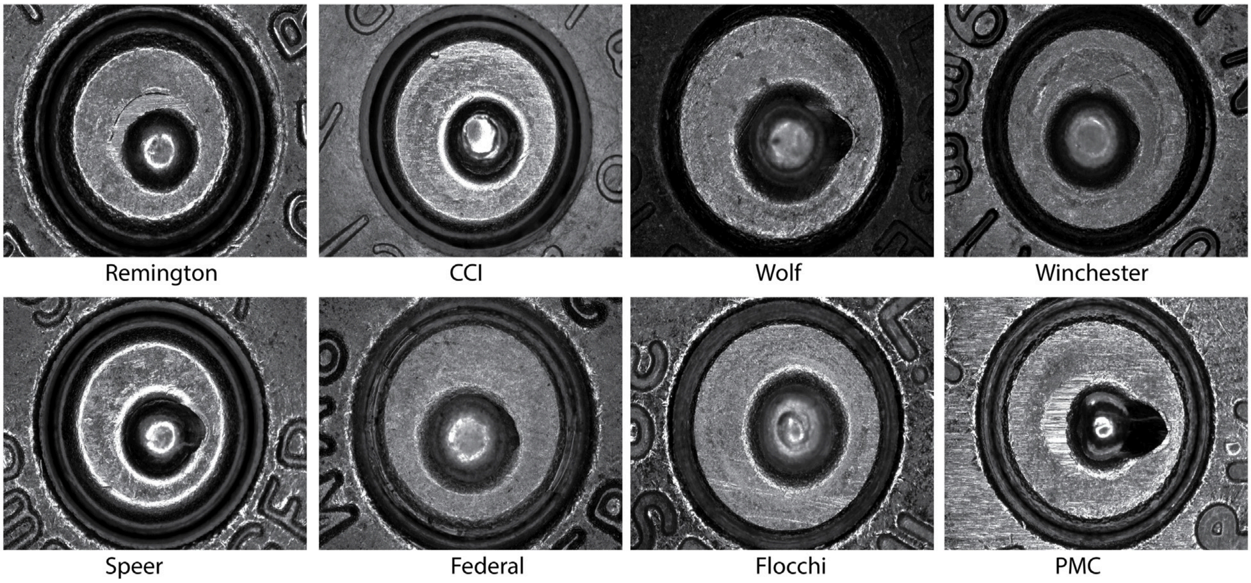


Fig. 3. Example images from different cartridge cases.

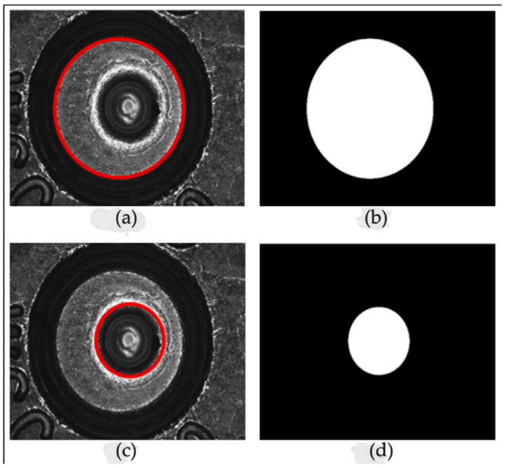


Fig. 4. Ground truth mask generation. (a) A cartridge case image with a red circle showing the boundary of BFA with HSA, and (c) a red circle showing the boundary of FPA. (b,d) are the corresponding masks. (For interpretation of the references to colour in this figure caption, the reader is referred to the web version of this article.).

mask for the HSA was created by inverting the joint mask for the BFA and FPA, Fig. 4(b), i.e. by setting black pixels to white and white pixels to black.

The original resolution of the NIST dataset is 2592 × 1944. Ground truth masks were created at this resolution where ROIs are close to circles. Our hardware setup includes an Intel Core i7-8700 CPU (48 GB RAM) and an NVIDIA TITAN Xp GPU (12 GB VRAM). Due to memory constraints, it is not feasible to use full-resolution images (2,592 × 1,944) directly for training even with a batch size of one. Additionally, our code expects square inputs, as we employ a patch-based training approach. To address these limitations, we resize the full-resolution images and ground truth masks to 512 × 512, enabling efficient training. For resizing, we use the Python OpenCV library with the cv2.INTER_AREA interpolation method. This method performs resampling by averaging pixel values over the target area, resulting in smoother and higher-quality results during down sampling. After segmentation is completed on the down-sampled image (512 × 512), the binary breech face and firing pin masks are resized back to the original resolution (2,592 × 1,944) using the cv2.INTER_NEAREST interpolation method. This method preserves the exact binary values (0 or 255) by simply replicating the nearest pixel values without introducing intermediate grey levels, ensuring the masks remain clean and accurate. These steps are illustrated in Fig. 5.

The set of masks for the FPA, BFA and both areas were collected in a dataset which was partitioned into training (70%), validation (10%) and testing (20%) datasets. The selection was performed within each type of cartridge case in Table 1, which also shows the distributions of cartridge cases in these sets. The actual partition resulted in a split of 72%, 8%, and 20% for training, validation and testing. The dataset containing the masks is publicly available in [19]. The file name of each mask contains the name of its corresponding NIST cartridge case image.

2.2. Deep learning models

2.2.1. UNet

UNet architecture is very robust and has successfully segmented various targets in medical imaging [16]. Compared to medical imaging, segmentation of cartridge cases is less complex because there are only two regions to detect and both have a circular border compared to multiple asymmetrical regions in a medical image. Therefore, we use the complexity of the UNet architecture for our task by reducing the number of encoder and decoder layers to two instead of four in the original UNet. The code was obtained from Optic-Disc-Unet in the DeepTrial repository [20]. This modification decreases the training time whilst still providing accurate segmentations. The modified architecture, Fig. 6, consists of an encoder and a decoder path typical of a UNet architecture.

