IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 11 June 2024, accepted 3 August 2024, date of publication 12 August 2024, date of current version 21 August 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3442728

RESEARCH ARTICLE

Improving Armed People Detection on Video Surveillance Through Heuristics and Machine Learning Models

ALONSO JAVIER AMADO-GARFIAS, SANTIAGO ENRIQUE CONANT-PABLOS, (Member, IEEE), JOSÉ CARLOS ORTIZ-BAYLISS, (Member, IEEE), AND HUGO TERASHIMA-MARÍN, (Senior Member, IEEE)
Tecnologico de Monterrey, School of Engineering and Sciences, Monterrey, Nuevo Leon 64849, Mexico
Corresponding author: Santiago Enrique Conant-Pablos (sconant@tec.mx)

ABSTRACT Much research aims to enhance weapon detection by applying different techniques to object detection models. However, little research focuses on identifying armed people through real-time surveillance cameras. The proposed solution involves the development of algorithms for identifying people carrying handguns (pistols and revolvers). We have chosen the YOLOv4 model to detect people, guns, and faces. Then, we extract information from YOLO related to real-time videos, such as bounding box coordinates, distances, and intersection areas between firearms and the people in each video frame to recognize the armed people. There are some challenges to overcome, for example, occlusion, hidden handguns, and people close to each other. It allows us to develop and compare different types of solutions. We proposed three heuristics and seven machine-learning models. The heuristics are the method of centers, the method of intersections, and the method of distances. Furthermore, the machine learning models are Random Forest Classifier, Multilayer Perceptron, k-Nearest-Neighbors, Support Vector Machine, Logistic Regression, Naive Bayes, and Gradient Boosting Classifier. The Random Forest Classifier presented the best performance reaching an accuracy of 85.44%, a precision of 87.07%, a recall of 88.68%, and an F1-score of 87.87%.

INDEX TERMS Armed people detection, machine learning, heuristics, computer vision.

I. INTRODUCTION

This research aims to optimize video surveillance camera systems to detect armed people. We employ a popular object detection algorithm in computer vision and deep learning to detect people, faces, and handguns in video. The research challenge seeks to identify people with handguns and distinguish their faces. Thus, we propose to automate the detection of armed people to reduce the reaction time to a crime and improve the efficiency of the supervision carried out by security personnel.

Surveillance cameras have become essential for securing places such as homes, businesses, and streets. Most cameras do not have people supervising them and are used to try to recognize those responsible after crime. These types of systems represent a useless method to prevent different crimes. Conversely, some of these systems count on security personnel supervising them, relying on someone efficiently monitoring the system to warn security personnel when necessary, increasing reaction time. People who supervise these systems are exposed to fatigue and different distractions. In this research, we focus on identifying people carrying weapons through the video surveillance camera system, specifically handgun detection, which includes pistols and revolvers.

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

Pistols and revolvers are the firearms that usually are used to execute different crimes. According to the report made by the United Nations Office on Drugs and Crime (UNODC) [1], which includes information from 81 countries, the most seized firearms due to their illicit uses are the following: pistols (39%), shotguns (25%), rifles (18%), revolvers (14%), submachine guns (3%), and machine guns (1%). In addition, UNODC [2] has determined that in the American continent, the homicide rate for men between 18 and 19 years old is 46 out of every 100 thousand. Furthermore, firearms are involved far more often in homicides in the Americas than in other parts of the world. These statistics support the decision that the scope of this research focuses on the use of handguns.

For the model to be valid, it must present a reasonable speed at which the detections can be processed. Therefore, the object detector model used in this research is YOLOv4 [3]. This model applied to the MSCOCO dataset achieved 65 frames per second and an average precision (AP) of 43.5% on a Tesla v100 GPU. The model has been trained using three classes: handguns, people, and faces. We have made up a dataset of 5000 images that collect images from different internet sources. It contains people captured by surveillance cameras, close-up pictures of handguns, and people carrying handguns. Furthermore, some handgun pictures were extracted from the Internet Movie Firearms Database (IMFDB) [4]. These are images of firearms taken from Hollywood movies and video games.

YOLOv4 generates bounding boxes around each of the given classes. In this research, we take advantage of these bounding boxes. We propose three heuristics to determine the people who carry weapons: the deterministic method of centers that measures distances between bounding boxes, the deterministic method of intersections that quantifies intersection areas between bounding boxes, and the deterministic method of distances that detects the location of bounding box centers inside other bounding boxes. We also propose seven machine learning models to detect armed people: a Random Forest Classifier, a Multilayer Perceptron, a k-Nearest-Neighbors model, a Support Vector Machine, a Logistic Regression model, a Naive Bayes model, and a Gradient Boosting Classifier.

There are several challenges to tackle when detecting armed people. Among those challenges, we can highlight the following. Occlusion is a common issue presented in video surveillance, and it happens when two or more objects appear, one in front of the other in the same direction as the camera. Hence, the camera captures partial portions of two or more objects in the same region. Another challenge is presented when several people are close by since it is difficult for the algorithm to identify which one carries an object. Furthermore, there are some situations when the person carrying the handgun hides it around his body or clothes, making it difficult for the algorithm to continue identifying the armed people.

This paper proposes a contribution of different ML models and deterministic methods as a solution to face the challenges described above. Thereby, we present an analysis of the behavior of each of these algorithms, seeking an optimal solution for armed people detection. In addition, aware that these algorithms can be perfected, we make the codes, models, videos, and datasets available at https://github.com/AlonsoJAG/armed_people_detection.

The remainder of this document is organized as follows: Section II describes previous related works on object detection models to recognize firearms. Furthermore, it describes research related to linking weapons with their owner. Section III presents the methodology applied during the research development. It contains a labeling process technique, object detection model, heuristics, and ML models to identify armed individuals. Section IV describes the results, and Section V discusses the results of each method and model. Finally, Section VI brings a general conclusion and future works to take advantage of the different areas of opportunity.

II. RELATED WORKS

This section describes the state of the art into two subsections: firearms detection and armed people detection.

A. FIREARM DETECTION

Because of their ability to make predictions very quickly, the literature contains many examples of works that use one-stage detectors, such as YOLO [3], [5], [6], [7]. However, there is a tradeoff between the speed of the predictions and the accuracy achieved. Some authors prioritize speed over accuracy. Thus, they apply these models to obtain fast results in firearms detection. Accordingly, de Azevedo Kanehisa and de Almeida Neto [8] used the YOLO algorithm to detect firearms reaching 70% of the mean average precision (mAP) with their dataset. Likewise, Duran-Vega et al. [9] employed Temporal Yolov5, an architecture based on Quasi-Recurrent Neural Networks [10]. The temporal information was extracted from the video to improve the results of handgun detection. Additionally, they explored two temporal data augmentation techniques based on Mosaic and Mixup. Furthermore, Veranyurt and Sakar [11] proposed a deep-learning solution to detect and locate hidden pistols through thermal images in real-time video surveillance cameras using their dataset. They developed multiple deep-learning architectures for image classification and segmentation. VGG-19 presented the best performance in concealed gun detection, achieving an F1-score of 84%. Then, applying the YOLOv3 model, they obtained the highest average precision (AP) value of 95%. Hashi et al. [12] developed different deep-learning models to detect firearms. They compared the deep learning object classification models VGG-19, ResNet, and GoogleNet to select the best backbone for building the best object detection model between faster region-based convolutional neural networks (R-CNN) and YOLOv6. The best performance was ResNet50, which achieved an average accuracy of 92%. YOLOv6 achieved

the highest mAP and inference speed compared to the faster R-CNN. Another research that relies on the efficiency of YOLO to detect firearms and bladed weapons is the one developed by Boukabous and Azizi [13]. They compared different object detection techniques, such as YOLOv5, faster R-CNN, and single-shot multi-box detector (SSD). YOLOv5 achieved the optimal balance between mAP and inference speed for real-time prediction.

