IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 8, 2020, accepted June 30, 2020, date of publication July 13, 2020, date of current version July 24, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3008943

Bullet Impact Detection in Silhouettes Using Mask R-CNN

RICHAR FERNÁNDEZ VÍLCHEZ AND DAVID MAURICIO
AI Group, National University of San Marcos, Lima 15081, Peru
Corresponding author: David Mauricio (dmauricios@unmsm.edu.pe)

ABSTRACT The bullet impact detection in the silhouette plays an important role in the institutions which have a shooting range, specially where are regulated the license to carry a gun and where the evaluation manual process could present distortion in the ability's evaluation of the shooter and like a consequence the license's issuance. This paper proposes a method for an automatic detection of the bullet impacts in silhouettes based on deep learning and image processing, which consist of the following steps: pre-processing, impacts detection, edge detection and evaluation results. The experiments about 600 silhouettes with 2401 bullet impacts images of the proposed and implemented method considering the models Resnet 50 and Resnet 101 for Mask R-CNN show that Resnet 50 get better results than Resnet 101, achieving 97.6 %, 99.5 %, and 97.9 %, of accuracy, precision and recall, respectively, above the methods Circular Hough Transform, Circlet Detection, Random Sample Consensus, Randomized Hough Transform, Randomized Circle Detection, Support Vector Machine, Faster R-CNN, MnasNet and YOLO. Also, the results show 100 % of effectiveness in the edge detection and the count of the detected bullet impacts.

INDEX TERMS Edge detection, bullet impact detection, image processing, Mask R-CNN.

I. INTRODUCTION

The shooting range is an outdoor or indoor space, conditioned to practice the gunshot. The countries which have governmental institutions to issue license to carry a gun have these spaces which get the required characteristics and safety measures according to the current standards. The process to get the license, generally, contemplate a psychological test to make sure of the person's mental health, a theoretical test which objective is to evaluate the normative knowledge and basic concepts related to a gun, a practical test to determine the ability of handling, the gun supply and basic safety rules, and, finally, the test which measure the skill of the shooter (shooting test). Generally, this process is applied in some countries [1]–[4].

To take the shooting test, the people must follow the given security obligations for the regulatory institution, besides, it must respect the instructions of the responsible of the shooting range rigorously. The evaluation of the test is done standing in the shooting line, with a gun which contains a certain amount of ammunitions, in front of the silhouette placed on according to the current standards of the institution (see Fig. 1). To pass this test, the shooter must reach the shooting threshold inside the target that is in the shooting silhouette, which is determined with the detection of the bullet impact in a silhouette (DBIS) [4].

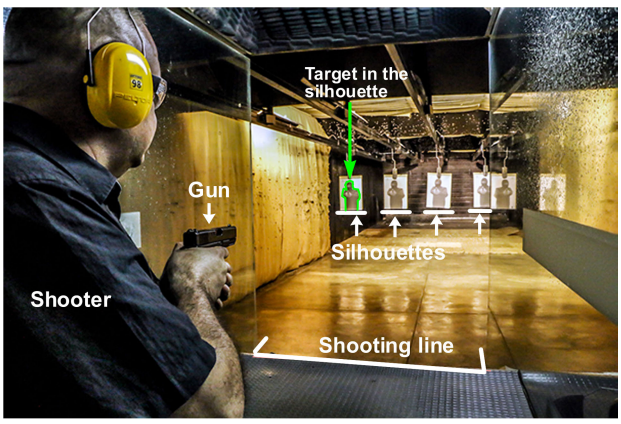

FIGURE 1. A shooter during the evaluation of a test in a governmental institution.

In some countries where the evaluation process of the shooting test is performed manually, there is a supervisor who is in charge of signaling if the candidate passes or fails the shooting test and visualizing the bullet impacts detection

The associate editor coordinating the review of this manuscript and approving it for publication was Massimo Cafaro.

and count in a shooting silhouette [4]–[6], who becomes a vulnerable being to acts of corruption, which generates the granting of gun licenses to unfit people. During 2019, in some regions such as Latin America, it was reported that more than 1 in 5 people paid bribes when they used a public service [7].

An alternative to mitigate errors in the shooting evaluation process is given by automation of this service, using artificial intelligence techniques to solve DBIS. However, an exhaustive review on Web of Science and Scopus, where the keywords were used [gunshot detection, gunshot recognition, bullet recognition, bullet impact recognition, gunshot pattern, gunshot localization, bullet impact detection], shows nowadays there is no publication for DBIS through computing techniques. The few studies related to DBIS are oriented to the detection of the shooter [8], bullet impact analysis of woven fabrics [9], [10], shooting detection through accelerometer [11], and particle residue analysis [12].

However, nowadays, there are several machine learning techniques for the processing of images that have achieved high precision for a number of tasks, such as the classification of images, the detection and locating of objects, one of them is Mask R-CNN, which has achieved precisions of 97.8 %, 95.78 %, 98.5 %, and 85 % by solving problems of detection of fake images [13], detection of skin burns regions [14], detection of workers and danger zones in a building [15], and detection of breast lesions [16], respectively. So, Mask R-CNN could be used to solve DBIS.

In this paper, it's proposed the bullet impact detection in silhouettes based on Mask R-CNN along with image processing techniques. This techniques combination based on deep learning makes it possible to detect, segment and recognize the bullet impacts accurately, while the image processing techniques help in identifying the edge of the silhouette in order to evaluate the shooter's accuracy.

This work is organized into 6 sections. In the section II, a background and literature review are performed about circular detection, Mask R-CNN and edge detection. In the section III, the method for automatic bullet impact detection is proposed: pre-processing, impact detection, edge detection and evaluation results. The section IV describes the automatic bullet impact detection through the Mask-R-CNN implementation, edge detection, and evaluation results. Model validation is presented in the section V. Finally, the Section VII presents the conclusions and future works.

II. BACKGROUND AND LITERATURE REVIEW

The bullet impact detection in silhouettes involves tasks to detect, locate, and identify the bullet hole region, as well as evaluating the accuracy of the shooting to identify whether it impacted in or out the silhouette target. This means that the challenge of the problem focuses on performing an object classification (Are the objects impacts on the silhouette?), object location (where are the impacts?), and object instance segmentation (What portion of the image is the bullet hole?). Because of it, studies related to shooting and impacts, Mask R-CNN and edge detection are reviewed.

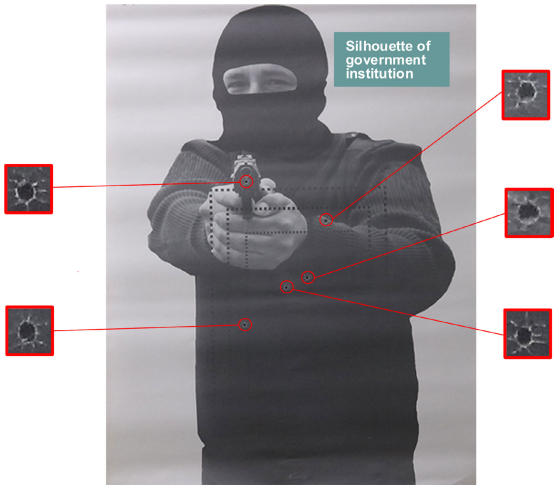

FIGURE 2. Real silhouette used in a shooting range of a governmental gun control institution.