The input is a 512 × 512 pixel greyscale image which is passed through a 3 × 3 padded convolution layers with 32 channels followed by a Leaky Rectified Linear Unit (Leaky ReLU) activation function to each channel. These two operations are repeated three times. The next step is downsampling from 512^2 to 256^2 filters using a 2 × 2 max-pooling operation. These operations (the three convolutions with Leaky ReLU and the max pooling) are repeated with the number of channels in the convolutions doubled to 64 resulting in 64 128^2 filters. The encoder layers end here. The output of the encoder layer is passed on to the transition layers which consists of three 3 × 3 padded convolution layers with 128 channels, each followed by a Leaky ReLU activation function. The decoder, at each step, also applies three transformations.

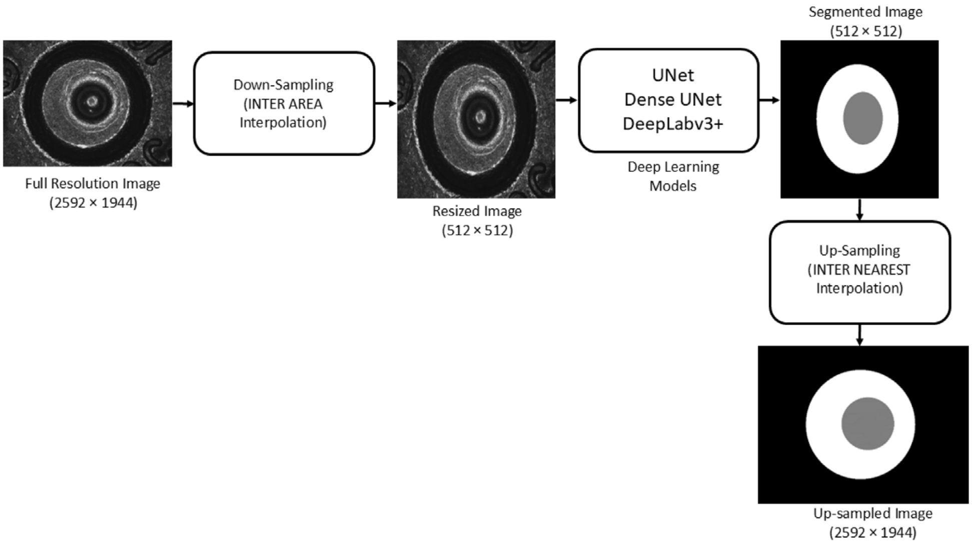


Fig. 5. Image downsampling and upsampling in training and prediction. The ground truth masks were created from the original resolution images (2,595 × 1,944 pixels).

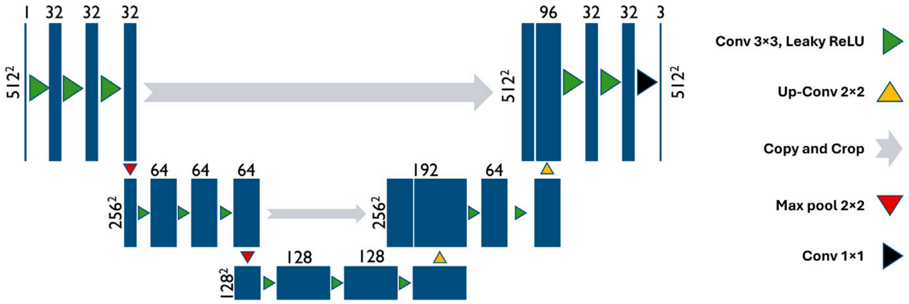


Fig. 6. UNet architecture.

The path begins with an up-sampling of the output of the transition layer. This is followed by a concatenation of the output of encoder convolution steps with 64 channels, then these are then passed through two 3 × 3 padded convolution layers of 64 channels. These three operations are repeated, this time the convolution channels are halved to 32. Finally, the output is passed through a 1 × 1 padded convolution followed by a Leaky ReLU activation. The loss function used was categorical cross entropy.

In the original UNet architecture, the encoder and decoder paths consist of four down sampling and four up sampling operations, respectively. The architecture in this research consists of two down sampling operations in the encoder path and two up sampling operations in the decoder path. The number of channels in each step of the encoder and encoder paths in this article, Fig. 6, is also reduced compared to the original architecture in Figure 1 in [16].

The total number of parameters of the UNet used here is 667,299 where 666,339 are trainable and 960 are non-trainable parameters (fixed values). The model was trained for 86 epochs with a batch size of 6 images. An early stopping strategy was applied to the training with a *patience* of 20 epochs, i.e. if there were no improvement in validation loss for 20 epochs, the training stopped and the best weights were saved for each model. The model was implemented in Python 3.6 using Keras 2.2.4 and TensorFlow-GPU 1.10, running on a computer with an Intel Core i7-8700 CPU and an NVIDIA TITAN Xp GPU. The training took 2.71 h.

2.2.2. Dense UNet

Dense UNet is a variation of the UNet architecture that has also been successful in segmenting medical images [17]. It uses a Dense Block (DB), described below, to make the architecture more robust to the vanishing gradient problem. The encoder path starts with a 512 × 512 greyscale image which is input to a 3 × 3 padded convolution layer with 32 channels followed by a Rectified Linear Unit (ReLU) activation function. The output is then passed to a DB, which comprises two 3 × 3 padded convolution layers where each layer is connected to the previous layers and shares the feature maps. The output of the DB undergoes a 2 × 2 max-pooling operation which takes the input (512^2) and outputs filters with half the height and width (256^2). The DB followed by a max pooling operation is repeated twice, where the convolution layer of the DB comprises 64 channels and each max pool reduces the height and width of the input by half. The encoder part of the architecture ends here and the output passes through a transition layer comprising of a DB with 3 × 3 padded convolution layers and 64 channels. The decoder path starts by up-sampling the output from the transition layer consisting of 64 64^2 filters, doubling the input's height and width of the filters to obtain 64 128^2 filters, which are concatenated with the encoder layer's corresponding output (64 64^2 filters). This results in 128 128^2 filters. This is then passed to a DB with 3 × 3 padded convolution layers and 64 channels. The up-sampling (concatenation with encoder output and DB operation) is repeated twice where the number of channels in the convolutions for the two DB are 64 and 32 respectively. Every up-sampling step doubles the height and width of its input. Finally, the output is passed through a 1 × 1 padded convolution followed by a ReLU activation. The loss function used was categorical cross entropy.

