Expert Systems With Applications 212 (2023) 118698

Contents lists available at ScienceDirect

Expert Systems With Applications

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

Review

A comprehensive study towards high-level approaches for weapon detection using classical machine learning and deep learning methods

Pavinder Yadav, Nidhi Gupta *, Pawan Kumar Sharma

Department of Mathematics and Scientific Computing, National Institute of Technology, Hamirpur, Himachal Pradesh, 177005, India

ARTICLE INFO

Keywords:
Weapon detection
Deep learning
Machine learning
Computer vision
Security and surveillance

ABSTRACT

Surveillance systems do not give a rapid response to deal with suspicious activities such as armed robbery in public places. Consequently, there is a need for technology that can recognize criminal activities from Closed Circuit Televisions (CCTV) footage without the need of human help. Various high-performance computing algorithms have been developed but are limited to specific conditions. In this paper, we have identified gaps between existing technologies for weapon detection. The automatic detection of guns/weapons could help in the investigation of crime scenes. A new and difficult area of study is identifying the specific type of firearm used in an attack known as intra-class detection. The study examines and classifies the strengths and shortcomings of several existing algorithms using classical machine learning and deep learning approaches, employed in the detection of different kinds of weapons. We have thoroughly compare and analyze the performance of several recent state-of-the-art methods on different datasets along with their future scope. We observed that deep learning techniques beat traditional machine learning techniques in terms of speed and accuracy.

1. Introduction

Nowadays, Closed Circuit Televisions (CCTVs) are widely being used in society to prevent crimes and identify suspicious activities. With the fast development of CCTV cameras, inspecting and analyzing them becomes more difficult for a human operator, and taking any necessary action based on the video input from the remote camera. When several people are involved in video input, it becomes expensive and ineffective. According to several studies, human operators develop video blindness and tend to miss up to 95% of the screen action after 20 to 40 min of intensive monitoring (Velastin et al., 2006). This overall results in a significant quality reduction and poor productivity, which leads to an inaccurate detection rate of up to 83% (Ainsworth, 2002). Researchers have developed a number of computer vision-based automatic weapon detection systems in response to the proliferation of high-powered computers and the availability of high-speed internet. Object detection, in particular, has become a demanding study field in the last decade, and it has been used in a variety of applications including foreground moving targets detection (Minaeian et al., 2018), human activity recognition (Singh & Vishwakarma, 2019), marine surveillance (Jeong et al., 2018), pedestrian identification (Jin et al., 2016), weapon detection (Olmos et al., 2018), and many more. Some merely need to identify items that take up a significant portion of the scene, while others require the detection of many objects of different sizes. The outcomes vary according to the size of the object, with small objects having poorer outcomes compared to large objects. These findings are well reflected in the challenges like ImageNet (Deng et al., 2009), Common Objects in COntext (COCO) (Lin et al., 2014), and open images (Kuznetsova et al., 2020), where small objects are observed with 38% less accuracy in detection than larger objects. The reason is that small objects have fewer pixels on the image, which means that they do not show up very often, either because they are not labeled or because they are not well represented in the training phase of the process.

The crime rate is relatively high in nations where a person has access to armory (e.g. a pistol). Information from many sources revealed the effects of criminal actions, which ranged from murder to theft, resulting in the loss of valuable lives, infrastructure devastation, and disrupted law and order circumstances. Standard CCTV cameras are used for monitoring at certain places, but the monitoring process is very laborious. Nowadays, guns are available in a wide range of styles and sizes, which makes them difficult to identify in real-time. In this situation, deep learning ushered in a breakthrough. However, with time, researchers have created various models to identify weaponry using deep learning methods. The detection of weapons has become

* Corresponding author.
E-mail addresses: pavinder_phdmath@nith.ac.in (P. Yadav), nidhi@nith.ac.in (N. Gupta), psharma@nith.ac.in (P.K. Sharma).

https://doi.org/10.1016/j.eswa.2022.118698
Received 5 March 2022; Received in revised form 2 August 2022; Accepted 24 August 2022
Available online 31 August 2022

a difficult task despite the existence of several advanced state-of-the-art deep learning techniques. A thorough examination of several techniques has been carried out in this article.

Detection of small objects in aerial and satellite images has been previously addressed by modifying the architecture of the network (Sommer et al., 2017), data augmentation for small objects (Tong et al., 2020), or adding perceptual generative adversarial networks for better image resolution (Li et al., 2017). Although these techniques observed higher precision in small object detection, they were limited to certain applications such as traffic signs or satellite images only.

Earlier, the object detection procedure was divided into three phases: (i) generating proposals, (ii) extracting feature vectors, and (iii) classifying regions. The main objective was to find a zone of interest in an image that might include objects of any size. For safeguard information, multiple scales were used to reduce the size of the input images and the multi-scale windows curve was employed for transition between the images. The second step was to get a feature vector of fixed length from the sliding window in order to secure specific information about the area enclosed. Low-level visual descriptors such as the Harris Corner Detector (Harris & Stephens, 1988), Histogram of Gradients (HOG) (Dalal & Triggs, 2005), Scale Invariant Feature Transform (SIFT) (Lowe, 1999), or Speeded Up Robust Features (SURF) (Bay et al., 2006) were used to encode feature vectors, which exhibited fitness to scale, illumination, and rotational variance. Finally, the area classifiers were trained to assign labels per different categories to the covered regions in the third phase. Because of their high performance on small-sized training data, Support Vector Machines (SVM) (Hearst et al., 1998) were commonly used. Additionally, in the classification stage, various classification approaches such as bagging (Opitz & Maclin, 1999), cascade learning (Dalal & Triggs, 2005), and Adaboost (Freund & Schapire, 1996) were utilized, resulting in better detection accuracy.

In particular, Deep Convolutional Neural Networks (DCNNs) have outperformed all other machine learning approaches in object detection in the last few years. It takes a lot of work to find higher level features in data with traditional methods (Guo et al., 2016). DCNNs models do this automatically. Convolutional layers, non-linear activation functions such as ReLU, pooling layers, and fully-linked layers make up DCNNs. The convolutional layers extract a variety of characteristics from the source images. Following that, fully connected layers learn from these characteristics. Another advantage of utilizing such a design is that it may be reused partially or fully for similar applications. It is possible because of the theory of transfer learning, which cuts down on model building time and eliminates the need for a large dataset.

Object detection models based on deep learning methods are majorly divided into two groups: (i) one-stage detectors like You Only Look Once (YOLO) (Redmon et al., 2016) and its versions like YOLOv2 (Redmon & Farhadi, 2017), and YOLOv3 (Redmon & Farhadi, 2018), and (ii) two-stage detectors like Region-based Convolutional Neural Network (R-CNN) (Girshick et al., 2014) and its versions like Fast R-CNN (Girshick, 2015), and Faster R-CNN (Ren et al., 2017). Without the need for a cascading area classification phase, one-stage detectors produce categorical predictions of items on each position of the feature maps. Two-stage detectors start with a proposal generator that generates a small number of proposals and extracts characteristics from each one, and then use region classifiers to predict the suggested category of the object. One-stage detectors are substantially more time-efficient and have more applicability in real-time identification, but two-stage detectors produce better results on public benchmark datasets. In this study, we cover the fundamental concepts of major approaches and review each of these methods in a methodical manner.

For a few decades, researchers have been striving to develop an automatic weapon detection system based on computer vision algorithms. In a potentially perilous situation, that person carries a knife or another firearm in his hand rather than any other body parts. It is needless to say that, normally, guns can only be operated by hand while committing any crime. Therefore, the vision system is expected to be trained to access the ideal weapon image or a form that is comparable to that weapon. The major goals of such a detection system are as follows:

- To create an automatic alarm system that can alert surveillance security personnel in real-time, resulting in a quick response; and
- To classify different types of weapons, which can provide a crucial information for forensic investigation.

Deep learning revolutionized the creation of weapon detection systems. In this regard, several studies have been conducted and various models have been developed to identify firearms.

The main objective of this comprehensive study is to identify research gaps in the field of weapon detection and identification and to thoroughly study existing datasets, their limitations, and future research directions. This article represents the huge number of contributions of a significant articles in a structured and systematic manner. This survey can provide readers with a comprehensive understanding of weapon detection using deep learning, as well as perhaps drive future research efforts on weapon detection approaches and their benefits. Overall, this article examines the strengths and shortcomings of the various existing approaches, and offers a detailed assessment of the open issues with forecasting of future prospects.