Conversely, other researchers have applied a two-stage detector prioritizing accuracy over speed, such as faster R-CNN [14], to obtain better accuracy in detecting firearms. Also, Fernandez-Carroble et al. [15] applied Faster R-CNN with two types of Convolutional Neural Networks (CNN), GoogleNet [16], and SqueezeNet [17]. These were pre-trained on the ImageNet dataset and fine-tuned on a custom dataset of 3000 images. SqueezeNet reached better performance, achieving an 85.4% AP50. Likewise, Verma and Dhillon [18] presented an automatic gun detection system from a cluttered scene using Faster R-CNN with VGG-16 backbone applying the IMFDB dataset [4]. Meanwhile, Olmos et al. [19] aimed to minimize the false positives by building a training dataset according to the results of a CNN classifier. Consequently, it asses the best classification model employing a sliding window and region proposal approach.

Another technique to detect firearms is to apply only CNNs. The research conducted by Egiazarov et al. [20] used a group of semantic CNNs to detect a weapon. These networks broke down the problem of detecting and locating a gun into a set of smaller problems related to the individual parts of a weapon. The authors argued that it offers a straightforward way to address situations where firearms are partially concealed, lack certain distinctive features, or are modified. Moreover, it used simpler neural networks dedicated to specific tasks requiring fewer computational resources and can be trained in parallel. Furthermore, Berardini et al. [21] proposed two CNNs working together, the first for people detection, which guides a second for handguns and knife detection. The proposed solution was deployed on an NVIDIA Jetson Nano edge device connected to an IP camera. The results based on COCO average precision were 79.30% and 5.10 frames per second.

Thereby, Lim et al. [22] presented an improved deep multi-level feature pyramid network that addresses the difficulty of inferring firearms. They trained with a dataset in a multi-level multi-scale object detector (M2Det). Experiments with their video surveillance dataset showed that the proposed model achieved an accuracy of 87.42%.

Besides, Grega et al. [23] presented a novel method to detect firearms or knives in video surveillance. Their work used a background subtraction algorithm to analyze the footage frame by frame. It recognizes image differences between consecutive frames. As image differences leave multiple artifacts due to image flickering and changes in illumination, they applied erosion and dilation to support removing these artifacts and focus further steps of the algorithm on the foreground part of the image.

Some recent studies have used the pose estimation technique to forecast the position and direction of people, generating artificial skeletons on them for various purposes. This technique has also been used for weapons detection. It seeks to reinforce the detection of weapons generated by the object detection models. An example of this technique is the one used by Salido et al. [24], who proposed reducing the number of non-detections of weapons (false negatives) without increasing the number of false positives, incorporating information associated with the pose of the people who carry the weapon. Their research aimed to avoid detection errors due to the small size of the guns' images, partial occlusions, and the poor quality of the pictures. Thus, they applied the object detector to the original images in search of weapons and added the poses of the people holding the guns to these same images to compensate for the problems capturing the gun images. The authors used three object detection models: Faster R-CNN, RetinaNet, and YOLOv3. Using their dataset (1220 images), he achieved with YOLOv3 an improvement in average accuracy from 88.49% (without posing) to 90.09% (with posing).

Meanwhile, Velasco-Mata et al. [25] proposed more in-detail research into using pose estimation by armed people. They proposed combining the object detector with the individual's pose information to improve handgun detection. It presented that the combinational architecture takes two inputs: the result of the object detection network and the pose information. Hence, regions detected as handguns by mistake can be removed from the final result because those regions do not match with any human on the scene. The results improved over a handgun detector by leveraging the human pose. It reached a maximum improvement of 17.5% in AP of the proposed combinational model over the baseline handgun detector (YOLOv3). Similar research was proposed by Ruiz-Santaquiteri et al. [26], who combined the weapon's appearance and the human pose's information in a single architecture. First, key pose points are estimated to extract regions of the hand and generate binary pose images, which are the inputs to the model. Each input is then processed on different subnets and combined to produce the gun bounding box. Results show that the proposed combined model improves handgun detection from 4.23 to 18.9 in AP. Thereby, Ruiz-Santaquiteria et al. [27] proposed an automatic handgun detection based on a combination architecture that harnesses body pose estimation and gun appearance features. The architecture contained CNN and transformers.

Likewise, Chatterjee and Chatterjee [28] aimed their research at analyzing hand posture patterns for recognizing a person holding a weapon. They proposed different ML models to classify guns and non-guns. Furthermore, they introduced a metric learning approach instead of classification, called the fuzzy discernible feature selection (FDFS) technique. Its best result was obtained by a Deep Neural Network together with FDDS, which achieved a test accuracy of 93%.

B. DETECTION OF ARMED PEOPLE

There is little research on identifying armed people on video surveillance. Most studies aim to detect guns rather than the people carrying them. However, the work by Agarwal et al. [29] exhibits some similarities with ours. They proposed a model to detect explosives hidden inside abandoned objects. Hence, they classified abandoned and unattended objects separately and backtracked to identify the owner and find the last known location of the owner in a social environment using visual surveillance in real-time. Their work used short- and long-term models to determine which objects have been abandoned. It applied MobileNet-Single Shot Multi-Box Detector (SSD) and Histogram of Oriented Gradients (HOG) with Support Vector Machines (SVM) to detect the people. Besides, it applied a graphical user interface (GUI) to determine the region of interest (RoI) and exclude undesirable regions. The authors used Scale-invariant feature transform (SIFT) [30] to extract the features of all the humans in a frame. To track the person making a match of the stored features in consecutive frames, it used Fast Library for Approximate Nearest Neighbors (FLANN) [31].

McPartlin describes a similar work, where he sought to identify the owner of abandoned luggage [32]. The European Commission Framework 7 program funded a research & development project called "Surveillance of Unattended Baggage and the Identification and Tracking of the Owner" (SUBITO), which aimed to detect abandoned baggage, and identify the owner, determine the location or path they followed. This project was divided into two areas: image analysis and threat assessment. Image analysis included three algorithms: object detection, object tracking, and object classification. Threat assessment was related to two algorithms: observation and analysis and threat classification. The general criteria determining belongings were that the bag owner was the closest person in appearance, that the bag was unattended if the owner was not within two meters, and that the bag was abandoned if it was unattended for 30 seconds.

The following research presented a theory similar to the heuristics we set out in this research. Moura et al. [33] proposed to apply the Intersection Over Union (IoU) concept to identify weapon carriers or shoots in a video frame using YOLOv5. However, the proposal varied the original theory of the IoU. This technique was used for two different classes: person and weapon. It infers that the person with the firearm in the same video frame with the highest IoU is the armed person.

III. METHODOLOGY

In this section, we describe methods, techniques, and experiments that we have designed to obtain the primary goal of identifying the carriers of handguns. We have divided the methodology into three stages. The first stage is the labeling process technique of our image dataset, which is applied to train the object detection model. The second stage shows the characteristics of training the YOLOv4 object detection model. The third stage consists of detecting armed people, which includes the different heuristics and ML models proposed to identify armed people.

A. LABELING PROCESS TECHNIQUE

The labeling process is essential to get optimal results with each of the armed people detection methods. We use the LabelImg program [34] to label manually the classes required to identify armed people: people, faces, and handguns. Furthermore, we propose to use the set theory, a novel way of tagging the images of our dataset, to solve the issue of recognizing the person and the face of the person carrying a handgun. The images in the dataset have been labeled considering the people bounding box as a universal set containing two items: handguns and faces. Hence, guns have been considered within the people's bounding box and, in the same way, their faces. Figure 1 shows this technique.

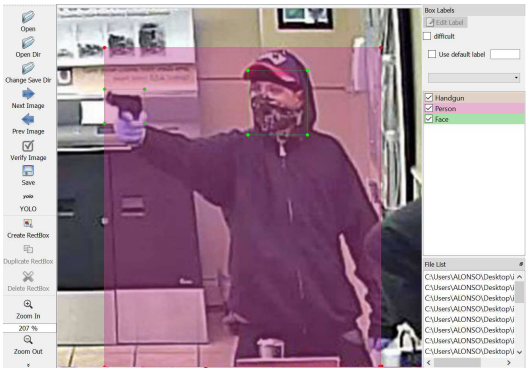

FIGURE 1. An example of how the images in the dataset have been labeled. This technique considers the people bounding box as a universal set, which always contains the gun and face bounding boxes inside.

