Multimedia Tools and Applications (2024) 83:19109–19127
https://doi.org/10.1007/s11042-023-16231-x

# A deep-learning framework running on edge devices for handgun and knife detection from indoor video-surveillance cameras

**Daniele Berardini[1] · Lucia Migliorelli[1] · Alessandro Galdelli[1] · Emanuele Frontoni[2] · Adriano Mancini[1] · Sara Moccia[3]**

Received: 1 August 2022 / Revised: 24 May 2023 / Accepted: 4 July 2023 /
Published online: 26 July 2023


**Abstract**
The early detection of handguns and knives from surveillance videos is crucial to enhance people's safety. Despite the increasing development of Deep Learning (DL) methods for general object detection, weapon detection from surveillance videos still presents open challenges. Among these, the most significant are: (i) the very small size of the weapons with respect to the camera field of view and (ii) the need of a real-time feedback, even when using low-cost edge devices for computation. Complex and recently-developed DL architectures could mitigate the former challenge but do not satisfy the latter one. To tackle such limitation, the proposed work addresses the weapon-detection task from an edge perspective. A double-step DL approach was developed and evaluated against other state-of-the-art methods on a custom indoor surveillance dataset. The approach is based on a first Convolutional Neural Network (CNN) for people detection which guides a second CNN to identify handguns and knives. To evaluate the performance in a real-world indoor environment, the approach was deployed on a NVIDIA Jetson Nano edge device which was connected to an IP camera. The system achieved near real-time performance without relying on expensive hardware. The results in terms of both COCO Average Precision (AP = 79.30) and Frames per Second (FPS = 5.10) on the low-power NVIDIA Jetson Nano pointed out the goodness of the proposed approach compared with the others, encouraging the spread of automated video surveillance systems affordable to everyone.

**Keywords** Video surveillance system · Deep learning · Indoor weapon detection · Edge computing · Single board computer

## 1 Introduction

Crimes and violent activities involving handheld weapons are widespread across the world and this is a significant problem for society. According to [33], handgun and knife-based

✉ Daniele Berardini
d.berardini@pm.univpm.it

Extended author information available on the last page of the article

violence results in the 76% of total homicides, worldwide. The great relevance of such problem boosted the diffusion of Video Surveillance Systems (VSS), nowadays extensively used in both public and private places (e.g., airports, hospitals, house, and offices). Early detection of handheld guns and knives in VSS may allow for a prompt intervention of security officers, leading to a significant reduction of violent crimes and homicides. To date, such systems mainly rely on human operators to monitor video streams 24 hours a day, 7 days a week. This procedure, besides being time-consuming, poses issues including the cost of personnel, which is often not affordable, and the level of operator attention, which inevitably decreases as operator fatigue increases [5].

In recent years the field of automatic object detection has been extensively studied and several approaches have been proposed in literature. With the increasing popularity of Deep Learning (DL), automatic object detection approaches based on such paradigm progressively replaced earlier approaches based on classical computer vision techniques (e.g., DPM [7], Selective Search [32]), outperforming them in terms of both speed and reliability.

Despite the advances in the development of generic object detectors, weapon detection in video surveillance still remains an open problem [9, 38]. Compared with generic object detection, additional challenges arise that need to be addressed. The prominent one is the small size of the objects to be detected [6]. In fact, when it comes to detecting weapons from videos coming from a Closed-Circuit Television (CCTV), the size of the objects to be localized is often very small with respect to the Field of View (FoV) of the camera. In addition, the object of interest is usually several meters away from the spot where the camera is placed, further decreasing its relative size compared to the FoV. In generic object detection the small object issue can be partially mitigated with the use of increasingly powerful DL-based architectures [22, 37, 40] coupled with higher input image resolution, as discussed in [30]. On the downside, such architectures (i) must be trained on very large datasets to be effective and (ii) need to be deployed on expensive computational resources to run inference, which limits the applicability in the actual domain. As for (i), in the video-surveillance field, the lack of real-world reference datasets for weapon detection hampers the possibility of developing and applying burdensome detection architectures [38]. In order to enhance localization of small weapons and to overcome the data-lack issue, more robust solutions could be used based on video object segmentation techniques designed to require a weaker form of supervision [13, 39]; but the high complexity of such solutions would again lead to (ii). Given this, to enable the widespread deployment of automatic VSS while keeping costs affordable to everyone, there is an emerging need to develop approaches for weapon detection that can be executed with low-cost and computationally limited hardware.

Another major challenge in the detection of handheld weapons in video surveillance regards the efficiency. Object detectors in automatic VSS should be able to process data in near real-time, quantifiable in the range of 3-5 Frames per Second (FPS), considering the application needs [2]. This would guarantee a prompt response (e.g., alarms or communication with the competent authorities) in the event of affirmative detection.

In this perspective, a solution based on the edge computing paradigm, employing low-cost devices for the network-deployment phase, would allow to lower costs and mitigate privacy issues related to non-local data processing (e.g., cloud-based solutions). Moreover, an edge-computing solution would also enhance efficiency by reducing latency [1, 14].

Summing up the challenges and requirements in this still relatively unexplored research field, there is the need to find an effective yet efficient approach for small handheld weapons detection in CCTV under resource-constrained settings.

Driven by these considerations, this work presents a DL-based approach oriented to the edge computing paradigm for handgun and knife detection from indoor surveillance videos.

In comparison with the state of the art, the innovative contribution of the work is the proposal of an approach robust to the small-object size challenge yet deployable on edge devices with limited computing capacity – and consequently costs. The approach leverages a first CNN to obtain a prior detection of the people in the frame, and a second CNN to perform a subsequent detection of the potential handgun or knife within each person's bounding box. The edge device to deploy the proposed approach is the NVIDIA® Jetson Nano, a low-cost hardware platform oriented to DL [4]. For the intended purposes, a custom real-world indoor dataset consisting of 2425 annotated frames from 52 videos was collected. The dataset was obtained from recordings of a CCTV camera placed within a building's rooms.

The rest of the paper is organized as it follows. Section 2 reviews existing works in weapon detection from CCTV cameras, with a particular focus on handheld weapons. Section 3 describes the dataset and the data acquisition phase, the implemented approach as well as the experimental setup that was used and the deployment on the edge. Section 5 shows the experiments carried out and the results obtained. Section 6 presents the discussion of results. Finally, Section 7 concludes the paper along with considerations on future work. For enabling fair comparisons we made our codes available at GitHub[1].

## 2 Related work

Despite the popularity of generic object detection, research effort in automatic handguns and knives detection from surveillance videos is quite limited, especially in on-the-edge settings with Single Board Computers (SBCs). Among the seminal works in this field, the authors in [10] present an approach for firearms and knives detection from CCTV images based on visual descriptors and Machine Learning (ML). In particular, knife detection algorithm relies on sliding window technique followed by MPEG-7 based feature extraction and Support Vector Machine (SVM) for classification. Firearm detection also involves an image pre-processing step using background subtraction and Canny edge detection algorithms, in addition to sliding window and a classification based on MPEG-7 region shape descriptor. Both detection algorithms are evaluated on a custom-built dataset. Despite the valuable contribution, the use of sliding window approach and other time-consuming techniques limits the applicability in a real-world scenario.