A. RELATED WORKS

There are studies related to shooting in the military field for the measurement and identification of the small-caliber bullets orientation before it reaches the impact surface [17], and the use of acoustic location techniques of shooting to be able to identify the shooter [8]. In the forensic field, images of potential residues of particles from the shooting, taken with a hyperspectral camera, are analyzed in order to identify automatically through the use of machine learning algorithms the existence of samples of these residues [12]. In other areas, a literature review about bullet impacts study has been performed in woven fabrics [10], where the penetration resistance of the bullet impact is reviewed. On the other hand, the bullet impacts in woven packages are analyzed in order to develop a model for ballistic interaction simulation for multilayer woven packages [9]. Also, the shooting is detected using portable accelerometers to demonstrate the use of these sensors and identify signals corresponding to the use of guns [11].

As it can note, the studies which were found take several approaches, however, they give a little information about DBIS, which it is an image processing problem. An important aspect to solve DBIS is the detection of a bullet impact hole in a silhouette, and generally it has a circular shape (see Fig. 2), for that reason we considered to review the studies about the detection of circular shapes in images. One of these techniques is Circular Hough Transform (CHT), used in [18]–[20] to detect objects with circular shapes in each research. In [18], CHT is used to locate the coronary root in the aorta, in order to segment the coronary tree from CT images. In [19] and [20], is used to detect and count nano particles and red blood cells, respectively, in order to solve the required manual work problem in the counting of the mentioned objects.

In other studies, a Support Vector Machine (SVM) classifier is used to recognize patterns of bunches in vineyard [22], thus achieving an accuracy of 95.5 %. In [23], a new Circlet

Transform (CTT) is implemented for the analysis of microscopic images, detecting and counting red blood cells, achieving an accuracy of 93.3 %. On the other hand, the Random Sample Consensus (RANSAC) algorithm is proposed for the circular detection of transparent contact lenses using an SVM image classifier [24], achieving an accuracy of 90.63 %. In the same way, in [25], the same algorithm is used for the automatic gauge detection to find the most fitted ellipse to the targeted object. Randomized Hough Transform (RHT) is implemented in [26] for the identification of elliptical forms of the optic disc as part of the early glaucoma diagnostic in an expert system based on fuzzy logic, with an accuracy of 96 %. On the other hand, [27] introduces Randomized Circle Detection (RCD), implementing a specific voting strategy based on the curvature of the isophotes.

Studies based on deep learning also show effectiveness in detecting and locating objects, such as [28], [29], using Faster R-CNN. Whereas in [28] Faster R-CNN is used for the automatic detection of rice seedlings with an accuracy and recall of 98.9 % and 91.9 %, respectively, in [29] is used for the oil tank detection of satellite images with an accuracy and recall of 97.36 % and 94.66 %, respectively. In [30], [31], You Only Look Once (YOLO) is proposed, with works that contribute to detection response speed. In [30], YOLO is used with 2 different focal length cameras for vehicle and license plate detection with accuracies of 90.38 % and 86.1 %, respectively, whereas in [31] is used for the fabric defect detection with effectiveness rates of 97.81 %. On the other hand, Mask R-CNN and UniNet 2.0 are proposed for iris recognition [32], where a precision of 96.8 % is achieved. Other studies use deep learning focused on the size of the trained model, such as [33], which uses a MnasNet architecture for mobile phones.

B. MASK R-CNN

Mask Regional Convolutional Neural Network (Mask R-CNN) is an effective object detection technique based on Faster R-CNN [34], which complements and extends its architecture to introduce an additional layer of segmentation per each instance. He *et al.* [35] conceptualizes it in a simple way: while Faster R-CNN provides 2 outputs for each detected object (the labeled class and bounding box), Mask R-CNN provides a third output which is called Mask.

This technique based on a deep neural network achieves the segmentation of instances for each bounding box using a fully connected layer (FC) in parallel with the location and identification, in this way, it is capable of generating the proposed areas of regions where the objects might be detected. Mask R-CNN, which is seen from a high level, consists of 3 parts: the backbone, a convolutional neural network (usually ResNet, VGG or ConvNet) with the function of getting the characteristics of the analyzed image from a low level as edges, until a high level like people, animals, among others. The second is a light neural network called RPN (Region Proposal Network) that is responsible for receiving the map of features generated by the backbone in order to find the regions of interest (RoI), so, the areas that can contain the objects. With the following regions proposed by RPN, they are sent to the third scenario, an FC layer and a fully convolutional network (FCN) to generate 3 outputs for each RoI, the classification of the object in the region of interest, the bounding box (with the purpose to refine the location and size of the bounding box) and the mask. This last scenario generates scores for bounding boxes, classification, and segmentation masks. The overall architecture of the Mask R-CNN is shown in Fig. 3.

Works related to this technique allow the recognition of different objects, for example, a model capable of detecting two types of image manipulations: copy-move and splicing, in order to perform the detection, location and segmentation of the fake images in forensic analysis, where an average accuracy of 97.8 % is achieved [13]. Also, an automated detector of strawberry based on Mask R-CNN in order to improve the manual harvest in the strawberry industry, with an average accuracy rate of 95.78 % and a recall of 95.41 % [36]. On the other hand, there is a model for the automatic detection, segmentation and classification of breast lesions with ultrasound images, where an accuracy of 85 % is achieved [16]. Yang *et al.* [15], in order to avoid accidents in building constructions when fall heavy objects, proposes a method that recognizes, locates and segments workers and danger zones, and then measures the safety distance, in this way it reaches an accuracy of 98.5 %.

C. EDGE DETECTION

The image processing techniques provide a set of tools and methods for processing and analyzing images for object recognition, edge detection, and so on. As the scope of this work is to detect the accurate impacts within the target of the silhouette, it is considered to detect the limit of the figure, i.e., to detect its edges that serve as a reference to evaluate whether the impact is inside or outside it.

The studies found [15], [20], [21], [37]–[40] are supported on the Canny algorithm to process the images and achieve the edge detection, besides, to present proposals for algorithm improvement. In [38] is used to mitigate the problems of noisy and broken edges. On the other hand, in [37], a modified Canny algorithm is used to segment the images with a high amount of noise in order to obtain the limit of the object. In [40], an edge detection option based on Artificial Neural Network (ANN) is proposed where the Canny algorithm is used for neural network training, i.e., it achieves a preprocessing function of the image before using other techniques. In [39], Canny plays a principal role in detecting the edges prior to the circular detection using power histograms, which is processed as a pre-circular step using CHT. In the case of [20], its objective is the detection of particles using Canny for edge detection, which is processed as a previous step to the circular detection using CHT. Also, in [21], a combination of Canny and CHT is used to detect eye exudates, pre-processed and segmented region to achieve the final result using a convolutional neural network.

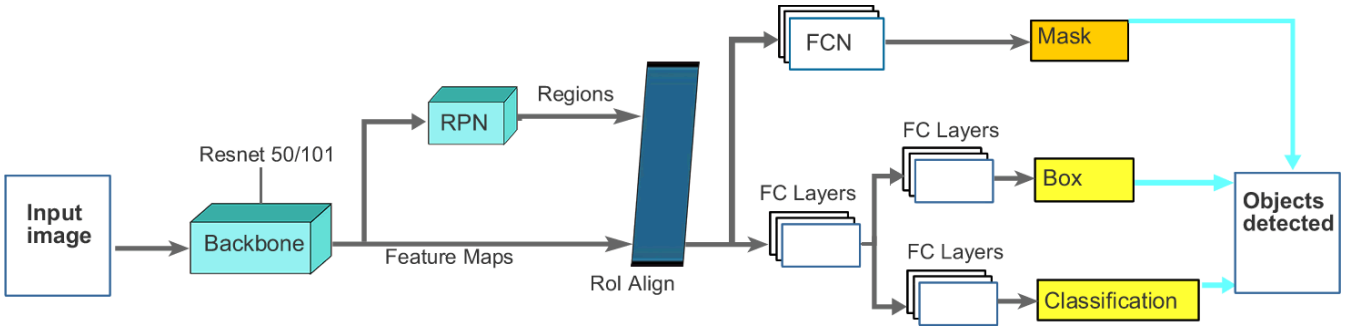

FIGURE 3. Mask R-CNN Network Architecture.