We propose three heuristics to identify armed people using the bounding boxes' centers, distances, and intersection areas. In addition, we present seven ML models with the same purpose. However, applying these solutions requires an optimal labeling process, which means that the bounding boxes of faces and guns are always inside the bounding box of people to correctly extract information related to distances, intersection areas, and centers. Therefore, we applied the labeling technique to our dataset, which includes 5,000 images. These images have been downloaded from different public sources and consist of some simulated images that show armed people, besides authentic images of armed people at a crime scene. Likewise, the dataset includes people images and handgun images. Moreover, some revolvers and pistols in the dataset were downloaded from the Internet Movie Firearms Database (IMFDB) [4]. The photos are in color and have different sizes and resolutions.

B. OBJECT DETECTION MODEL

After executing the labeling process, we used our dataset to train the YOLOv4 object detector. We trained it from scratch to recognize faces, handguns, and people in the video.

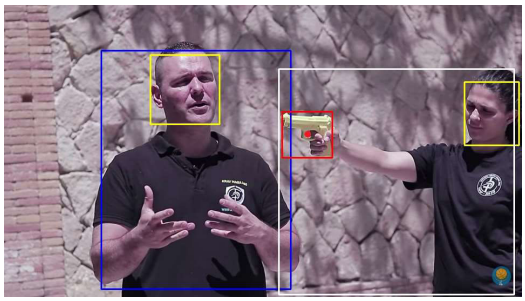

FIGURE 2. An example of how DMC identifies armed people.

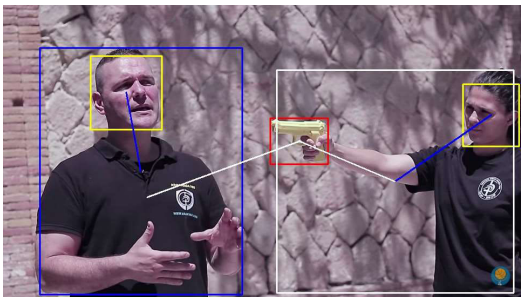

FIGURE 3. An example of how DMD identifies armed people.

FIGURE 4. An example of how DMI identifies armed people. The large intersection area between the handgun's bounding box and the woman's bounding box indicates that she is armed.

We randomly divided our dataset into 4,000 images for training and 1,000 for testing. Afterward, we downloaded YOLOv4 from Alexey Bochkovskiy's GitHub [35] (YOLOv4 creator's GitHub repository). Furthermore, we trained YOLOv4 for 6,000 iterations. This experiment used a Dell Xps 8930-2018 with Nvidia GeForce GTX 1050 TI and a 3+1 voltage regulator module (VRM). In addition, it has 32 GB of RAM and an 8th generation Intel (R) Core I7-8700-3.2 GHZ with six cores.

We chose this object detection model because it delivers optimal performance between speed and precision. It has the following architecture: as a backbone CSPDarknet53 [36], an additional module called Spatial Pyramid Pooling (SPP) [37]. Furthermore, it applies a path-aggregation Nek (PANet) [38], and as head, YOLOv3 [7] (anchor-based). Besides, this model applied Bag-of-Freebies and Bag-of-Specials methods, which enhanced the model's performance during the detector training. This object detector model reached 65 FPS and 43.5% AP (65.7% AP50) on a Tesla v100 using the MSCOCO dataset.

C. ARMED PEOPLE DETECTION

We propose three heuristics and seven ML models to identify armed individuals. The heuristics are the Deterministic Method of Centers (DMC), Deterministic Method of Distances (DMD), and Deterministic Method of Intersections (DMI). These heuristics are described in the following lines: DMC identifies the armed people using a handgun's bounding box center. When a handgun's bounding box center is within the person's bounding box, it means the person is the handgun owner. Figure 2 shows an example of how DMC works. In this example, we detect the presence of a gun with YOLOv4 and calculate its bounding box. We also detect two subjects (a man and a woman) and calculate their corresponding bounding boxes. Since the center of a handgun's bounding box is inside the woman's bounding box, the system determines that the woman has a handgun. Then, DMC concludes that the woman is armed while the man is not.

Although DMC works well in many cases, it exhibits a limited performance when the gun's bounding box lies within the bounding boxes of two or more people. As a way to address such a limitation, we proposed DMD. It relies on the Euclidean distance to identify armed people. The method calculates the Euclidean distances between the centers of the subjects' bounding boxes and the center of a handgun's bounding box. The shortest distance determines positive belonging, as shown in Figure 3, where the distance between the centers of a handgun's bounding box and the centers of the man and woman are 215.75 and 187.14 pixels. With DMD, this frame is interpreted as in the previous case: the woman is armed while the man is unarmed.

The third heuristic, DMI, identifies armed people by measuring the intersection area between the bounding box of the identified subjects and a handgun's bounding box. The largest intersection area defines the armed person. Figures 4 and 5 show how DMI works. Figure 4 shows that the woman is armed because her bounding box presents the highest intersection area related to the handgun's bounding box (7,945.95 pixels^2). Conversely, Figure 5 shows a minimal intersection area between the man's bounding box and the handgun's bounding box (1,431.95 pixels^2), meaning that the man is unarmed.

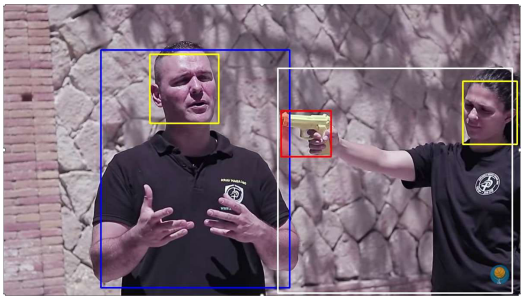

FIGURE 5. An example of how DMI identifies armed people. The small intersection area between the handgun's bounding box and the man's bounding box indicates that he is unarmed.

The seven machine learning (ML) models we build to detect armed people are a Random Forest Classifier (RFC), a Multilayer Perceptron (MLP), a k-Nearest-Neighbors model (KNN), a Support Vector Machine (SVM), a Logistic Regression model (LR), a Naive Bayes model (NB), and a Gradient Boosting Classifier (GBC).

The hyper-parameters considered for the ML models are as follows. RFC employs ten estimators and ten folds for cross-validation. It uses an entropy criterion and applies a maximum depth of two. MLP has four hidden layers, each with 25 neurons. It trained through 500 iterations. It applies the Adam solver to adjust the weights. KNN operates through three neighbors, and the weights of each neighbor are uniform. The algorithm for calculating the nearest neighbors is automatic. SVM works with a linear kernel, and the regularization parameter C is 1.0. LR employs one versus remainder (OvR) scheme, the inverse of the regularization strength C is equal to 100, the solver is the limited memory Broyden Fletcher Goldfarb Shanno (LBFGS), and the maximum number of iterations is 10000. NB requires the Bernoulli Naive Bayes classification algorithm. GBC works with 100 estimators, a maximum depth of six, a learning rate equal to 0.1, a subsample of 1.0, and Friedman-MSE as a criterion. MLP, KNN, and SVM require data standardization by applying the preprocessing module StandarScaler. To obtain better performance metrics for each ML model, they trained and tested with different proportions between their training and test sets. MLP, KNN, and SVM employed 75% (9,489 cases) of the dataset (12,652 cases) for training and 25% (3,163 cases) for testing. LR, NB, and GBC employed 70% (8,856 cases) for training and 30% (3,796 cases) for testing. Finally, RFC employed 80% (10,121 cases) for training and 20% (2,531 cases) for testing.

The ML models have been trained with the dataset detailed in Table 1. The dataset has 20 predictors within which the main characteristics of the heuristics have been considered. These are the center of intersection that expresses whether the center of the handgun's bounding box is inside the person's bounding box, the intersection area between the handgun bounding box and the people bounding boxes, the handgun bounding box area, and the distances between the handgun center bounding box and the people center bounding boxes. In this way, the models learn the advantages and disadvantages of each heuristic. Moreover, the models could recognize in which frame it is convenient to prioritize particular predictors, surpassing the efficiency of the heuristics.

The ML models used in this work operate in two different phases. The first phase is data extraction, while the second is armed people detection. The data extraction data phase begins when the model receives the streaming from the video surveillance camera. The video surveillance streaming enters the object detector model, adding information from the bounding boxes of detected people, faces, and guns to the video. This information is extracted and sent to the next phase. The data extraction phase is illustrated in Figure 6.