In recent years, the increasing popularity of DL prompted the diffusion of new general-purpose architectures for object detection. Among the state-of-the-art object detectors some of the most widely used architectures are Faster R-CNN, based on a two stage detection process (i.e., a first region proposal stage followed by the object localization and classification), and one-stage object detectors (i.e., direct object localization and classification) like the You Only Look Once (YOLO) family [25–27] and Single Shot multibox Detector (SSD) [19].

Following this trend, more recent works in the field of handguns and knives detection from surveillance videos focused on exploiting such general-purpose detection architectures.

In [34] a handheld gun detection approach based on DL is proposed. The authors exploited a Faster R-CNN with a VGG16 backbone and compared its performance against several ML methods on the Internet Movie Firearms Database (IMFDB), proving its superiority. Similarly, the authors in [20] compared a sliding window and a region proposal approach both based on a VGG16 Convolutional Neural Network (CNN) classifier, with the latter outperforming the former in terms of speed and detection accuracy on a custom-built dataset.

[1] https://github.com/daniebera/on-the-edge-weapon-detection

These works have the merits of having highlighted the validity of DL over standard ML methodologies in video surveillance field. However, the datasets they use are not too representative of a real-world scenario. Indeed they are mainly made up of static images not acquired by CCTV and firearms are often the biggest and the only object in the foreground. These properties simplifies excessively the problem, invalidating the obtained results. Moreover, results obtained in [20] on test videos show a high number of false negatives (i.e., missed detections).

While the previous work focused exclusively on the detection of guns, the authors in [8] addressed the detection of both guns and knives. To this end, a Faster R-CNN was trained on a custom-built dataset obtained by collecting data from various sources, including part of the dataset in [20] for guns and COCO[2] images for knives. Both GoogleNet and SqueezeNet CNNs were tested as backbones for the Faster R-CNN. While the latter obtained results comparable with [20] for gun detection, it performed poorly on knives. On the other hand, the GoogleNet-based architecture outperformed the others in knives detection, even though with relatively low detection performances. In addition to the same issues on the data composition as the previous works, the proposed solution needs two distinct architectures to be effective both for handguns and knives, limiting the applicability in resource-constrained settings.

In the last few years research effort in general-purpose object detection also focused on the development of DL modules to be added on top of CNN architectures to improve detection performances. To this end, one of the most popular components is Feature Pyramid Network (FPN) [17], which combines high and low-resolution features among different CNN layers, improving the detection at different scales. Following the improvements over the existing state-of-the-art, more recent works in handguns and knives detection from CCTV adopted DL architectures with the integration of such components to tackle the issues related to small object sizes [9, 16]. Authors in [16] proposed an approach for handgun detection from CCTV based on a single stage object detector which integrates a multi-level FPN to enhance localization ability. The approach was tested on a custom dataset containing 5500 images with handguns extracted from CCTV videos. In [9] a Faster R-CNN with a FPN was exploited to perform gun detection on CCTV images. The training was performed on several combinations of non-CCTV data from [20], custom CCTV data and synthetic data. The evaluation on CCTV data highlighted that while the addition of synthetic training data slightly improved the results, the addition of non-CCTV data even decreased detection performances on small objects.

Although both works use components on top of the detection architectures to improve performances on their respective CCTV datasets, the results obtained in terms of weapon detection and inference speed do not allow to translate their approaches into the real-world domain. This is clearly expressed by the authors themselves in the conclusions of [9].

With an eye towards focusing on computationally undemanding detection architectures, the authors in [21] present a comparative analysis of the one-stage detectors YOLOv5 and YOLOv7 [36] on a custom dataset of people, handguns, rifles and knives, with images from Google Open Images Dataset, Roboflow Public Dataset and local sources. YOLOv5 outperformed YOLOv7, but the overall results were rather poor, hindering the real-world implementation of the approach. Moreover, as demonstrated by the image samples shown in the paper, the dataset does not mirror a real-world domain. Table 1 summarizes the state of the art approaches with their methodological details and limitations.

To take a first step towards the resolution of the still open issues in the field of automatic video surveillance, this work proposes an on-the-edge approach relying on a prior focus on

[2] https://cocodataset.org/

**Table 1** Summary of the state-of-the-art approaches in weapon detection

| Work | Method | Dataset | Limitations |
|---|---|---|---|
| Grega et al., 2016 | Sliding Window + MPEG-7 + SVM, | custom, | burdensome for edge devices, |
| | handgun and knife classification | CCTV, | not real-time |
| | | released w/o box-level annotations | |
| Verma et al., 2017 | VGG16 Faster RCNN, | IMDB, | unrealistic dataset, |
| | handgun detection | non-CCTV, | suffers from small objects, |
| | | public | burdensome for edge devices |
| Olmos et al., 2018 | VGG16-based region proposal approach, | custom, | unrealistic dataset, |
| | handgun detection | non-CCTV, | suffers from small objects, |
| | | released | burdensome for edge devices |
| Fernandez-Carrobles et al., 2019 | SqueezeNet Faster RCNN, | custom/COCO/Olmos et al., 2018 | unrealistic dataset, |
| | gun and knife detection | non-CCTV, | low knife-detection performance |
| | | not released | |
| Lim et al., 2019 | Multi-Level FPN-based single-stage detector, | custom, | burdensome for edge devices |
| | handgun detection | CCTV, | |
| | | not released | |
| González et al., 2020 | ResNet50 Faster RCNN with FPN, | custom, | burdensome for edge device, |
| | handgun detection | Synthetic/CCTV/non-CCTV, | low speed for real-time domain |
| | | released | |
| Olorunshola et al., 2023 | YOLOv5, | custom/Google Open Images, | unrealistic dataset, |
| | person, handgun, rifle and knife detection | non-CCTV | suffers from small objects |
| | | not released | |

the people for handgun and knives detection from CCTV video. The approach was trained and evaluated on a custom fully-CCTV dataset, built with appropriate expedients to overcome the limits highlighted in the state-of-the-art in terms of dataset composition. To the best of the authors' knowledge, this is the first approach for on-the-edge handgun and knives detection

to be executed on a SBC while maintaining high detection performances on CCTV data. Figure 1 shows the workflow of the proposed approach.