The paper is organized into seven sections. Section 2 provides a detailed description of existing publicly available datasets. Classical machine learning approaches adopted for weapon detection are discussed in Section 3. In Section 4, deep learning approaches are described in detail. Furthermore, the comparative analysis of various classical machine learning and deep learning methods is discussed in Section 5. The key contributions of the study are highlighted in Section 6. The future extents related to the real-time weapon detection methods are given in Section 7.

2. Publicly available datasets

Table 1 shows the public datasets that can be used to classify and recognize weapons, providing the year of publication, the number of images in each dataset, the resolution of images or videos, and the types of used weapons.

2.1. IMFDBs

The Internet Movie Firearm Database contains a huge picture collection of weapons. It is a powered-wiki-managed online repository that is publicly available at the site IMFDBs. It comprises about 4,50,000 pictures of the weapon, some of which are displayed in Fig. 1. Several thousand images from the movie sequences or video games are used to create the database, which reflects the limited number of pictures with close-up views of weapons. The weapons that were obliterated by darkness or made unseeable due to blurriness or size are also included here. IMFDBs is an excellent dataset for firearms since it has a wide range of gun pictures in various unconstrained orientations and positions.

2.2. Knives images database

This database (Grega et al., 2016) comprises 12,899 pictures of knives, which are classified into two categories: (i) positive examples containing 3559 images, and (ii) negative examples containing 9340 images. The images containing a knife shown in Fig. 2(a) belong to the positive class, and the images shown in Fig. 2(b) belong to negative examples considering all circumstances. It is considered that a knife that is not being wielded by a person is less hazardous. It can also be overlooked during processing, resulting in a slew of false alarms. Because carrying knives in public is banned in Poland, the photographs were taken either indoors or via vehicle windows. All images in this database have a resolution of 100×100 pixels.

Table 1
Statistics of datasets that are publicly available.

Database	Publication/site	Year	# Images/videos	Resolution	Types of weapon
IMFDBs	IMFDBs	...	4,50,000[a]	Variable size	Handguns, rifles, knives
Knives images database	Grega et al. (2016)	2016	12,899	100 × 100	Knives
Gun movies database	Grega et al. (2016)	2016	7 Videos	640 × 480	Guns
Dataset of Olmos et al.	Olmos et al. (2018)	2018	3000	Variable size	Guns
Sohas_weapon	Pérez-Hernández et al. (2020)	2020	17,684	Variable size	Guns, knives
Dataset created by mock attack	González et al. (2020)	2020	5149	1920 × 1080	Handguns and rifles
Synthetic dataset	González et al. (2020)	2020	2500	1920 × 1080	Guns
ITU firearm dataset	Iqbal et al. (2021)	2021	10,973	480 × 800	Guns

[a]Approximate.

Fig. 1. Sample images from IMFDBs database (IMFDBs).

(a) (b)

Fig. 2. Knives images database (a) Positive samples (b) Negative samples (Grega et al., 2016).

2.3. Database of firearm videos

The Gun Movies Dataset is a video collection recorded by surveillance cameras. Grega et al. (2016) created this dataset by simulating a gun-shooting situation due to the lack of real-life gun-shooting footage. As a result, this dataset comprises CCTV footage with an actor and seven different video recordings. The training and testing sets were identical in size, with each recording lasting for 8.5 minutes and yielding roughly 12,000 frames. Sixty percent of each set contains negative examples that do not include a handgun but other objects in a hand, while the remaining forty percent contains positive examples that include a firearm visible to an observer. A few images from this dataset are shown in Fig. 3.

2.4. Dataset of Olmos et al.

Olmos et al. (2018) developed two databases for knives and handguns. The knife dataset contains 12,869 images. Each image contains multiple variations of cold steel weapons of various sorts, forms, colors, sizes, and materials, placed near and far from the camera, partially occluded by the hand or anything else. There are three sets of databases, one with 102 classes and 9261 images, which is suitable for the classification tasks. The second dataset comprises 3000 images of weapons with extensive contextual information that may be used to make detection. The third dataset comprises 608 images, with 304 of them being handgun images. This dataset may be used for classification as well as detection tasks. Some of the sample images are shown in Fig. 4(a).

2.5. Sohas_weapon

The authors of Pérez-Hernández et al. (2020) created a dataset called Sohas_weapon to investigate six tiny objects that are frequently handled in a similar manner to a weapon, namely handguns, knives, smartphones, bills, handbags, and cards. They employed a variety of surveillance cameras to capture images. Among them, 10% of the images were obtained from web sources. For detection, all of these images were manually annotated. The dataset contains different types of knives with different shapes. A few images of the dataset are shown in Fig. 4(b).

Fig. 3. Frames extracted from Gun movie dataset of Grega et al. (2016).

2.6. Dataset created by mock attack and synthetic dataset

During a simulated assault, this dataset was collected and manually annotated by González et al. (2020). The infrastructure of the dataset is made up of three security cameras that are strategically placed at the same location to cover two distinct pathways and one entrance, generating diverse situations. A total of 5149 frames were retrieved from movies at a rate of two frames per second (607, 3511, and 1031 frames from cam1, cam5, and cam7, respectively). In addition, they constructed a fictitious dataset by simulating a section of a city and an educational facility within it using the Unity Game Engine. Several cameras capture the motions of eleven distinct models and seven animations that make up the cast of characters. The images add eleven distinct items to the produced datasets: four different pistols, five different rifles, a knife, and a smartphone. The creation of synthetic data might assist the network focus on the item to be discovered. Therefore, this dataset is not realistic. Fig. 5(a) shows an example of the dataset collected by mock attack. Fig. 5(b) illustrates a few sample images shown from synthetic dataset.

2.7. ITU Firearms Dataset (ITUF)

The ITUF dataset (Iqbal et al., 2021) contains images of guns and rifles in a variety of settings, including being targeted, laid out on tables, transported, or stored in racks. Web scraping was used to acquire the images for the dataset. Weapons, battles, pistols, film titles, firearms, firearm variations, shooters, corps, guns, and rifles were among the phrases used in the dataset. The data-driven algorithms can overcome garment variances, body posture variations, weapon position and size variations, changing light conditions, and both indoor and outdoor situations. These mentioned variables result in a strong prior for the data-driven algorithms. Images were eliminated from the findings, which were not associated with weaponry, as well as cartoons and duplicates. There are 10,973 completely annotated firearm images in the final clean collection, with 13,647 firearm occurrences. Fig. 6 depicts a few sample images from the dataset.

3. Classical machine learning methods

This section covers the detailed description of algorithms that use classical machine learning approaches (Haralick & Shapiro, 1985). Some of these approaches are listed in Table 2. Edge detection is the process of identifying parts of an image where objects could be distinguished from others. The edge detection has the advantage of reducing the quantity of data required to analyze the image. Edge detection works effectively with sharp images, while noisy images enhance complexity and make it a challenging task to be resolved.

3.1. Active appearance models (AAMs)

A statistical model of the form and pixel intensities (texture) throughout the object can be expressed as an Active Appearance Models (AAMs) (Cootes et al., 1998) in general. The phrase *appearance* refers to the combination of form and texture, while *active* refers to the employment of an algorithm to match the shape and texture model in fresh images. Objects of interest are manually tagged with so-called landmark points in the images from the training set to characterize the form during the training phase. The algorithm may be divided into four stages:

1. *Select a starting reference shape.*
2. *Align the reference shape with all other forms.*
3. *Recalculate the aligned forms mean shape.*
4. *If the mean form distance from the reference shape exceeds a threshold, set the mean shape as the reference shape and return to step 2. Otherwise, the mean shape should be returned.*