Subsequently, the ML models receive twenty measures related to people and handguns. The phase of armed people detection receives the measurements obtained from each video frame. Those measurements go into the ML model and detect the armed people presented in the video. The information related to people and faces is then fed into the face ML model to identify the faces of the armed people. Figure 7 shows the detection phase of armed people.

We trained our ML models on a dataset created from three videos with a total duration of three minutes and 28 seconds. The videos show up to four people with up to five guns. We processed the videos, and they have generated 12,652 records. The records have been compiled frame by frame, generating each record by taking the data related to the first person with the data corresponding to the first gun, then the first person with the second gun, thus combining all the people and weapons present in the frame. Therefore, the number of records per frame depends on the number of people and handguns. The records represent each case in our dataset, where the ground truth indicates whether the person is armed or unarmed. Hence, the number of armed people is 4,228, and 8,424 are unarmed. Please note that the heuristics require no training because they obey specific rules that regulate their behavior. However, the ML models and heuristics require a dataset that allows us to evaluate their performance.

For this reason, we used a 37-second video to test their effectiveness. The video features two people with a pistol. The handgun changes carrier during the video. In some moments of the video, both people fight over handgun possession. We previously processed the video, extracting information from each frame to create a dataset. The video resolution is 1920 x 1080 pixels, and the split video resulted in 1,135 frames. The test video was processed using the same technique as the training videos. It generated 639 records. It associates the information corresponding to each person with each handgun presented in the video frame. Each record represents information about a person with one of the weapons. The records show 380 armed people and 259 unarmed people. Table 2 illustrates the general details of the datasets.

IV. EXPERIMENTS AND RESULTS

This section shows the results of the object detection model used to identify the following classes: people, faces, and handguns, which are mandatory for armed people detection. Likewise, it describes the behavior of the results of heuristics and ML models for armed people detection.

A. RESULTS FOR OBJECT DETECTION

The implementation of YOLOv4 has followed the instructions given by its creator, Bochkovskiy, in his GitHub repository. Figure 8 shows the training results of the object detection model. The x-axis represents the number of iterations through the training. Likewise, the y-axis represents the loss value presented in each iteration. The Mean Average Precision (mAP) value is represented through

FIGURE 6. The first phase in our ML models corresponds to data extraction.

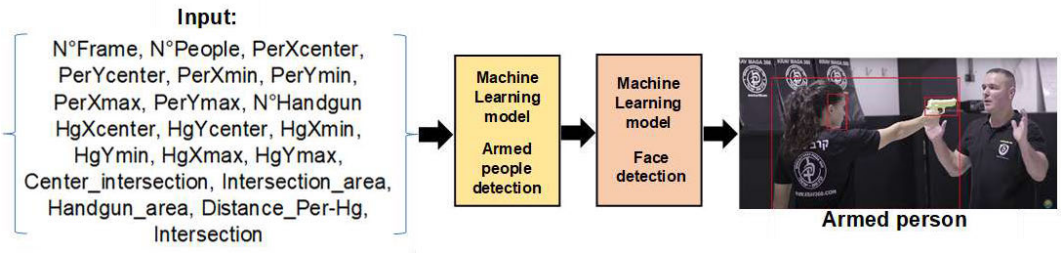

FIGURE 7. The second phase in our ML models corresponds to armed people detection.

TABLE 1. Dataset predictors considered for this work.

N°	Predictor	Description	Units
1	N°Frame	Video frame number	Units
2	N°People	Amount of people	Units
3	PerXcenter	Central coordinate of the X axis of the person.	Pixels
4	PerYcenter	Central coordinate of the Y axis of the person.	Pixels
5	PerXmin	Minimum coordinate of the X axis of the person.	Pixels
6	PerYmin	Minimum coordinate of the Y axis of the person.	Pixels
7	PerXmax	Maximum coordinate of the X axis of the person.	Pixels
8	PerYmax	Maximum coordinate of the Y axis of the person.	Pixels
9	N° Handgun	Number of handguns	Units
10	HgXcenter	Central coordinate of the X axis of the handgun.	Pixels
11	HgYcenter	Central coordinate of the Y axis of the handgun.	Pixels
12	HgXmin	Minimum coordinate of the X axis of the handgun.	Pixels
13	HgYmin	Minimum coordinate of the Y axis of the handgun.	Pixels
14	HgXmax	Maximum coordinate of the X axis of the handgun.	Pixels
15	HgYmax	Maximum coordinate of the Y axis of the handgun.	Pixels
16	Center intersection	Predictor that expresses whether the center of the handgun's bounding box is inside the person's bounding box.	Binary
17	Intersection area	Intersection area between the handguns' bounding boxes and the people bounding box.	$Pixels^2$
18	Handgun area	Handgun bounding box area.	$Pixels^2$
19	Distance Per-Hg	Distances between the handgun center bounding boxes and the people center bounding boxes.	Pixels
20	Intersection	Position of the handgun relatives with the people.	Categorical

the red line. We trained the YOLOv4 model for 6,000 iterations, reaching an mAP of 89%, reducing the loss by 2.1856 with a growing tendency to improve. This trend suggests that increasing the number of images in our dataset

TABLE 2. Characteristics of the datasets considered for this work.

Dataset ID	Purpose	Length	Records	Source
1	Training	00h 01min 01s	12,652	DeportesUncomo [39] (00h 00min 36s: 00h 01min 37s)
2		00h 01min 10s		Own development
3		00h 01min 17s		Own development
4	Testing	00h 00min 37s	639	DeportesUncomo [39] (00h 03min 59s: 00h 04min 36s)

and the number of iterations makes it possible to obtain better results.

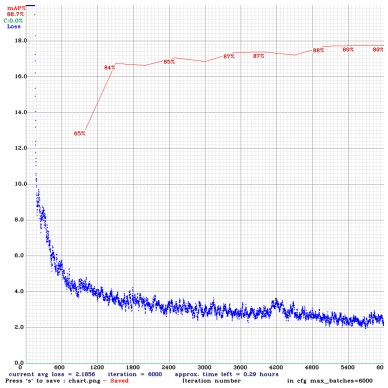

FIGURE 8. YOLOv4 Training results.

In general, YOLOv4 provided competent results in assessments performed with the test set, as illustrated in Table 3. However, some elements can be improved. One such element is the people class's average precision (AP), which presented the lowest score of the three categories. A potential explanation for this performance is that people can take different positions, unlike handgun and face classes. Also, based on the camera locations, the human bodies do not always appear complete. Therefore, it hinders the network learning process. This difficulty is overcome by increasing the number of images of people in different positions, both with full and partial bodies.

Moreover, we detected 1,988 true positives (TP), 526 false positives (FP), and 483 false negatives (FN). True negative (TN) is not a metric applicable to measuring the performance of an object detector. It is because the TN should measure all parts of the image where the model correctly claimed not to detect any of the classes (faces, people, and weapons). It is not measurable since it cannot be verified with the labeled classes in the dataset images (Ground Truth). These results originate from the test set and represent quite competent results. Furthermore, the precision, recall, and F1-score are 79%, 80%, and 80%, respectively, which means that precision and recall metrics maintain equilibrium and that the model works highly efficiently. The average intersection over union (IoU) equals 65.5%, representing a high percentage of coincidence between the model predictions and the Ground Truth. Finally, the mAP is close to 89%, which is a high score for model detection. Hence, these results are adequate for the next phase of this research related to armed people identification.

TABLE 3. YOLOv4 training metrics.

Classes	AP	TP	FP	FN
Handgun	0.9071	697	161	-
Person	0.8708	643	243	-
Face	0.8834	648	122	-
Total	-	1988	526	483

B. RESULTS OF THE HEURISTICS AND ML MODELS

Table 4 shows the results of applying the heuristics to the training videos. These methods do not require any training. However, this experiment is relevant for comparison with the training results of ML models. The true positives are the number of people correctly identified as armed by the different heuristics. Thus, true negatives are people correctly identified as unarmed. In addition, false positives describe the number of people misidentified as armed, while false negatives represent cases where the methods incorrectly recognized an unarmed person. DMI, DMC, and DMD achieved accuracies of 80.87%, 78.61%, and 72.91%, respectively.