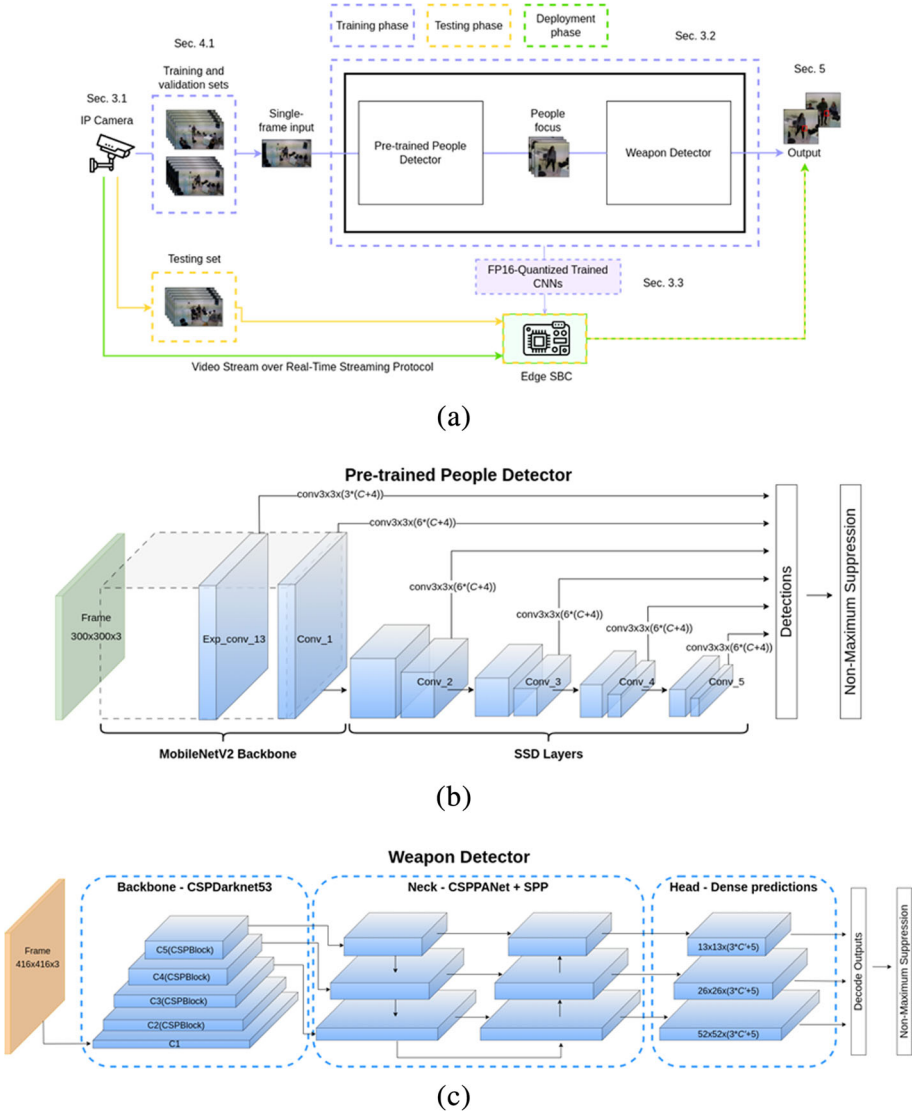



**Fig. 1** (a) Workflow of the proposed approach for indoor handgun and knife detection. After proper dataset preparation (described in Section 4.1) the weapon detector was trained using the output of the people detector (as detailed in Section 3.2) and the mean average precision performance was computed on the test set. Both convolutional neural networks were quantized in half-precision (i.e., FP16 quantization) and deployed in the NVIDIA Jetson Nano (as in Section 3.3) for real-time processing of the IP camera video stream. The details on the convolutional structure of (b) the people detector and (c) the weapon detector are shown, too

## 3 Materials and methods

### 3.1 Data acquisition

The CCTV dataset to train the DL algorithm for dangerous object detection was collected using a Dahua® 4MP Bullet Network Camera, a commercial Internet Protocol (IP) camera connected to a LAN with Power over Ethernet connection. An IP camera was chosen to have a good compromise between cost, performance, and quality in terms of image resolution and compression technology. The camera was mounted in a top corner of a nearly empty room to acquire video sequences in which a variable number of subjects were left free to move, with one of them holding a weapon (i.e., a knife or a handgun).

The acquisition sessions were carried out using a custom-built Python script, collecting a total of 52 video sequences of 30s each. The camera frame rate was set to 10 FPS at the default resolution of 1280×720 pixels, resulting in a total of 300 frames for each video sequence. Across the 52 collected video sequences, 19 different subjects appear holding a handgun or a knife. In addition, the same subject appearing in multiple videos uses different clothing. The average number of people per video is 2.88, with standard deviation of 1.05.

The dataset was gathered to simulate an indoor real-world application domain, so to overcome the limitations found in the datasets used in [20, 34]. Some of the proposed dataset challenges are shown in Fig. 2.

The major one is the very small size of the objects to be localized compared to the whole image (i.e., ∼0.1% of the image area, computed on the average ground-truth boxes areas), due to the distance of the people from the camera. A challenge related to the previous one

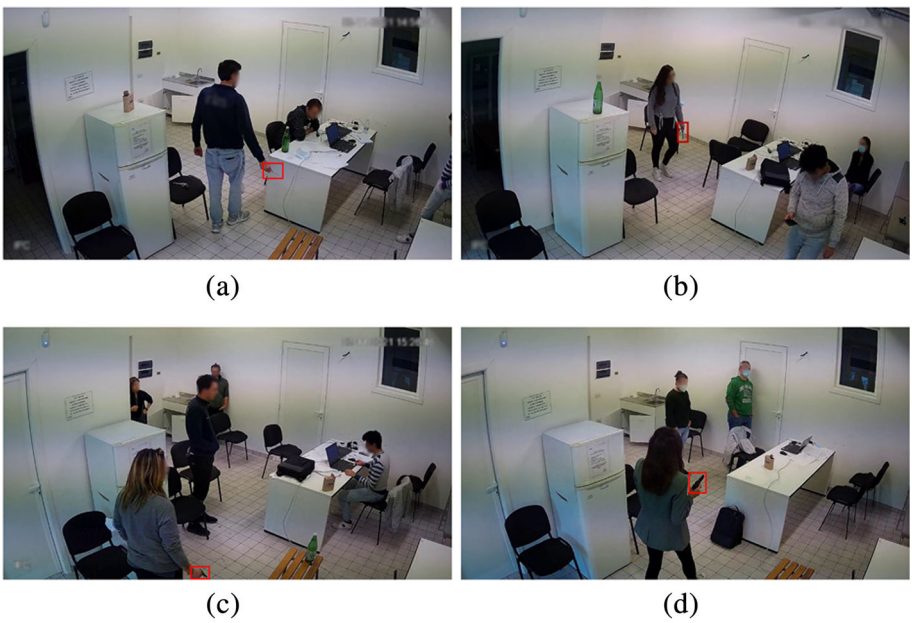



**Fig. 2** Sample of frames from the dataset extracted from a recording are shown to highlight the related challenges (e.g., multiple people, different dangerous and non-dangerous objects, distance from camera). For visualization purposes only, the hand-held guns and knives have been pointed out in red

is the poor contrast of the objects with respect to the background. Other challenges includes the presence of multiple people possibly holding non-dangerous objects (e.g., smartphone as in Fig. 2b), the different person's poses (e.g., sitting or standing as in Fig. 2a and c) and orientation (e.g., Fig. 2b with respect to Fig. 2a), the motion blur of the frames extracted from video sequences and the variability both in terms of subjects and intra-class objects (i.e., the use of different objects belonging to the same class).

### 3.2 Double-step deep-learning approach

The proposed approach is based on the observation that the weapon must necessarily be carried by a human subject to be dangerous. Thus, a double-step detection is proposed, with a prior detection of the people within each frame and a subsequent detection of the potential handgun or knife within each person's bounding box. Each step relies on a specific DL architecture with the aim of maximizing the speed-accuracy trade-off. The architectural choices also take into account the SBC hardware constraints in terms of memory footprint, as this is a primary concern when dealing with edge devices.

To carry out the prior people's detection, the SSD MobileNetV2 network [19, 29] was used, since it represents a good compromise between computational speed and people detection accuracy. Although its detection performance on the COCO dataset is lower than other architectures [29], it achieves nearly-optimal results when the performance evaluation is restricted on the *person* category, as shown in [23]. The SSD meta-architecture was chosen since it performs object localization by adopting a single-stage approach as opposed to other two-stage architectures which enhances accuracy to the detriment of speed (e.g., [28]). This allowed to reduce inference time.

