IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 12 October 2024, accepted 18 November 2024, date of publication 22 November 2024, date of current version 4 December 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3504483

RESEARCH ARTICLE

Automatic Selection of Machine Learning Models for Armed People Identification

ALONSO JAVIER AMADO-GARFIAS, SANTIAGO ENRIQUE CONANT-PABLOS, (Member, IEEE), JOSÉ CARLOS ORTIZ-BAYLISS, (Member, IEEE), AND HUGO TERASHIMA-MARÍN, (Senior Member, IEEE)

School of Engineering and Sciences, Tecnológico de Monterrey, Monterrey, Nuevo Leon 64849, Mexico

Corresponding author: Santiago Enrique Conant-Pablos (sconant@tec.mx)

ABSTRACT This research aims to improve the automatic identification of armed people in surveillance videos. We focus on people armed with pistols and revolvers. Furthermore, we use the YOLOv4 to detect people and weapons in each video frame. We developed a series of algorithms to create a dataset with the information extracted from the bounding boxes generated by YOLOv4 in real-time. Thereby, we initially developed six-armed people detectors (APD) based on six machine learning models: Random Forest Classifier (RFC-APD), Multilayer Perceptron (MLP-APD), Support Vector Machine (SVM-APD), Logistic Regression (LR-APD), Naive Bayes (NB-APD), and Gradient Boosting Classifier (GBC-APD). These models use 20 predictors to make their predictions. These predictors are computed from the bounding box coordinates of the detected people and weapons, their distances, and areas of intersection. Based on our results, the RFC-APD was the best-performing detector, with an accuracy of 95.59%, a recall of 94.51%, and an F1-score of 95.65%. In this work, we propose to create selectors for deciding which APD to use in each video frame (APD4F) to improve the detection results. Besides, we implemented two types of APD4Fs, one based on a Random Forest Classifier (RFC-APD4F) and another in a Multilayer Perceptron (MLP-APD4F). We developed 44 APD4Fs combining subsets of the six APDs. Both APD4F types outperformed most of the independent use of all six APDs. A multilayer perceptron-based APD4F, which combines an MLP-APD, a NB-APD, and a LR-APD, presented the best performance, achieving an accuracy of 95.84%, a recall of 99.28% and an F1 score of 96.07%.

INDEX TERMS Machine learning, armed people detection, computer vision, object detection, YOLO.

I. INTRODUCTION

Surveillance cameras are the primary tool against criminals in public and private facilities. There are two types of these systems, supervised and unsupervised. The first of these relies on the efficiency of security personnel and their capacity to detect crime and subsequent capture of criminals rapidly. However, people tend to be affected by fatigue and various distractions. Conversely, unsupervised video surveillance camera systems are not helpful as a crime prevention method. They will only serve to recognize criminals after the offense is executed. Both types of video surveillance camera systems exhibit inefficiencies, as they do not facilitate the early identification of criminals. Thus, we propose automating the detection of armed people by applying computer vision techniques, object detection, and machine learning (ML) models. The proposed system can identify armed people and extract their facial images. This provides a timely alarm in the event of an incident involving weapons, as well as being a tool that facilitates the identification of those involved. Therefore, it reduces the reaction time to the occurrence of a crime. It makes it a novel research, as it does not focus on detecting handguns but armed people.

Considering the variety of types of weapons used by criminals, we wanted to focus on the most commonly used

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

to commit crimes. Whence, we have based on the report carried out by the United Nations Office on Drugs and Crime (UNODC). It developed a report containing information from 81 countries about the most firearms seized [1]. It concluded the following: pistols (39%), shotguns (25%), rifles (18%), revolvers (14%), submachine guns (3%), and machine guns (1%). According to this report, handguns are the most common weapon used by criminals, which includes two types of guns: pistols and revolvers. These represent 53% of the weapons commonly used in crimes. For this reason, we have focused on detecting armed people with handguns.

Detecting armed people has some challenges, such as occlusion, hidden weapons, and people close to each other. Occlusion occurs when some objects face each other in the same direction as the camera. In this way, the camera captures incomplete portions of the objects in the same region. The concealed firearm situation occurs when a person hides firearms between his body and clothing. People close to each other happen when people are very near, and it is difficult to realize who is carrying the gun. These challenges encourage the search for new, reliable solutions. Hence, this research applies the object detection model, YOLO, in its fourth version (YOLOv4) [2]. We chose the model above because it presents an optimal balance between frames per second (FPS), mean average precision, and computational cost. However, the main objective of this research is to identify armed people and not just detect firearms. Although some works have aimed to improve object detection models to recognize weapons through video surveillance, little research has aimed at identifying armed people. In our work, we only use YOLOv4 to extract information related to each class. That information then feeds the APDs to identify armed individuals.

We propose six APDs to identify armed people. These are Random Forest Classifier (RFC-APD), Multilayer Perceptron (MLP-APD), Support Vector Machine (SVM-APD), Logistic Regression (LR-APD), Naive Bayes (NB-APD), and Gradient Boosting Classifier (GBC-APD). We trained such APDs using a dataset with bounding box coordinates, distances, and intersection areas between handguns and the people in each video frame as main predictors. However, we hypothesize that all APDs have strengths and weaknesses. Therefore, we can train ML models to select the optimal APD based on the specific features presented in each video frame. Thus, we developed the APD4F. We implemented with two ML models, MLP-APD4F and RFC-APD4F. These models identify and apply the best of the six APDs to each situation in the video. Furthermore, being aware that these algorithms can be refined, we make the codes, models, videos, and datasets available at https://github.com/AlonsoJAG/automatic_model_selector

The main contributions of this research are the following:

- Developed a methodology for creating Armed People Detector selectors (APD4F) that leverage characteristics of individuals and weapons in each video frame to enhance armed individual identification.
- Designed and evaluated several APD4F models, notably an MLP-based APD4F model $MLP_{MLP+NB+LR}$ which achieved a performance of 95.84% accuracy, 93.07% precision, 99.28% recall, and an F1-score of 96.07%. This model offers reliable detection of armed individuals, providing stable performance metrics despite varying characteristics within video frames.
- Compiled a dataset of 8,533 records derived from four videos, three of which were specifically created for this study, complemented by the dataset from Amado et al. This combined dataset was used for training and testing the APD4F models.

The remainder of this document is structured as follows: Section II describes previous related work on object detection models for recognizing weapons. Furthermore, it describes research related to identifying armed people. Section III presents the methodology for developing this research. It contains the technical characteristics of the APDs and the APD4F. Section IV describes and analyzes the results. Finally, Section VI provides the conclusions and some ideas for future work.

II. RELATED WORKS

This section reviews the literature related to our research topic. However, there is little research on detecting armed people. Current research mainly aims to modify or apply techniques to improve firearm detection capabilities. Our work investigates beyond, and we propose detecting weapons and associating their belonging to the people in the video frame. We revise the literature from two perspectives: the detection of handguns and the detection of the people who carry those guns —the armed people.

A. HANDGUN DETECTION

The object detection model is the most crucial part of the research carried out for weapons detection. There is a wide variety of models. However, we can classify them into two groups according to the number of stages they use. Real-time weapons detection requires a balance between speed and precision. Most of the research experiments with YOLO (one stage) [2], [3], [4], [5], [6], [7], [8], [9], [10] and R-CNN variants (two stages) [11], [12], [13]. However, as can be seen in the research detailed below, YOLO has stood out for providing fast detection times, which is relevant in security issues, such as armed people detection. Furthermore, despite its velocity, it pays attention to its precision.

An example of research dedicated to improving the performance of the object detector for detecting firearms is that presented by Duran-Vega et al. [14]. They applied a quasi-recurrent neural network architecture. Furthermore, they used Temporal YOLOv5. It allows us to extract temporal information from the video. Additionally, they used data augmentation combined with mosaic and mixup to get better performance. Likewise, Hashi et al. [15] worked on weapon detection, comparing the efficiency of combining different models and object detectors. The models used

were VGG19, ResNet50, and Google Net v3, and the object detectors were Faster R-CNN and YOLO. The model with the best performance was the one presented by ResNet50. YOLOv6 achieved the highest mAP and inference speed compared to the faster R-CNN. Another research that compares the effectiveness of different object detection models for weapon detection is the one developed by Boukabous and Azizi [16]. In addition, their proposal identifies bladed weapons. They applied YOLOv5, faster R-CNN, and Single-Shot Multi-box Detector (SSD). The best object detector model was YOLOv5. It obtained optimal results in mAP and inference speed for real-time predictions. Likewise, Bhatti et al. [17] applied two approaches: sliding window/classification and region proposal/object detection. The models used were VGG16, Inception-V3, Inception-ResnetV2, SSDMobileNetV1, Faster RCNN Inception-ResnetV2 (FRIRv2), YOLOv3, and YOLOv4. The model with the best performance was YOLOv4. Narejo et al. [18] presented similar research. They used YOLOv2 and YOLOv3 for weapon detection. Likewise, they applied their dataset to train the object detectors.

Additionally, Mehta et al. [19] used YOLOv3 for fire and handgun detection based on an anomaly detection system. It presented a validation loss of 0.2864, with a detection rate of 45 frames per second. It has been benchmarked on datasets like UGR, FireNet, and IMFDB with accuracies of 82.6%, 86.5%, and 89.3%, respectively. Another application that seeks to identify weapons in real-time is the proposal by Jain et al. [20]. They applied two object detection models, Faster R-CNN and single-shot multi-box detector (SSD). Furthermore, Veranyurt and Sakar [21] developed research for concealed handgun identification. They applied it to thermal images in video surveillance cameras in real-time. The experiments applied various models for classification and segmentation. VGG-19 and YOLOv3 achieved the best performance. Berardini et al. [22] proposed a framework deployed on an NVIDIA Jetson Nano edge device connected to an IP camera for handgun and knife detection. They applied two CNNs, the first to detect people and the second to detect handguns and knives. Another research dedicated to weapon detection through the application of YOLOv5s and Faster R-CNN is the one presented by Ashraf et al. [23]. In their research, they additionally applied areas of interest. Their model preprocesses the frames, removing the background from the image using the Gaussian blur algorithm. YOLOv5s presented better performance, achieving high recall and detection speed. Lim et al. [24] presented a research example of the faster R-CNN improvement in weapons identification. They introduced an improved multi-level deep feature pyramid network (M2Det). Their goal was to infer firearms from a non-canonical perspective. Besides, González et al. [25] applied Faster R-CNN with Feature Pyramid Network and ResNet-50 for weapon detection. The model was trained with real and synthetic images generated from the research. They focused on the results of the application of synthetic data in the object detector. It resulted in a weapons detection system with near real-time operation (90 ms inference time with an NVIDIA GeForce GTX-1080Ti card).