The Dense UNet architecture used in this research, Fig. 7, is also different from the original architecture [21]. In the original architecture, there are four down sampling and four up sampling operations in the encoder and decoder paths. Whereas the architecture adopted in

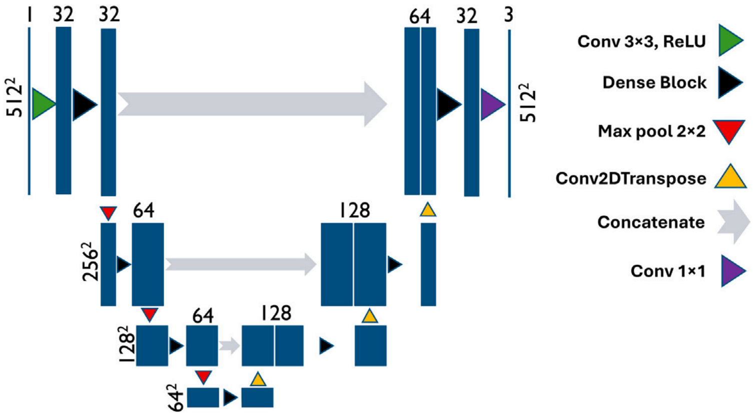


Fig. 7. Dense UNet architecture.

this research has three down sampling and up sampling operations in each path. There are also slight differences in the Dense Block (DB): the original architecture used four padded convolution blocks, however, our employed architecture had two padded convolution blocks. The number of channels in each step of the employed architecture was also reduced compared to the original architecture. The architecture reported in Figure 3 of [17] is also different from the one used in this research. The work in [17] uses two transition down and two transition up operations after DB, each consisting of Batch Normalizations, ReLU, 1 × 1 padded convolution and 3 × 3 average pooling for down or up sampling. Our work uses max-pooling operations for down sampling and convolutions 2D transpose for up sampling operations. The DB in [17] is also significantly different from the one employed in our work, which comprises three composite functions, each consisting of BN, ReLU, and a 3 × 3 padded convolution. There are also a series of skip connections that connect the output of one composite function to the outputs of all previous composite functions.

The total number of parameters in this research is 587,747 where 585,891 are trainable and 1,856 are non-trainable parameters (fixed values). The model was trained for 69 epochs with a batch size of 6 images. An early stopping strategy was applied to the training with a patience of 20 epochs. The model was implemented in Python 3.6 using Keras 2.2.4 and TensorFlow-GPU 1.10, running on a computer with an Intel Core i7-8700 CPU and an NVIDIA TITAN Xp GPU. The training took 1.76 h.

2.2.3. Modified DeepLabv3+

The third model applied in this research is a modified version of the DeepLabv3+ architecture [18,22] developed by Google, Fig. 8. Like UNet, the algorithm consists of an encoder and decoder architecture. The key elements of the algorithms are a Convolution Block (CB), Fig. 9, and an Atrous Spatial Pyramid Pooling (ASPP), Fig. 10.

A CB consists of five 3 × 3 padded convolution layers with dilation rates of 4,6,8,10 and 12 respectively. The number of channels, denoted by x in Fig. 9, in each of these convolutions is the same and is an input argument when the CB is called. The filter dimensions, y^2, are the same as the input filters. A Leaky ReLU activation function follows each convolution. The input is passed through a 3 × 3 padded convolution and the result of this is added to the output of the fifth convolution layer. and it is passed through a Leaky ReLU activation function.

In an ASPP, the input simultaneously passes through three layers. The first two layers are 3 × 3 padded convolution layers with dilation rates of 1 and 6 respectively. a Leaky ReLU activation function follows each convolution. The third layer is average pooling followed by 3 × 3 padded convolution transpose layers. The outputs of the three layers are concatenated and passed through a Leaky ReLU activation function. The specific places where DeepLabv3+ uses BBs and ASSPs are shown in Fig. 8. The loss function used was categorical cross entropy.

In the original Deeplabv3+ architecture, modified aligned Xception is used as a main feature extractor prior to ASPP, whereas in the Deeplabv3+ architecture in this article, we have used a series of CBs and max pooling layers before the ASPP layers. The architecture of the employed model before the ASPP also has several skip connections that concatenate the low-level features, which are then fed into the decoder part of the architecture.

The total number of parameters in this research is 2,986,691 where 2,980,483 are trainable and 6,208 are non-trainable parameters (fixed values). The model was trained for 34 epochs with a batch size of 6 images. An early stopping strategy was applied to the training with a patience of 20 epochs. The model was implemented in Python 3.6 using Keras 2.2.4 and TensorFlow-GPU 1.10, running on a computer with an Intel Core i7-8700 CPU and an NVIDIA TITAN Xp GPU. The training took 1.18 h.

2.3. Performance measures

We monitor pixel-wise accuracy, $(TP+TN)/(TP+TN+FP+FN)$, across epochs for both training and validation, where TP, TN, FP and FN are true positives and negatives and false positives and negatives, respectively. The performance of the segmentation methods is assessed using two well-known and commonly used measures: the Sørensen–Dice ($DICE$) and the Intersection over Union (IoU) coefficients. They are defined as,

$$DICE = \frac{2TP}{2TP + FP + FN}, \tag{1}$$

and

$$IoU = \frac{TP}{TP + FP + FN}. \tag{2}$$

Both measures score the overlap between the ground truth and the predicted segmentation masks. If the two images coincide exactly, $FP = FN = 0$ and $DICE = IoU = 1$, while if the two images have no overlap, $TP = 0$ and $DICE = IoU = 0$. IoU penalizes more FP and FN than $DICE$.