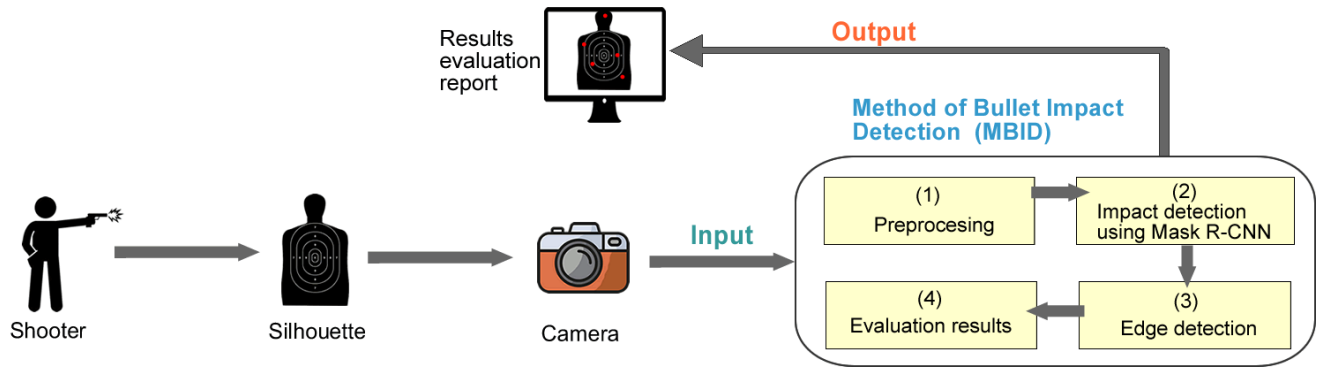

FIGURE 4. Method for resolving DBIS.

III. DETECTION OF REAL IMPACTS IN SHOOTING SILHOUETTES

To solve DBIS, a 4-step method is proposed, based on images processing techniques and artificial intelligence: 1) Pre-processing, 2) Impact detection, 3) Edge detection and 4) Evaluation results. Finally, each step performs a specific task to obtain another image with the detected impacts and edges, which it will be shown in order to know the performance of a shooting test in a shooting range. Fig. 4 shows the method and how DBIS is resolved.

The process for resolving DBIS through the method is as follows: the shooter is positioned on the shooting line of the shooting range and performs a regulated number of shootings in the silhouettes (test). Once the test is completed, the image capture of the silhouette is performed using a device such as a camera, and then it's processed by the proposed method of bullet impact detection (MBID), which it also generates a report of evaluation results.

A. METHOD OF BULLET IMPACT DETECTION (MBID)

In MBID, each component is executed in the indicated order in Fig. 4, where each output of the previous component is the input of the next component. This means that the interaction among components is a sequence of activities that run one after the other.

The pre-processing component will give as output a more suitable image, improving certain characteristics of the figure such as the color and size, in addition to make it possible to perform faster processing operations. The impact detection will output an image with the bullet impacts detected and drawn in the silhouette using Mask R-CNN, while the edge detection helps to get an image with the detected edges in the shooting silhouette, then to identify and count the successful impacts in the next evaluation result component. The last component generates the results report.

B. PRE-PROCESSING

This component includes a set of techniques to initiate image processing. The purpose of pre-processing is to obtain, from a source image, to another whose result is more appropriate, adapting particular characteristics of the image and obtain better results in its processing. The steps that run on this component are to convert the image to grayscale, and then resize it and so that they all have the same size. The final result of this component is a prepared image to detect the impacts.

1) GRAYSCALE CONVERSION

This image processing technique is highly recurring when it's performed objects detections in images by converting a color image to grayscale in order to gain processing speed, because the color increases the complexity and time when it's processed. In [20]–[22], [41], the grayscale is part of the initial method of processing of an image which it served as a reference to include it in MBID.

2) IMAGE RESIZING

This technique refers to the resizing of an image and it is applied in MBID in case the image does not have the dimensions of 768 × 1024 pixels, which are the dimensions of the captured image by the camera and in several experiments have given better results. To avoid distorting the image, the same percentage is applied to both sides of the image and that percentage is calculated by reference to the width of the image. Just like [21], [22], [41]–[43], the resizing is part of the object detection techniques.

C. IMPACT DETECTION

Impact detection is one of the main components which will allow to identify the impacts on shooting silhouettes in order to account for all the made impacts. The process flow starts with receiving the pre-processed image, loads the trained model, runs the model for detection, and draws on the detected bullet impacts in the silhouette.

This component uses Mask R-CNN and was established a simple classification to predict whether the area to be identified in the silhouette is a bullet impact or not. Mask R-CNN delivers 3 proposals: the classification (class), the bounding box of the bullet hole (bbox) and the mask of pixels of the hole (mask). The classification is unique because it will only be known whether it is a bullet impact or not, the box delimits the impact region, whereas the mask takes the pixels from the bullet hole, delimiting the object.

The trained model Mask R-CNN is loaded into this component, as well as its set configurations: the backbone network (based on Resnet), batch size, learning rate, mask shape, among other configurations detailed in the section IV-A. The image processed in the previous component will be continued the impact detection flow using the loaded Mask R-CNN model, thus, both of them: bounding boxes and masks are drawn on the processed image (see Fig. 5). The left silhouette shows an image containing 4 real impacts performed on a shooting test, and other shapes or elements that can confuse and complicate the counting of bullet impacts, even it has done it manually. The right silhouette is one after using the impact detection model, which shows the 4 detected bullet impacts, delimiting and segmenting the region of the bullet hole.

D. EDGE DETECTION

This component in MBID seeks to detect the edges of the shooting silhouette, in other words, to delimit the object to identify the region in which the impacts are considered successful, and outside of them as unsuccessful. All impacts will be accounted for and categorized (successful and unsuccessful) in the evaluation result component. The steps which are included in this component to apply: image noise, dilation, erosion, image threshold, and final edge detection. This set of techniques will facilitate the edge detection making that different elements from the silhouette such as logos of the institution, written in the silhouette, as well as the same holes, disappear and the edge detection

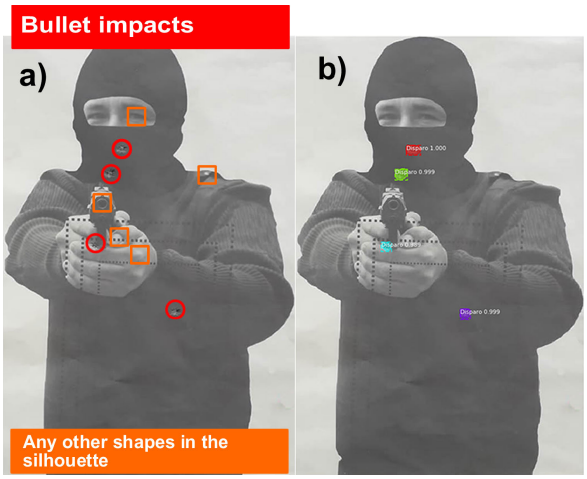

FIGURE 5. a) Silhouette to detect, b) Silhouette with detected impacts using Mask R-CNN.

will be more efficient. Each step of this component is detailed below:

1) IMAGE NOISE

Median Filter technique is used to reduce the noise in an image. New methods based on this technique are presented in [40], [45], in order to remove high noise density.