Currently, some research uses pose estimation by generating artificial skeletons on the image of people's bodies. These skeletons are applied as tools. The goal is to predict various situations, such as people's behavior. Below is detailed research on how this technique was used to predict armed people. The research presented by Salido et al. [26] sought to reduce false negatives in weapons detection by adding information related to the pose of armed people. Adding the pose of the people seeks to compensate for the small size of the firearms. They verified that adding the skeleton to the armed person images and then processing it by the object detector improved the detection. This test was executed with Faster R-CNN, YOLOv3, and Retinanet. YOLOv3 presented the best performance. 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. [27] proposed a combined method for verifying the existence of a weapon. First, the object detector must detect the weapon. Then, that weapon must match the person's pose for the presence of a gun to be considered a true positive. If this match does not exist, the existence of a weapon is ruled out. The authors achieved a 17.5% improvement in AP using the combined method rather than applying YOLOv3 alone. Ruiz-Santaquiteria et al. [28] aimed to overcome the problem of the small size of the guns and proposed a combined architecture that uses the estimation of the body pose and the characteristics of the appearance of the weapon. It developed through the application of architectures based on transformers and CNNs. Another similar research is presented by Ruiz-Santaquiteria et al. [29], where they proposed the application of the weapon's appearance and the human pose's information on the same platform. The architecture recognizes and extracts people's wrists and contrasts them with their poses. The results exceeded the AP initially obtained from 4.23% to 18.9%. Likewise, Chatterjee and Chatterjee [30] applied the poses, particularly in the hands, to identify whether a person is holding a weapon. Their research was carried out using various ML models to classify the presence of guns. Another research that focuses on the regions of people's hands for weapon detection is presented by Lamas et al. [31]. It estimates people's body poses. The innovative aspect of this research is applying the adaptive pose factor, which allows optimal localization of the hand regions. This technique relies on the distance between the body and the camera.

Transformers have emerged as state-of-the-art in deep learning, initially designed for Natural Language Processing (NLP) tasks [32]. Recently, their application has expanded into Computer Vision (CV) [33] through Vision Transformers (ViT) [34], which excels in tasks such as image classification and object detection. For instance, Tummala et al. employed

Vision Transformers for brain tumor classification from MRI images [35], while Zheng and Jiang applied ViTs to traffic sign classification [36]. Furthermore, Yang et al. introduced a modality fusion vision transformer for collaborative classification using hyperspectral and LiDAR data [37]. ViTs have also been explored for weapon detection, as demonstrated by Son et al., who applied Swin Transformer [38] alongside models such as Mask R-CNN for detecting various weapon types [39]. Additionally, Singh et al. compared CNN and ViT models for firearm classification, where Swin-ViT achieved a precision of 97.55% [40]. Nayak et al. proposed a weapon detection model using YOLOv8 with a GhostNet backbone, a C3 transformer module, and a neck (YOLO-GTWDNet), designed for both image and live video stream processing [41].

B. ARMED PEOPLE DETECTION

Agarwal et al. [42] proposed a model to detect explosives hidden inside abandoned objects. This research classifies abandoned objects with the aim of identifying the owner. It also seeks to know the last known position of the owner. This application works in real time. They applied short- and long-term models to determine abandoned objects. Additionally, for person detection, they used MobileNet-Single Shot Multi-Box Detector (SSD) and Histogram of Oriented Gradients (HOG) with Support Vector Machines (SVM). Besides, it employed region of interest (RoI) and excluded undesirable regions. They obtained the features of all the humans in a frame through Scale-invariant feature transform (SIFT) [43]. Then, they matched these stored features from consecutive frames in order to track the person by applying Fast Library for Approximate Nearest Neighbors(FLANN) [44]. Likewise, McPartlin and Lowe's research aimed to identify the owners of abandoned luggage [45]. In addition, to determine the location and the path followed by the person. The research lines were based on image analysis using three algorithms: object detection, object tracking, and object classification. In addition, there was threat assessment, which included two algorithms: observation and analysis and threat classification. The criteria used to classify abandoned luggage were that the owner was not within two meters for at least 30 seconds. Furthermore, Moura et al. [46] proposed to apply the Intersection Over Union (IoU) concept to two classes of objects: people and weapons. The person who, together with the weapon, shows the highest IoU in the same video frame is the armed person. They applied YOLOv5 as an object detector.

Amado-Garfias et al. proposed a method for identifying armed individuals through heuristics and machine learning models [47]. They extract information related to individuals and weapons present in each video frame using YOLO in real-time, which subsequently informs their models and heuristics [1]. The Random Forest Classifier demonstrated the best performance, achieving an accuracy of 85.44%, precision of 87.07%, recall of 88.68%, and an F1-score of 87.87%. We utilized the same hyperparameters from this study for our APDs, enabling us to benchmark the results between the APDs and the APD4Fs, thereby enhancing the findings presented in the previous work.

The task of detecting armed persons is more complex than just identifying weapons. It is a challenge due to different limitations, such as the proximity between people, occlusions, and hidden weapons. Since little research addresses this topic, it represents a significant opportunity for future work.

III. METHODOLOGY

This section describes the training, hyperparameters, and operating processes of the APDs and APD4Fs. We have divided the methodology into three stages. The first stage explains the operation and characteristics of the APDs. The second stage shows the procedure and features of the different APD4Fs. The third stage describes the attributes of the datasets used to train and test the APDs and the APD4Fs.

Figure 1 illustrates the workflow underlying our approach. Specifically, it demonstrates how the models developed in our previous research by Amado et al., which serve as our Armed People Detectors (APDs), are applied within this study. These APD models were imported to support the 44 APD4Fs we propose here. The APD4Fs were also trained and configured to interface with YOLO for object detection. The detector, whether used in real-time or not, provides the essential data that the APD4Fs utilize for predictive analysis.

A. ARMED PEOPLE DETECTOR

We trained the APDs to identify armed people by analyzing information from the video frame. They comprise six ML models: RFC-APD, MLP-APD, SVM-APD, LR-APD, GBC-APD, and NB-APD. The APDs receive 20 measurements of the bounding boxes generated by YOLO in real-time. We trained YOLOv4 from scratch to identify three classes: handguns, people, and faces. The dataset was randomly divided into 4,000 images for training and 1,000 images for testing, sourced from various internet platforms. The images include close-ups of firearms, armed and unarmed individuals, and actual crime scene footage. Additionally, some gun images were obtained from the Internet Movie Firearms Database (IMFDB) [48]. YOLOv4 was trained over 6,000 iterations, achieving a mean Average Precision (mAP) of 89% and a loss of 2.1856. APDs process the information and identify the armed people involved in each video frame. Then, the information related to people and faces is fed into the face ML model to recognize the faces of the armed people. The training dataset for the APD was originally 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 then generated 12,652 records. The dataset has 20 predictors, which include the corners and center coordinates of the bounding boxes, areas, and distances between the handgun and the person's bounding boxes.

The predictors are detailed as follows.

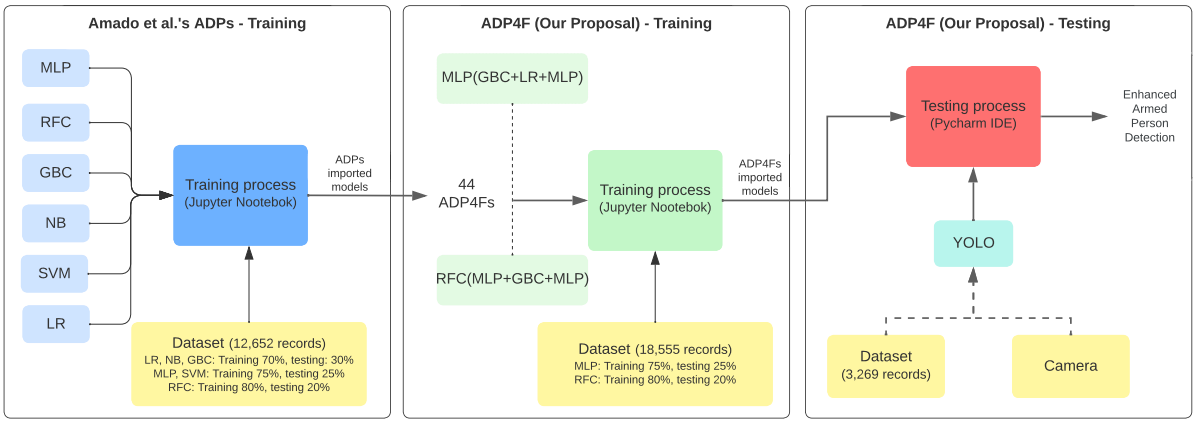


FIGURE 1. Description of the research methodology workflow.