MobileNetV2 was adopted as backbone for features extraction to further increase the speed of the SSD. MobileNetV2 is a lightweight CNN with a $3\times3$ convolutional layer followed by 19 inverted residual blocks [29], made up of three $1\times1$, $3\times3$, $1\times1$ convolutions interleaved with batch normalization and ReLU6 activation function, with a residual connection [12] between the $1\times1$ layers. The peculiar blocks' structure reduces the number of network parameters, thus increasing inference speed. The SSD meta-architecture stacks on top of the MobileNetV2 six output convolutional blocks, obtaining six different scales of detection for each input image.

An intermediate processing on each person's bounding box within the frame was performed before the weapon detection step. In particular, with the aim of preserving the objects' aspect ratio, a square crop from the original image was computed, according to both the center and the maximum side (between width and height) of each person's bounding box. Each crop was then fed to the subsequent step, resizing it according to the needs.

Once the prior information on each person's location within the frame was obtained, the subsequent step performed the detection of potential weapons carried by a subject. The YOLOv4-Cross-Stage-Partial (CSP) network [35] was implemented for handgun and knife detection due to its ability in detecting objects with respect to other state-of-the-art detectors, while attaining a good inference speed in resource-constrained hardware. Such results are highlighted by the comparison in both [3] and [29] on the COCO dataset. The YOLOv4-CSP was designed starting from the YOLOv4 network, originally introduced in [3]. As regards the backbone, the YOLOv4-CSP exploits the existing CSPDarknet53 (i.e., a Darknet53 with CSP stages each made up of 1,2,8,8,4 residual layer, respectively) and converts only the first CSP stage into an original Darknet residual layer for efficiency purposes. Instead, as regards the neck, it introduces CSP connections in the Path Aggregation Network architecture of

the YOLOv4 by transforming the original reversed darknet layers of YOLOv4 in reversed CSP darknet layers, maintaining the Spatial Pyramid Pooling module. As output layers, in its original configuration YOLOv4-CSP has three $1 \times 1$ convolutions with 255 filters each, so as to obtain detection at three different scales. As a result, the YOLOv4-CSP obtained a substantial gain in terms of trade-off between speed and accuracy, making it suitable for challenging detection tasks in resource-constrained settings.

To accomplish the detection on two classes (i.e., *knife*, *gun*), each of the three original output layer of the YOLOv4-CSP was replaced with a $1 \times 1$ convolution having $3 \times (2+5) = 21$ filters. In this way, each output layer provides a $n \times n \times 21$ map in which each of the $n \times n$ spatial locations encodes the information (i.e., coordinates, class scores and probability of containing an object or *objectness* score) of 3 candidate bounding boxes, so that 3 candidate bounding boxes $\times$ (2 class scores + 4 bounding box coordinates + 1 *objectness* score) = 21. The final selection of the most promising bounding boxes among candidates was performed according to the non-maximum suppression algorithm.

### 3.3 On-the-edge deployment

The SBC to deploy the DL framework was the NVIDIA® Jetson Nano Developer Kit[3]. The Jetson Nano allows to run low power Artificial Intelligence (AI) applications and has higher computational capabilities with respect to its competitors thanks to its 4 GB RAM, 4-cores ARM A57 CPU and on-board GPU with 128 CUDA cores based on a Maxwell micro-architecture. To enhance the inference speed of the proposed approach, NVIDIA® TensorRT[4] (TRT) was used. TRT is a framework that helps to optimize DL models and converts each model in a serialized engine to be next used for higher performance inference on NVIDIA® GPUs.

The MobileNetV2 model was converted in a pretty straightforward way from Tensorflow to the Universal File Format (UFF) for TRT-compatibility and from UFF to an optimized TRT engine. Instead, the YOLOv4-CSP model was converted from Keras to Open Neural Network Exchange (ONNX), an open format to represent AI models and to enable framework interoperability. Then, a TRT engine was created from the ONNX-like model. Since an internal default parameter of a Keras layer (i.e., upsampling) caused incompatibility issues with TRT, the Keras model structure was re-implemented with a custom layer. Moreover, two custom plugins in TRT were used to allow post processing of the model predictions and to apply Non-maximum Suppression algorithm, otherwise not supported in TRT engine.

Among the optimizations applied, both the DL models were post-training quantized to FP16, meaning that the models' constants (e.g., weights and bias) were converted from full precision (32-bit) to reduced precision (16-bit) floating point data type. The models quantization allowed to halve the model size, further improving the inference speed on the Jetson Nano.

Once the TRT engines were obtained, a pipeline was implemented to acquire frames from the IP camera and to process them first with the SSD Mobilenetv2 and then (potentially) with the YOLOv4-CSP. To carry out the data exchange between the IP camera and the Jetson Nano, a camera handler was implemented. Using camera-specific parameters (e.g., camera url, compression decoding standard to use, ...) the camera handler opens a video stream via Real Time Streaming Protocol (RTSP) and starts to receive video data as consecutive frames, which are then forwarded to the DL algorithms for processing. The people detector

[3] https://developer.nvidia.com/embedded/jetson-nano-developer-kit

[4] https://developer.nvidia.com/tensorrt

and the weapon detector were implemented as distinct processes that communicate via the Transmission Control Protocol (TPC), enabling them, in principle, to be physically decoupled (i.e., in an edge computing architecture with multiple nodes each process can communicate with the others independently from its physical location). The entire pipeline was designed to be suitable even in multi-camera settings via the opening of multiple camera-to-device RTSP video streams.

## 4 Experimental protocol

### 4.1 Dataset preparation

The acquired data were processed to create the dataset to train DL algorithms. To this goal, each video was trimmed, after manually checking the actual start and end of the individual acquisition session (i.e., when the person with the dangerous object in hand started and stopped walking). For each video sequence, in order to increase differences between consecutive video frames, the obtained frames were sampled at the rate of 1 every 4, obtaining a total of 2425 frames. Then, each frame was manually labeled using LabelMe[5], a publicly available annotation tool. Each annotation was performed by drawing a bounding box to tightly enclose the weapon and by assigning it to the correct object class, either *knife* or *gun*. Table 2 summarizes main information on the dataset, including number of frames and videos for each class (i.e., *knife* and *gun* classes).

During the inference, in the second step of the proposed approach, the YOLOv4-CSP for handguns and knives detection takes as input a square crop centered on each person's location instead of the original frame. Thus, the dataset was further processed to enable a faster training of the algorithm. A new dataset was constructed by square cropping on the original frame (having a resolution of 1280×720 pixels) the detected person holding the weapon. Each crop was then resized to 416×416 pixels (i.e., the network's input size) and the ground-truth bounding box coordinates were adjusted accordingly, to obtain the dataset used in training phase.

Table 3 summarizes the splitting strategy. The split was explicitly performed at video level (i.e., without mixing frames extracted from the same video across train, validation and test set) to attenuate possible bias. As a result of this strategy, the 78.3%, 8.3% and 13.4% of the available frames were used for training, validation and testing, respectively.