3.2. Harris corner detector

It is a corner detection technique commonly utilized in computer vision algorithms to extract corners and infer image features in detail (Harris & Stephens, 1988). Instead of applying shifting patches for every 45° angle, the Harris corner detector takes into account the differential of the corner values with respect to directions directly, allowing it to more effectively distinguish between edges and corners. The concept is to imagine a tiny window surrounding each pixel in the image. By moving each window, a minimal fraction in a specific direction, it is possible to determine the amount of change in the values of pixels. Eq. (1) expresses the change function $E(m,n)$ as the total sum of all squared differences,

$$E(m,n) = \sum_{x,y} w(x,y)[I(x+m,y+n) - I(x,y)]^2 \tag{1}$$

where m,n are the pixels x,y are the coordinates, and I is value of the intensity of pixels in 3×3 window. A feature of the image is regarded to be any pixels with high $E(m,n)$ values, as determined by a threshold value in the image. For corner detection, we must maximize the function value $E(m,n)$. Subsequently, the second term should also be maximized. Eq. (2) is determined by applying Taylor Series Expansion to Eq. (1) and then performing a number of mathematical operations on the result.

$$E(m,n) \approx [m,n]P \begin{bmatrix} u \\ v \end{bmatrix} \tag{2}$$

where,

$$P = \sum w(x,y) \begin{bmatrix} I_x^2 & I_xI_y \\ I_xI_y & I_y^2 \end{bmatrix} \tag{3}$$

where, I_x and I_y are the derivatives of the image in the (x,y) directions, respectively. The corner response has been calculated by Eq. (4)

$$Q = det(P) - c\left(trace(P)^2\right) \tag{4}$$

where, c is an constant and α and β are eigenvalues of P. As a consequence, as seen in Fig. 7, the eigenvalues indicate whether an area is a corner, a flat surface, or an edge.

- *When $|Q|$ is small, which occurs when α and β are small, the area is flat.*
- *When $Q < 0$, the region is considered to be an edge, which happens when $\alpha \gg \beta$ and vice versa.*
- *When Q is large, the area is considered a corner, which happens when α and β are both large and $\alpha \sim \beta$.*

(a) (b)

Fig. 4. A few samples from the (a) Dataset of Olmos et al. (2018) (b) Sohas_weapon dataset (Pérez-Hernández et al., 2020).

(a) (b)

Fig. 5. (a) Dataset collected during mock attack (b) A few sample images from synthetic dataset (González et al., 2020).

Fig. 6. Sample images from ITU Firearm Dataset (Iqbal et al., 2021).

Table 2
List of classical machine learning approaches and respective area of application.

Method	References	Year	Area of application
Active Appearance Models (AAMs)	Cootes et al. (1998)	1988	Knives detection
Harris corner detector	Harris and Stephens (1988)	1988	Knives and gun detection
Scale Invariant Feature Transform (SIFT)	Lowe (1999)	1999	Knives and gun detection
Haar Cascades	Viola and Jones (2001)	2001	Knives detection
Speeded Up Robust Features (SURF)	Bay et al. (2006)	2006	Knives and gun detection

3.3. Scale Invariant Feature Transform (SIFT)

Scale Invariant Feature Transform (SIFT) (Lowe, 1999) extracts a high number of distinct possible key-points from an image which are invariant to various perspectives, scaling, rotation, light variations, and noise. Fig. 8 shows the architecture of the SIFT algorithm.

There are basically four stages in this algorithm:

- Scale-space detection: *The input image is scanned to find the regions of interest that are either local maxima or local minima and invariant to transformations like rotation and scaling.*
- Key-point localization: *In this phase, the most stable key-points are identified and poor contrast and outliers are eliminated from the key-points. Outliers, low-contrast pixels, and poorly located key-points along an edge are all removed using the taylor series.*

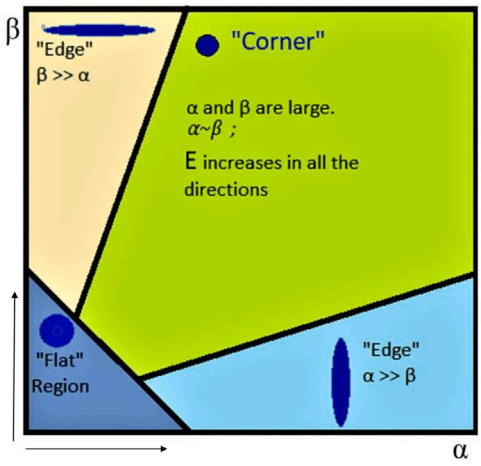

Fig. 7. Classification of image point using eigenvalue of Q.

- Orientation: *After finding the most stable key points in phase one, the local image gradient direction is used to assign the orientation of each key-point to it.*
- Key-points description: *Suitable feature points are described as stored intensity samples in the neighborhood in the final phase. All of these key-points are unique and unaffected by affine transformations or changes in illumination.*

3.4. Speeded Up Robust Features (SURF)

Speeded Up Robust Features (SURF) (Bay et al., 2006) is a feature extraction method that operates in a similar way to SIFT but is significantly faster. The SURF method is a reliable detector and description of prospective key-points of interest. The architecture of the SURF algorithm is shown in Fig. 9.

The SURF detector makes use of the Hessian Matrix to discover interest key-points. The Hessian Matrix (H) is quick and accurate calculation method. The Hessian Matrix (H) for $x = (x, y)$ is expressed by Eq. (5):

$$\mathrm{H}(x,\sigma) = \begin{bmatrix} D_{xx}(x,\sigma) & D_{xy}(x,\sigma) \\ D_{xy}(x,\sigma) & D_{yy}(x,\sigma) \end{bmatrix} \tag{5}$$

where, the Gaussian second order derivative is convolutioned with the integral image I yields $D(x, \sigma)$.

SURF descriptor: Interest key-points are characterized by using Haar wavelet responses to assign orientation to each key-point. A square area is built around each key-point for the description of key-points. The chosen square region is then subdivided into 4×4 subregions. After that, the four descriptor components d_x, d_y, $|d_x|$, and $|d_y|$ are assessed, d_x represents the horizontal haar wavelet response, whereas d_y represents the vertical Haar wavelet response. $|d_x|$ and $|d_y|$ are both absolute values of horizontal and vertical directions, respectively.

3.5. Haar Cascades

Viola and Jones (2001) presented the Haar Cascades detector, which is a successful fusion of three fundamental principles. To begin, a large collection of characteristics is needed that can be calculated in a short and consistent time. This feature-based strategy reduces in-class variability while increasing inter-class variability. Secondly, using a boosting method allows the salient features to be selected and the classifier to be trained at the same time. Afterward, a quick and efficient detection method is made possible by building a chain of more complex classifiers.

As illustrated in Fig. 10, the method employs edge and line detection features as well as center–surround features. At each level of the procedure, the number of the features employed to evaluate the image increases. With only 200 basic features, Wilson and Fernandez (2006) were able to recognize a human face with 95% accuracy rate.

4. Deep learning methods

Deep neural networks are used in deep learning algorithms, such as Faster R-CNN (Ren et al., 2017), Single Shot multibox Detector (SSD) (Liu et al., 2016), and YOLO (Redmon et al., 2016). Fig. 11 depicts the key advancements and achievements in deep learning-based object detection algorithms from the year 2012. One of the most significant benefits of deep learning algorithms is that they do not require hand-crafted features such as edge and corner detection. During the training phase, these algorithms learn characteristics on the fly. Therefore, these algorithms require a large quantity of data to be trained. However, a large amount of data may also be used to detect covered objects. The data must be labeled beforehand as training for SSD, Faster R-CNN, and YOLO. It is also sometimes referred to as supervised learning. Several deep learning methods are given in Table 3 with their advantages and shortcomings.

4.1. Backbone networks

Object detectors based on deep neural networks make use of backbone networks to extract high-level information from input images. Deep neural networks are most commonly used as image classifiers to perform on large-scale image classification datasets like the ImageNet classification dataset. In most image classifiers, the final classification layers are eliminated, and the remaining layers are employed as backbone networks, and further detection layers are added to the backbone networks to construct comprehensive object detectors. The major design goals of backbone networks are to increase detection accuracy and processing efficiency. The following are some of the most widely used backbone networks:

- VGGNets (Simonyan & Zisserman, 2014) with convolutional layers that employ tiny filters of 3×3 pixels, followed by 2×2 max pooling. VGG-16 contains thirteen convolutional layers, whereas VGG-19 has sixteen convolutional layers. VGG was the winner of the ImageNet Challenge in the year 2014, and it is still one of the most popular networks today.
- Residual networks (ResNets) (He et al., 2016) were presented as a way to train very deep networks using residual blocks. Residual networks come in a variety of shapes and sizes. ResNet50 and ResNet101 are the most popular variants. ResNet is substantially more comprehensive than VGGNet.
- Inception networks (Szegedy et al., 2015, 2016) that boosted network size and scope without adding to the computational costs. Convolution layers of 1×1, 3×3, and 5×5 filter sizes, as well as max pooling layers, are layered in parallel in the Inception module. Many scales of features may be retrieved at the same time in a single layer. VGGNet is substantially slower than Inception networks.
- DenseNet (Huang et al., 2017) is a network in which each layer is densely linked to all other levels in a forward manner, allowing all later layers to utilize lower level characteristics. The vanishing-gradient problem can be solved with DenseNet.
- The ZFNet (Zeiler & Fergus, 2014) is a classic convolutional neural network. The design was inspired by showing intermediate feature layers and the operation of a classifier. The filter widths and strides of the convolutions are comparatively shorter than in several previous architectures.