2.4. Post-processing

The predicted FPA and BFA set union occasionally contains small irregular patches in addition to the expected circular region. Post-processing aims at removing these small patches. Post-processing calculates the *circularity* and *area* of the connected components of the predicted area. The circularity takes values between zero and one, where a value of one corresponds to a perfect circle. Both circularity and area were calculated using the Matlab function `regionprops` from the image processing toolbox. The union of FPA and BFA is circular and would score a high circularity value. Post-processing is applied to the 512 × 512 pixel predicted masks and to the predicted masks after

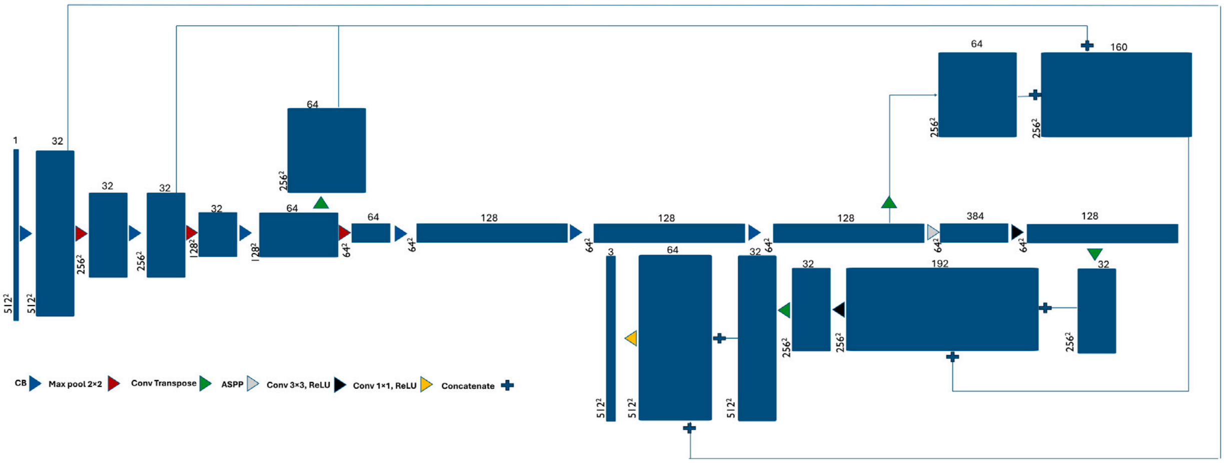


Fig. 8. Modified DeepLabv3+ architecture.

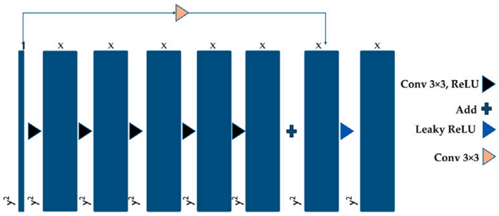


Fig. 9. DeepLabv3+ convolutional block.

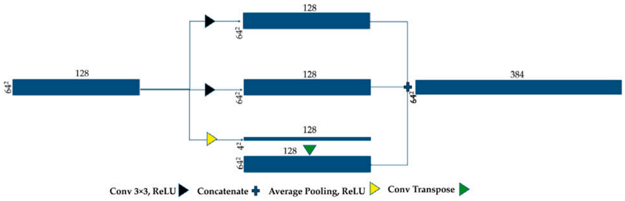


Fig. 10. Atrous spatial pyramid pooling.

upsampling. Post-processing selects regions with a circularity greater than 0.6 and an area greater than 5,000 pixels for both original-resolution images and 512 × 512 pixel images. The performance of the CNNs is evaluated before and after post-processing for both resolution images.

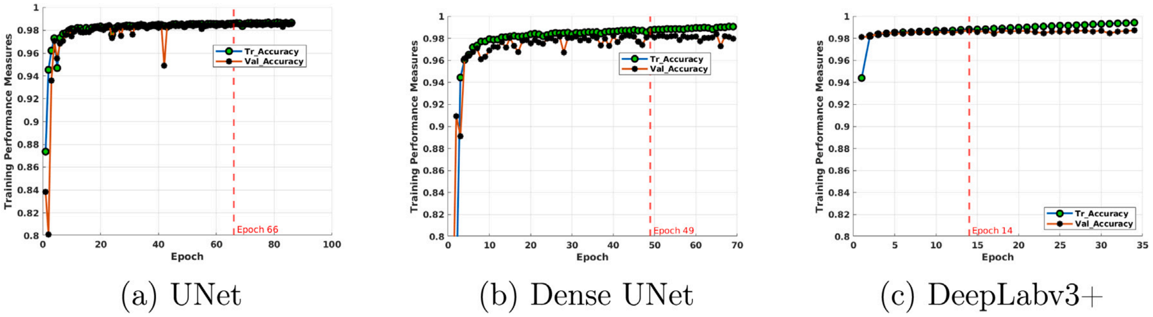
3. Results

A total of 1,703 512 × 512 pixel images were used in this study. A data split of 72% : 8% : 20% (1227, 137 and 339 images) was employed for training, validation and testing for the three algorithms (UNet, Dense UNet and Modified DeepLabv3+). The estimated parameters and models can be obtained from: github.com/LRCFS/Cartridge-Case-Segmentation. Fig. 11 shows the algorithms' training and validation accuracy by epoch. It is evident from the figure that all the algorithms reached around 98% accuracy in just 10 epochs and maintained a steady state until the end of training. The steady-state behaviour of the accuracy, coupled with its high value, suggests that the models have achieved optimal performance. The training and validation accuracies of the modified DeepLabv3+ started high (94% and 98%) and remained largely consistent throughout training, with only a slight increase in training accuracy. In contrast, the accuracies of both Dense UNet and UNet showed a large increase at the beginning of training and plateau at around 10 epochs.

Fig. 12 shows the training and validation loss by epoch. It is clear from this figure that both the training and validation losses for each algorithm decreased as the epoch number increased. However, the pattern is different for each algorithm. The training loss for the modified DeepLabv3+ reduced steadily from the beginning to the end of the training, whereas the validation loss decreased only slightly in the beginning and fluctuated by a small margin during the rest of the training. Both training and validation losses for UNet followed a similar trend where they rapidly decreased until approximately 10 epochs, after which validation loss reaches a steady state and training loss steadily decreased until the end of the training. Like UNet, the training and validation loss for Dense UNet also decreased sharply until approximately 8 epochs, after which the training loss decreased steadily until the end of the training and the validation loss fluctuated by a noticeable margin and only decreased very slightly. The training and validation losses for Dense UNet and DeepLabV3+ showed a tendency to overfit. This could be due to the limited number of training samples (n = 1,227) and the fact that breech face and firing pin patterns are not particularly complex. However, we employed early stopping with a patience of 20 and selected the final model based on the lowest validation loss. The best validation losses and the corresponding number of epochs for UNet, Dense UNet, and DeepLabV3+ were 0.032 at epoch 66, 0.4314 at epoch 49, and 0.0307 at epoch 14, respectively (Fig. 12).