- **N°Frame**. The corresponding video frame number. (measured in units).
- **N°People**. The number of people in the frame. (measured in units).
- **PerXcenter**. The central coordinate of the X axis of the person (measured in pixels).
- **PerYcenter**. The central coordinate of the Y axis of the person (measured in pixels).
- **PerXmin**. Minimum coordinate of the X axis of the person. (measured in pixels).
- **PerYmin**. Minimum coordinate of the Y axis of the person. (measured in pixels).
- **PerXmax**. Maximum coordinate of the X axis of the person. (measured in pixels).
- **PerYmax**. Maximum coordinate of the Y axis of the person. (measured in pixels).
- **N° Handguns**. Number of handguns (measured in units).
- **HgXcenter**. Central coordinate of the X axis of the handgun. (measured in pixels).
- **HgYcenter**. Central coordinate of the Y axis of the handgun. (measured in pixels).
- **HgXmin**. Minimum coordinate of the X axis of the handgun. (measured in pixels).
- **HgYmin**. Minimum coordinate of the Y axis of the handgun. (measured in pixels).
- **HgXmax**. Maximum coordinate of the X axis of the handgun. (measured in pixels).
- **HgYmax**. Maximum coordinate of the Y axis of the handgun. (measured in pixels).
- **Center intersection**. Predictor that expresses whether the center of the handgun's bounding box is inside the person's bounding box (binary measured).
- **Intersection area**. Intersection area between the handguns' bounding boxes and the people bounding box (measured in pixels^2).
- **Handgun area**. Handgun bounding box area (measured in pixels^2).
- **Distance Per-Hg**. Distances between the handgun center bounding boxes and the people center bounding boxes (measured in pixels).
- **Intersection**. Position of the handgun relative to the people (categorical measured).

In addition, APDs are tested with test videos that last 2 minutes and 29 seconds. Processing this video generated a dataset of 3,269 records.

Moreover, the hyper-parameters considered for the six APDs are as follows: GBC-APD has 100 estimators and a learning rate of 0.1. Likewise, it uses a maximum depth of six, a subsample of 1.0, and Friedman-MSE as a criterion. We trained MLP-APD through 500 iterations. It employs four hidden layers of 25 neurons, each one. Furthermore, it applies the Adam optimizer. For training LR-APD, we used a maximum number of iterations of 10,000. It has one versus remainder (OvR) scheme, and the inverse of the regularization strength C equals 100. The solver is the limited memory Broyden Fletcher Goldfarb Shanno (LBFGS). RFC-APD has ten estimators and ten folds for cross-validation. It applies an entropy criterion and uses a maximum depth of two. SVM-APD uses a regularization parameter C equal to 1.0 and a linear kernel. NB-APD works with the Bernoulli Naïve Bayes classification algorithm.

Furthermore, we have applied data standardization to the MLP-APD and SVM-APD. We used the preprocessing module from scikit-learn, called `StandardScaler`. We developed the models in the Jupyter Notebook, where standardization was applied. However, the models were imported to receive the data generated by YOLO. These data also had to be standardized in the case of MLP-APD and SVM-APD. Consequently, standardization of input data in real-time was necessary. Therefore, we use the formula applied by the `StandardScaler` function, $z = (x - u)/s$, where x represents the input data to be standardized, u represents the mean, and s is the standard deviation of the training samples. We have experimented with all APDs,

with training and testing ratios in the following proportions: 75% 25%, 70% 30%, and 80% 20%. The data have been distributed in different proportions to improve the results of the models. The best results have been obtained, keeping the proportions as detailed below: MLP-APD and SVM-APD employed 75% (9,489 cases) for their training and 25% (3,163 cases) for testing of the dataset (12,652 cases). LR-APD, NB-APD, and GBC-APD handled their training with 70% (8,856 cases) and testing with 30% (3,796 cases). Finally, RFC-APD used 80% of the dataset for training (10,121 cases) and 20% for testing (2,531 cases).

B. ARMED PEOPLE DETECTOR FOR EACH FRAME

The main goal of APD4Fs is to identify and apply the best APDs to a specific situation presented in the video. APD4Fs are designed by combining two or three APDs. We developed 44 APD4Fs using two ML models, Random Forest and Multilayer Perceptron. As previously mentioned, the APDs are RFC-APD, MLP-APD, SVM-APD, LR-APD, NB-APD, and GBC-APD. We developed the APD4F dataset, taking the APD with the highest probability of correctly predicting the armed or unarmed person in each case of the training video as the ground truth. The prediction probability is determined through the `scikit-learn` function, `predict_proba`. The ground truth is exchanged between two or three APDs according to the APD4F configuration. The hyperparameters of APD4Fs are as follows: RFC-APD4F presents ten estimators and ten folds for cross-validation. It considers an entropy criterion and a maximum depth of four. We used 80% of the dataset for training and the remaining 20% for testing. MLP-APD4F works with four hidden layers, each with 25 neurons. We trained it through 500 iterations by applying the Adam solver to adapt the weights. MLP-APD4F used 75% for training and 25% of the dataset for testing. As in the APDs, the MLP-APD4F training data was standardized using the `scikit-learn` preprocessing module called `StandardScaler`. Therefore, it is also necessary to standardize the real-time data that is delivered to the imported model by applying the standardization formula detailed in the previous subsection. The distribution of the records in the dataset is illustrated in Table 1.

APD4Fs function in two distinct phases. The first phase entails data extraction, during which an object detector processes video surveillance signals to identify three classes: guns, people, and faces. The algorithm then extracts information from the bounding boxes and passes this data to the next phase.

In the second phase, model selection and armed people detection, the system receives the data from the previous stage and inputs it into the APD4F. Based on the characteristics of the various predictors, the system selects the most suitable APD. Subsequently, the chosen APD processes the same data to distinguish between armed and unarmed individuals. Additionally, data pertaining to individuals and faces is forwarded to a facial recognition model to identify armed individuals. Both phases are visually represented in Figure 2.

The dataset used to train the APD4F differs for each of the 44 model combinations. This is because each APD has different probabilities of predicting each record in the dataset. This way, when we combine the APDs through APD4F, the APD with the highest probability becomes the ground truth. Furthermore, records that could not be resolved by APDs were removed from the dataset. Thus, the APD4F learns to choose the APDs that correctly resolve each record in the dataset. Thereby, the datasets for each APD4F have different numbers of records. The dataset used to train the APD4Fs consists of five videos that are five minutes and 18 seconds. Using the processed videos, we generated 18,555 records. The APD4Fs were executed on three test videos. These have a total duration of two minutes and 29 seconds long. These videos make up our test set with a total of 3,269 records. A detailed view of the datasets used in this work is presented in Table 2.

C. CHARACTERISTICS OF THE DATASET APPLIED TO THE TRAINING AND TESTING OF THE APDS AND APD4FS

The dataset was created by processing each video frame. We extracted data from each frame related to combining all people with all weapons. Thus, the records are made up by grouping the data of the first person with the data corresponding to the first weapon. Then, the first person with the second weapon, and so on, taking all the people and handguns. Consequently, the number of records per frame depends on the number of people and guns. The ground truth used for APDs indicates whether the person is armed or unarmed. Meanwhile, the applied ground truth for the APD4F is represented by the APDs with the highest probability of correctly predicting the record. The datasets used for both techniques, APD and APD4F, have the same predictors. They only vary in the ground truth and the amount of data used for training and testing.

The dataset for this research was created from several videos for training and testing. The training set consists of five videos. We developed four of them, and the remaining one was downloaded from YouTube [49]. The training videos have a total duration of five minutes and 18 seconds. All training videos have a resolution of 1920 $\times$ 1080 pixels. These videos were processed and generated 18,555 records. Among the people in the videos, 7,203 is armed and 11,352 are unarmed. We trained the APD4Fs using all the videos in the training set. However, the APDs used only three videos. These three videos represent 12,652 records. The number of armed people is 4,228, and the number of unarmed people is 8,424. The test set consists of three videos, one created by the authors. The other two videos were downloaded from YouTube [49], [50]. All test videos have 1920 $\times$ 1080 pixels of resolution, except for the one-minute and five-second video downloaded from YouTube. It has a resolution of 854 $\times$ 480 pixels. The test videos have a total duration of two minutes and 29 seconds. These videos represent 3,269 records. The number of armed people is 1,678, and the number of unarmed people is 1,591. We tested the APD4Fs

TABLE 1. Number of records assigned to each APD in the dataset for training the MLP-APD4Fs and RFC-APD4F.

APD4F/APD	MLP	LR	GBC	NB	RFC	SVM
$MLP/RFC_{GBC+LR+MLP}$	15,050	1,838	1,632			
MLP/RFC_{LR+GBC}		5,259	13,199			
MLP/RFC_{LR+NB}		895		16,848		
$MLP/RFC_{LR+NB+GBC}$		225	1,744	16,519		
$MLP/RFC_{MLP+GBC}$	15,995		2,350			
$MLP/RFC_{MLP+GBC+RFC}$	15,980		2,319		213	
$MLP/RFC_{MLP+GBC+SVM}$	15,955		2,250			307
MLP/RFC_{MLP+LR}	15,996	2,480				
$MLP/RFC_{MLP+LR+SVM}$	15,960	2,307				230
MLP/RFC_{MLP+NB}	13,015			5,485		
$MLP/RFC_{MLP+NB+LR}$	12,997	151		5,362		
$MLP/RFC_{MLP+NB+SVM}$	15,109			2,666		727
$MLP/RFC_{MLP+RFC}$	17,575				919	
$MLP/RFC_{MLP+RFC+SVM}$	17,459				30	222
$MLP/RFC_{MLP+SVM}$	17,486					976
$MLP/RFC_{RFC+GBC}$			17,911		536	
MLP/RFC_{RFC+LR}		16,564			1,799	
MLP/RFC_{RFC+NB}				17,127	470	
$MLP/RFC_{RFC+SVM}$					5,551	11,993
$MLP/RFC_{SVM+GBC}$			17,427			1,020
MLP/RFC_{SVM+LR}		14,360				3,305
MLP/RFC_{SVM+NB}				17,054		657

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

	Purpose	Length	Number of records	Source	Applied
1	Training	1min 01s	18,555	YouTube [49] (00h 00min 38s : 00h 01min 37s)	APD/APD4F
2		1min 10s		Own development	APD/APD4F
3		1min 17s		Own development	APD/APD4F
4		1min 00s		Own development	APD4F
5		0min 50s		Own development	APD4F
6	Testing	0min 37s	3,269	YouTube [49] (00h 03min 59s : 00h 04min 36s)	APD/APD4F
7		0min 47s		Own development	APD/APD4F
8		1min 05s		YouTube [50]	APD/APD4F

and APDs on all the videos of the test set. Table 2 details the datasets' distinctive features.