4.2. Two-stage detectors

Broadly, detectors are divided into one-stage detectors and two-stage detectors. In this section, two-stage detector models are discussed in detail as below.

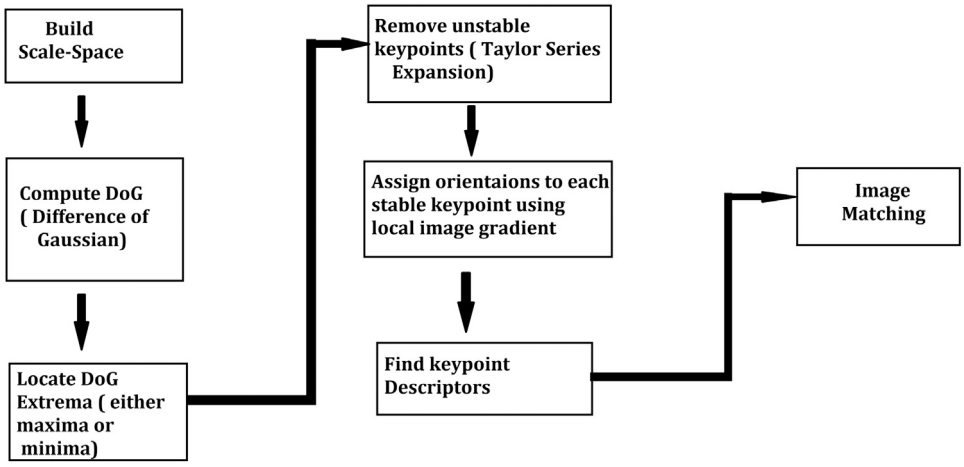

Fig. 8. Architecture of SIFT (Lowe, 1999).

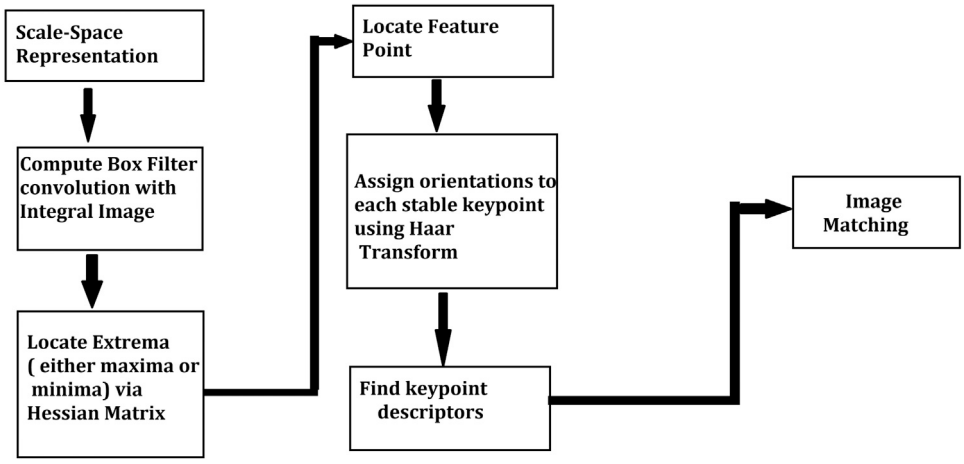

Fig. 9. Architecture of SURF (Bay et al., 2006).

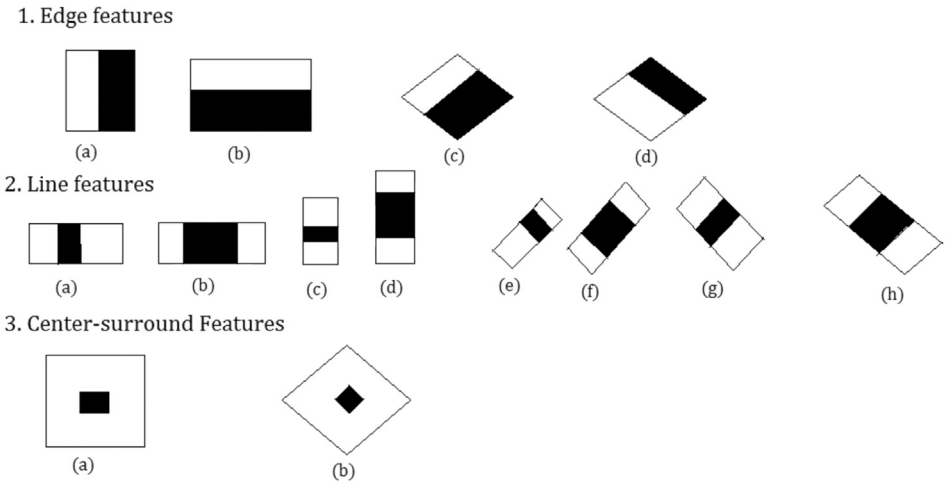

Fig. 10. Haar-like feature employs edge or line detection characteristics (Wilson & Fernandez, 2006).

4.2.1. R-CNN

One of the most significant drawbacks of the conventional method based on the sliding window method is that it reads every available portion of the image. Within the image, the object of interest might be in multiple spatial positions and have different aspect ratios. This will necessitate the selection and processing of a large number of areas and hence increase the processing time. R-CNN (Girshick et al., 2014) resolves this issue by employing a selective search approach. This technique generates 2000 region suggestions, commonly known as "region extraction". There are 4096-dimensional feature vectors generated by warping the areas into squares and forwarding them to a convolutional neural network. These characteristics are passed on to

Table 3
Highlights and shortcomings of deep learning methods using one–two stage approaches.

Method	Publication	Approach	Highlights and shortcomings
R-CNN	Girshick et al. (2014)	Two-sage	**Highlights:** Significant improvement in performance over previous state-of-the-art methods; The first method which combine CNN and RP methods **Shortcomings:** Training is costly in terms of both space and time; Testing is time-consuming
Fast R-CNN	Girshick (2015)	Two-stage	**Highlights:** Create a layer for ROI pooling; First method which enables training to end-to-end detector (ignoring RP generation) **Shortcomings:** The new bottleneck is revealed to be external RP computation; For real-time applications, it is still too slow
Faster R-CNN	Ren et al. (2017)	Two-stage	**Highlights:** Instead of selective search, propose RPN for producing nearly high-quality and cost-free RP; By sharing convolution layers, combine Fast RCNN and RPN into a single network; Introduce multiscale anchor boxes and translation invariant as RPN references **Shortcomings:** It is not a simplified procedure; Training is complex; For real-time applications, it is still too slow
YOLO	Redmon et al. (2016)	One-stage	**Highlights:** The first highly effective unified detector; YOLO can run at 45 FPS; Drop the process of RP completely; Framework for detection that is both efficient and elegant; Dramatically faster than previous detection techniques **Shortcomings:** Localization of small object is difficult; The accuracy of the detector falls far short of that of previous detectors
SSD	Liu et al. (2016)	One-stage	**Highlights:** To detect objects using convolution layers of multi-scale, it effectively combines YOLO and RPN ideas; First efficient and accurate unified detector; Can run at 59 FPS; Faster and significantly more accurate than YOLO **Shortcomings:** It is ineffective at detecting objects of small size
FPN	Lin et al. (2017)	Two-stage	**Highlights:** Superior to Faster-RCNN while maintaining high accuracy; Using a bank of specialized convolution layers, create a set of position sensitive score maps **Shortcomings:** For real-time applications, it is still too slow; Training is not a simplified process
YOLOv2	Redmon and Farhadi (2017)	One-stage	**Highlights:** It employs a variety of existing strategies to boost both accuracy and speed; Propose a faster DarkNet-19; In real time, YOLOv2 can identify over 9000 object classes **Shortcomings:** It is ineffective at detecting objects of small size
YOLOv3	Redmon and Farhadi (2018)	One-stage	**Highlights:** YOLOv3 improves detection object accuracy by referring to the concept of residual network; It employs darknet-53 to generate a feature maps of small-size **Shortcomings:** it is ineffective at detecting objects of small size
YOLOv4	Bochkovskiy et al. (2020)	One-stage	**Highlights:** Backbone of the model employs Bag-of-Specials (BoS) and Bag-of-Freebies (BoF), which improves performance with Cross Stage Partial Darknet-53; Neck has improved Spatial Pyramid Pooling, resulting in output of fixed-size independent of size of input **Shortcomings:** Training is not a streamlined process; it is ineffective at detecting objects of small size