TABLE 4. Performance metrics obtained by the heuristics on the training videos.

Methods	DMC	DMI	DMD
TP	3184	3127	2566
TN	6762	7105	6659
FP	1662	1319	1765
FN	1044	1101	1662
Accuracy	78.61%	80.87%	72.91%
Precision	65.7%	70.33%	59.24%
Recall	75.3%	73.95%	60.69%
F1-Score	70.17%	72.1%	59.96%

Like the training video, the test video has been divided into frames to analyze the results of each method. The split video resulted in 1,135 frames. However, only 376 are valid because the rest are qualified as occlusion or no detection. The occlusion occurs when the handgun does not appear on the frame because an object blocks it from the camera's view. The armed people detection models only work when YOLOv4 detects the presence of a handgun in the frame. Each method is evaluated according to the number of people correctly classified in each frame.

Table 5 presents the results of the heuristics applied to the test video. DMI presents the best performance, reaching an accuracy of 81.53%. This model correctly identified 335 armed and 186 unarmed people. Besides, it made 45 errors detecting armed and 73 errors identifying unarmed people. Regarding its performance, the following method is the DMC, which reached an accuracy of 76.05%. DMC identified correctly 337 armed and 149 unarmed people. DMC also produced 110 errors identifying unarmed and 43 errors identifying armed people. DMD presents the worst performance, reaching an accuracy of 51.64% and correctly identifying 225 armed and 105 unarmed people. Moreover, it made 154 errors identifying unarmed people and 155 errors identifying armed people. The results hold the same performance position as the experiment with the training videos.

TABLE 5. Performance metrics obtained by the heuristics on the test video.

Methods	DMC	DMI	DMD
TP	337	335	225
TN	149	186	105
FP	110	73	154
FN	43	45	155
Accuracy	76.05%	81.53%	51.64%
Precision	75.39%	82.11%	59.37%
Recall	88.68%	88.16%	59.21%
F1-Score	81.49%	85.02%	59.28%

Table 6 depicts the main metrics that evaluated the training of each ML model. The model with the best accuracy is the GBC with 99.31%. The next model with the best performance is the MLP, which achieved an accuracy of 99.02%, then KNN reached 98.89%, LR got 92.78%, SVM delivered 91.65%, NB achieved 90.83%, and RFC reached 90.79%. Regardless of the differences, we can conclude that all seven models performed competently.

Table 7 shows the results of applying the seven ML models to the test video. The RFC presents the best performance, reaching an accuracy of 85.44%. The next model with the best performance is the MLP, which delivered an accuracy of 79.18%. KNN achieved 76.83%, SVM presented an accuracy of 76.05%, LR got 75.11%, NB reached 72.77%, and GBC presented the worst performance with an accuracy of 71.67%.

Starting from the hypothesis that each heuristic and ML model has certain advantages and disadvantages for predicting cases of armed people with particular characteristics, we evaluated their performance in different ranges of intersection areas (Pixels^2) of the people present in each frame on the test video. The results of this experiment are presented in Table 8 and illustrated in Figure 10. They allow us to identify which of the proposed techniques performs best at different intervals of intersection areas between people. The first interval shows us 338 cases where there were no intersection areas. In this interval, the DMI obtained the best accuracy. In general, the superiority of the machine learning models over heuristics is evident for the rest of the intervals.

To better understand the performance of the ML models, we have obtained their respective Receiver Operating Characteristic (ROC) curves and their Area Under the Curve (AUC) metrics. Table 9 shows each ML model's AUC and average precision score (APS) applied to the test video. These metrics are essential because they give us an idea of the performance of binary classifiers such as those proposed in this research. Figure 9 presents an overview of the ROC curve and the AUC calculated based on the probabilities in the predictions of the ML models. They show that RFC performs best, outperforming the other models with an AUC of 82.8%. The probability of predictions is calculated for each ML model differently, affecting the ROC curve's accuracy. This effect is most evident in the case of MLP and KNN, where the probabilities tend to be 0 or 1.

V. DISCUSSION

We have divided the discussion section into two stages: deterministic methods and ML models for armed people detection. Three approaches perform deterministic methods: DMC, DMD, and DMI. The ML models have seven approaches: RFC, MLP, KNN, SVM, LR, NB, and GBC.

A. DETERMINISTIC METHOD

The results show that detecting armed people in real time through a surveillance camera system using deterministic methods and ML models is possible. Regarding the deterministic methods, the DMI presents the best accuracy, reaching 81.53% as shown in Table 5. It performs better, allowing a considerable approach between people and minimizing overlapping. It does its job correctly until people get close enough, causing the intersection area to be more significant in the person not carrying the handgun, generating a false positive. Also, when the intersection areas are the same for both people, an error occurs when detecting multiple people armed with a single handgun.

However, it fails when people are very close, unlike the other two deterministic methods that present the same problems when people are not so close. It allows the DMI to identify the armed person at relatively short distances among the people in the video. The results of our experiments showed that the DMI did not present any multiple detections of armed people for a single handgun in a frame.

Conversely, the DMD identifies armed people by measuring in pixels the distance between the center of the person's bounding box and the center of the handgun's bounding box through the Euclidean distance formula. The shortest distance determines the armed people. However, it tends to fail when a person extends his arm to aim the handgun. In that case, the distance between the center of his bounding box and the center of the handgun bounding box increases. This effect means that when the armed person approaches and points the handgun at another person, the distance of the gun quickly becomes less between the center of the handgun's bounding box and the center of the person's bounding box who is threatened, so it is considered incorrectly armed.

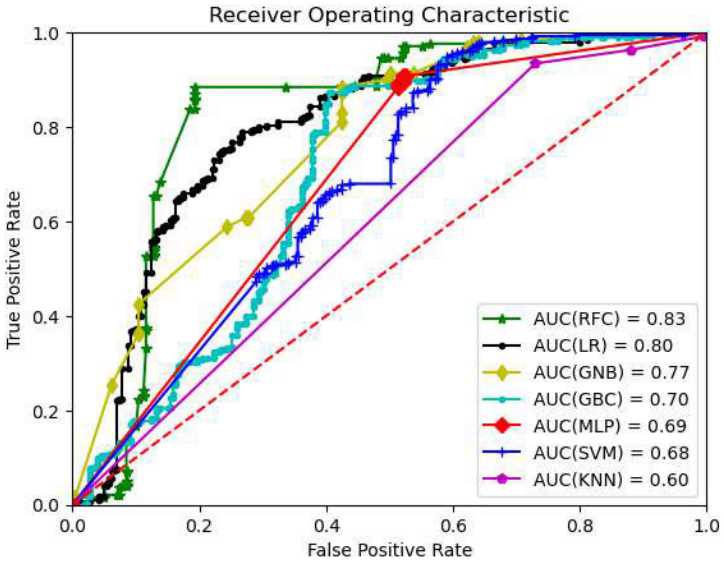

FIGURE 9. ROC curve applied to the results of the ML models on the test video.

TABLE 6. Performance metrics obtained by the ML models on the training videos.

Models	RFC	MLP	KNN	SVM	LR	NB	GBC
TP	800	1036	1039	964	1232	1275	1285
TN	1498	2096	2089	1935	2290	2173	2485
FP	212	10	17	171	210	327	15
FN	21	21	18	93	64	21	11
Accuracy	90.97%	99.02%	98.89%	91.65%	92.78%	90.83%	99.31%
Precision	79.05%	99.04%	98.39%	84.93%	85.43%	79.58%	98.84%
Recall	97.44%	98.01%	98.29%	91.2%	95.06%	98.37%	99.15%
F1-Score	87.28%	98.52%	98.34%	87.95%	89.99%	87.99%	98.99%

TABLE 7. Performance metrics obtained by the ML models on the test video.

Models	RFC	MLP	KNN	SVM	LR	NB	GBC
TP	337	344	344	337	321	367	300
TN	209	162	147	149	159	98	158
FP	50	97	112	43	100	161	101
FN	43	36	36	110	59	13	80
Accuracy	85.44%	79.18%	76.83%	76.05%	75.11%	72.77%	71.67%
Precision	87.08%	78%	75.43%	88.68%	76.24%	69.5%	74.81%
Recall	88.68%	90.52%	90.52%	75.39%	84.47%	96.57%	78.94%
F1-Score	87.87%	83.8%	82.29%	81.49%	80.15%	80.83%	76.82%

TABLE 8. Performance of heuristics and ML models through the areas of intersection between people in the test video.