All experiments were conducted on a Dell XPS 8930-2018 computer equipped with an Nvidia GeForce GTX 1050 TI GPU and an eighth-generation Intel (R) Core I7-8700 processor operating at 3.2 GHz with six cores.

IV. EXPERIMENTS AND RESULTS

The results section is divided into two subsections. The first consists of the detection of armed people using APDs. The second describes the results of the APD4Fs to select the most optimal APD model. In addition, it explains the results of the APD4Fs for detecting armed people.

A. RESULTS OF THE ARMED PEOPLE DETECTOR

Table 3 depicts the metrics obtained during the training of the APDs. The model with the best accuracy is the GBC-APD with 99.31%. The next model with the best performance is the MLP-APD, which achieved an accuracy of 99.02%, followed by LR-APD, SVM-APD, and NB-APD, with 92.78%, 91.65%, and 90.83%, respectively. Regardless of the variation in performance in terms of accuracy achieved by the models, the general results are within what we consider a reasonable range. From this, we can conclude that all six models have learned efficiently from the training dataset.

Table 4 presents the results of applying each of the six APDs to the test dataset. The RFC-APD exhibits the best performance, reaching an accuracy of 95.59%. The methods SVM-APD, LR-APD, NB-APD, and GBC-APD were also competent with accuracies of 94.52%, 94.31%, 94%, and 93.66%, respectively. Conversely, MLP-APD showed the worst performance with an accuracy of 78.43%. All APDs, except for MLP-APD, exceeded 90% accuracy. Experimenting with the test dataset, which contains videos with features different from the training dataset, exhibits the efficiency of the models. This experiment demonstrates the capacity of the APDs to identify armed people.

Another experiment to evaluate the effectiveness of APDs and APD4Fs is to measure the accuracy using distance intervals between people. We obtain it by measuring the intersection area generated between the people bounding boxes in square pixels. The intersection areas are extracted from the videos in the test dataset, ranging from 0 to 91,605.8 pixels^2, and divided into 11 intervals. The number of intervals was

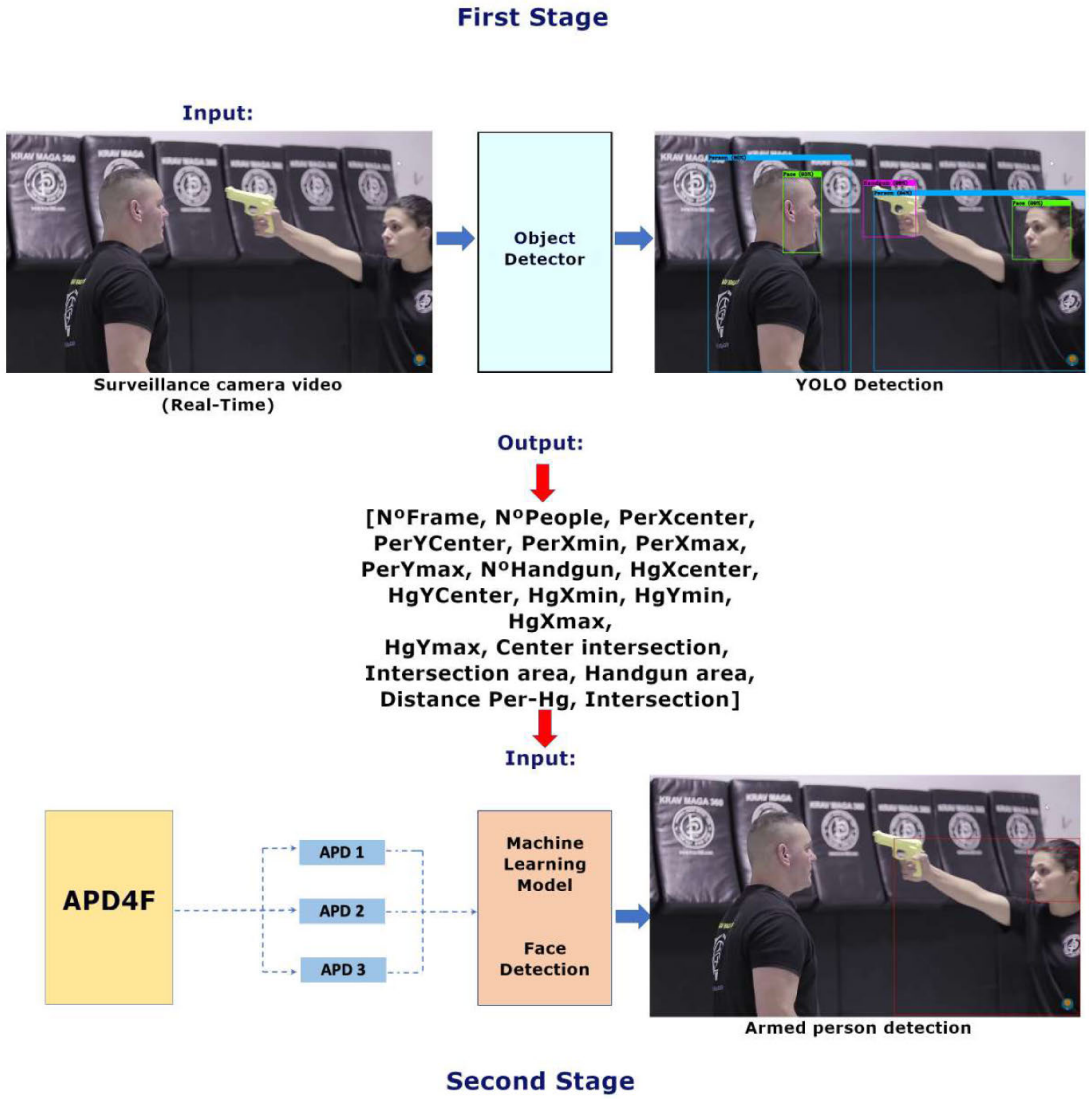


FIGURE 2. APD4F phases: extracting data from a video frame, and model selection and armed people detection.

TABLE 3. Performance metrics obtained by APDs on the training set.

Models	RFC	MLP	SVM	LR	NB	GBC
Accuracy	90.79%	99.02%	91.65%	92.78%	90.83%	99.31%
Precision	79.05%	99.04%	84.93%	85.43%	79.58%	98.84%
Recall	97.44%	98.01%	91.2%	95.06%	98.37%	99.15%
F1-Score	87.28%	98.52%	87.95%	89.99%	87.99%	98.99%

determined by Sturges' rule, with the following formula: $k = 1 + 3.322\ log(n)$, where n represents the experiment sample. We decided to create an interval for the cases where there were no areas of intersection between the people. These represent 2,757 cases of the test set. For the remaining 512 records, we applied Sturges' rule, resulting in 9.99 intervals. Therefore, we have 11 intervals, one for those that do not have an area of intersection between

TABLE 4. Performance metrics obtained by APDs on the test set.

Methods	RFC	MLP	SVM	LR	NB	GBC
Accuracy	95.59%	78.43%	94.52%	94.31%	94%	93.66%
Precision	96.82%	84.09%	97.28%	93.77%	90.09%	93.49%
Recall	94.51%	71.51%	92.32%	95.23%	99.22%	94.21%
F1-Score	95.65%	77.29%	94.73%	94.5%	94.44%	93.85%

TABLE 5. Performance of APDs through the areas of intersection between people in the test set. The best result per row has been highlighted using bold.

Intersection area between people (Pixels^2)		Number of records	Accuracy					
			RFC	MLP	SVM	GBC	LR	NB
0	0	2,757	97.82%	80.26%	97.64%	97.24%	97.89%	**98.62%**
58.73	834.51	51	80.39%	64.70%	**98.03%**	**98.03%**	**98.03%**	90.19%
874.63	1,062.41	52	82.69%	53.84%	**100.00%**	98.07%	98.07%	96.15%
1,062.99	1,282.18	52	82.69%	53.84%	96.15%	96.15%	**98.07%**	**98.07%**
1,283.4	2,802.23	51	80.39%	70.58%	90.19%	**92.15%**	86.27%	86.27%
2,907.36	7,119.68	51	98.03%	**100.00%**	98.03%	**100.00%**	82.35%	74.51%
7,188.75	22,987.57	51	72.54%	**74.51%**	72.54%	72.54%	**74.51%**	52.94%
23,017.87	32,675.78	50	**90.00%**	84.00%	76.00%	66.00%	80.00%	48.00%
32,874.75	36,845.13	52	**78.84%**	71.15%	48.07%	40.38%	51.92%	46.15%
36,866.78	56,366.17	52	**84.61%**	63.46%	48.07%	38.46%	38.46%	48.07%
56,662.43	91,605.88	50	**86.00%**	50.00%	50.00%	42.00%	42.00%	50.00%

people and 10 for those that do have areas of intersection. However, to maintain order in the graphs, we have tried to keep the same amount of data in each interval where there are areas of intersection. Likewise, the accuracy of each interval is graphed in the arithmetic mean calculated on the upper and lower areas of the intersection of each interval. Note that a larger intersection area between people makes it difficult to discriminate between armed and unarmed people. Table 5 shows the accuracy of the eleven intervals applied to APDs. In the first interval, when there are no intersection areas, the APD presenting the best accuracy is NB-APD, with 98.62%. The following models in terms of accuracy were LR-APD, reaching an accuracy of 97.89%, and RFC-APD, with 97.82%. In the interval that presents the largest intersection area (56,662.43 - 91,605.8 pixels^2), the model with the highest accuracy is RFC-APD with 86%, followed by SVM-APD, MLP-APD, and NB-APD, all with 50%. As expected, we can notice that some APDs perform better when there is no intersection area between people, others with a slight intersection area, and others with a large intersection area. Then, the intersection area is not a criterion that determines how well an APD performs.