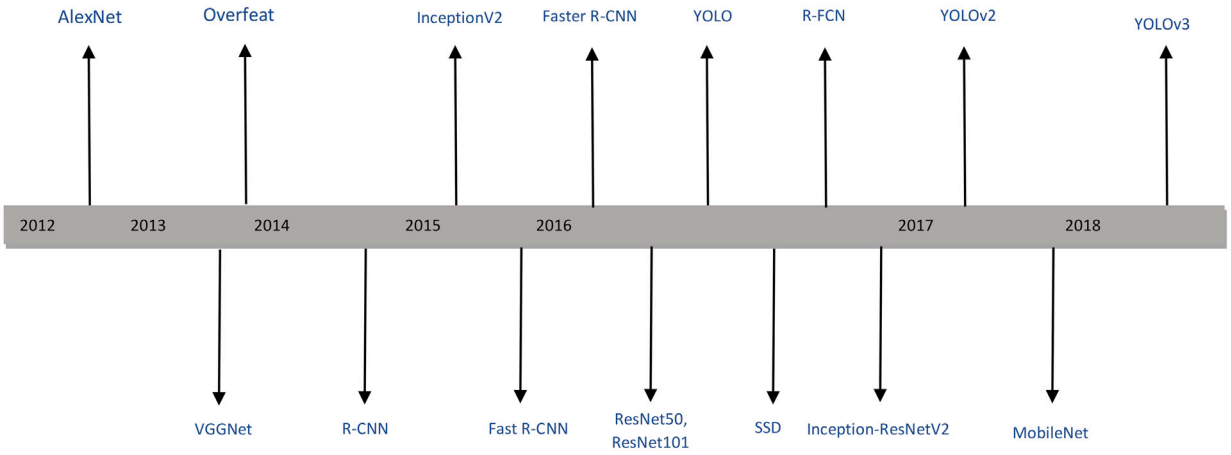

Fig. 11. Milestones in object detection using deep convolutional neural networks.

SVM, which categorizes the areas at the final stage. It also uses a regression technique to determine the bounding boxes of the categorized objects detected in the image. The drawbacks of this method include the fact that it takes a long time to train because it must categorize 2000 areas for each image. It is difficult to use in real time because each image takes around 47 seconds to process.

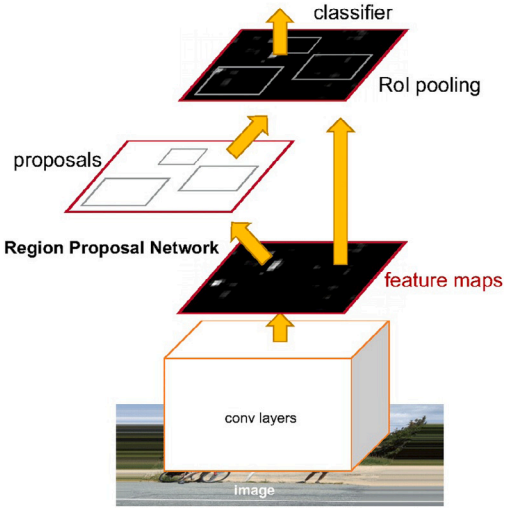

Fig. 12. Architecture of Faster R-CNN (Ren et al., 2017).

4.2.2. Fast R-CNN

Fast R-CNN (Girshick, 2015) addresses the issue of R-CNN and develops a significantly faster algorithm. The steps are similar to R-CNN, but instead of calculating the areas, the image is sent directly to CNN, which generates feature maps. The area proposals are detected and, using the Region-Of-Interest (ROI) pooling layer, they are warped into squares. Using this convolutional feature map, the shape is converted to a fixed size and transmitted to the fully connected layer. A softmax layer is used to predict the class and bounding box using the ROI feature vector. This method is considerably faster than R-CNN, since it does not create 2000 suggested regions.

4.2.3. Faster R-CNN

In the Faster R-CNN (Ren et al., 2017) configuration, there are two phases. The feature map of the original image is created in the first stage using feature extraction (VGG, ResNet, Resnet-V2, Inception, etc.). The feature map from a chosen in-between convolutional layer is used by the Region Proposal Network (RPN) to predict proposal areas with objectness scores and locations. Time-saving software is used to make a score that approximates the chance that a thing will be an object or not. Box regressions are also done for each of the proposals, using a robust loss function.

The second step uses ROI pending to crop features from the same intermediate feature map in order to determine the position of the proposed areas. The regional feature map for each proposed region is given to the remainder of the network to forecast the less specific score and improve the box location. Using this technique, it is possible to skip entering each proposed region into the front-end CNN in order to calculate the regional feature map. However, each proposed region must be entered individually into the database of the network. As a result, the speed of detection is proportional to the number of RPN proposal areas. The architecture of the Faster R-CNN is shown in Fig. 12.

4.2.4. Feature Pyramid Network (FPN)

The dilemma was addressed by the Feature Pyramid Network (FPN) (Lin et al., 2017), which determined that bottom-level feature maps contain spatial information rather than semantic information. Also, later layers of a deep neural network contain high-level semantic information rather than spatial information. FPN used CNN's network structure to create a bottom-up and top-down path with wide extent. A CNN was utilized to process an input image in the bottom-up section, and a pooling layer was employed to reduce the size of feature maps. The extracted features were up-sampled in the top-down section to the same size as in the bottom-up section. FPN created integrated image features that boost detection performance substantially, mainly for small objects.

4.3. One-stage detectors

4.3.1. SSD

SSD (Liu et al., 2016) is one-stage object detection network that use a single forward CNN to predict item class and position. SSD achieved performance standards in terms of speed and accuracy for object detection tasks, achieving over 74% mean Average Precision (mAP) at 59 fps on standard datasets. The basic architecture of SSD is shown in Fig. 13.

In general, the SSD consists of three sections:

1. *The fundamental convolutional layer, which contains ResNet, ResNetv2, VGG, inception, and other feature extraction networks. The intermediate convolutional layer creates a layer-scale feature map, which splits the receptive field into a large number of small cells, assisting in the identification of small objects.*
2. *An extra convolutional layer is linked to the last layer of the basic convolutional network. Larger-scale multi-scale feature maps are produced.*
3. *A prediction convolutional layer employing a tiny convolutional kernel predicts bounding box location and confidence for several categories.*

4.3.2. YOLO

YOLO (Redmon et al., 2016) focused on speeding up the object detection methods. The region proposal was deleted since the object detection issue was regarded as a regression issue. It divides the input image into 7×7 pixels, with each pixel being used to estimate where the center of an object could lie, rather than using pre-defined anchors for object portions. Each cell projected bounding box locations, class probabilities, and scores for each bounding box. YOLO is a real-time object detector that can detect things at a rate of 45 fps, which is extremely fast in comparison to the previous object detection models. On the other hand, only class probability was estimated within each cell. It cannot handle a large number of ground truth objects and does not work well with items that are partially localized in one cell and has poor localization accuracy due to bounding box sizes and proportions. On the COCO dataset, YOLO produces a mAP with a rate of around 54.30%.

4.3.3. YOLOv2

YOLOv2 (Redmon & Farhadi, 2017) proposes several improvements to the first version of YOLO. The completely connected layers are eliminated, and the anchor boxes approach is used to forecast bounding boxes to improve the recall. Unsupervised learning approaches are used to construct bounding box sizes and proportions directly using training data. The bounding box analysis forecasted the position in relation to the left top location of the cell, resulting in predicting limits of 0 and 1. Batch normalization, high-resolution classifications, and multi-resolution training are among the other strategies offered by this version. All of the strategies have significantly increased detection accuracy while maintaining high speed.