Intersection area between people (Pixels^2)		Amount of data	Accuracy									
			DMI	DMC	DMD	RFC	MLP	KNN	SVM	LR	NB	GB
0	0	338	93.19%	88.46%	80.17%	88.46%	85.79%	84.02%	88.46%	87.57%	89.94%	89.94%
327.95	4,152.85	31	74.19%	70.96%	61.29%	70.96%	83.87%	80.64%	70.96%	77.41%	74.19%	74.19%
4,161.59	9,294.77	31	96.77%	93.54%	80.64%	93.54%	90.32%	83.87%	93.54%	90.32%	74.19%	93.54%
9,295.99	22,343.49	30	60%	63.33%	33.33%	63.33%	73.33%	56.66%	63.33%	66.66%	46.66%	63.33%
22,458.9	24,789.52	30	93.33%	90%	0%	93.33%	93.33%	90%	90%	90%	50%	73.33%
25,163.05	33,131.79	30	80%	56.66%	6.66%	83.33%	70%	66.66%	56.66%	66.66%	46.66%	56.66%
33,525.48	35,830.92	30	66.66%	50%	6.66%	76.66%	70%	60%	50%	46.66%	46.66%	40%
35,967.8	41,381.68	30	63.33%	46.66%	0%	83.33%	70%	60%	46.66%	50%	46.66%	33.33%
41,568.83	49,216.55	30	56.66%	46.66%	0%	80%	63.33%	63.33%	46.66%	33.33%	46.66%	40%
49,475.26	72,396.13	30	40%	50%	0%	76.66%	53.33%	73.33%	50%	36.66%	50%	63.33%
74,920.21	91,605.88	29	51.72%	51.72%	3.44%	100%	48.27%	51.72%	51.72%	51.72%	51.72%	24.13%

Consequently, this method presents the worst performance, reaching an accuracy of 51.64%, as shown in Table 5. However, the number of errors due to multiple armed person detections for a single handgun presented in a frame was equivalent to one person. It is because, like the DMI, it is unlikely that the distances between the center of the people and the center of the handgun bounding box will be the same.

Meanwhile, the DMC identifies the armed people using the center of the handgun's bounding box. When the center of the handgun's bounding box is within the person's bounding

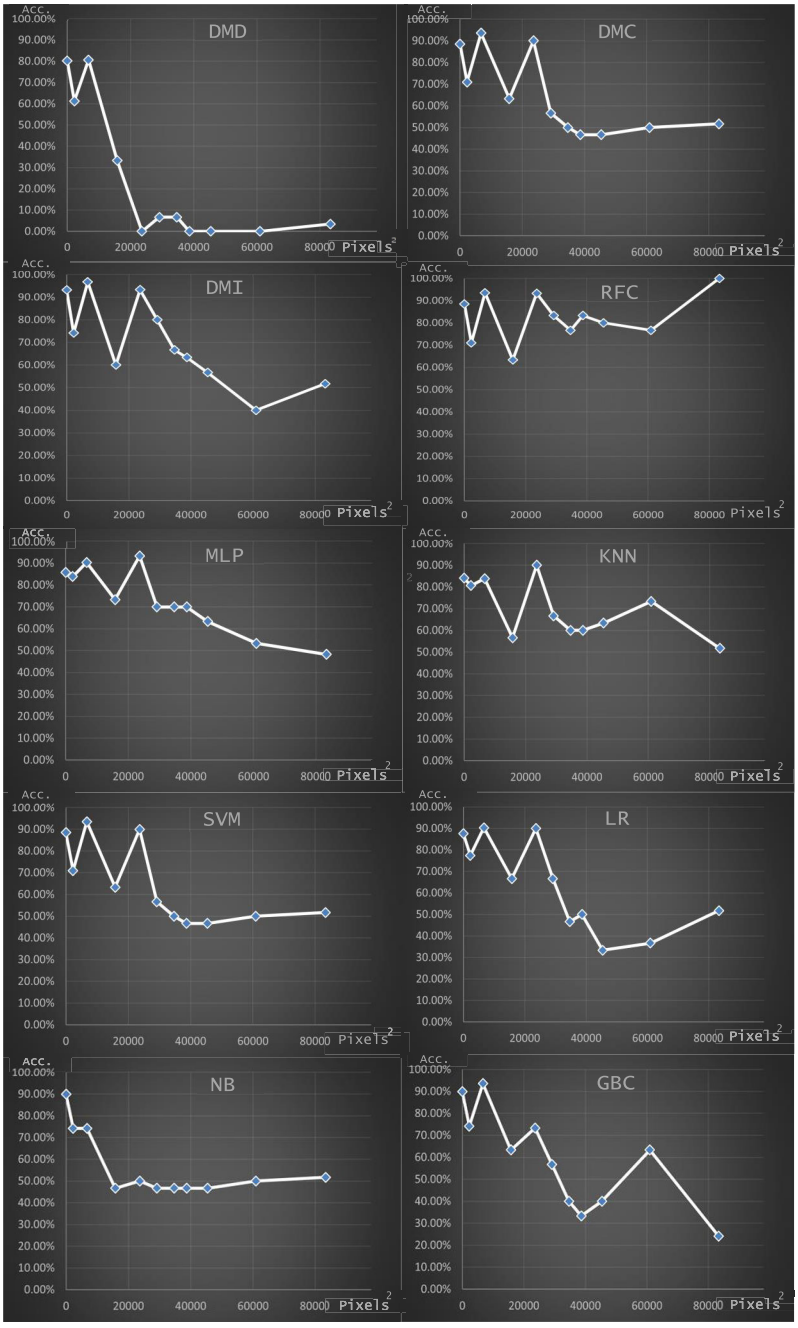

FIGURE 10. Accuracy of heuristics and ML models across intervals of intersection areas between people in the test video.

TABLE 9. Area under the curve accuracy on the test video.

Models	AUC	Avg. Precision
RFC	82.8%	80.1%
LR	80.04%	78.27%
GNB	77.11%	79.14%
GBC	70.23%	70.67%
MLP	69.06%	70.58%
SVM	67.86%	69.98%
KNN	60.08%	65%

box, this person is considered armed. DMC obtained an accuracy of 76.05%. The drawback of this method is that when the people are closed, the center of the gun's bounding box is inside the bounding box of both people. It generated 93 multiple detections of armed people for a single handgun presented in a frame. It represented the highest number of this kind of errors made for deterministic methods.

Figure 11 shows the same frame of the test video processed by each deterministic method. The bounding boxes painted in white on the people are the ones the system recognizes as armed people. The bounding boxes painted in blue are those the system recognizes as unarmed people. Likewise, the system generates a red bounding box on the handguns. This image shows the superiority of the DMI over the other deterministic methods. The DMC, when it presents people

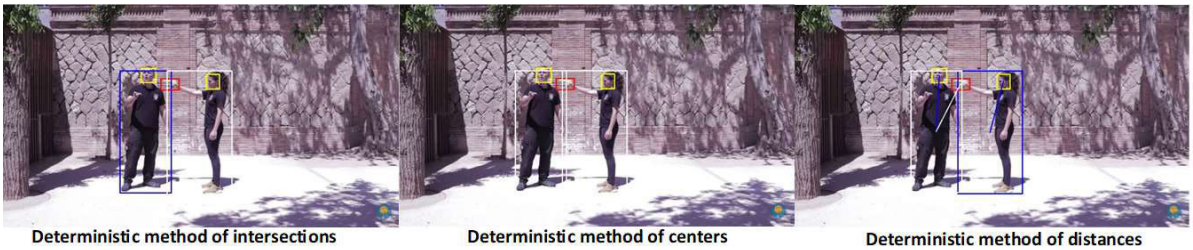

FIGURE 11. Detection nearby armed people.

very close to each other, causes the center of the handgun's bounding box to be inside the bounding boxes of both people, and it erroneously indicates that both are armed. In this case, the DMD wrongly identifies the unarmed person as armed. It is because when the person extends his arm to aim the gun, the distance between the centers of the bounding boxes of the handgun and the unarmed person becomes less than the distance between the gun and the armed person.