By applying the Median filter on this component, it can achieve an effect that blur on the image in order to reduce the external elements in the silhouette and in this way to improve the edge detection results.

2) IMAGE DILATION

Dilation is a morphological operation of the image processing which increases the size of objects by using a structuring element as an input [46]. In [47], a method based on this technique is proposed to improve image recognition smoothing regions of the image.

This component is applied in order to further disappear the external elements to the silhouette using a rectangular dilation kernel. One effect that is achieved with this process is to dilate the silhouette, therefore only a low number of iterations is defined in order not to distort the image.

3) IMAGE EROSION

The erosion, like dilation, is a morphological operation, which performs the opposite action by decreasing the size of objects. This technique in MBID aims to recover the dilation of silhouette size after applying the dilation procedure, in addition to contributing with the disappearance of elements outside the silhouette.

4) IMAGE THRESHOLD

The thresholding process is used to segment an image based on pixel intensity variation. If the pixel value is above the value of a threshold, it is set to a foreground value, otherwise, a background value [48]. Whereas the conventional threshold uses the same value for all pixels, an adaptive

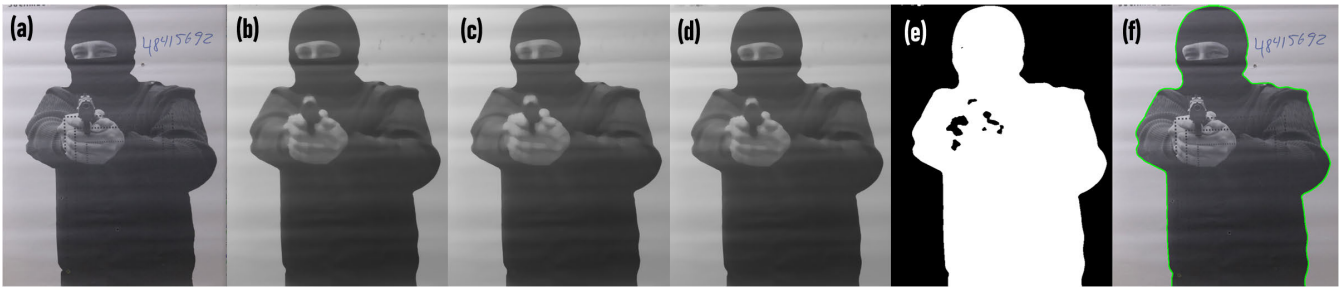

FIGURE 6. Results of applying "Edge Detection" component. a) Pre-processed silhouette, b) Silhouette with Median Filter, c) Dilated silhouette, d) Eroded silhouette, e) Silhouette with adaptive threshold, f) Silhouette with edges found and drawn on the initial silhouette.

threshold determines it dynamically for a pixel [49], making it better suited to the contrasts and grayscale tones of an image. In [50], a filter based on this technique is proposed to eliminate the noise in corrupt digital images in order to restore them.

In MBID, the adaptive threshold is used to separate the image by regions based on the variation in intensity between the silhouette pixels and the background pixels.

5) EDGE DETECTION

The edges are the set of lines that go along the boundaries of an image, and applying this technique in MBID means identifying the edges of the shooting silhouette. The combinations of all the previous techniques in this component allow to reduce the amount of information that is processed significantly, and therefore the edge detection is processed with relevant pixels that make the result more effective. Once the edge detection process is completed, the border is drawn in the image.

The result of applying these 5 steps are shown in Fig. 6.

E. EVALUATION RESULTS

This component allows to qualify the impacts made by the shooter. It is considered a successful impact if it is within the region of the silhouette, otherwise, it is considered a failed or unsuccessful impact. Then the results are drawn on the shooting silhouette (see Fig. 7).

IV. IMPLEMENTATION

A. MASK R-CNN

The detection model Mask R-CNN uses a pre-trained model based on the COCO dataset, which contains the trained weights of a huge number of images from different categories used in object detection and segmentation. This pre-trained model was used to initialize the backbone of the Mask R-CNN model.

The Mask R-CNN backbone can be built with residual networks, such as ResNet 47, 50, 71 or 101, which each one also has different numbers of layers. For the detection model training, Resnet 50 and 101 were configured with the defined values in Table 1.

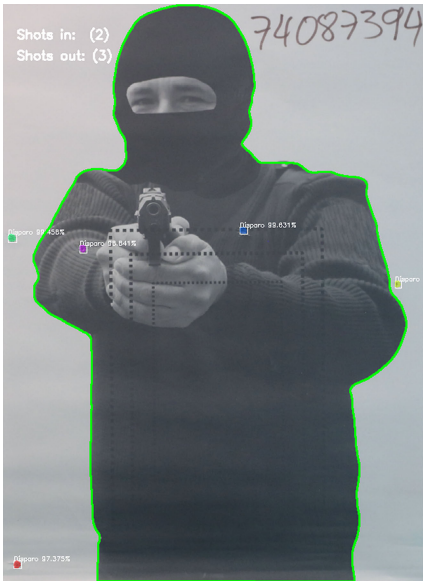

FIGURE 7. Final result of testing a shooting silhouette with MBID.

TABLE 1. Parameters configured for the Mask R-CNN model.

Parameter	Value
Epochs	80
Steps per epoch	50
Optimizer	SGD
Learning rate	0.001
Learning momentum	0.9
Weight decay	0.0001
Batch size	2
Backbone sttrides (the strides of each layer of the FPN Pyramid)	[4, 8, 16, 32, 64]
Mask shape	28x28

Mask R-CNN was implemented with the Python language and the open source Tensorflow and Keras libraries for the deep learning.

B. EDGE DETECTION AND EVALUATION RESULTS

The edge detection and evaluation results components are implemented with the OpenCV library. For Image noise, its implemented *MedianBlur* function is used, which a positive odd value is sent. The configured value is 13 and represents the size of the kernel that performs the image scan.

For image dilation and erosion, a structuring element (of any shape, or size) is used that calculates the shape of a pixel closeness over a maximum value taken for dilation and a minimum value for erosion. In MBID, it was configured with a value of 2×2 and the formulas used in the library for dilation and erosion are defined in (1) and (2), respectively.

$$dst(x, y) = max_{(x',y'):element(x',y')\neq0} src(x + x', y + y') \quad (1)$$

$$dst(x, y) = min_{(x',y'):element(x',y')\neq0} src(x + x', y + y') \quad (2)$$

where *src* is the image to be processed, *element* is the structural element used for dilation or erosion, *dst* the output image, *max* is the maximum value over the given kernel area, *min* is the minimum value, (x', y') it is the coordinate of the pixel to erode or dilate, and (x, y) is the dilated or eroded pixel.

For the adaptive threshold, a function that requires additional values that condition the output of the image is used, which are detailed in Table 2, and the default formula of the OpenCV library is also used.

TABLE 2. Parameters configured for adaptive threshold processing using OpenCV.

Parameter	Description	Value for OpenCV
maxValue	Value that represents if the pixel value is greater than the threshold value	255
adaptiveMethod	Type of adaptive threshold algorithm to use (Mean/Gaussian)	ADAPTIVE_THRESH_GAUSSIAN_C
thresholdType	Threshold Type (Binary/Inverse binary)	THRESH_BINARY_INV
blockSize	Size of a neighborhood of pixels which is used to calculate the threshold value for the pixel	2041
C	Constant value subtracted from the mean or weighted mean	9
src	Image to process	Eroded image
dst	Destination image processed	-

Also, the *findContour* function has been considered with the *CHAIN_APPROX_SIMPLE* method to get the edges of the image. Lastly, the evaluation of results was implemented with the *pointPolygontest* function, which determines whether a point is inside, outside, or at the same edge. This function brings a positive value if the point is inside, negative, outside, or equal to zero or on the same edge.