4.3.4. YOLOv3

In order to keep low translation variance, SSD chooses the early layers to create large-scale feature maps specifically used to find smaller objects. Feature maps generated by early layers are complex enough, hence resulting in poor performance on smaller objects. To address these mentioned issues, YOLOv3 (Redmon & Farhadi, 2018) enhances the accuracy of detection of objects by referring to the notion of residual network. It is a one-stage method that also works efficiently

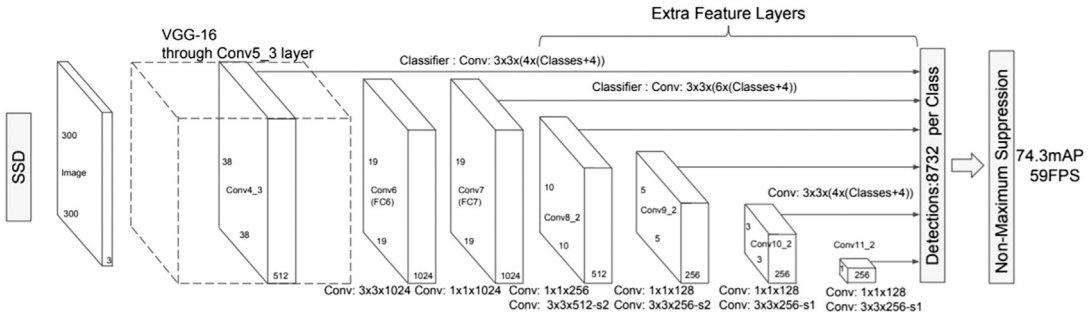

Fig. 13. Architecture of SSD (Liu et al., 2016).

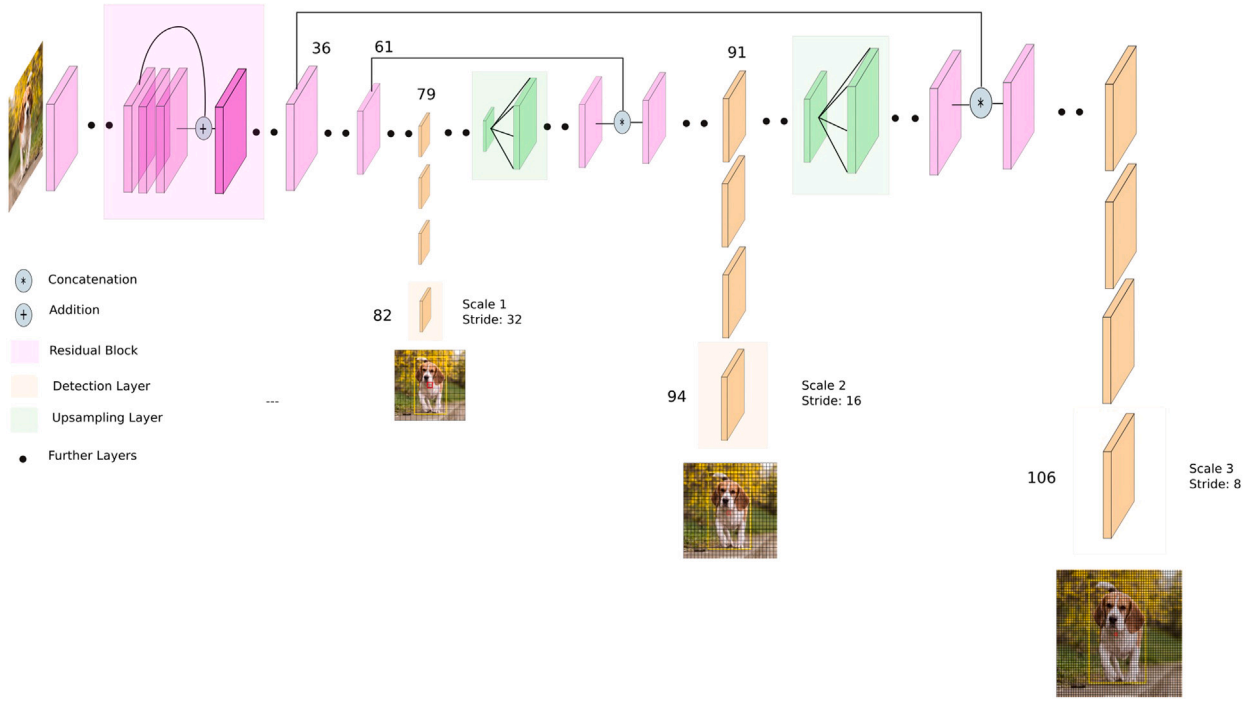

Fig. 14. YOLOv3 architecture (Redmon & Farhadi, 2018).

with respect to detecting speed. The architecture of YOLOv3 is depicted in depth in Fig. 14. It generates a small-scale feature map that is a 32-fold lower resolution version of the original image using Darknet-53, omitting the last three layers. To detect big objects, a small-scale feature map is employed. The small-scale feature map is up-sampled and concatenated with the feature map generated by previous layers. A large-scale feature map is generated by YOLOv3, as opposed to SSD choosing the previous layers to build large-scale feature maps. For the detection of small-sized objects, a large-scale feature map including position information from previous layers and complicated features from deeper levels is employed. The feature map scales are 8, 16, and 32 times down-sampled as compared to the original image, respectively. Softmax is used to predict single-level classification, but YOLOv3 predicts multilevel classification for each bounding box by using separate sigmoid functions for each of the boxes.

4.3.5. YOLOv4

YOLOv4 (Bochkovskiy et al., 2020) is an improved version of YOLOv3. It splits images into regions and further processes the probabilities for each region and bounding boxes using a single neural network on the entire image. Bag-of-Specials (BoS) and Bag-of-Freebies (BoF) are two distinct packages used in the model's backbone to improve performance with Cross Stage Partial Darknet53. The trade-off between these two factors affects the performance efficiency. While BoS is used to increase inference cost by a minimal amount while considerably enhancing object detection accuracy, BoF is used to just raise the cost of training while keeping the cost of inference low. The neck of YOLOv4 has improved Spatial Pyramid Pooling, which creates a fixed-size output independent of input size.

5. State-of-the-art for weapon detection methods

Weapon detection has emerged as a captivating topic in the field of object detection methods. Various systems have been developed for detecting weapons like pistols, rifles, and knives, each with its own set of advantages and limitations. Misclassification, intra-class detection, a dynamic background, occlusion, and varying illuminations are amongst the major issues which make the identification of handguns, knives, and other weapons difficult. A thorough examination of several techniques has been carried out to cover the previous weapons detection systems to the most recent models as shown in Fig. 15.

In the literature, several algorithms have been suggested like Harris interest point detector (Harris & Stephens, 1988), SIFT (Lowe, 1999), SURF (Bay et al., 2006) and deep learning techniques like Faster R-CNN (Ren et al., 2017), SSD (Liu et al., 2016), and YOLO (Redmon et al., 2016) for the detection task. The generalized weapon detection model is shown in Fig. 16.

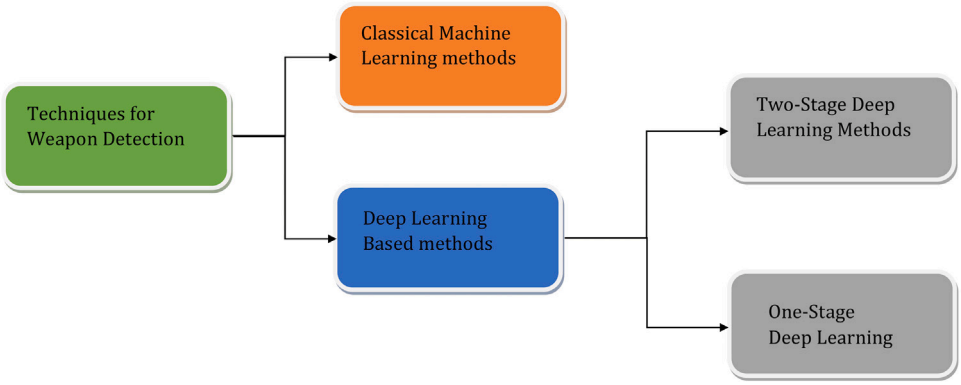

Fig. 15. A classification of computer vision algorithms for detecting weapons.