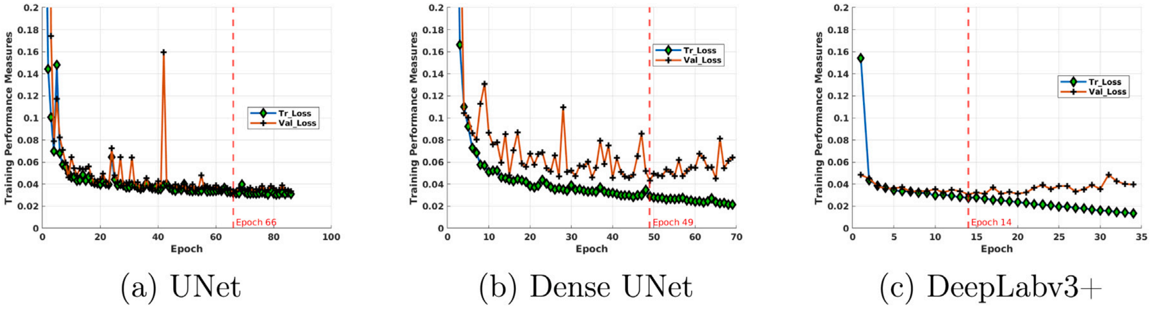
The performance of the algorithms was evaluated on 339 512 × 512 pixel test images and their corresponding 339 original resolution (2,529 × 1,944 pixel) images. The original-resolution segmentation was performed by downsampling the image to 512 × 512 pixels and feeding it to the network for prediction, then it was resized back to its original size using inter-nearest interpolation with OpenCV image processing toolbox (Fig. 5), followed by post-processing. The evaluation of the algorithms for 512 × 512 pixel images is summarized in Table 2. The classification performances are divided into three categories. In the first and second categories, we assessed the performance of FPA and BFA individually, and in the third category, we assessed the performance of these two areas together. The performance metrics are DICE and IoU. The post-processing effect on these metrics is reported in columns with titles that include the suffix "-P".

The results show that the algorithms perform very well in all the categories (DICE ≥ 95.4% and IoU ≥ 91.5%). The modified DeepLabv3+ outperformed UNet and Dense UNet. This may be because it uses a dilated convolution operation at different rates which effectively increase the receptive field of the filter without increasing the number of parameters. This enables more focus on specific regions of an image, improving feature extraction efficiently. In our study, the performance of the modified DeepLabv3+ model was only slightly better than the other algorithms. The UNet also slightly outperformed Dense



(a) UNet (b) Dense UNet (c) DeepLabv3+

Fig. 11. Training and validation accuracy.



(a) UNet (b) Dense UNet (c) DeepLabv3+

Fig. 12. Training and validation loss.

Table 2
Algorithms performance using 512 × 512 images. DICE and IoU scores are averaged over the 339 testing images. "-P" in a column title means that post-processing has been applied.

Algorithm	Area	DICE	DICE-P	IoU	IoU-P
DeepLabv3+	FPA	0.969	0.969	0.941	0.941
Dense UNet		0.959	0.961	0.922	0.925
UNet		0.963	0.964	0.930	0.931
DeepLabv3+	BFA	0.971	0.971	0.944	0.944
Dense UNet		0.963	0.963	0.929	0.930
UNet		0.970	0.970	0.942	0.943
DeepLabv3+	Both	0.992	0.992	0.985	0.985
Dense UNet		0.990	0.990	0.980	0.981
UNet		0.992	0.993	0.985	0.985

Table 3
Algorithms performance using $2,592 \times 1,944$ images. DICE and IoU scores are averaged over the 339 testing images. "-P" in a column title means that post-processing has been applied.

Algorithm	Area	DICE	DICE-P	IoU	IoU-P
DeepLabv3+	FPA	0.969	0.969	0.940	0.941
Dense UNet		0.958	0.949	0.921	0.915
UNet		0.963	0.961	0.929	0.929
DeepLabv3+	BFA	0.971	0.971	0.944	0.944
Dense UNet		0.964	0.954	0.931	0.923
UNet		0.970	0.967	0.942	0.940
DeepLabv3+	Both	0.992	0.992	0.985	0.985
Dense UNet		0.990	0.990	0.981	0.981
UNet		0.992	0.992	0.985	0.985

UNet. Dense UNet uses densely connected convolutional networks that produce a deep network structure by allowing each layer to remain connected to all preceding layers, improving feature reuse, deeper feature learning and propagation. One explanation for the performance of the Dense UNet in comparison to the UNet may be the simplicity of the segmentation task in this study. The study involved only three classification categories, and the greyscale input images all shared the same resolution, with no significant variation in pixel values between them. Performance differences between the algorithms are more likely to emerge when the training objectives are challenging and the datasets are diverse. It is interesting to observe that the result of each algorithm is better for the combined region of FPA and BFA. This can be attributed to the absence of misclassifications in the boundary of FPA and BFA. This result is important because deep learning algorithms would use the combined breech and firing pin area. It is also interesting to observe that post-processing only improves the results by a very small margin for UNet and Dense UNet. However, the removal of small misclassified areas, may have an impact on features detected by deep learning algorithms.

The evaluation of the algorithm for the original resolution images is reported in Table 3. Their performance is almost identical to that of 512 × 512 pixel images. However, the post-processing, in this case, reduces the performance of FPA and BFA segmentation for UNet and Dense UNet. For modified DeepLabv3+, the post-processing does not improve the results.