Table 6 shows the errors made by each of the six APDs. In addition, it shows us how many of the mistakes made by one APD have been correctly predicted by the other five models. The APD with the best performance in predicting the cases from the videos in the test dataset is the RFC-APD, which produced 144 wrong predictions. From these 144 errors, the other methods correctly predicted, at most, 87 cases: MLP-APD (87 cases), NB-APD (79 cases), LR-APD (50 cases), GBC-APD (45 cases), and SVM-APD (31 cases). These results confirm that, although the APDs can become highly accurate, there are cases where they will fail, but other models will succeed.

B. RESULTS OF THE ARMED PEOPLE DETECTOR FOR EACH FRAME

The following experiment consisted of training the APD4Fs and measuring their performance in choosing the most appropriate APD. They learned to recognize the characteristics of the predictors in each frame and to identify the APD with the highest probability of predicting armed people. We want to highlight that this experiment focuses on recognizing the ability of the APD4Fs to select the APD with the highest likelihood of identifying armed people rather than on the effectiveness of the identification itself. Table 7 presents the performance achieved by the APD4Fs for videos within the training dataset. The model with the best accuracy is MLP_{RFC+NB}, which reached 99.9% of accuracy. MLP_{SVM+NB} is the next best-performing model with 99.81% and followed by MLP_{LR+NB} with 99.61%. Even the worst model, $RFC_{MLP+NB+SVM}$, showed an acceptable performance, achieving an accuracy of 86.97%. So, the evidence suggests that the models have learned the features of the video frames, allowing them to predict, with acceptable performance, the best APD for each situation.

This time, we evaluate the models' performance on unseen videos (those from the test dataset). So, we applied each of the APD4Fs to the videos in the test dataset. This experiment, unlike the one explained in the previous paragraph, aims to measure the efficiency of the APD4Fs in detecting armed people. The table shows the results of the APD4Fs to identify armed people, through the joint use of several APDs. The results of this experiment are shown in Table 8. The model with the best performance is $MLP_{MLP+NB+LR}$, reaching an accuracy of 95.84%. The next model with the best performance is the RFC_{MLP+NB} with an accuracy of 95.8%, followed by MLP_{MLP+LR} with 95.53%. The interesting thing about this experiment is that it shows

TABLE 6. Errors and solutions generated by the APDs in the test dataset.

Models		RFC	MLP	SVM	LR	NB	GBC
Wrong predictions		144	705	179	186	196	207
Correct predictions by other models	RFC	-	601	66	92	131	108
	MLP	87	-	83	95	99	112
	SVM	31	601	-	42	71	48
	LR	50	599	35	-	77	77
	NB	79	589	54	67	-	92
	GBC	45	610	20	56	81	-

TABLE 7. Performance metrics obtained by APD4Fs on the videos from the training dataset.

Models	Accuracy	Precision	Recall	F1-Score
$MLP_{GBC+LR+MLP}$	96.45%	91.54%	91.23%	91.31%
MLP_{LR+GBC}	97.39%	96.78%	96.83%	96.81%
MLP_{LR+NB}	99.61%	98.91%	97.05%	97.96%
$MLP_{LR+NB+GBC}$	98.91%	94.56%	97.84%	96.14%
$MLP_{MLP+GBC}$	97.47%	95.05%	93.47%	94.24%
$MLP_{MLP+GBC+RFC}$	97.70%	94.86%	93.11%	93.97%
$MLP_{MLP+GBC+SVM}$	97.25%	92.76%	87.73%	90.07%
MLP_{MLP+LR}	98.22%	95.52%	96.99%	96.24%
$MLP_{MLP+LR+SVM}$	98.14%	94.57%	91.04%	92.74%
MLP_{MLP+NB}	97.57%	97.25%	96.92%	97.08%
$MLP_{MLP+NB+LR}$	97.79%	96.62%	97.26%	96.94%
$MLP_{MLP+NB+SVM}$	98.59%	97.64%	94.89%	96.22%
$MLP_{MLP+RFC}$	99.43%	97.90%	95.85%	96.85%
$MLP_{MLP+RFC+SVM}$	99.39%	90.27%	89.52%	89.89%
$MLP_{MLP+SVM}$	99.17%	95.88%	95.88%	95.88%
$MLP_{RFC+GBC}$	99.04%	92.84%	89.73%	91.22%
MLP_{RFC+LR}	97.27%	93.21%	91.06%	92.10%
$\boldsymbol{MLP_{RFC+NB}}$	**99.9%**	**99.53%**	**98.71%**	**99.12%**
$MLP_{RFC+SVM}$	99.47%	99.36%	99.42%	99.39%
$MLP_{SVM+GBC}$	98.59%	92.18%	94.82%	93.45%
MLP_{SVM+LR}	97.73%	96.56%	95.95%	96.25%
MLP_{SVM+NB}	99.81%	98.73%	98.73%	98.73%
$RFC_{GBC+LR+MLP}$	89.20%	91.45%	62.23%	67.40%
RFC_{LR+GBC}	89.08%	88.96%	83.96%	85.94%
RFC_{LR+NB}	97.91%	98.50%	80.57%	87.28%
$RFC_{LR+NB+GBC}$	94.48%	97.30%	72.96%	81.35%
$RFC_{MLP+GBC}$	90.97%	91.61%	65.36%	70.80%
$RFC_{MLP+GBC+RFC}$	89.44%	62.02%	40.31%	42.95%
RFC_{MLP+LR}	94.31%	94.49%	80.31%	85.56%
$RFC_{MLP+LR+SVM}$	93.72%	96.26%	72.47%	81.14%
RFC_{MLP+NB}	90.18%	91.66%	84.74%	87.29%
$RFC_{MLP+NB+LR}$	90.03%	91.72%	82.65%	86.37%
$RFC_{MLP+NB+SVM}$	86.97%	57.53%	48.01%	50.51%
$RFC_{MLP+RFC}$	97.02%	94.08%	68.39%	75.52%
$RFC_{MLP+RFC+SVM}$	96.85%	90.78%	72.03%	76.95%
$RFC_{MLP+SVM}$	96.34%	98.13%	66.58%	73.95%
$RFC_{RFC+GBC}$	98.78%	91.90%	84.43%	87.78%
RFC_{RFC+LR}	90.77%	95.36%	51.84%	51.13%
RFC_{RFC+NB}	98.66%	99.32%	76.50%	84.29%
$RFC_{RFC+SVM}$	98.03%	97.57%	97.91%	97.74%
$RFC_{SVM+GBC}$	97.28%	91.88%	80.70%	85.32%
RFC_{SVM+LR}	89.27%	83.62%	78.70%	80.81%
RFC_{SVM+NB}	98.33%	98.50%	78.18%	85.45%
$RFC_{MLP+GBC+SVM}$	89.81%	62.28%	41.67%	44.79%

that using APD4Fs we obtain a higher accuracy than applying the APDs independently. Below are the APDs and their respective accuracies, which integrate the configuration of the three APD4Fs that presented the best performance: MLP obtained 78.43%, NB reached 94% and LR achieved 94.31%. These APDs showed lower accuracy than those delivered when working together in the APD4Fs.

Furthermore, similar to the experiment performed with APDs on the distance intervals between people described in the previous subsection, Table 9 shows the results of the same experiment but applied to the four best performance APD4Fs. The best performance in the first interval is RFC_{MLP+NB} with 98.58%, followed by $MLP_{MLP+NB+LR}$ with 98.54%, MLP_{MLP+LR} reached 97.96%,

TABLE 8. Performance metrics obtained by APD4Fs on the videos from the test dataset.