B. MACHINE LEARNING MODELS FOR ARMED PEOPLE DETECTION

We trained the ML models on a dataset containing the main features of deterministic methods. The dataset includes the corners and center coordinates of the bounding boxes, areas, and distances between the different classes. These predictors are detailed in Table 1. The ML models aim to identify the convenience of using particular characteristics of the deterministic methods according to the scenario presented in the video frames. In that sense, ML models must be able to take advantage of the best deterministic methods.

Table 6 shows the training results of the ML models using the three training videos in the dataset. All the models presented metrics above 90%. Even GBC and MLP obtained an accuracy of 99.31% and 99.02% correspondingly. It represents optimal performance for predicting armed people. Meanwhile, the deterministic methods were applied to the three training videos, reaching the following results. The DMI obtained an accuracy of 80.87%, the DMC 78.61%, and the DMD 72.91%, whose metrics are detailed in Table 4. The ML models performed better during their training with the same dataset used by deterministic methods. Table 7 illustrates the performance metrics of the ML models in the test video. These obtained optimal results, showing that ML models have learned from deterministic methods. RFC outperformed all three deterministic methods. MLP and KNN surpassed the deterministic method with the second-best performance, DMC. Thereby, SVM achieved the same accuracy as DMC. Furthermore, all ML models successfully passed the DMD.

Although, in the training process, the GBC presented the highest accuracy of the ML models, with the test video, it delivered the lowest. This effect is attributed to the fact that GBC entered a state of overfitting. GBC became too biased in the training data and could not generalize its predictions with new data.

The ML models used in this research were developed on the Jupyter Notebook platform. The dataset used for the MLP, KNN, and SVM training process was standardized before training using the function StandardScaler from the Scikit-Learn library. However, we imported the models into our general system to receive the input data from YOLO's live stream. It implies that the input data must be in the same conditions as the training process. Consequently, it was mandatory to standardize the input data in real-time, so we have applied the mathematical formula used by the StandardScaler function according to $z = (x - u)/s$, where x represents the input data to be standardized, u stands for the mean, and s is the standard deviation of the training samples. The drawback of this solution is that we are using the mean and standard deviation of our training dataset, but it does not correspond to the real-time input data. Therefore, it generates a slight inaccuracy in the prediction of our MLP, KNN, and SVM models.

Additionally, to better understand the behavior of ML models and heuristics, the accuracy is calculated by intervals of intersection area of people in the test video. It is shown in Table 8 and illustrated in Figure 10. In this way, it is evident that when there are no intersection areas between people, the best accuracy is presented by DMI, which obtained 93.19%. In this same interval, the best ML models are NB and GB, with an accuracy of 89.94%. However, by increasing the intersection area between people, the performance of ML models over heuristics improves.

Comparing the heuristics and the ML model with the best accuracy, we will realize that DMI presents a slight improvement in accuracy only in the first three intervals (0 - 9,294.77 Pixels^2). This improvement does not exceed 3.23% for the two intervals that present intersection areas. In the rest of the intervals (9,295.99 - 91,605.88 pixels^2), RFC exceeds the accuracy presented by the DMI by a more significant margin as the intersection area increases. In the last interval (74,920.21 - 91,605.88 pixels^2), RFC obtained an accuracy of 100%, and the DMI obtained 51.72%.

In this experiment, it is worth highlighting that although RFC obtained the best accuracy in general, in the first five intervals (0 - 24,789.52 pixels^2), there were ML models that equaled or surpassed it. It is an interesting fact as it shows that the ML models have advantages and disadvantages in certain intersection areas. No other ML model surpassed its accuracy in the subsequent six intervals (25,163.05 - 91,605.88 pixels^2). It demonstrates its

superiority by identifying armed people when they are very close.

Furthermore, another metric that allows us to appreciate the superiority of RFC is the ROC curve and its respective AUC. RFC obtained an AUC equal to 82.8% and an APS of 80.1%. The next model with the best performance is LR, which obtained an AUC of 80.04% and an APS of 78.27%. Although MLP had the second-best accuracy and KNN the third, their AUCs are not optimal due to their way of calculating the probability in their predictions. They do not have much variance in their probabilities, most of which are 0 or 1. It causes their ROC curve to be made with only a few coordinates, reducing the AUC.

VI. CONCLUSION

The results presented in this document show that it is possible to identify armed people in real time through a surveillance camera system like the one proposed in this work. The concept of heuristics has generated a dataset with 12,652 records, which have been used to train the ML algorithms. These algorithms show a better performance, overcoming the heuristics. Therefore, the ML models have learned the best characteristics and performances of the heuristics through the dataset.

The ML models can work together with YOLO to identify automatically armed people. Consequently, these depend on YOLO performance because if the object detection model fails to recognize the handguns or the people, it would be impossible to determine who is armed. Although we trained YOLO with 5,000 images for this research, 254 frames of the test video presented problems detecting handguns. Therefore, the machine's algorithm could not identify armed people in those frames. Hence, to overcome these issues, we plan to migrate to more current versions of YOLO, such as YOLOv9 [40] for future research. Meanwhile, we plan to increase the number of images in the dataset, specifically with natural scenes of armed people taking them from video surveillance cameras. It is necessary to complement the close-up pictures since, in actual cases, it requires recognizing guns that are further away from the camera and look smaller.

This research aims at identifying armed people through video in real time. However, this task is challenging when people are close to each other, and handguns are occluded. The different methods and models presented tackle these challenges. However, in occlusion, they cannot identify the armed persons because YOLO cannot detect the handguns. Therefore, for future research, we will use recurrent neural networks, such as Long Short Term Memory (LSTM) [41], to predict the coordinates of handguns when they enter an occlusion state. Hence, it will be possible to have a prediction of armed people at all times.

Furthermore, although we have identified that the RFC is the ML model with the best accuracy, other models make better predictions in certain situations delivered in video. Therefore, it is possible to generate an automatic model selector that identifies and applies the best of the seven ML models to a specific condition presented on video. Thus, we would have the seven ML models working together, improving the accuracy of detecting armed people.

REFERENCES

[1] *Global Study on Firearms Trafficking*, United Nations Office Drugs Crime (UNODC), Mar. 2020.
[2] *Global Study on Homicide: Homicide Trends, Patterns and Criminal Justice Response*, United Nations Office Drugs Crime (UNODC), Vienna, Austria, Jul. 2019.
[3] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," 2020, *arXiv:2004.10934*.
[4] *Internet Movie Firearms Database*. Accessed: Jan. 18, 2023. [Online]. Available: https://www.imfdb.org/wiki/Main_Page
[5] 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.
[6] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6517–6525.
[7] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.
[8] R. Kanehisa and A. Neto, "Firearm detection using convolutional neural networks," in *Proc. 11th Int. Conf. Agents Artif. Intell.*, 2019, pp. 707–714.
[9] M. Alberto Duran-Vega, M. Gonzalez-Mendoza, L. Chang, and C. Daniel Suarez-Ramirez, "TYOLOv5: A temporal YOLOv5 detector based on quasi-recurrent neural networks for real-time handgun detection in video," 2021, *arXiv:2111.08867*.
[10] J. Bradbury, S. Merity, C. Xiong, and R. Socher, "Quasi-recurrent neural networks," 2016, *arXiv:1611.01576*.
[11] O. Veranyurt and C. O. Sakar, "Concealed pistol detection from thermal images with deep neural networks," *Multimedia Tools Appl.*, vol. 82, no. 28, pp. 44259–44275, Nov. 2023.
[12] A. O. Hashi, A. A. Abdirahman, M. A. Elmi, and O. E. R. Rodriguez, "Deep learning models for crime intention detection using object detection," *Int. J. Adv. Comput. Sci. Appl.*, vol. 14, no. 4, pp. 1–20, 2023.
[13] M. Boukabous and M. Azizi, "Image and video-based crime prediction using object detection and deep learning," *Bull. Electr. Eng. Informat.*, vol. 12, no. 3, pp. 1630–1638, Jun. 2023.
[14] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, 2015, pp. 1–24.
[15] M. Fernandez-Carrobles, O. Deniz, and F. Maroto, "Gun and knife detection based on faster R-CNN for video surveillance," in *Proc. Iberian Conf. Pattern Recognit. Image Anal.*, 2019, pp. 441–452.
[16] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.
[17] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50× fewer parameters and 0.5MB model size," 2016, *arXiv:1602.07360*.
[18] G. K. Verma and A. Dhillon, "A handheld gun detection using faster R-CNN deep learning," in *Proc. 7th Int. Conf. Comput. Commun. Technol.*, Nov. 2017, pp. 84–88.
[19] R. Olmos, S. Tabik, and F. Herrera, "Automatic handgun detection alarm in videos using deep learning," *Neurocomputing*, vol. 275, pp. 66–72, Jan. 2018.
[20] A. Egiazarov, V. Mavroeidis, F. M. Zennaro, and K. Vishi, "Firearm detection and segmentation using an ensemble of semantic neural networks," in *Proc. Eur. Intell. Secur. Informat. Conf. (EISIC)*, Nov. 2019, pp. 70–77.
[21] D. Berardini, L. Migliorelli, A. Galdelli, E. Frontoni, A. Mancini, and S. Moccia, "A deep-learning framework running on edge devices for handgun and knife detection from indoor video-surveillance cameras," *Multimedia Tools Appl.*, vol. 83, no. 7, pp. 19109–19127, Jul. 2023.
[22] J. Lim, M. I. Al Jobayer, V. M. Baskaran, J. M. Lim, J. See, and K. Wong, "Deep multi-level feature pyramids: Application for non-canonical firearm detection in video surveillance," *Eng. Appl. Artif. Intell.*, vol. 97, Jan. 2021, Art. no. 104094.
[23] M. Grega, A. Matiolański, P. Guzik, and M. Leszczuk, "Automated detection of firearms and knives in a CCTV image," *Sensors*, vol. 16, no. 1, p. 47, Jan. 2016.