V. VALIDATION

A. DATASET

In this work, shooting silhouettes were used to perform the training and validation tasks of the Convolutional Model Mask R-CNN, and the testing of MBID too. The silhouettes were digitized in JPG format with a resolution of 2680 × 3600 pixels and then they were resized to the size of 768 × 1024 pixels. The camera was in a perpendicular position to each used silhouette (with real impact) 2 meters away, and they were taken at a shooting range of a governmental gun control institution.

A total of 600 images were obtained in 20 days, which were used as follow: 81 % (486 images) for training, 9 % (54 images) for validation, and the rest 10 % (60 images) for subsequent tests. The images to be trained were accompanied by annotations with the masks of the bullet holes using the VIA annotation tool (VGG Image Annotator).

The dataset has silhouettes with different amounts of bullet impacts, in other words, there are silhouettes from 1 to 5 impacts, which are the thresholds of the institution where the silhouettes were collected (see Table 3).

TABLE 3. Distribution of dataset bullet impacts.

Silhouettes	Number of silhouettes	Impacts inside the target	Impacts outside the target	Total impacts
With 1 impact	43	30	13	43
With 2 impacts	54	83	25	108
With 3 impacts	76	193	35	228
With 4 impacts	113	403	49	452
With 5 impacts	314	1428	142	1570
TOTAL	600	2137	264	2401

B. SCENARIOS

The implementation of Mask R-CNN was worked for 2 experimental scenarios. Scenario 1 considers all the impacts (inside and outside the target), this is to say, 2401 impacts, and scenario 2 only considers impacts inside the target, that is 2137 impacts (see Table 3). The purpose of the first scenario is to train the Mask R-CNN model with the mayor quantity of impacts in the silhouettes, whereas the second is to improve the model by training impacts of the same type (only impacts inside the target).

C. CROSS VALIDATION

In order to achieve the best performance of the convolutional model, the cross validation method was used with 540 selected images randomly. The Cross validation k-fold sets the "k" like a unique parameter, referring to the number of groups in which the data sample is divided, and in this work it's considered k-fold equal to 10, thus, 10 groups will be performed, 1 to validate and 9 to train.

D. LOSS FUNCTION

At the end of the training process, the values of the recommended loss function by the author of Mask R-CNN were obtained [29], who establishes the loss of multiple tasks, shown in (3).

$$L_{final} = L_{class} + L_{bbox} + L_{mask} \quad (3)$$

The 3 lost L_{class}, L_{bbox} and L_{mask} correspond to the classification loss, bounding box and mask, respectively. These values are individually calculated for each region of interest and the sum of all of them defines the final loss (L_{final}) of Mask R-CNN.

The L_{class} value indicates the loss which it gives to the wrong classification of the object to be detected, showing how safe the trained model is to predict the classification labels, i.e., how accurate the model is to identify whether it is an impact or not. The L_{bbox} value indicates the associated loss with the bounding box and how far away it is from the true box, which means how accurate the model is to locate the impact inside the silhouette. Lastly, L_{mask} indicates how well the model segments the objects, this corresponds to the impact masks created in the shooting silhouette. For the first and second scenario, these values are shown in Table 4 and 5, respectively.

TABLE 4. Loss values of the first scenario for Resnet 50 and 101.

Loss function	Resnet 50		Resnet 101	
	Training Loss	Validation Loss	Training Loss	Validation Loss
L_{class}	0.0226	0.0152	0.0242	0.0452
L_{bbox}	0.1597	0.1575	0.1608	0.1340
L_{mask}	0.2892	0.2656	0.2865	0.2858
L_{final}	0.4715	0.4383	0.4715	0.4650

TABLE 5. Loss values of the second scenario for Resnet 50 and 101.

Loss function	Resnet 50		Resnet 101	
	Training Loss	Validation Loss	Training Loss	Validation Loss
L_{class}	0.0183	0.0037	0.0168	0.0188
L_{bbox}	0.1716	0.1829	0.1636	0.5199
L_{mask}	0.2912	0.2859	0.2809	0.4521
L_{final}	0.4811	0.4725	0.4613	0.9908

The results show that the loss present a low value in the first scenario regardless of the classification model to be used, this is explained because in the first scenario the dataset additionally includes 12.3 % of bullet impact images than the second scenario. In addition, the results show for both scenarios that the Resnet 50 has lower loss than Resnet 101, with 5.74 % lower in the first scenario. The graphs in Fig. 8 show a declining trend of loss types with a trend of stabilization since the epoch number 60, with less deviation in training than in validation. Besides, the lost classification L_{class} is less than the L_{bbox} and L_{mask} loss, this is explained because it is considered a single type of object to be classified (the bullet impact).

VI. RESULTS

The performed experiments are presented to demonstrate the efficiency of the trained model and the used image processing techniques to determine how successful a shooting was, for that reason, 60 shooting silhouettes were used which contain a total of 249 bullet impacts that were not used in the training and validation process. The tests were performed on a PC with NVIDIA GeForce GTX 1060 for GPU acceleration, Intel (R) Core (TM) i7-8750H CPU and 16 GB of memory.

The several experiments for the Mask R-CNN model are evaluated with the measures: Accuracy (AC), Precision (PR), and Recall (RC) which are defined in (4), (5) and (6), respectively.

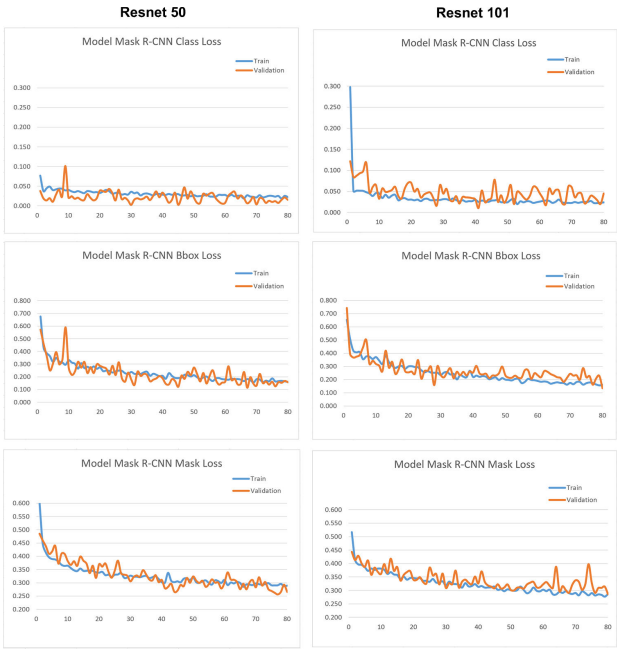

FIGURE 8. Loss function L_{class}, L_{bbox} y L_{mask} of Resnet 50 and 101 of the first scenario.

$$AC = \frac{TP + TN}{TP + FP + TN + FN} \tag{4}$$

$$PR = \frac{TP}{TP + FP} \tag{5}$$

$$RC = \frac{TP}{TP + FN} \tag{6}$$

where $TP = True\ Positives$, $TN = True\ Negatives$, $FP = False\ Positives$ y $FN = False\ Negatives$. These values will be obtained from the detection results, presented in a confusion matrix, to get to detail explicitly, where the model is right, omits or confuses the class with other objects.