Models	Accuracy	Precision	Recall	F1-Score
$\boldsymbol{MLP_{MLP+NB+LR}}$	**95.84 %**	**93.07%**	**99.28%**	**96.07%**
$MLP_{GBC+LR+MLP}$	95.32%	94.05%	97.02%	95.51%
MLP_{LR+GBC}	94.00%	93.79%	94.57%	94.18%
MLP_{LR+NB}	93.76%	91.17%	97.25%	94.11%
$MLP_{LR+NB+GBC}$	93.88%	92.32%	96.06%	94.15%
$MLP_{MLP+GBC}$	94.06%	91.17%	97.91%	94.42%
$MLP_{MLP+GBC+RFC}$	95.16%	92.69%	98.33%	95.43%
$MLP_{MLP+GBC+SVM}$	94.80%	93.43%	96.66%	95.02%
MLP_{MLP+LR}	95.53%	93.82%	97.73%	95.73%
$MLP_{MLP+LR+SVM}$	94.40%	93.63%	95.59%	94.60%
MLP_{MLP+NB}	94.95%	92.47%	98.15%	95.23%
$MLP_{MLP+NB+SVM}$	90.51%	84.65%	99.58%	91.51%
$MLP_{MLP+RFC}$	94.58%	93.60%	96.00%	94.79%
$MLP_{MLP+RFC+SVM}$	94.58%	92.52%	97.31%	94.85%
$MLP_{MLP+SVM}$	94.73%	93.77%	96.12%	94.93%
$MLP_{RFC+GBC}$	94.15%	93.81%	94.87%	94.34%
MLP_{RFC+LR}	94.67%	93.77%	96.00%	94.87%
MLP_{RFC+NB}	94.00%	91.46%	97.51%	94.39%
$MLP_{RFC+SVM}$	94.67%	93.51%	96.30%	94.89%
$MLP_{SVM+GBC}$	93.45%	93.31%	93.98%	93.64%
MLP_{SVM+LR}	94.52%	93.44%	96.06%	94.74%
MLP_{SVM+NB}	94.58%	91.81%	98.21%	94.9%
$RFC_{GBC+LR+MLP}$	82.22%	92.16%	71.45%	80.49%
RFC_{LR+GBC}	93.85%	93.67%	94.39%	94.03%
RFC_{LR+NB}	94.00%	90.09%	99.22%	94.44%
$RFC_{LR+NB+GBC}$	94.40%	91.27%	98.51%	94.75%
RFC_{RFC+LR}	94.55%	93.75%	95.76%	94.75%
$RFC_{MLP+GBC}$	81.89%	85.21%	78.30%	81.61%
$RFC_{MLP+GBC+RFC}$	81.24%	85.01%	77.05%	80.83%
RFC_{MLP+LR}	86.57%	93.05%	79.79%	85.91%
$RFC_{MLP+LR+SVM}$	82.22%	92.22%	71.39%	80.48%
RFC_{MLP+NB}	95.80%	93.75%	98.39%	96.01%
$RFC_{MLP+NB+LR}$	94.70%	92.53%	97.55%	94.98%
$RFC_{MLP+NB+SVM}$	82.53%	85.27%	79.73%	82.41%
$RFC_{MLP+RFC}$	88.71%	87.20%	91.41%	89.26%
$RFC_{MLP+RFC+SVM}$	86.11%	88.29%	84.08%	86.14%
$RFC_{MLP+SVM}$	80.05%	84.66%	74.67%	79.35%
$RFC_{RFC+GBC}$	93.57%	93.38%	94.16%	93.76%
RFC_{RFC+NB}	94.00%	91.13%	97.99%	94.44%
$RFC_{RFC+SVM}$	94.67%	93.51%	96.30%	94.89%
$RFC_{SVM+GBC}$	93.76%	93.40%	94.51%	93.95%
RFC_{SVM+LR}	94.73%	93.32%	96.66%	94.96%
RFC_{SVM+NB}	94.00%	90.09%	99.22%	94.44%
$RFC_{MLP+GBC+SVM}$	79.84%	84.54%	74.31%	79.09%

and $MLP_{GBC+LR+MLP}$ obtained 97.71%. However, in the last interval, all four models received the same accuracy of 50%. Figure 3 has been included to visualize the behavior of the metrics provided by APD and APD4F through the variations of the intersection areas of the test videos. In this experiment, it is evident that the APD4F, unlike the APDs, does not present as much variation in the accuracies presented in the different intervals. Table 10 shows the standard deviation calculated from the accuracies reached by the APDs and the best four APD4Fs in intervals of the intersection areas. The total standard deviation obtained by the APDs is 1.47, while the APD4Fs obtained 0.17. These results prove that there is much more variation in the APD's accuracy than in those shown by the APD4Fs. It is positive since it allows the APD4Fs to obtain more uniform results, maintaining optimal accuracies.

Figure 4 represents the importance of the predictors in the blue bars and their standard deviation in the black line of the four RFC-APD4Fs with the best accuracies obtained in this research. We obtained these results through the `feature_importances_` attribute. Unfortunately, this attribute does not apply to MLPs. This experiment was performed to analyze the importance of the predictors in APD4F when predicting the APD applied in each frame. This experiment shows that RFC_{MLP+NB} and $RFC_{MLP+NB+LR}$ prioritize determining the location of people in the video frame. Therefore, it attaches relevance to *PerYmax*, *PerXmin*, or *PerXcenter* predictors. Conversely, RFC_{SVM+LR} and $RFC_{RFC+SVM}$ prioritize weapon-related predictors, such as *intersection_area* between people and weapons and *Intersection_No_intersection*, indicating that there is no intersection between the person and the gun. Despite the top priority of $RFC_{RFC+SVM}$ is *PerXmin*, its subsequent four priorities are geared toward weapon position. It allows us to understand how APD4Fs manage predictors to select the most suitable APD.

TABLE 9. Performance of the best APD4Fs through the areas of intersection between people in the test dataset.

Intersection area between people (Pixels^2)		Number of records	Accuracy MLP (MLP+NB+LR)	RFC (MLP+NB)	MLP (MLP+LR)	MLP (GBC+LR+MLP)
0	0	2,757	98.54%	**98.58%**	97.96%	97.71%
58.73	834.51	51	92.15%	90.19%	94.11%	**98.03%**
874.63	1,062.41	52	**98.07%**	94.23%	**98.07%**	96.15%
1,062.99	1,282.18	52	98.07%	**100.00%**	98.07%	98.07%
1,283.4	2,802.23	51	**92.15%**	**92.15%**	**92.15%**	88.23%
2,907.36	7,119.68	51	94.11%	90.19%	98.03%	**100%**
7,188.75	22,987.57	51	74.51%	72.54%	74.51%	**76.47%**
23,017.87	32,675.78	50	80.00%	**84.00%**	**84.00%**	**84.00%**
32,874.75	36,845.13	52	69.23%	**71.15%**	**71.15%**	69.23%
36,866.78	56,366.17	52	**63.46%**	**63.46%**	**63.46%**	**63.46%**
56,662.43	91,605.88	50	**50.00%**	**50.00%**	**50.00%**	**50.00%**

TABLE 10. Standard deviation calculated from the intervals of the intersection areas.

Intersection area between people		Number of records	Standard deviation APD	APD4F
0	0	2,757	0.07191897	0.00431152
58.73	834.51	51	0.13469136	0.03347337
874.63	1,062.41	52	0.17934447	0.01838260
1,062.99	1,282.18	52	0.17488432	0.00965000
1,283.40	2,802.23	51	0.07843500	0.01960000
2,907.36	7,119.68	51	0.1095083	0.04349401
7,188.75	22,987.57	51	0.08379131	0.01604418
23,017.87	32,675.78	50	0.15073155	0.02000000
32,874.75	36,845.13	52	0.1507717	0.01108513
36,866.78	56,366.17	52	0.17767408	0.00000000
56,662.43	91,605.88	50	0.16476246	0.00000000
	Total		**1.4765135**	**0.17604081**

V. DISCUSSION

This research highlights the advantages of using APD4Fs over APDs. This section explains the reasons behind the APD4F's superiority, benefits, and applications. We divide the discussion into two parts. Firstly, we analyze the results presented by the APDs in the detection of armed people. Secondly, we describe the outcomes of the APD4Fs for selecting the most adequate APD model. Furthermore, the results of the APD4Fs in detecting armed people are detailed.

A. DISCUSSION ON ARMED PEOPLE DETECTOR

APDs performed well on the videos of both the training and test datasets. Overall, the accuracy range within the training set was between 99.02% and 90.79%. Meanwhile, the experiments with the test set yielded results between 95.59% and 93.66%, except for MLP-APD, which obtained 78.43%. When we analyze the errors made by APDs, we realize that models do not always make mistakes in the same cases given in the database. Moreover, in most cases, there is an APD that correctly predicts the error presented by another model. This effect is illustrated in Table 6. It means that each model has certain advantages in making predictions for specific features of the video frame. To demonstrate this, we experimented to evaluate the accuracy in different intervals of intersection areas applied to the videos in the test dataset. These experiments showed us that some APDs perform better than others in certain intervals despite not presenting the best accuracy in general.

An example of these advantages is the one presented by NB-APD, which has the best accuracy in the interval where there are no intersection areas between people. However, it had one of the lowest general accuracy of the APDs. It achieved, for the first interval, an accuracy of 98.62%. Furthermore, the SVM-APD, GBC-APD, and LR-APD presented the best accuracy in the second interval, reaching 98.03%. It covers 58.73 to 834.51 pixels^2. The best accuracy in the interval of 874.63 to 1,062.41 pixels^2 was 100%, reached by SVM-APD. NB-APD and LR-APD achieved the best accuracy from 1,062.99 to 1,282.18 pixels^2, with 98.07%. Similarly, GBC-APD reached the best accuracy within the range of 1,283.4 to 2,802.23 pixels^2, with 92.15%. GBC-APD and MLP-APD got 100% accuracy in the range of 2,907.36 to 7,119.68 pixels^2. MLP-APD and LR-APD acquired an accuracy of 74.51% within the range of 7,188.75 to 22,987.57 pixels^2. RFC-APD presented the best accuracy in the last four consecutive intervals. It delivered 90%, 78.84%, 84.61%, and 86% respectively. These intervals span from 23,017.87 to 91,605.88 pixels^2.

In this experiment, we can see that all six models have excelled in certain intersection area intervals. These results support the hypothesis of the present research: the use of

FIGURE 3. Metrics of the APDs and APD4Fs using intervals of intersection areas between people on the test dataset.