Fig. 13 shows examples of the segmentation results of UNet, Dense UNet and modified DeepLabv3+, with and without post-processing, for 512 × 512 and $2,592 \times 1,944$ pixel resolution images. The grey, white and black areas represent FPA, BFA and HSA respectively. The second and fourth rows in the figure show the output of the three CNNs and their upsampled versions, respectively. The third and fifth rows highlight the resulting images when post-processing is applied. The three CNNs perform well in this example, except for a small patch on the top right-hand corner of the image. The post-processing successfully removes misclassified patches and enhances the robustness of classification.

Fig. 14 shows an example of the ground-truth and predicted masks for the three regions and their corresponding IoU and DICE scores. The FPA and BFA together has the largest IoU and DICE scores. The FPA has slightly smaller scores than the FPA and BFA together. The BFA has smaller scores because it has two boundaries, one with FPA and the other with HSA.

Fig. 15 shows the segmentation results for five examples of cartridge case images at 512 × 512 pixel resolution using the best-performing algorithm, Deeplabv3+. The predicted masks for all 339 images in the test set at 512 × 12 resolution for the three CNNs, before and after post-processing, are provided in [23]. The grey, white and black areas represent FPA, BFA and HSA respectively. The predicted segmentation masks for the cartridge case image in the last column contain

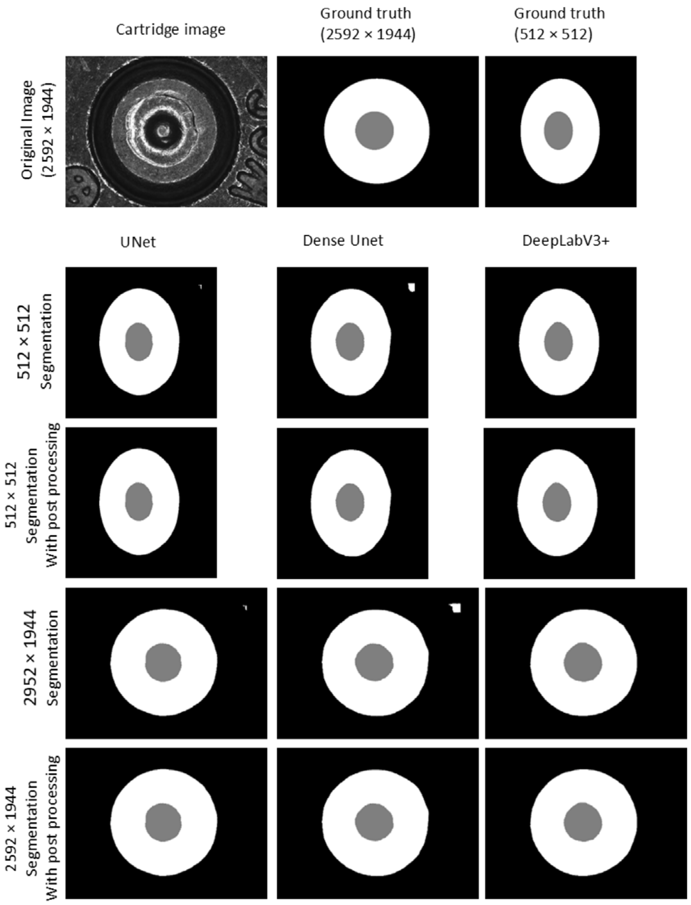

Cartridge image
Ground truth (2592 × 1944)
Ground truth (512 × 512)
Original Image (2592 × 1944)
UNet
Dense Unet
DeepLabV3+
512 × 512 Segmentation
512 × 512 Segmentation With post processing
2952 × 1944 Segmentation
2592 × 1944 Segmentation With post processing

Fig. 13. Segmentation results for cartridge case CWRBF0950, from the NIST database, using UNet, Dense UNet and DeepLabv3+ with and without post-processing (Black: HSA, White: BFA and Grey: FPA).

a small section attributed to the BFA but disconnected from it. The post-processing algorithm rectified this: it is no longer present in the post-processed image.

The predicted segmentation masks for the fourth image have a small section attributed to the FPA which is part of the BFA. The post-processing algorithm did not rectify this because the section is connected to FPA. Only two of the 339 (0.6%) images tested had this artefact. This highlights an important aspect of using an automated system: diagnostics. One measure that can be used for this purpose is the circularity of the segmentation masks. The circularity for all FPA masks but the two images with this artefact is greater than 0.85, while the circularities for the FPA predicted masks in the two images with the artefact are 0.48 and 0.57. One method of dealing with these images is highlighting them to an operator so that masks can be produced manually. Another method is to develop an extra post-processing step to remove the artefact. The method depends on the number of images that are processed. We chose the former method because of the small percentage of images with this artefact. There are also a couple of predicted masks in the supplementary materials that have small patches. These could be removed by extending the post-processing algorithm or the diagnostics.

The FPA of the cartridge case image in the last column has a protuberance on the right side. The predicted FPA mask does not contain the protuberance, however, the BFA does. This feature is distinctive and, although it is not part of the FPA, it would be considered for classification purposes in the BFA.

4. Discussion and conclusions

Firearm examiners use impressions on cartridge case bases, among other information, to address whether the same firearm may have fired two cartridge cases. The headstamp area contains information about the make and calibre of the bullet. Once this information is taken into consideration, a firearm examiner compares the breech face and firing pin areas. There is a growing research interest in developing automated methods for automated comparison based on features in these areas. This requires removing the head stamp area from images of cartridge case bases. This article aimed to segment high-resolution images into three sections, headstamp, breech face and firing pin areas using deep learning algorithms.

Three algorithms were tested: UNet with three encoder–decoder layers, Dense UNet and modified DeepLabv3+ on images with a 512 × 512 pixel resolution from the NIST cartridge case database. The algorithms performed well, returning DICE scores over 95% where modified DeepLabv3+ performed best of the three. A post-processing step was added that removed small areas outside the regions of interest and improved the robustness of the segmentation. Performance was also calculated for the upsampled predicted masks with and without post-processing, resulting in a performance similar to that of the 512 × 512 pixel resolution images.