To improve DL algorithms' generalization capabilities, online data augmentation strategies were implemented on the training dataset. The applied data-augmentation transformations were: the change in brightness level as to simulate a scenario where artificial and natural lighting might change throughout the day, and the horizontal flipping to switch the weapon grip.

### 4.2 Training settings

The SSD MobileNetV2 was implemented using Tensorflow. The available weights obtained with the pre-training of the model on the COCO dataset were exploited for inference. The YOLOv4-CSP network was implemented and trained in Keras, a Python library running on

[5] https://github.com/wkentaro/labelme

**Table 2** Number of annotated frames (i.e., prior data-augmentation application) and number of video sequences related to each class of interest

| | N. of frames | N. of videos |
|---|---|---|
| *knife* | 1118 | 22 |
| *gun* | 1307 | 30 |
| total | 2425 | 52 |

top of TensorFlow. To train the YOLOv4-CSP the fine tuning methodology was applied. Starting from the pre-trained weights on COCO dataset, the model was trained with stochastic gradient descent (SGD) for 300 epochs using an initial learning rate of 0.001 and a batch size of 16. The learning rate reduction on plateau policy was applied with a reduction factor of 0.5 after 10 epochs with no improvements on the validation loss. Early stopping was also applied, with training termination after 75 epochs with no improvements on the validation loss. The optimal combination of batch size, optimizer and initial learning rate was found after the tuning of each hyper-parameter through manual search. The best weights configuration among epochs was retrieved according to the lowest loss value achieved on the validation set. The training was performed on a GPU NVIDIA® GeForce RTX™ 3090.

### 4.3 Ablation study and comparison with other architectures

Table 4 outlines the ablation studies conducted, including the DL models used in each step and the name of each approach evaluated. As a first ablation study, the use of SSD Mobilenetv2 in both steps of the proposed approach was investigated (i.e., SSD-MobilenetV2$^2$), to evaluate the impact on the detection performances. Since the work aims at developing an approach to maximize the speed-accuracy trade-off, also the use of YOLOv4-CSP in both steps was investigated (i.e., YOLOv4-CSP$^2$), mainly to evaluate its influence on the inference speed.

The proposed double-step approach for handguns and knives detection was compared also with the state-of-the-art methods in [9, 20] developed for weapons detection task, as well as with other popular object detectors.

Moreover, to point out the impact of using different image input sizes on detection performances, further comparison with state-of-the-art detectors with varying input sizes was performed. The rationale for such comparison lies on the fact that in general-purpose object detection the size of the input image can affect the performance. As a matter of fact, bigger input sizes often leads to more accurate but slower detection while smaller input sizes leads to faster but less accurate detection [31].

For a fair comparison, all the approaches were investigated using the same data splitting and were trained on the same computational hardware.

**Table 3** Number of video sequences related to train, validation and test datasets for each class

| | Train | Validation | Test |
|---|---|---|---|
| *knife* videos (frames) | 17 (870) | 2 (98) | 3 (150) |
| *gun* videos (frames) | 24 (1030) | 2 (103) | 4 (174) |
| total videos (frames) | 41 (1900) | 4 (201) | 7 (324) |

In round brackets is given the number of total frames in each set for each class, obtained by summing the number of labeled frames of each video belonging to the set considered

**Table 4** Proposed ablation study

| Name of the architecture | First-step | Second-step |
|---|---|---|
| SSD-MobilenetV2[2] | SSD-MobilenetV2 | SSD-MobilenetV2 |
| YOLOv4-CSP[2] | YOLOv4-CSP | YOLOv4-CSP |
| Proposed | SSD-MobilenetV2 | YOLOv4-CSP |

### 4.4 Performance metrics

To validate the proposed approach and compare it against the other state-of-the-art approaches, embracing the main literature in the field [3, 19], the detection performance was assessed using the standard COCO detection metrics as follows:

- Average Precision (AP) as primary metric computed as the average AP over the *knife* and *gun* classes and over 10 Intersection over Union (IoU) thresholds from 0.50 to 0.95 with a step size of 0.05 (0.50:0.95);
- AP50 as the AP computed at IoU 0.50, corresponding to the primary PASCAL VOC[6] metric;
- AP75 as a strict metric computed as the AP at IoU 0.75;
- APs and APm as the AP at IoU 0.50:0.95 for small (where object area $< 32^2$ pixels) and medium (where $32^2 <$ object area $< 96^2$ pixels) objects, respectively. The APl for large objects was not included since the weapons' related bounding-box area is always smaller than $96^2$ pixels in the collected dataset.

To further evaluate the presented approaches, efficiency metrics were also computed. Specifically, (i) floating point operations (GFLOPs) were computed when comparing the proposed approach with the others in the state of the art and (ii) inference speed on the Jetson Nano board ($FPS_{nano}$) in terms of FPS was computed for the ablation studies. Following the literature in closer fields [3, 15, 18, 24], both GFLOPs and FPS were plotted against AP. Additionally, to assess if significant differences exist among the approaches in the ablation studies, the one-way ANOVA (significance level = 0.05) with post hoc test was conducted. The considered population for each approach was the set of APs computed individually for each video in the test set.

## 5 Results

Table 5 summarizes the performance comparison in terms of AP, APm, APs, AP50, AP75 and $FPS_{nano}$ of the approaches in the ablation study.

The proposed approach achieved the highest results in all the COCO metrics, with an AP = 79.30 averaged over all classes, as well as an AP50 = 99.60, which represents the PASCAL VOC traditional metric computed at a single IoU of 0.50. The YOLOv4-CSP[2] approach obtained the same results of the proposed one for all the COCO metrics, while it achieved the worst results in terms of inference speed ($FPS_{nano}$ = 2.80) on the Jetson Nano board. In contrast, with the use of the SSD-MobilenetV2[2] approach the inference speed reached the highest value ($FPS_{nano}$ = 13.60), but the AP dropped significantly (AP = 21.20 with 58.10 points drops), along with all the other COCO metrics. In particular, the SSD-MobilenetV2[2]

[6] http://host.robots.ox.ac.uk/pascal/VOC/

**Table 5** COCO standard evaluation metric and inference speed comparisons for the ablation study. Best results are in **bold**

| Ablations | AP | APm | APs | AP50 | AP75 | $FPS_{nano}$ |
|---|---|---|---|---|---|---|
| SSD-MobilenetV2[2] | 21.20 | 26.50 | 19.40 | 56.80 | 8.90 | **13.60** |
| YOLOv4-CSP[2] | 79.30 | 50.10 | 49.30 | 99.60 | 93.90 | 2.80 |
| Proposed | **79.30** | **50.10** | **49.30** | **99.60** | **93.90** | 5.10 |

approach resulted in very low performance when computed at IoU of 0.75 (AP75 = 8.90) with a drop of 85.00 points with respect to the proposed approach. Significant differences were found (p-value $< 0.05$) between the approaches in the ablation studies. The speed-accuracy trade-off of the proposed approach with respect to the ablations is shown in Fig. 3.

When compared with the other state-of-the-art single-step approaches, the proposed one obtained by far the best performances for all the COCO metrics (shown in Table 6). Moreover, the proposed approach required GFLOPs = 26.35, achieving the best results in terms of trade-off between complexity and detection performance (as pointed out in Fig. 4).