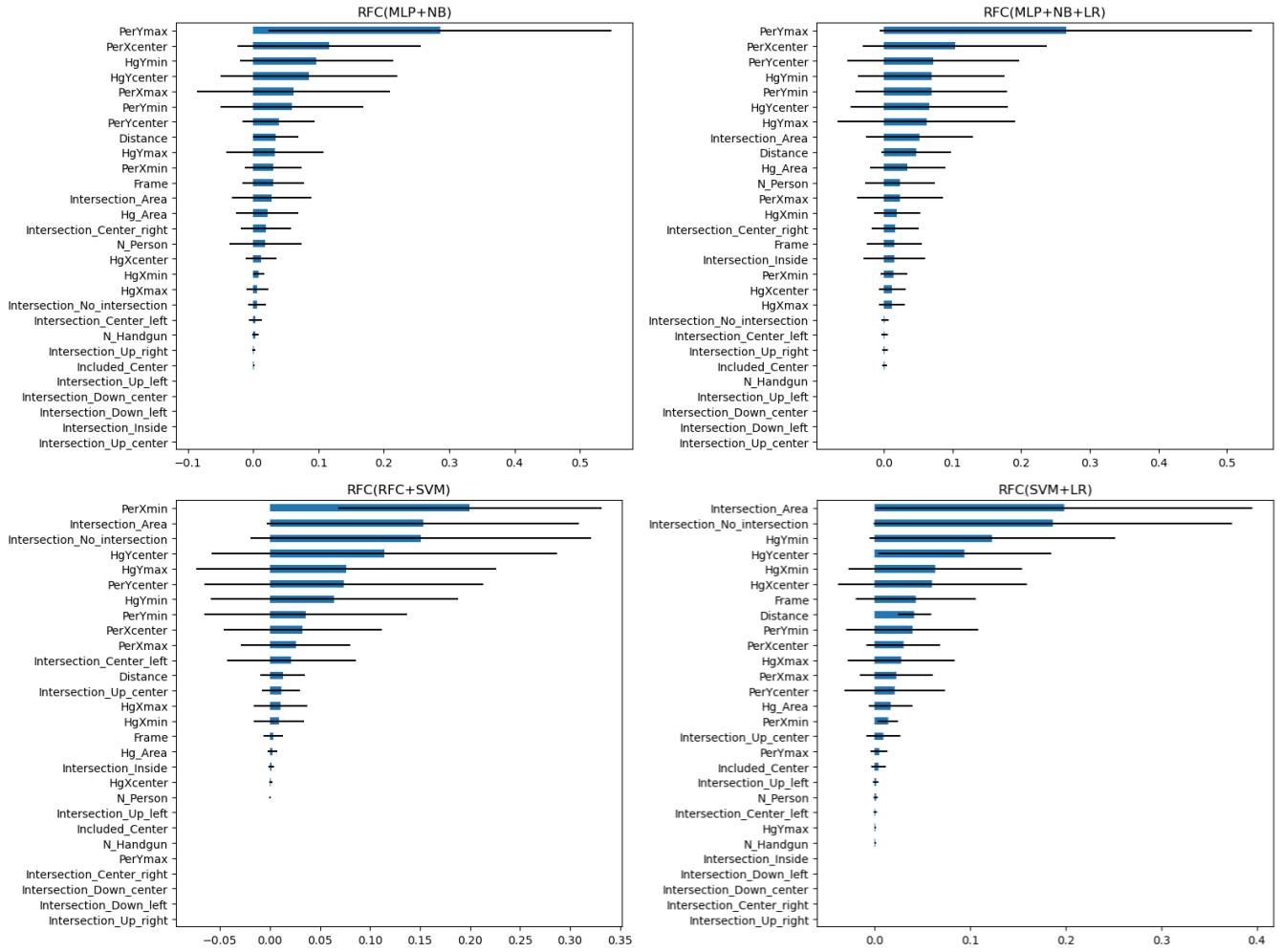


FIGURE 4. **Importance of the predictors used by the four RFC-APD4Fs that presented the highest accuracies.**

APD4Fs, for the analysis of the features of each video frame and the selection of the best APD improves the accuracy of the detection of armed people.

B. DISCUSSION ON ARMED PEOPLE DETECTOR FOR EACH FRAME

The results of the APD4Fs training are fairly high in accuracy. The accuracy range of the 44 APD4Fs is between 99.9% and 86.97%. The results with the test dataset are also satisfactory, achieving accuracies that range from 95.84% to 79.84%. These results show that APD4Fs learned the exiting patterns within the video frames. APD4Fs are able to identify the APD that performs the best under particular scenarios based on the features of the video frames.

We have also shown the ability of the APD4F predictors to prioritize predictors that identify the APD most likely to predict the cases present in a video frame. Figure 4 shows the importance of the predictors in the four RFC-APD4Fs with the best performance. The two models that contain MLP-APD and NB-APD, $RFC_{MLP+NB+LR}$ and RFC_{MLP+NB}, prioritize the *PerYmax* and then the *PerXcenter*, which means that, to make their predictions, they prioritize the people's positions. On the other hand, of the two models that contain SVM-APD, $RFC_{RFC+SVM}$ gives more importance to *PerXmin*, and the *Intersection_Area* between the people and the weapons. Furthermore, it prioritizes the *Intersection_No_intersection*, which indicates that there are no weapons inside the bounding box of the people. RFC_{SVM+LR} gives greater importance to the *Intersection_Area* and the *Intersection_No_intersection*. It is similar to the results presented by $RFC_{RFC+SVM}$. These last two APD4Fs prioritize the weapon's position in the video frame. This experiment demonstrates that the APD4F has inferred the most relevant characteristics that will allow them to make the most optimal selection. APD4Fs prioritize the predictors that increase the probability of correctly selecting an APD. Therefore, there are similarities in the predictor's prioritization with the same APDs.

Another experiment that allows us to demonstrate the qualities of the APD4Fs is the calculation of its performance in the intervals of the intersection areas between the people

in the test videos, whose results are shown in Table 9. If we compare the best APD against the best APD4F, we observe that $MLP_{MLP+NB+LR}$ obtained better accuracy in six of the intervals, while RFC-APD only in five. If we compare RFC-APD with the second-best performing APD4F, RFC_{MLP+NB} and RFC-APD outperformed each other in five intervals and obtained the same accuracy in one. We performed the same experiment with the next best-performing APD4F, resulting in MLP_{MLP+LR} outperforming RFC-APD in six intervals and equaling it in one of them. Likewise, $MLP_{GBC+LR+MLP}$ outperformed RFC-APD in six intervals out of five. APDs have variable performance; some of them are better at predicting cases where the weapon is very close to the person and others where it is not so close. Therefore, using a single APD to predict all cases present in a video can bias the results. The APD4Fs allow us to generalize the predictions and not focus on the abilities of a single APD. Proof of this is in the results obtained when calculating the standard deviation by intervals and total in the accuracies presented by the APDs and the four best APD4Fs. The detailed results are provided in Table 10. The general standard deviation of the APDs is 1.47, while that of the APD4Fs is 0.17. Therefore, in addition to the fact that the APD4Fs present a better accuracy, they also generalize the results presented by the APDs that integrate them, maintaining a minimum variation.

Furthermore, the experiment conducted with the APD4Fs and the APDs with the videos from the test dataset showed that the APD4Fs outperformed the APDs. However, to go deeper into the results, it is necessary to analyze various metrics. In this research, we have focused on using accuracy, but other metrics also show the superiority of APD4F, such as recall. This metric applies to what we require for this automatic recognition system for armed people since it evaluates the elements correctly identified as positive from the total number of true positives (TP). Conversely, precision evaluates the cases correctly identified as positive from a total number of elements identified as positive. It means from the number of cases predicted as positive and not from the total number of positive cases, as recall does. Considering that in this research, we want to evaluate the cases of armed people from the total number of cases presented, the most appropriate is recall. In addition, if we consider the slight imbalance between the armed and unarmed classes in the test dataset, it is also suitable to use the F1 score.

The best APD4F model was the $MLP_{MLP+NB+LR}$, which achieved an accuracy of 95.84%, a recall of 99.28%, and an F1 score of 96.07%. The second-best metrics were achieved by RFC_{MLP+NB} with an accuracy of 95.8%, recall of 98.39%, and F1 score of 96.01%. The two best APD4Fs outperformed the best APD, RFC, which achieved an accuracy of 95.59%, a recall of 94.51%, and an F1 score of 95.65%. The next APD with the highest accuracy was the SVM-APD, with 94.52%, a recall of 92.32%, and an F1 score of 94.73%. 17 APD4Fs surpassed this. The other APDs performed even lower, evidencing the superiority of the APD4Fs.

Based on our analysis, we conclude that the APD4Fs surpass the conventional APDs in three critical areas. First, the performance metrics—accuracy, precision, recall, and F1-score—of the APD4Fs significantly exceed those of the individual APDs, highlighting an enhanced capability in detecting armed individuals. Second, the APD4Fs demonstrate an ability to prioritize predictors effectively, optimizing APD selection based on the characteristics within each frame. This prioritization is evident in the strong correlation observed between predictor relevance and the APD selection, indicating that APD4Fs can adaptively identify and utilize significant features across video frames. Third, the APD4Fs maintain a high level of stability in their performance, unlike the individual APDs, whose accuracy tends to vary with changes in inter-individual distances within the frames. This consistent use of the most effective APD by APD4Fs ensures uniform detection accuracy across a range of scenarios, making them a superior choice for real-world applications.

VI. CONCLUSION

The APD4Fs demonstrated superior performance compared to the APDs. It was evident that each APD has specific advantages for the specific situations presented in the video frames. Although they have been trained with the same dataset, due to the characteristics of each APD, some have learned better how to resolve predictions according to specific situations. This difference supports the need for a model selector to identify the best APD for each situation. This research proved that APD4Fs identify video frame characteristics that are favorable for selecting each APD.

Likewise, APD4Fs avoid the generation of biases present in the APD. Some APDs perform optimally with specific videos and not with others. The APD4Fs generalize the predictions using the most appropriate APD for each case. It provides more stable and efficient results. The APD4Fs presented in this research are combinations of two or three APDs due to the small size of our dataset. The APD4Fs could contain more combinations of APDs. However, a bigger dataset would be necessary to avoid a class imbalance in the APD's ground truth. Consequently, the results can be improved by increasing the number of cases in the training dataset [51].

APD4Fs and APDs are effective for detecting armed people. However, they work exclusively when the object detector has identified a weapon in a video frame. These results could be improved by migrating to a more modern object detector, such as YOLOv9 [3]. It would allow us to identify weapons with greater precision. Therefore, we could detect armed people with better effectiveness.

The detection of armed people is limited to the identification of the weapon by the object detector. However, this task has some challenges, such as the small size of the guns, occlusion, and the presence of people with hidden weapons. Therefore, for future research, we plan to rely on recurrent neural networks, such as Long Short Term Memory [52] and Transformers [32], to predict the future position of the guns

when it is not detected. Thus, we would have the armed people identified permanently in real-time.