Table 6 shows the confusion matrix for the 2 proposed Mask R-CNN classification models (Resnet 50 and 101) and for YOLOv3, MnasNet, Faster R-CNN (Resnet 50 and VGG), SVM, RANSAC (Circle and Ellipse), RHT, CTT, RCD and CHT, where DUM and NDUM refer to the impact detection and non-detection, respectively. All these techniques have been considered because they are part of the state-of-the-art for object detection with efficient results, in order to compare all of them with the proposal. Also, the results for Mask R-CNN (Resnet 50 and 101) have been considered for the second scenario because it does not deserve to show the results for the other techniques since some are invariant in the size of the dataset and others worsen its efficiency when it is reduced. RCD was implemented by reference to [54], changing the thresholds because the bullet impact has a different size from those tested in the reference, just like [55] for CTT. CHT was implemented with the OpenCV library, taking as a reference part of the method proposed in [56]. In the tests using

TABLE 6. Confusion Matrix of the first and second scenario.

Method	DUM TP	DUM FP	NDUM FN	NDUM TN	Total
First scenario					
Mask R-CNN Resnet 50	244	1	5	0	250
Mask R-CNN Resnet 101	240	8	9	0	257
YOLOv3	210	0	39	0	249
MnasNet	182	0	67	0	249
Faster R-CNN Resnet 50	161	7	88	0	256
Support Vector Machine	153	27	96	0	276
Faster R-CNN VGG	137	1	112	0	250
RANSAC Circle	119	584	130	0	833
RANSAC Ellipse	113	784	136	0	1033
Randomized Hough Transform	107	459	142	0	708
Circlet Transform	88	221	161	0	470
Randomized Circle Detection	62	647	187	0	896
Circular Hough Transform	44	345	205	0	594
Second scenario					
Mask R-CNN Resnet 50	243	2	6	0	251
Mask R-CNN Resnet 101	194	3	55	0	252

RANSAC [52] and RHT [53] it was necessary to process the image by sections because it was difficult to complete the process due to the high calculation generated by the number of iterations performed in these algorithms. YOLOv3 was implemented with the imageAI library [57], MnasNet using the Google tool called AutoML, Faster R-CNN by reference to [58] but implementing the Resnet 50 backbone. SVM was implemented by reference to [59] with a 3-stage method and detection using HOG (Histogram of Oriented Gradients) descriptors.

It can observe that YOLO and MnasNet are the only ones which detect 249 impacts, in other words, the other methods detect 1 to more false positives, for example confusing the muzzle gun as one more impact. This can be explained because YOLO has an architecture that does not search for interesting regions in the image that could potentially contain the object, otherwise it accesses to the whole image in predicting boundaries, training it end-to-end with a loss optimizing [51]. On the other hand, MnasNet architecture [33] implements multi-objective optimization that aims to achieve high precision and speed.

RANSAC, RHT, CTT, RCD and CHT achieve lower detection results, this can be explained because these algorithms need preprocessing, such as setting thresholds for the specific problem or even the image. As the dataset was collected over a period of 20 days and at different time points, the silhouettes have different types of brightness or color shades, which could influence the number of impacts and false positives detected. The number of false positives can also be explained because the bullet impact does not always have a well-defined circular shape and its edge can have different openings, grooves and slits resulting from the bullet impact. These techniques also detect circular shapes in the silhouettes, but many of them do not correspond to bullet impacts, for example, the target person's eye in the silhouette.

Mask R-CNN is the model that has the best result, followed by the YOLOv3, MnasNet and Faster R-CNN Resnet

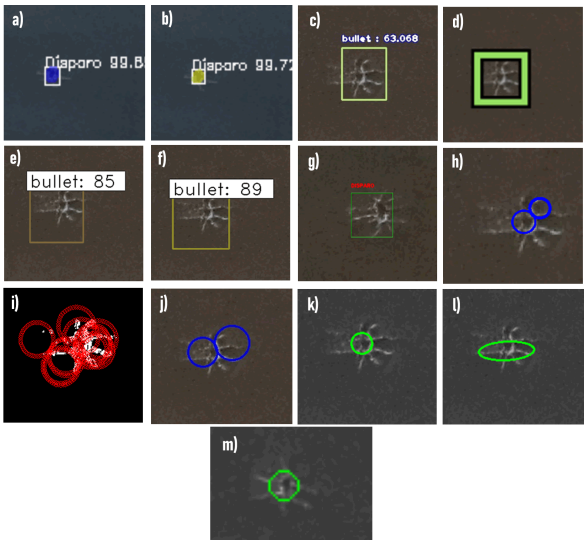

FIGURE 9. Bullet impact detection using different methods. a) Mask R-CNN Resnet 50, b) Mask R-CNN Resnet 101, c) YOLOv3, d) MnasNet, e) Faster R-CNN Resnet 50, f) Faster R-CNN VGG, g) SVM, h) CTT, i) RHT, j) RCD, k) RANSAC Circular, l) RANSAC Elíptica, m) CHT.

TABLE 7. Accuracy, precision and recall of the first scenario.

Method	Accuracy (AC)	Precision (PR)	Recall (RC)
Mask R-CNN Resnet 50	97.6 %	99.5 %	97.9 %
Mask R-CNN Resnet 101	93.3 %	96.7 %	96.3 %
YOLOv3	82.4 %	100 %	82.4 %
MnasNet	73.1 %	100 %	73.1 %
Faster R-CNN Resnet 50	62.9 %	95.8 %	64.7 %
Support Vector Machine	55.4 %	85 %	61.5 %
Faster R-CNN VGG	54.8 %	99.3 %	55 %
Circlet Transform	18.7 %	28.5 %	35.3 %
Randomized Hough Transform	15.1 %	18.9 %	42.9 %
RANSAC Circular	14.3 %	16.9 %	47.8 %
RANSAC Elipse	10.9 %	12.6 %	45.4 %
Circular Hough Transform	7.40 %	11.31 %	17.67 %
Randomized Circle Detection	6.9 %	8.74 %	24.9 %

50 models, this can be explained because Mask R-CNN achieves good results with few images [14], unlike the others that need a dataset with more training images. Likewise, the Mask R-CNN models have better results in terms of correct detection (TP + TN) in the first than the second scenario, this can be explained because in the first scenario the models are trained with more images quantities than in the second.

Table 7 shows that Resnet 50 has better values than the other methods, achieving 99.5 % of PR, and 97.9 % of RC, i.e., it does not identify 5 bullets from 249 bullet impacts. The Mask R-CNN Resnet 101 model also has good results, and very superior to other techniques. On the other hand, the edge detections and the evaluation results were also verified, thus, no inconveniences were found for the 60 silhouettes, i.e., the edges were correctly generated and the bullet impacts were counted.

VII. CONCLUSIONS

A method called MBID (Method of Bullet Impact Detection) has been introduced to solve DBIS based on deep learning and image processing, which consists of 4 steps:

1) Pre-processing, 2) Impact detection, 3) Edge detection and 4) Evaluation of results. MBID is an easy method to implement, since there are several available libraries and many of them mature to address each of the steps.