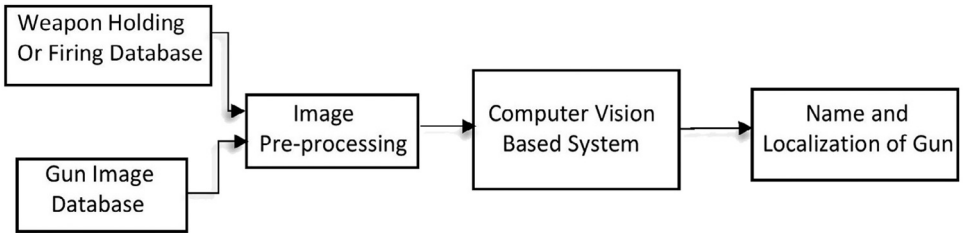

Fig. 16. Basic model of computer vision-based weapon detection system.

5.1. Weapon detection using classical machine learning methods

The Haar Cascades were developed by Żywicki et al. (2011) to identify hazardous equipment such as knives. Positive and negative sample images were used in the training phase to demonstrate the presence and absence of the target item, respectively. A total of 1560 positive and 6518 negative samples were used in the training phase. Positive sample criteria include information about angle, illumination, dynamic background, knife in hand, variety of blades, and varied grips. To improve the performance, three training sets were constructed in the experiment, among which the third training set observed the best result. However, the results were not satisfying as the true positive rate was 46%, which is a relatively small score for the detection in real-time.

Glowacz et al. (2013) introduced an AAM based method for object detection like knives. The goal was to determine whether or not a knife could be seen in the given image. They utilized the Harris corner detection method (Harris & Stephens, 1988) to identify tip-of-the-knife. The number of discovered corners was determined on the basis of a predefined threshold. All knife tips were identified at the lowest threshold of 204, and for all images, the mean value of the threshold at which the knife-tip is tagged as a corner was 217. The overall classification accuracy of this model was 92.50%. However, as AAMs are not invariant to the rotation, the method works only if the tip of the knife is visible in the images. Kmieć et al. (2012) introduced an approach employing the Harris corner detection technique and AAM initialized with shape-specific interest points. The model failed to recognize the knife in three images out of 40 positive images. This method only works when the tip of the knife is visible in the image.

Tiwari and Verma (2015a) used the Harris Interest Point Detector (HIPD) and Fast Retina Keypoint (FREAK) to develop a new approach for detecting firearms. A hybrid technique utilized both concepts. It included color-based segmentation to eliminate irrelevant images or colors from the image, and Harris Interest Point Detector and FREAK to detect the gun. For color-based segmentation, the K-means clustering method was used. Morphological processing is applied to each image in order to extract boundaries and close tiny gaps. To discover the resemblance with the gun, the interest point feature of the object boundary was extracted and compared to the stored description. When the similarity score exceeds 50%, the system provides a warning. The model was assessed in terms of accuracy after testing it against various sessions as well as negative images. This method has an overall accuracy of 84.26%. Later on, Tiwari and Verma (2015b) enhanced their work by proposing a technique for detecting firearms in which they used SURF. These extracted features of the object boundary was compared to the stored descriptions to find the resemblance with the gun. The system elevates the warning when it receives a resemblance of greater than 50%. The authors also discussed several challenges such as gun rotation, orientation, and variation as well as light, shadow, noise, real-time processing power, information loss owing to 3D to 2D transformation, partial or complete occlusion of the gun, and deformation. Following that, morphological closure and boundary extraction were done, resulting in an image that displays the general structure of the item while hiding the interior details covered by a rectangular box. Although SURF feature extraction is not faster than other techniques like Harris and SIFT, it can handle images irrespective of scale, orientation, or other characteristics. SURF first finds interest-points (such as corners and blobs) and then uses the Hessian Matrix to produce descriptors for each. Finally, a similarity score was calculated between the stored description of the gun and that of the blob. The SURF characteristics of an object border are utilized to compare the forms of items. A total of 25 pictures were utilized, out of which 15 were with positive samples. Overall, the true positive rate of the model was 86.67%. However, since these systems were time-consuming and complicated, they were unable to be used for real-time weapon detection.

The comparative analysis in terms of true positive rate between classical machine learning methods is shown in Fig. 17. It can be explicitly concluded from the graph that AAMs are an efficient method over others, having a true positive rate of 92.50%.

A detailed description of the work based on classical machine learning methods is summarized in Table 4. The specifics of the datasets and the outcomes that were acquired are detailed in this table. From the results, it can be concluded that the approaches utilized by Glowacz et al. (2013) and Kmieć et al. (2012) resulted in the highest true positive rates.

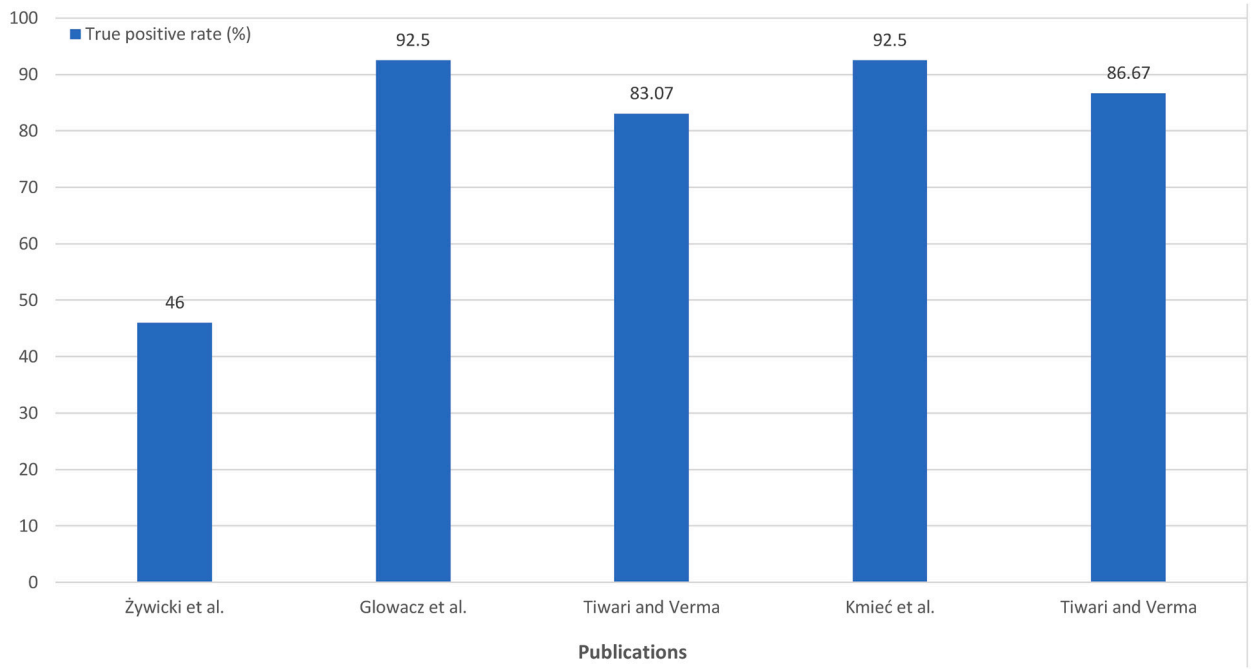

Fig. 17. Performance analysis between classical machine learning methods.

Table 4
Detailed results based on classical machine learning methods.

Publication	# Images in positive test set	# Correctly classified positive images	True positive rate	# images in negative test set	# Misclassified negative images	False positive rate	Classification accuracy
Żywicki et al. (2011)	1560	–	46.00%	6518	–	–	–
Kmieć et al. (2012)	40	37	92.50%	40	0	0%	92.50%
Tiwari and Verma (2015a)	65	54	83.07%	24	3	8.33%	84.26%
Glowacz et al. (2013)	40	37	92.50%	40	0	0%	92.50%
Tiwari and Verma (2015b)	15	13	86.67%	10	0	0%	86.67%

Classical machine learning methods need a lot more human interaction to produce results. These systems have problems with the reliability of their database, where guns make up the majority of the picture. This does not accurately show how real-life events with a handgun work. As a result, these systems are not suitable for continuous monitoring in situations where the images retrieved from CCTV recordings are complicated owing to various variables or when there are open regions with a large number of objects. In such complicated scenarios, these conventional methods fail to provide better accuracy for weapon detection.