REFERENCES

[1] *Global Study on Firearms Trafficking*, United Nations Office Drugs Crime (UNODC), Vienna, Austria, Mar. 2020.

[2] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," 2020, *arXiv:2004.10934*.

[3] C.-Y. Wang, I.-H. Yeh, and H.-Y. M. Liao, "YOLOv9: Learning what you want to learn using programmable gradient information," 2024, *arXiv:2402.13616*.

[4] 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.

[5] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6517–6525.

[6] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.

[7] C. Li, L. Li, H. Jiang, K. Weng, Y. Geng, L. Li, Z. Ke, Q. Li, M. Cheng, W. Nie, Y. Li, B. Zhang, Y. Liang, L. Zhou, X. Xu, X. Chu, X. Wei, and X. Wei, "YOLOv6: A single-stage object detection framework for industrial applications," 2022, *arXiv:2209.02976*.

[8] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 7464–7475.

[9] G. Jocher, A. Chaurasia, and J. Qiu. (2023). *Ultralytics YOLOv8*. Accessed: Mar. 9, 2024. [Online]. Available: https://github.com/ultralytics/ultralytics

[10] G. Jocher. (2020). *Ultralytics YOLOv5*. Accessed: Mar. 1, 2024. [Online]. Available: https://github.com/ultralytics/yolov5

[11] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.

[12] 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. 2016.

[13] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.

[14] M. A. Duran-Vega, M. Gonzalez-Mendoza, L. Chang, and C. D. Suarez-Ramirez, "TYOLOv5: A temporal YOLOv5 detector based on quasi-recurrent neural networks for real-time handgun detection in video," 2021, *arXiv:2111.08867*.

[15] 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. 300–306, 2023.

[16] 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.

[17] M. T. Bhatti, M. G. Khan, M. Aslam, and M. J. Fiaz, "Weapon detection in real-time CCTV videos using deep learning," *IEEE Access*, vol. 9, pp. 34366–34382, 2021.

[18] S. Narejo, B. Pandey, D. E. Vargas, C. Rodriguez, and M. R. Anjum, "Weapon detection using YOLO V3 for smart surveillance system," *Math. Problems Eng.*, vol. 2021, no. 1, May 2021, Art. no. 9975700.

[19] P. Mehta, A. Kumar, and S. Bhattacharjee, "Fire and gun violence based anomaly detection system using deep neural networks," in *Proc. Int. Conf. Electron. Sustain. Commun. Syst. (ICESC)*, Jul. 2020, pp. 199–204.

[20] H. Jain, A. Vikram, Mohana, A. Kashyap, and A. Jain, "Weapon detection using artificial intelligence and deep learning for security applications," in *Proc. Int. Conf. Electron. Sustain. Commun. Syst. (ICESC)*, Jul. 2020, pp. 193–198.

[21] 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.

[22] 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. 2024.

[23] A. Ashraf, H. Abdul, M. Imran, A. Qahtani, A. Alsufyani, O. Almutiry, A. Mahmood, M. Attique, M. Habib, and R. Mohamed, "Weapons detection for security and video surveillance using CNN and YOLO-v5s," *Comput. Mater. Continua (CMC)*, vol. 70, no. 4, pp. 2761–2775, 2022.

[24] 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.

[25] J. González, C. Zaccaro, J. Álvarez-García, L. Morillo, and F. Caparrini, "Real-time gun detection in CCTV: An open problem," *Neural Networks*, vol. 132, pp. 297–308, Dec. 2020.

[26] 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.

[27] 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.

[28] 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.

[29] 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.

[30] 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.

[31] A. Lamas, S. Tabik, A. C. Montes, F. Pérez-Hernández, J. García, R. Olmos, and F. Herrera, "Human pose estimation for mitigating false negatives in weapon detection in video-surveillance," *Neurocomputing*, vol. 489, pp. 488–503, Jun. 2022.

[32] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[33] N. Parmar, N. Parmar, A. Vaswani, J. Uszkoreit, L. Kaiser, N. Shazeer, A. Ku, and D. Tran, "Image transformer," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 4055–4064.

[34] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16×16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.

[35] S. Tummala, S. Kadry, S. Bukhari, and H. Rauf, "Classification of brain tumor from magnetic resonance imaging using vision transformers ensembling," *Current Oncol.*, vol. 29, no. 10, pp. 7498–7511, 2022.

[36] Y. Zheng and W. Jiang, "Evaluation of vision transformers for traffic sign classification," *Wireless Commun. Mobile Comput.*, vol. 2022, Jun. 2022, Art. no. 3041117.

[37] B. Yang, X. Wang, Y. Xing, C. Cheng, W. Jiang, and Q. Feng, "Modality fusion vision transformer for hyperspectral and LiDAR data collaborative classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 17, pp. 17052–17065, 2024.

[38] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin Transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Montreal, QC, Canada, Oct. 2021, pp. 9992–10002, doi: 10.1109/ICCV48922.2021.00986.

[39] D. T. Son, N. T. K. Tram, and V. T. Anh, "Weapon detection using Swin transformer," in *Proc. Int. Conf. Adv. Technol. Commun. (ATC)*, Da Nang, Vietnam, Oct. 2023, pp. 328–333, doi: 10.1109/atc58710.2023.10318911.

[40] L. Singh, G. Singh, N. Agnihotri, and B. Dhaliwal, "Firearm classification based on CNN, vision transformer and Swin transformer models," in *Proc. Int. Conf. Commun., Comput. Sci. Eng. (IC3SE)*, May 2024, pp. 1–5, doi: 10.1109/IC3SE62002.2024.10592921.

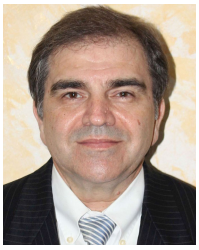
[41] R. Nayak, U. C. Pati, S. K. Das, and G. K. Sahoo, "YOLO-GTWDNet: A lightweight YOLOv8 network with GhostNet backbone and transformer neck to detect handheld weapons for smart city applications," *Signal, Image Video Process.*, vol. 18, no. 11, pp. 8159–8167, 2024.

[42] 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.* Cham, Switzerland: Springer, 2020, pp. 221–232.

[43] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[44] M. Muja and D. G. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration," in *Proc. VISAPP*, 2009, vol. 2, nos. 331–340, p. 2.
[45] M. McPartlin and D. G. Lowe, "Surveillance of unattended baggage and the identification and tracking of the owner," *7th Res. Framework Programme*, vol. 2, pp. 1–53, Oct. 2011.
[46] N. S. Moura, J. M. Gondim, D. B. Claro, M. Souza, and R. de Cerqueira Figueiredo, "Detection of weapon possession and fire in public safety surveillance cameras," in *Proc. Anais Do 18th Encontro Nacional de Inteligência Artif. E Comput.* Porto Alegre, Brazil: Sociedade Brasileira de Computação, 2021, pp. 290–301.
[47] A. J. Amado-Garfias, S. E. Conant-Pablos, J. C. Ortiz-Bayliss, and H. Terashima-Marín, "Improving armed people detection on video surveillance through heuristics and machine learning models," *IEEE Access*, vol. 12, pp. 111818–111831, 2024, doi: 10.1109/ACCESS.2024.3442728.
[48] *Internet Movie Firearms Database*. Accessed: Jan. 18, 2023. [Online]. Available: https://www.imfdb.org/wiki/Main_Page
[49] deportesUncomo. *Cómo Defenderse De Amenaza De Pistola - Técnicas Krav Maga*. Accessed: Feb. 10, 2023. [Online]. Available: https://www.youtube.com/watch?v=xrXkD2uWw_o
[50] elimparcialTV. *Mujer Policía Mata A Asaltante Durante Festejo Por Día De Las Madres En Brasil*. Accessed: Jun. 1, 2023. [Online]. Available: https://www.youtube.com/watch?v=sr58HURkw90&rco=1
[51] L. Alzubaidi, J. Bai, A. Al-Sabaawi, J. Santamaría, A. S. Albahri, B. S. N. Al-dabbagh, M. A. Fadhel, M. Manoufali, J. Zhang, A. H. Al-Timemy, Y. Duan, A. Abdullah, L. Farhan, Y. Lu, A. Gupta, F. Albu, A. Abbosh, and Y. Gu, "A survey on deep learning tools dealing with data scarcity: Definitions, challenges, solutions, tips, and applications," *J. Big Data*, vol. 10, no. 1, p. 46, Apr. 2023.
[52] 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 bachelor's degree in naval maritime sciences from the Peruvian Naval Academy, in 2006, the bachelor's degree in systems engineering from Universidad Científica del Sur, Lima, in 2019, and the master's degree from Tecnológico de Monterrey, 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 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 with 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 the Mexican National System of Researchers and the Mexican Academy of Computing.

JOSÉ CARLOS ORTIZ-BAYLISS (Member, IEEE) was born in Culiacán, Sinaloa, Mexico, in 1981. He received the B.Sc. degree in computer engineering from Universidad Tecnológico de la Mixteca, in 2005, the B.Sc. degree in project management from Universidad Virtual del Estado de Guanajuato, in 2019, the M.Sc. degree in computer sciences from Tecnológico de Monterrey, in 2008, the M.Ed. degree from Universidad del Valle de México, in 2017, the M.Ed.A. degree from Instituto de Estudios Universitarios, in 2019, and the Ph.D. degree from Tecnológico de Monterrey, in 2011. 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 the Mexican National System of Researchers, the 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 a strategic focus on intelligent systems. He has been a 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, characterization of problems and algorithms, constraint handling and applications of artificial Intelligence, and machine learning. He is a member of the National System of Researchers (Rank II), the Mexican Academy of Sciences, and the Mexican Academy of Computing.