The approach in [20] (i.e., Faster-RCNN-VGG16$_{640\times640}$) achieved low values for all the metrics, and particularly for AP, APs and AP75, with 10.40, 6.70 and 3.10, respectively. The same holds for the approach in [21], with AP = 10.10, AP50 = 23.30 and AP75 = 7.80. The approach in [9] (i.e., Faster-RCNN-ResNet50-FPN$_{1280\times720}$) required the highest GFLOPs (i.e., 223.68), while obtained the nearest performance to the proposed approach with AP = 30.50, AP50 = 67.60 and AP75 = 20.80, yet showing consistent degradation in performance when the IoU threshold increases from 0.50 to 0.75. Moreover, decreasing the input size on the same architecture led to a significant reduction of all the metrics (AP = 15.80 and AP

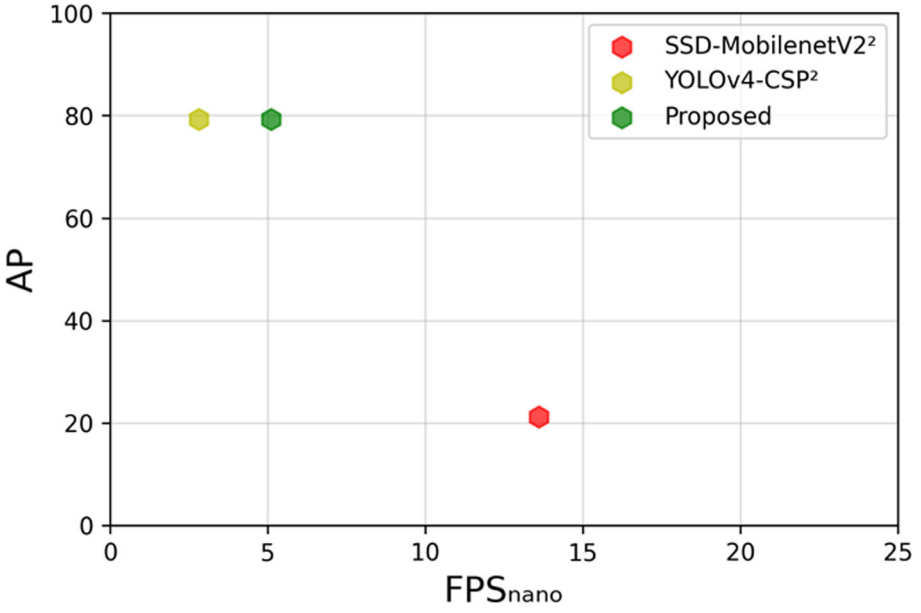



**Fig. 3** Comparison of the speed-accuracy trade-off in terms of frame per second on the Jetson Nano ($FPS_{nano}$) and Average Precision (AP) for the ablation study

**Table 6** COCO standard evaluation metric for comparisons between the proposed approach and other state-of-the-art architectures. Best results are in **bold**

| Compared Approaches | AP | APm | APs | AP50 | AP75 | GFLOPs |
|---|---|---|---|---|---|---|
| Faster-RCNN-VGG16$_{640x640}$ [20] | 10.40 | 14.50 | 6.70 | 28.20 | 3.10 | 138.12 |
| Faster-RCNN-ResNet50-FPN$_{1280x720}$ [9] | 30.50 | 34.70 | 27.50 | 67.60 | 20.80 | 223.68 |
| Faster-RCNN-ResNet50-FPN$_{640x640}$ | 15.80 | 21.20 | 10.20 | 39.50 | 8.80 | 99.64 |
| Faster-RCNN-ResNet50-FPN$_{416x416}$ | 7.00 | 9.60 | 4.50 | 19.50 | 1.40 | 44.56 |
| SSD-MobilenetV2$_{416x416}$ | 9.80 | 17.50 | 2.90 | 26.40 | 5.10 | **1.18** |
| YOLOv4-CSP$_{416x416}$ | 9.00 | 11.50 | 1.10 | 32.00 | 1.90 | 25.17 |
| YOLOv5$_{416x416}$ [21] | 10.10 | 17.20 | 3.60 | 23.30 | 7.80 | 47.90 |
| Proposed | **79.30** | **50.10** | **49.30** | **99.60** | **93.90** | 26.35 |

= 7.00 for Faster-RCNN-ResNet50-FPN$_{640\times640}$ and Faster-RCNN-ResNet50-FPN$_{416\times416}$, respectively)

In particular, when evaluating the approach in [9] using the same input size as the proposed approach (i.e., $416 \times 416$ pixels), the worst results were obtained in terms of AP, APm, AP50 and AP75 with values 7.00, 9.60, 19.50 and 1.40, respectively.

Both the architectures SSD-MobilenetV2 and YOLOv4-CSP in single-step settings (i.e., trained to directly detect the weapons from the original frames) obtained very low performance, with the worst value on small objects achieved by YOLOv4-CSP (APs = 1.10). On

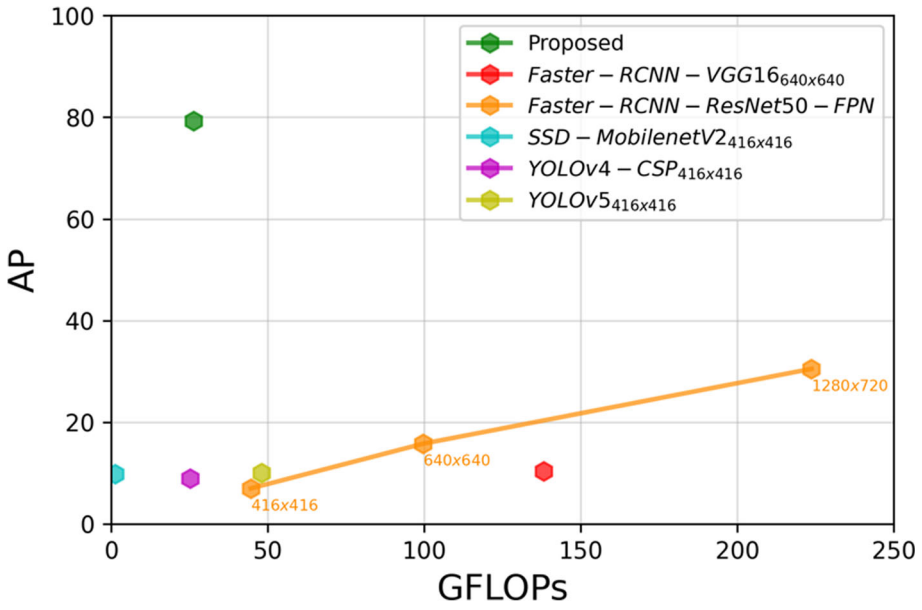
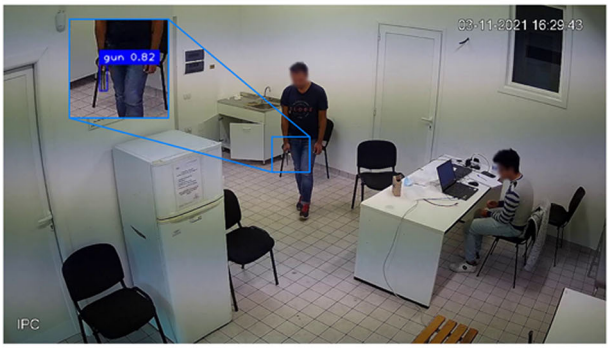