The MBID was implemented using the Resnet 50 and Resnet 101 models of Mask R-CNN for impact detection, and an experiment was carried out with 600 shooting silhouettes used in a gun control institution, which contains a total of 2401 images of bullet impacts. The testing results with 249 images of bullet impacts which were not used in training and validation show that the Resnet 50 model obtains better results than Resnet 101, achieving 97.6 %, 99.5 % and 97.9 %, for accuracy, precision and recall, respectively. In addition, deep learning techniques such as YOLOv3, MnasNet, Faster R-CNN, and other techniques such as Support Vector Machine, Circlet Transform, Randomized Hough Transform, RANSAC, Circular Hough Transform and Randomized Circle Detection, achieving lower results than the proposal. These results are explained because the deep learning techniques need a dataset with a larger number of images to improve accuracy results, whereas the other techniques need a particular thresholding setting for each image tested, also, the bullet impacts do not always have a defined circular shape and their edges, in general, they have openings, grooves and slits, carrying out a difficult detection (false negatives), and the silhouettes present objects with circular shapes, confusing these as a bullet impact (false positives). Likewise, the results of edge detection and bullet impact counting show 100 % effectiveness.

REFERENCES

[1] G. Spain. (2018). *Pruebas Para Las Licencias de Armas*. Accessed: Feb. 7, 2020. [Online]. Available: http://www.guardiacivil.es/es/servicios/armasyexplosivo/controldearmas/pruebas/licenciasarmas/index.html

[2] República Portuguesa. (2006). *Regime Jurídico Das Armas E Suas Munições*. Accessed: Feb. 15, 2020. [Online]. Available: https://dre.pt/web/guest/legislacao-consolidada/-/lc/57453375/201307240100/57455586/diploma/indice

[3] ANMaC. *Solicitar La Portacón De Arma De Fuego, Trámites ANMaC*. Accessed: Feb. 15, 2020. [Online]. Available: https://www.argentina.gob.ar/solicitar-la-portacion-de-arma-de-fuego

[4] E. Perú. (2017). *Directiva N°22–2017-SUCAMEC: Directiva Que Regula la Evaluación Teórica, Práctica y de Tiro de Arma de Fuego*. Accessed: Mar. 7, 2020. [Online]. Available: https://busquedas.elperuano.pe/normaslegales/aprueban-la-directiva-n-22-2017-sucamec-directiva-que-regu-resolucion-no-1273-2017-sucamec-1593213-2/

[5] SENIAT. (2014). *Ley para el Desarme y Control de Armas y Municiones*. Accessed: Mar. 7, 2020. [Online]. Available: http://declaraciones.seniat.gob.ve/portal/page/portal/MANEJADOR_CONTENIDO_SENIAT/01NOTICIAS/00IMAGENES/GACETAOFICIAL40190ISLR.pdf

[6] LexiVox. (2014). *Decreto Supremo 2175*. Accessed: Mar. 7, 2020. [Online]. Available: https://www.lexivox.org/norms/BO-DS-N2175.html

[7] T. International. (2019). *Global Corruption Barometer: Latin America And The Caribbean 2019—Citizens' Views and Opinions of Corruption*. Accessed: Mar. 7, 2020. [Online]. Available: https://www.transparency.org/whatwedo/publication/global_corruption_barometer_latin_america_and_the_caribbean_2019

[8] S. Astapov, J. Berdnikova, J. Ehala, J. Kaugerand, and J.-S. Preden, "Gunshot acoustic event identification and shooter localization in a WSN of asynchronous multichannel acoustic ground sensors," *Multidimensional Syst. Signal Process.*, vol. 29, no. 2, pp. 563–595, Apr. 2018.

[9] R. Barauskas and A. Abraitienė, "Computational analysis of impact of a bullet against the multilayer fabrics in LS-DYNA," *Int. J. Impact Eng.*, vol. 34, no. 7, pp. 1286–1305, Jul. 2007.

[10] A. Tabiei and G. Nilakantan, "Ballistic impact of dry woven fabric composites: A review," *Appl. Mech. Rev.*, vol. 61, no. 1, pp. 1–13, Jan. 2008.

[11] C. E. Loeffler, "Detecting gunshots using wearable accelerometers," *PLoS ONE*, vol. 9, no. 9, Sep. 2014, Art. no. e106664.

[12] P. Głomb, M. Romaszewski, M. Cholewa, and K. Domino, "Application of hyperspectral imaging and machine learning methods for the detection of gunshot residue patterns," *Forensic Sci. Int.*, vol. 290, pp. 227–237, Sep. 2018.

[13] X. Wang, H. Wang, S. Niu, and J. Zhang, "Detection and localization of image forgeries using improved mask regional convolutional neural network," *Math. Biosci. Eng.*, vol. 16, no. 5, pp. 4581–4593, 2019.

[14] C. Jiao, K. Su, W. Xie, and Z. Ye, "Burn image segmentation based on mask regions with convolutional neural network deep learning framework: More accurate and more convenient," *Burns Trauma*, vol. 7, pp. 1–14, Feb. 2019.

[15] Z. Yang, Y. Yuan, M. Zhang, X. Zhao, Y. Zhang, and B. Tian, "Safety distance identification for crane drivers based on mask R-CNN," *Sensors*, vol. 19, no. 12, p. 2789, Jun. 2019.

[16] J.-Y. Chiao, K.-Y. Chen, K. Y.-K. Liao, P.-H. Hsieh, G. Zhang, and T.-C. Huang, "Detection and classification the breast tumors using mask R-CNN on sonograms," *Medicine*, vol. 98, no. 19, May 2019, Art. no. e15200.

[17] R. Decker, M. Duca, and S. Spickert-Fulton, "Measurement of bullet impact conditions using automated in-flight photography system," *Defence Technol.*, vol. 13, no. 4, pp. 288–294, Aug. 2017.

[18] Z. Gao, X. Liu, S. Qi, W. Wu, W. K. Hau, and H. Zhang, "Automatic segmentation of coronary tree in CT angiography images," *Int. J. Adapt. Control Signal Process.*, vol. 33, no. 8, pp. 1239–1247, Aug. 2019.

[19] Z. Ejaz, C. A. Ahmad, and H. Justin, "Automatic Red Blood Cell Detection and Counting System Using Hough Transform," *Indo Amer. J. Pharmaceutical Sci.*, vol. 5, no. 8, pp. 7913–7920, Jul. 2018.

[20] Y. Meng, Z. Zhang, H. Yin, and T. Ma, "Automatic detection of particle size distribution by image analysis based on local adaptive canny edge detection and modified circular Hough transform," *Micron*, vol. 106, pp. 34–41, Mar. 2018.

[21] K. Adem, "Exudate detection for diabetic retinopathy with circular Hough transformation and convolutional neural networks," *Expert Syst. Appl.*, vol. 114, pp. 289–295, Dec. 2018.

[22] R. Pérez-Zavala, M. Torres-Torriti, F. A. Cheein, and G. Troni, "A pattern recognition strategy for visual grape bunch detection in vineyards," *Comput. Electron. Agricult.*, vol. 151, pp. 136–149, Aug. 2018.

[23] O. Sarrafzadeh, H. Rabbani, A. M. Dehnavi, and A. Talebi, "Circlet transform in cell and tissue microscopy," *Opt. Laser Technol.*, vol. 124, Apr. 2020, Art. no. 106000.

[24] M. Kumar and N. B. Puhan, "RANSAC lens boundary feature based kernel SVM for transparent contact lens detection," *IET Biometrics*, vol. 8, no. 3, pp. 177–184, May 2019.

[25] B. Li, J. Yang, X. Zeng, H. Yue, and W. Xiang, "Automatic gauge detection via geometric fitting for safety inspection," *IEEE Access*, vol. 7, pp. 87042–87048, 2019.

[26] A. Soltani, T. Battikh, I. Jabri, and N. Lakhoua, "A new expert system based on fuzzy logic and image processing algorithms for early glaucoma diagnosis," *Biomed. Signal Process. Control*, vol. 40, pp. 366–377, Feb. 2018.