[24] J. Salido, V. Lomas, J. Ruiz-Santaquiteria, and O. Deniz, "Automatic handgun detection with deep learning in video surveillance images," *Appl. Sci.*, vol. 11, no. 13, p. 6085, Jun. 2021.
[25] A. Velasco-Mata, J. Ruiz-Santaquiteria, N. Vallez, and O. Deniz, "Using human pose information for handgun detection," *Neural Comput. Appl.*, vol. 33, no. 24, pp. 17273–17286, Dec. 2021.
[26] J. Ruiz-Santaquiteria, A. Velasco-Mata, N. Vallez, G. Bueno, J. A. Álvarez-García, and O. Deniz, "Handgun detection using combined human pose and weapon appearance," *IEEE Access*, vol. 9, pp. 123815–123826, 2021.
[27] J. Ruiz-Santaquiteria, A. Velasco-Mata, N. Vallez, O. Deniz, and G. Bueno, "Improving handgun detection through a combination of visual features and body pose-based data," *Pattern Recognit.*, vol. 136, Apr. 2023, Art. no. 109252.
[28] R. Chatterjee and A. Chatterjee, "Pose4Gun: A pose-based machine learning approach to detect small firearms from visual media," *Multimedia Tools Appl.*, vol. 83, no. 22, pp. 62209–62235, Sep. 2023.
[29] H. Agarwal, G. Singh, and M. A. Siddiqui, "Classification of abandoned and unattended objects, identification of their owner with threat assessment for visual surveillance," in *Proc. 3rd Int. Conf. Comput. Vis. Image Process.*, 2020, pp. 221–232.
[30] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004.
[31] M. Muja and D. G. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration," *VISAPP*, vol. 2, nos. 331–340, p. 2, 2009.
[32] M. McPartlin and D. G. Lowe, "Surveillance of unattended baggage and the identification and tracking of the owner," *Res. Framework Programme*, vol. 2, pp. 1–53, May 2011.
[33] N. S. Moura, J. M. Gondim, D. B. Claro, M. Souza, and R. Figueiredo, "Detection of weapon possession and fire in public safety surveillance cameras," in *Proc. Anais do 18th Encontro Nacional de Inteligência Artificial e Computacional*. Porto Alegre, Brazil: Sociedade Brasileira de Computação, 2021, pp. 290–301.
[34] D. Tzutalin. (2015). *HumanSignal/LabelImg*. Github Repository. Accessed: Nov. 25, 2022. [Online]. Available: https://github.com/heartexlabs/labelImg
[35] A. Bochkovskiy. (2020). *Darknet-YOLO V4, V3 and V2 for Windows and Linux*. Accessed: Dec. 5, 2021. [Online]. Available: https://github.com/AlexeyAB/darknet
[36] C.-Y. Wang, H.-Y. Mark Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, and I.-H. Yeh, "CSPNet: A new backbone that can enhance learning capability of CNN," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2020, pp. 1571–1580.
[37] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1904–1916, Sep. 2015.
[38] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8759–8768.
[39] deportesUncomo. *Cómo Defenderse de Amenaza de Pistola—Técnicas Krav Maga*. YouTube [Video File]. Accessed: Feb. 10, 2023. [Online]. Available: https://www.youtube.com/watch?v=xrXkD2uWw_o
[40] C.-Y. Wang, I.-H. Yeh, and H.-Y. Mark Liao, "YOLOv9: Learning what you want to learn using programmable gradient information," 2024, *arXiv:2402.13616*.
[41] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

ALONSO JAVIER AMADO-GARFIAS was born in Lima, Peru, in 1984. He received the B.S. degree in naval maritime sciences from the Naval School of Peru, in 2006, the bachelor's degree in systems engineering from the Scientific University of the South, Lima, in 2019, and the master's degree in information technology management from Monterrey Institute of Technology and Higher Education, Monterrey Campus, Mexico, in 2015, where he is currently pursuing the Ph.D. degree in computer science.

SANTIAGO ENRIQUE CONANT-PABLOS (Member, IEEE) received the B.Sc. degree in industrial engineering from the Instituto Tecnológico de Sonora and the M.Sc. degree in computer science and the Ph.D. degree in artificial intelligence from Tecnológico de Monterrey, in 2004. He is currently an Associate Research Professor with the School of Engineering and Sciences, Tecnológico de Monterrey; a Researcher with the group with a strategic focus on advanced artificial intelligence; and an Adjoint Member of the SOI-STEM Interdisciplinary Group, Institute for the Future of Education. His research interests include machine learning, computer vision, evolutionary and bio-inspired computation, hyper-heuristics design, and natural language processing. He is a member of Mexican National System of Researchers and Mexican Academy of Computing.

JOSÉ CARLOS ORTIZ-BAYLISS (Member, IEEE) was born in Culiacan, Sinaloa, Mexico, in 1981. He received the first B.Sc. degree in computer engineering from Universidad Tecnológica de la Mixteca, in 2005, the M.Sc. and Ph.D. degrees in computer sciences from Tecnológico de Monterrey, in 2008 and 2011, respectively, the M.Ed. degree from Universidad del Valle de México, in 2017, the second B.Sc. degree in project management from Universidad Virtual del Estado de Guanajuato, in 2019, and the M.Ed.A. degree from the Instituto de Estudios Universitarios, in 2019. He is currently an Assistant Research Professor with the School of Engineering and Sciences, Tecnológico de Monterrey. His research interests include computational intelligence, machine learning, heuristics, metaheuristics, and hyper-heuristics for solving combinatorial optimization problems. He is a member of Mexican National System of Researchers, Mexican Academy of Computing, and the Association for Computing Machinery.

HUGO TERASHIMA-MARÍN (Senior Member, IEEE) received the Ph.D. degree in informatics from Tecnológico de Monterrey, Monterrey Campus, in 1998. He is currently a Full Professor with the School of Engineering and Sciences and the Leader of the research group with strategic focus in intelligent systems. He has been the Principal Investigator of various projects for industry and CONACyT. He has published more than 100 research papers in international journals and conferences. He has supervised five Ph.D. dissertations and 34 master's thesis. His research interests include computational intelligence, heuristics, metaheuristics and hyper-heuristics for combinatorial optimization, the characterization of problems and algorithms, constraint handling, and the applications of artificial Intelligence and machine learning. He is a member of the National System of Researchers (Rank II), Mexican Academy of Sciences, and Mexican Academy of Computing.