A previous publication on segmentation of fired cartridge case images [12] used a set of 1,195 lower resolution (256 × 256 pixel) images. The data is proprietary and not publicly available. The data was augmented using translations, rotations, flipping and noise introduction to obtain a dataset of 3,945 images. The authors found that UNet with five encoder decoder layers trained with this augmented dataset performed very well. We aimed at a more computationally demanding problem of segmenting higher-resolution images of 512 × 512 pixel resolution so finer features can be extracted. The CNNs were trained with 1,703 images and without data augmentation. We found that UNet and Dense UNet performed well and that the modified DeepLabv3+

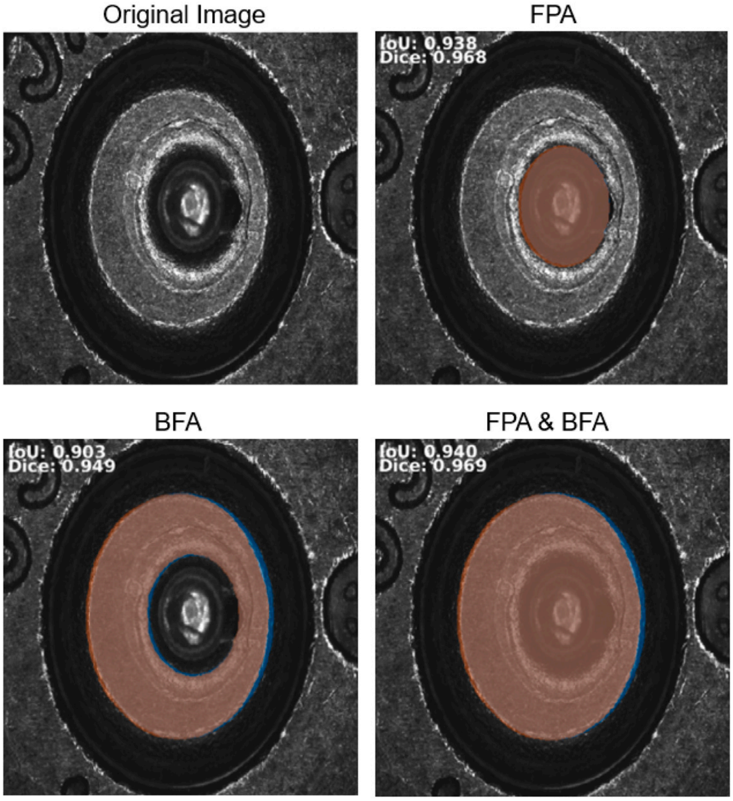


Fig. 14. Original image (CWRBF0005 from the NIST database), downsampled at 512 × 512 pixel resolution, and the superposition of ground-truth and predicted masks for the FPA, BFA and FPA and BFA together, also at 512 × 512 pixel resolution. The blue pixels are in the ground truth mask but not in the predicted mask. The red pixels are in the predicted mask but not in the ground-truth mask. The brown pixels are in both predicted and ground truth masks. (For interpretation of the references to colour in this figure caption, the reader is referred to the web version of this article.).

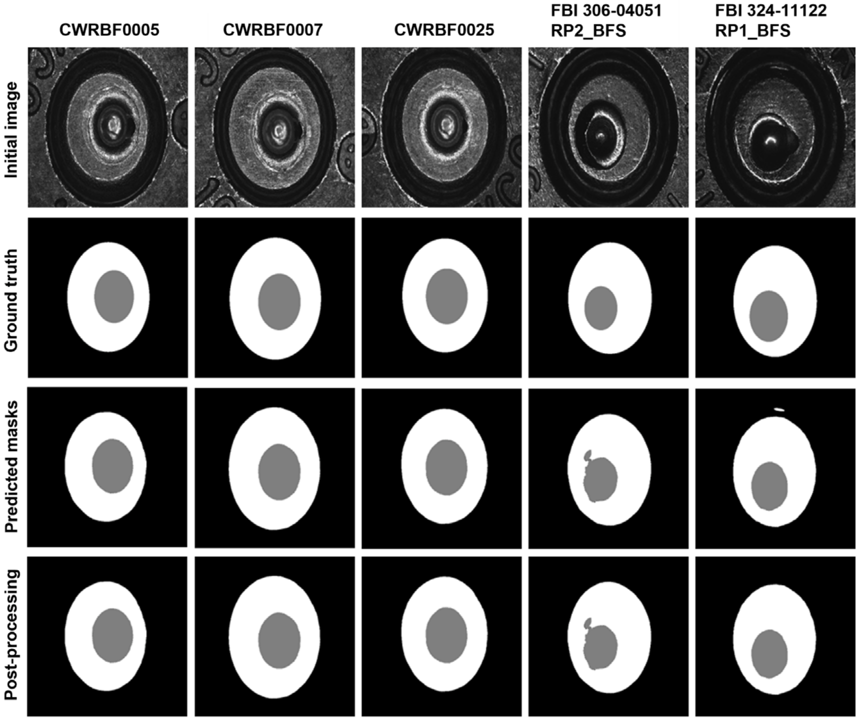


Fig. 15. Deeplabv3+ Segmentation masks for five cartridge case examples. The column names are the image names in the NIST database.

performed better. A segmentation task is a pixel base classification which means that each image contributes to $512^2 = 262,144$ data points and the modified DeepLabv3+ has about 3 million trainable parameters (in contrast with UNet and Dense UNet with about 0.7 and 0.6 million trainable parameters). This may be the reason why the modified DeepLabv3+ performed better.

The accuracies obtained in this article are not directly comparable to the accuracies reported in [12] because the datasets and methods are different. However, a comparison is informative about the accuracies that can be obtained for the methods and datasets. In our work, the best overall accuracy was obtained using Deeplabv3+: IoU = 94.1% and DICE = 96.9% for FPA, and IoU = 94.4% and DICE = 97.1% for BFA. The accuracy reported in [12] is slightly better than our work: IoU = 95.9% and DICE = 99.5% for FPA and IoU = 95.6% and DICE = 99.3% for BFA. We also calculated the accuracy for FPA and BFA combined: DICE = 99% and IoU = 98%. The work reported in [12] does not address the accuracy of the combined region. This is an important aspect because deep learning feature extraction algorithms would be applied to BFA and FPA combined region.