[27] T. De Marco, D. Cazzato, M. Leo, and C. Distante, "Randomized circle detection with isophotes curvature analysis," *Pattern Recognit.*, vol. 48, no. 2, pp. 411–421, Feb. 2015.

[28] F. Liu, Y. Yang, Y. Zeng, and Z. Liu, "Bending diagnosis of rice seedling lines and guidance line extraction of automatic weeding equipment in paddy field," *Mech. Syst. Signal Process.*, vol. 142, Aug. 2020, Art. no. 106791.

[29] M. Zalpour, G. Akbarizadeh, and N. Alaei-Sheini, "A new approach for oil tank detection using deep learning features with control false alarm rate in high-resolution satellite imagery," *Int. J. Remote Sens.*, vol. 41, no. 6, pp. 2239–2262, Mar. 2020.

[30] J. Liu and R. Zhang, "Vehicle detection and ranging using two different focal length cameras," *J. Sensors*, vol. 2020, pp. 1–14, Mar. 2020.

[31] J. Jing, D. Zhuo, H. Zhang, Y. Liang, and M. Zheng, "Fabric defect detection using the improved YOLOv3 model," *J. Engineered Fibers Fabrics*, vol. 15, Jan. 2020, Art. no. 155892502090826.

[32] Z. Zhao and A. Kumar, "A deep learning based unified framework to detect, segment and recognize irises using spatially corresponding features," *Pattern Recognit.*, vol. 93, pp. 546–557, Sep. 2019.

[33] M. Tan, B. Chen, R. Pang, V. Vasudevan, M. Sandler, A. Howard, and Q. V. Le, "MnasNet: Platform-aware neural architecture search for mobile," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2815–2823.
[34] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.
[35] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 386–397, Feb. 2020.
[36] Y. Yu, K. Zhang, L. Yang, and D. Zhang, "Fruit detection for strawberry harvesting robot in non-structural environment based on mask-RCNN," *Comput. Electron. Agricult.*, vol. 163, Aug. 2019, Art. no. 104846.
[37] J. Lin, T. Guo, Q. F. Yan, and W. Wang, "Image segmentation by improved minimum spanning tree with fractional differential and canny detector," *J. Algorithms Comput. Technol.*, vol. 13, Sep. 2019, Art. no. 174830261987359.
[38] M. Baştan, S. S. Bukhari, and T. Breuel, "Active canny: Edge detection and recovery with open active contour models," *IET Image Process.*, vol. 11, no. 12, pp. 1325–1332, Dec. 2017.
[39] B. Yuan and M. Liu, "Power histogram for circle detection on images," *Pattern Recognit.*, vol. 48, no. 10, pp. 3268–3280, Oct. 2015.
[40] J. Gu, Y. Pan, and H. Wang, "Research on the improvement of image edge detection algorithm based on artificial neural network," *Optik*, vol. 126, no. 21, pp. 2974–2978, Nov. 2015.
[41] S. S. Alahmari, D. Goldgof, L. Hall, H. A. Phoulady, R. H. Patel, and P. R. Mouton, "Automated cell counts on tissue sections by deep learning and unbiased stereology," *J. Chem. Neuroanatomy*, vol. 96, pp. 94–101, Mar. 2019.
[42] M. Woźniak and D. Połap, "Object detection and recognition via clustered features," *Neurocomputing*, vol. 320, pp. 76–84, Dec. 2018.
[43] J. Kozakiewicz, "Image analysis algorithm for detection and measurement of martian sand grains," *Earth Sci. Informat.*, vol. 11, no. 2, pp. 257–272, Jan. 2018.
[44] C. Zhang and K. Wang, "A switching median—Mean filter for removal of high-density impulse noise from digital images," *Optik*, vol. 126, nos. 9–10, pp. 956–961, May 2015.
[45] V. R. Vijaykumar, G. Santhana Mari, and D. Ebenezer, "Fast switching based median-mean filter for high density salt and pepper noise removal," *AEU Int. J. Electron. Commun.*, vol. 68, no. 12, pp. 1145–1155, Dec. 2014.
[46] X. Wang, Q. Zhao, and J. Tan, "Improved morphological band-pass filtering algorithm and its application in circle detection," *Math. Problems Eng.*, vol. 2018, pp. 1–9, May 2018.
[47] M. Barros Neiva, A. Vacavant, and O. M. Bruno, "Improving texture extraction and classification using smoothed morphological operators," *Digit. Signal Process.*, vol. 83, pp. 24–34, Dec. 2018.
[48] T. Y. Goh, S. N. Basah, H. Yazid, M. J. Aziz Safar, and F. S. Ahmad Saad, "Performance analysis of image thresholding: Otsu technique," *Measurement*, vol. 114, pp. 298–307, Jan. 2018.
[49] F. Yan, H. Zhang, and C. R. Kube, "A multistage adaptive thresholding method," *Pattern Recognit. Lett.*, vol. 26, no. 8, pp. 1183–1191, Jun. 2005.
[50] J. Varghese, "Adaptive threshold based frequency domain filter for periodic noise reduction," *AEU Int. J. Electron. Commun.*, vol. 70, no. 12, pp. 1692–1701, Dec. 2016.
[51] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.
[52] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981.
[53] P. Kultanen, L. Xu, and E. Oja, "Randomized Hough transform (RHT)," in *Proc. 10th Int. Conf. Pattern Recognit.*, Jun. 1990, pp. 631–635.
[54] T.-C. Chen and K.-L. Chung, "An efficient randomized algorithm for detecting circles," *Comput. Vis. Image Understand.*, vol. 83, no. 2, pp. 172–191, Aug. 2001.
[55] O. Sarrafzadeh, A. M. Dehnavi, H. Rabbani, N. Ghane, and A. Talebi, "Circlet based framework for red blood cells segmentation and counting," in *Proc. IEEE Workshop Signal Process. Syst. (SiPS)*, Oct. 2015, pp. 1–6.
[56] D. Koc-San, S. Selim, N. Aslan, and B. T. San, "Automatic citrus tree extraction from UAV images and digital surface models using circular Hough transform," *Comput. Electron. Agricult.*, vol. 150, pp. 289–301, Jul. 2018.
[57] J. Olafenwa and Moses. (2018). *ImageAI*. Accessed: May 21, 2020. [Online]. Available: https://github.com/OlafenwaMoses/ImageAI/
[58] Z. Song, L. Fu, J. Wu, Z. Liu, R. Li, and Y. Cui, "Kiwifruit detection in field images using faster R-CNN with VGG16," *IFAC-PapersOnLine*, vol. 52, no. 30, pp. 76–81, 2019.
[59] F. Zaklouta and B. Stanciulescu, "Real-time traffic sign recognition in three stages," *Robot. Auto. Syst.*, vol. 62, no. 1, pp. 16–24, Jan. 2014.

RICHAR FERNÁNDEZ VÍLCHEZ received the B.S. degree in computer science from Pedro Ruiz Gallo National University, Lambayeque, Peru. He is currently pursuing the M.S. degree in software engineering with the National University of San Marcos, Lima, Peru. His main research interest includes bullet impact detection techniques for the guns control institution of the Peruvian Government.

DAVID MAURICIO received the M.S. degree in applied mathematics and the Ph.D. degree in systems and computing engineering from the Federal University of Rio de Janeiro, Brazil. He was a Professor with the North State University Fluminense, Brazil, from 1994 to 1998. Since 1998, he has been a Professor with the National University of San Marcos. His research interests include mathematical programming, artificial intelligence, software engineering, and entrepreneurship.