**Fig. 4** Comparison of the complexity-accuracy trade-off in terms of floating point operations (GFLOPs) and Average Precision (AP) for the comparison against the state-of-the art approaches. The yellow values in the chart indicate the image input sizes for the Faster-RCNN-ResNet50-FPN architecture. The proposed approach outperforms the state-of-the-art weapon detectors while having fewer GFLOPs

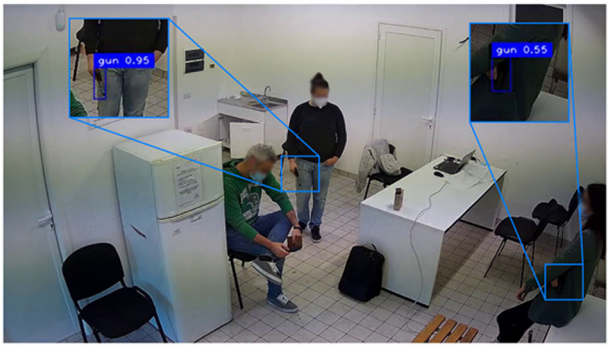
the other hand, the SSD-MobilenetV2 required the smallest amount of GFLOPs (i.e., 1.18), pointing out the lightweight design of the model.

Qualitative results of the proposed approach are shown in Fig. 5. The samples include weapons from both classes (i.e., *knife* in Fig. 5a and *gun* in Fig. 5b and c).


(a)




(b)




(c)

**Fig. 5** Samples of qualitative results. For the sake of clarity, each object detected has been zoomed in to point out both the predicted bounding box and the related classification score. Predicted *gun* and *knife* bounding boxes are highlighted in blue and red, respectively

## 6 Discussion

Automatic handheld weapon detection from CCTV plays a crucial role in preventing crimes and enabling a prompt response by law enforcement agencies. Despite its relevance, the survey of the literature highlighted the lack of effective yet efficient approaches in coping the open challenges in the field such as handling small-object sizes and achieving real-time responses [9] especially in on-the-edge settings. As a first step to solve such issues, the presented work addressed the challenging task of the on-the-edge indoor detection of handguns and knives while keeping near real-time performance.

The proposed double-step approach achieved satisfactory detection results with AP and AP50 equal to 79.30 and 99.60, respectively. The choice of YOLOv4-CSP as weapons detector in the second step allowed to obtain accurate detection with good localization capability, with marginal differences in AP for small and medium-sized objects. The impact of the YOLOv4-CSP as second-step detector is visible from the comparison with the SSD-MobilenetV2[2] approach. In the latter, the SSD-MobilenetV2 detector used in the second step was unable to achieve good localization at higher IoU and also suffered on small weapons detection (APs = 19.40), meaning that the feature extracted from the person's crop were not strong enough to localize challenging objects (e.g., very thin objects, objects with low background contrast). On the other hand, the SSD-MobilenetV2[2] approach achieved the best inference speed thanks to the higher lightness of the SSD-MobilenetV2 with respect to YOLOv4-CSP. Neverthless, the proposed approach still achieved the best speed/accuracy trade-off among the approaches in the ablation study. Its accuracy is also evidenced by the qualitative results, with high confidence in localizing and predicting each correct weapon class. Also, in extremely challenging scenarios (i.e., in Fig. 5c, the vaguely visible *gun* on the right side) the proposed approach localized the weapon, even if with lower confidence compared to other detections. The low confidence in such situations could be attributed to the detection hardness resulting from the low weapon/background contrast. In comparison with YOLOv4-CSP[2], while there is no difference in AP due to the simplicity of the people detection task for both YOLOv4-CSP and SSD-MobilenetV2 models (i.e., in the first step all the people were correctly identified in the frames), the speedup in the proposed approach is given by the use of the lighter model in the first step.

When compared with the state-of-the-art approaches, the proposed one achieved the highest performance. The low performances of [20] could be related to the hardness in the localization of handheld weapons whose size is very small compared to the frame size. In support of such a consideration, the worst metrics of [20] were the APs and the AP75. As regards [9], despite the addition of the FPN module slightly increased the detection ability on the small objects, the low performance paired with the high GFLOPs does not allow the use of the approach in the actual on-the-edge practice. Furthermore, reducing the input size makes the achieved result even worse. The state-of-the-art detectors (i.e., SSD-MobilenetV2 and YOLOv4-CSP) were evaluated at the same input size of the proposed approach and despite the small GFLOPs values highlight the small complexity of the approaches, they obtained very low performance. The poor results may be attributed to the small-sized weapons in the images, which almost disappear when the original frame size (i.e., $1280\times720$) is resized to match the detectors' input size (i.e., $416\times416$). In the proposed approach, thanks to the prior focus on the people, the size of the weapon with respect to the camera FoV does not affects the detection performances so heavily.

A limitation of the proposed approach lies in the dependence of its speed on the number of people in the FoV at the same time (i.e., the second step of the approach process an

image for each detected person), which ensure near real-time performance in non-crowded environments (e.g., home surveillance systems).

## 7 Conclusions

To the best of the authors' knowledge, this work proposes for among the first time in literature a DL-based approach for handgun and knife detection deployable on low-cost SBC devices. The proposed approach obtained satisfactory results in terms of effectiveness in localizing and recognising the weapons in indoor scenarios while achieving near real-time inference speed.

This moves toward the proposal of automatic VSSs both (i) reliable, which would enable their exploitation even without the need for continuous support from human operators, and (ii) able to give real-time responses even with the use of affordable and low-cost computing devices as to promote large-scale distribution, thereby increasing the safety and well-being of people.

Future improvement of the work deals with: (i) the collection of a similar dataset in outdoor scenarios and the testing of the proposed approach on it, (ii) the deployment of the DL approach on other SBC-type devices (e.g., Coral) as to test their efficiency performances, (iii) the use of tracking modules [41, 42] to assess someone's intentions by their moves once a weapon is detected, and (iv) the integration of the video-based extracted data with data coming from different sensing devices (e.g., Passive Infrared [11]) as to increase systems' reliability.

**Funding** Open access funding provided by Università Politecnica delle Marche within the CRUI-CARE Agreement.

**Data Availability Statement** The datasets generated during and/or analysed during the current study are not publicly available due to privacy restrictions but are available from the corresponding author on reasonable request.