We are very satisfied with the performance achieved and expect our method to be useful in the automated comparison of cartridge cases. However, publicly available datasets are necessary to evaluate the generalizability of our findings. We have made the ground-truth segmentation masks database publicly available. The dataset is split into training, validation and testing, to make it possible for other researchers to use the data as a benchmark and to compare their results with ours. We encourage researchers to share their data. Automated comparison of forensic pattern evidence (e.g. shoemarks, fingerprints, bullet striations) using machine learning is in its early stages. The extraction of region of interest from pattern evidence images will be required and a body of research in forensic image segmentation is needed. This article contributes to the body of research in the area of automated forensic pattern analysis.

CRediT authorship contribution statement

Muthu Rama Krishnan Mookiah: Writing – review & editing, Writing – original draft, Methodology, Formal analysis, Data curation, Conceptualization. **Roberto Puch-Solis:** Writing – review & editing, Writing – original draft, Supervision, Project administration, Methodology, Formal analysis, Data curation, Conceptualization. **Santo Farhan:** Writing – review & editing, Writing – original draft, Visualization, Validation, Methodology, Formal analysis, Conceptualization. **Busayo Ajala:** Writing – review & editing, Writing – original draft, Visualization, Validation, Project administration, Conceptualization. **Niamh Nic Daeid:** Writing – review & editing, Writing – original draft, Methodology, Funding acquisition, Formal analysis, Conceptualization.

Funding

The research was funded by The Leverhulme Trust, United Kingdom (RC-2015-011).

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships which have, or could be perceived to have, influenced the work reported in this article.

References

[1] Royal Society, Understanding Ballistics, a Primer for Court, Royal Society, 2021.
[2] J. Song, Z. Chen, T.V. Vorburger, J.A. Soons, Evaluating likelihood ratio (LR) for firearm evidence identifications in forensic science based on the congruent matching cells (CMC) method, Forensic Sci. Int. 317 (2020) 110502.
[3] U. Sakarya, O. Topçu, U.M. Leloğlu, M. Soysal, E. Tunalı, Automated region segmentation on cartridge case base, Forensic Sci. Int. 222 (1–3) (2012) 277–287.
[4] N. Basu, R.S. Bolton-King, G.S. Morrison, Forensic comparison of fired cartridge cases: Feature-extraction methods for feature-based calculation of likelihood ratios, Forensic Sci. Int. Synerg. 5 (2022) 100272.
[5] E. Hare, H. Hofmann, A. Carriquiry, Automatic matching of bullet land impressions, Ann. Appl. Stat. (2017) 2332–2356.
[6] F.P. León, Automated comparison of firearm bullets, Forensic Sci. Int. 156 (1) (2006) 40–50.
[7] M. Mookiah, R. Puch-Solis, N.N. Nic Daeid, Identification of bullets fired from air guns using machine and deep learning methods, Forensic Sci. Int. 349 (2023) 111734.
[8] M. Tong, J. Song, W. Chu, R.M. Thompson, Fired cartridge case identification using optical images and the congruent matching cells (CMC) method, J. Res. Natl. Inst. Stand. Technol. 119 (2014) 575–582.
[9] U. Sakarya, O. Topçu, U.M. Leloğlu, M. Soysal, E. Tunalı, Automated region segmentation on cartridge case base, Forensic Sci. Int. 222 (1–3) (2012) 277–287.
[10] C. Brein, Segmentation of cartridge cases based on illumination and focus series, in: Image and Video Communications and Processing 2005, vol. 5685, SPIE, 2005, pp. 228–238.
[11] X.H. Tai, W.F. Eddy, A fully automatic method for comparing cartridge case images, J. Forensic Sci. 63 (2) (2018) 440–448.
[12] M. Le Bouthillier, L. Hrynkiw, A. Beauchamp, L. Duong, S. Ratté, Automated detection of regions of interest in cartridge case images using deep learning, J. Forensic Sci. 68 (6) (2023) 1958–1971.
[13] S. Minaee, Y. Boykov, F. Porikli, A. Plaza, N. Kehtarnavaz, D. Terzopoulos, Image segmentation using deep learning: A survey, IEEE Trans. Pattern Anal. Mach. Intell. 44 (7) (2021) 3523–3542.
[14] J. Wu, R. Fu, H. Fang, Y. Liu, Z. Wang, Y. Xu, Y. Jin, T. Arbel, Medical sam adapter: Adapting segment anything model for medical image segmentation, 2023, arXiv preprint arXiv:2304.12620.
[15] NIST ballistic toolmark research database, 2024, https://tsapps.nist.gov/NRBTD/, (Accessed 24 October 2024).
[16] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, in: Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18, Springer, 2015, pp. 234–241.
[17] C. Wang, Z. Zhao, Q. Ren, Y. Xu, Y. Yu, Dense U-net based on patch-based learning for retinal vessel segmentation, Entropy 21 (2) (2019) 168.
[18] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, H. Adam, Encoder-decoder with atrous separable convolution for semantic image segmentation, in: Proceedings of the European Conference on Computer Vision, ECCV, 2018, pp. 801–818.
[19] M. Mookiah, R. Puch-Solis, S. Farhan, B. Ajala, N. Nic Daeid, Segmentation masks for breech and firing-pin areas for a NIST dataset of firearm cartridge case images, 2025, URL https://doi.org/10.15132/10000262.
[20] K. Xing, D. Herenu, Optic-disc-unet, GitHub, 2018, URL https://github.com/DeepTrial/Optic-Disc-Unet.
[21] S. Cai, Y. Tian, H. Lui, H. Zeng, Y. Wu, G. Chen, Dense-unet: a novel multiphoton in vivo cellular image segmentation model based on a convolutional neural network, Quant. Imaging Med. Surg. 10 (6) (2020) 1275–1285, http://dx.doi.org/10.21037/qims-19-1090.
[22] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, A.L. Yuille, Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs, IEEE Trans. Pattern Anal. Mach. Intell. 40 (4) (2017) 834–848.
[23] M. Mookiah, R. Puch-Solis, S. Farhan, B. Ajala, N. Nic Daeid, Predicted segmentation masks from UNet, dense UNet and DeepLabv3+ for breech and firing-pin areas for a NIST dataset of firearm cartridge case images, 2025, URL https://doi.org/10.15132/10000268.