## Declarations

**Conflict of interest statement** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.



## References

1. Berardini D, Mancini A, Zingaretti P, Moccia S (2021) Edge artificial intelligence: A multi-camera video surveillance application. In: International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, vol 85437, pp 007–07006. Am Soc Mech Eng

2. Bhangale U, Patil S, Vishwanath V, Thakker P, Bansode A, Navandhar D (2020) Near real-time crowd counting using deep learning approach. Procedia Computer Science 171:770–779
3. Bochkovskiy A, Wang C-Y, Liao H-YM (2020) Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934
4. Cass S (2020) Nvidia makes it easy to embed ai: The jetson nano packs a lot of machine-learning power into diy projects-[hands on]. IEEE Spectr 57(7):14–16
5. Cohen N, Gattuso J, MacLennan-Brown K (2009) CCTV Operational Requirements Manual 2009. Home Office Scientific Development Branch St, Albans, United Kingdom
6. Deng C, Wang M, Liu L, Liu Y, Jiang Y (2021) Extended feature pyramid network for small object detection. IEEE Transactions on Multimedia 24:1968–1979
7. Felzenszwalb PF, Girshick RB, McAllester D, Ramanan D (2010) Object detection with discriminatively trained part-based models. IEEE transactions on pattern analysis and machine intelligence 32(9): 1627–1645
8. Fernandez-Carrobles MM, Deniz O, Maroto F (2019) Gun and knife detection based on faster r-cnn for video surveillance. In: Iberian Conference on Pattern Recognition and Image Analysis, pp 441–452. Springer
9. González JLS, Zaccaro C, Álvarez-García JA, Morillo LMS, Caparrini FS (2020) Real-time gun detection in cctv: An open problem. Neural Netw 132:297–308
10. Grega M, Matiolański A, Guzik P, Leszczuk M (2016) Automated detection of firearms and knives in a cctv image. Sensors 16(1):47
11. Gu Z (2021) Home smart motion system assisted by multi-sensor. Microprocess Microsyst 80:103591
12. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proc IEEE Conf Comput Vis Pattern Recognit, pp 770–778
13. Huang P, Han J, Liu N, Ren J, Zhang D (2021) Scribble-supervised video object segmentation. IEEE/CAA Journal of Automatica Sinica 9(2):339–353
14. Khan WZ, Ahmed E, Hakak S, Yaqoob I, Ahmed A (2019) Edge computing: A survey. Futur Gener Comput Syst 97:219–235
15. Lee Y, Kim J, Willette J, Hwang SJ (2022) Mpvit: Multi-path vision transformer for dense prediction. In: Proc IEEE/CVF Conf Comput Vis Pattern Recognit, pp 7287–7296
16. Lim J, Al Jobayer MI, Baskaran VM, Lim JM, Wong K, See J (2019) Gun detection in surveillance videos using deep neural networks. In: 2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), pp 1998–2002. IEEE
17. Lin T-Y, Dollár P, Girshick R, He K, Hariharan B, Belongie S (2017) Feature pyramid networks for object detection. In: Proc IEEE Conf Comput Vis Pattern Recognit, pp 2117–2125
18. Li Y, Shao M, Fan B, Zhang W (2022) Multi-scale global context feature pyramid network for object detector. SIViP, 1–9
19. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu C, Berg AC (2016) Ssd: Single shot multibox detector. In: European Conference on Computer Vision, pp 21–37. Springer
20. Olmos R, Tabik S, Herrera F (2018) Automatic handgun detection alarm in videos using deep learning. Neurocomputing 275:66–72
21. Olorunshola OE, Irhebhude ME, Evwiekpaefe AE (2023) A comparative study of yolov5 and yolov7 object detection algorithms. Journal of Computing and Social Informatics 2(1):1–12
22. Qiao S, Chen L-C, Yuille A (2021) Detectors: Detecting objects with recursive feature pyramid and switchable atrous convolution. In: Proc IEEE/CVF Conf Comput Vis Pattern Recognit, pp 10213–10224
23. Rahmaniar W, Hernawan A (2021) Real-time human detection using deep learning on embedded platforms: A review. Journal of Robotics and Control (JRC) 2(6):462–468
24. Ramachandran P, Parmar N, Vaswani A, Bello I, Levskaya A, Shlens J (2019) Stand-alone self-attention in vision models. Advances in neural information processing systems 32
25. Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: Unified, real-time object detection. In: Proc IEEE Conf Comput Vis Pattern Recognit, pp 779–788
26. Redmon J, Farhadi A (2017) Yolo9000: better, faster, stronger. In: Proc IEEE Conf Comput Vis Pattern Recognit, pp 7263–7271
27. Redmon J, Farhadi A (2018) Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767
28. Ren S, He K, Girshick R, Sun J (2015) Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems 28
29. Sandler M, Howard A, Zhu M, Zhmoginov A, Chen L-C (2018) Mobilenetv2: Inverted residuals and linear bottlenecks. In: Proc IEEE Conf Comput Vis Pattern Recognit, pp 4510–4520
30. Tong K, Wu Y (2022) Deep learning-based detection from the perspective of small or tiny objects: A survey. Image Vis Comput 104471

31. Tulbure A-A, Tulbure A-A, Dulf E-H (2022) A review on modern defect detection models using dcnns-deep convolutional neural networks. J Adv Res 35:33–48
32. Uijlings JRR, Van De Sande KEA, Gevers T, Smeulders AWM (2013) Selective search for object recognition. Int J Comput Vis 104(2):154–171
33. United Nations Office on Drugs and Crime - UNODC (2019) Global Study on Homicide 2019. Vienna, https://www.unodc.org/unodc/en/data-and-analysis/global-study-on-homicide.html. Accessed on 29 Jan 2022
34. Verma GK, Dhillon A (2017) A handheld gun detection using faster r-cnn deep learning. In: Proceedings of the 7th International Conference on Computer and Communication Technology, pp 84–88
35. Wang C-Y, Bochkovskiy A, Liao H-YM (2021) Scaled-yolov4: Scaling cross stage partial network. In: Proc of the IEEE/cvf Conf Comput Vis Pattern Recognit, pp 13029–13038
36. Wang C-Y, Bochkovskiy A, Liao H-YM (2022) Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. arXiv preprint arXiv:2207.02696
37. Wang W, Dai J, Chen Z, Huang Z, Li Z, Zhu X, Hu X, Lu T, Lu L, Li H et al (2022) Internimage: Exploring large-scale vision foundation models with deformable convolutions. arXiv:2211.05778
38. Yadav P, Gupta N, Sharma PK (2022) A comprehensive study towards high-level approaches for weapon detection using classical machine learning and deep learning methods. Expert Syst Appl 118698
39. Zhang D, Han J, Yang L, Xu D (2018) Spftn: A joint learning framework for localizing and segmenting objects in weakly labeled videos. IEEE transactions on pattern analysis and machine intelligence 42(2):475–489
40. Zhang H, Li F, Liu S, Zhang L, Su H, Zhu J, Ni LM, Shum H-Y (2022) Dino: Detr with improved denoising anchor boxes for end-to-end object detection. arXiv preprint arXiv:2203.03605
41. Zhao J, Dai K, Wang D, Lu H, Yang X (2020) Online filtering training samples for robust visual tracking. In: Proceedings of the 28th ACM International Conference on Multimedia, pp 1488–1496
42. Zhao J, Dai K, Zhang P, Wang D, Lu H (2022) Robust online tracking with meta-updater. IEEE Transactions on Pattern Analysis and Machine Intelligence



## Authors and Affiliations

**Daniele Berardini[1] · Lucia Migliorelli[1] · Alessandro Galdelli[1] · Emanuele Frontoni[2] · Adriano Mancini[1] · Sara Moccia[3]**

Lucia Migliorelli
l.migliorelli@univpm.it

Alessandro Galdelli
a.galdelli@univpm.it

Emanuele Frontoni
emanuele.frontoni@unimc.it

Adriano Mancini
a.mancini@univpm.it

Sara Moccia
sara.moccia@santannapisa.it

[1] Department of Information Engineering, Università Politecnica delle Marche, Ancona, Italy

[2] Department of Political Science, Communication and International Relations, Università degli Studi di Macerata, Macerata, Italy

[3] Department of Excellence in Robotics and AI, Scuola Superiore Sant'Anna, Pisa, Italy