In traditional machine learning methodologies, the bulk of the applicable features must be set by a domain expert in order to minimize computational complexity and make patterns more transparent for learning techniques to efficiently work. The major advantage of deep learning algorithms is that they try to learn high-level features from data in an incremental manner. This reduces the feature extraction complexity as well as lowers the need of domain expertise in real-time applications.

5.2. Weapon detection using two-stage deep learning methods

The most often used measurements in computer vision are True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN). The number of images accurately labeled as positive images while employing classifiers to identify weapons is referred to as TP. The existence of a weapon in the input image is indicated by a positive. The term FP refers to an actual instance that is missed by the classifier. The properly classified negative image is indicated by TN, while the number of wrongly classified negative images is denoted by FN. The performance parameters namely, accuracy, precision, recall, and F1_score are measured by Eqs. (6), (7), (8), and (9), respectively as follows:

$$accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{6}$$

$$precision = \frac{TP}{TP + FP} \tag{7}$$

$$recall = \frac{TP}{TP + FN} \tag{8}$$

$$F1\_score = 2 * \frac{precision * recall}{precision + recall} \tag{9}$$

Olmos et al. (2018) designed an automated method for the detection of handguns to facilitate monitoring and control systems. The authors reframed the problem of weapon detection as a problem of reducing false positives, and discovered a solution by (i) building a key training dataset using the results of the DCNN classifier, and (ii) comparing the two approaches, namely the sliding window approach and the region proposal approach, to determine the best classification model. The dataset used in the experiment contains 3000 images of short guns with rich backdrop detail. The Faster R-CNN model with VGG-16 as a feature extractor produced the most promising results. After five consecutive true positives among the thirty situations, the automatic alarm system effectively activates the alarm. They also established a metric called Alarm Activation Time per Interval (AATpI) to assess the performance of the detection model. With an average time interval of AATpI = 0.2 s, the model correctly identified the gun in 27 scenarios. However, in three scenes, the detector was incapable of detecting handguns due to the same factors mentioned as before, like low contrast and poor

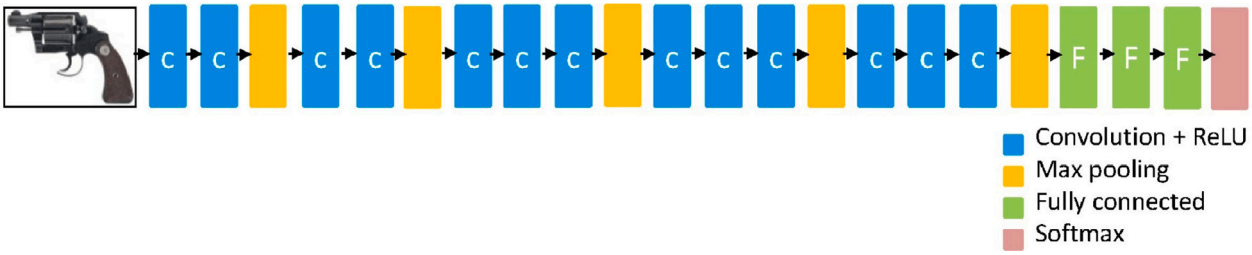

Fig. 18. Architecture of VGG-16 used in Olmos et al. (2018).

brightness of the frames, fast movements of the gun, or the guns not being visible in the forefront of the image. The precision score of this model was recorded at 84.21% with F1_score at 91.43%. As illustrated in Fig. 18, the architecture of VGG-16, contains 13 convolution layers and 3 fully connected layers, which is used as the feature extractor.

Verma and Dhillon (2017) utilized transfer learning to identify guns using a deep convolution network and a state-of-the-art feature area based CNN model. As a feature extractor, the system uses a CNN-based VGG-16 architecture followed by state-of-the-art classifiers trained on a typical gun database. The performance of the model was evaluated in a variety of situations, including different backgrounds with firearms, occlusion, and so on. The results show that SVM (Hearst et al., 1998) outperforms other classifiers with a classification accuracy of 92.60% and total accuracy was 93.10%. However, the model was built on a single CPU, which meant that training time was a major concern.

Gelana and Yadav (2018) proposed an image processing and machine learning-based weapon identification model. Their model consisted of six main elements: (i) RGB to gray-scale conversion was used to reduce the complexity of each frame and speed up the background subtraction process; (ii) Background subtraction: three alternative techniques to background subtraction and segmentation were used. The visual background extractor (Barnich & Van Droogenbroeck, 2011) and the improved Gaussian mixture model (Zivkovic, 2004) techniques, as well as the difference of frame background subtraction algorithm, are all used in this study (iii) Filtering operation: Dilation and erosion procedures were used on the extracted foreground object to eliminate tiny white noises caused by illumination fluctuations and to connect dissimilar parts in an image (iv) Segmentation/Edge Detection: The well-known Canny edge detection method (Canny, 1986) was employed for this purpose. The Canny algorithm inputs the filtered foreground object and outputs the information about edges (v) The sliding window approach substantially reduces the area evaluated by the learning algorithm. The size and slide step are chosen after several tests and are subjected to alteration in the future (vi) A tensorflow-based version of the CNN method was used to classify an item as either a treat (gun) or a non-treat (non-gun). After applying 30% split to the CNN training–testing dataset, 4000 negative and 1869 positive images comprised the dataset frame. The 585 positive and 1173 negative images among the 1758 images were used to test the algorithm. The most essential element in weapon detection was to reduce the frequency of false positives while maintaining detection sensitivity. The approach described in this work had a specificity of 99.73% for images containing non-gun items and a detection accuracy of 93.84% for images including gun objects.

Castillo et al. (2019) developed an automated cold steel weapon identification model for video surveillance that was based on a new brightness-directed preprocessing technique termed Darkening and Contrast at Learning and Test stages (DaCoLT) that enhances detection quality. The Faster R-CNN with Inception-ResNet-V2 (Szegedy et al., 2017) was the most accurate model with an F1_score of 95%. However, with a frame rate of 1.3 frames per second, it was not suited for near-real-time operations.

A unique binocular image fusion technique for reducing the frequency of false positives in the identification of firearms in surveillance films, was proposed by Olmos et al. (2019). They used a dataset of 3000 weapon images created by Olmos et al. (2018) for training and testing purposes. They compared the performance of Faster R-CNN with and without image fusion using four feature extractors, i.e., VGG-16, ResNet, Inception-Resnetv2, and Neural Architecture Search (NAS). Faster R-CNN (VGG-16 + ImageNet) has much greater accuracy, precision, recall, and F1_score compared to the previous existing methods. It achieved the overall highest accuracy of 80.62%. However, the most frequent cameras in CCTV systems are not dual cameras, so this method would not be appropriate for most retail establishments.

Pérez-Hernández et al. (2020) proposed a method utilizing binarization approach to improve the robustness, precision, and reliability of small item recognition. To enhance their detection accuracy in movies, they recommended adopting a two-level deep learning-based approach called Object Detection using Binary Classifiers. In which the first level selects potential areas from the input frame, while the second level employs a CNN-classifier that employs One-Versus-All (OVA) and One-Versus-One (OVO) binarization techniques. A firearm, a knife, a smartphone, a bill, a purse, and a card were used to create the database. The experimental study shows that the suggested technique reduces the incidence of false positives when compared to the baseline multi-class detection model. However, because this model was complicated and time-intensive, it could not be used to identify guns in real-time. The dataset collection had 560 images in total. As indicated in Table 5, the OVO model observed the highest precision of 93.87%.

Iqbal et al. (2021) proposed a weakly supervised Orientation Aware Object Detection (OAOD) approach using Axis-Aligned Bounding Boxes (AABB) for training and learning to recognize oriented object bounding boxes (OBB). The proposed OAOD differs from previous oriented object detectors in that it does not require OBB during training, which may or may not be available at any given time. To achieve the goal of training on AABB and identification of OBB, a multiphase method was utilized, with Stage-1 estimating AABB and Stage-2 estimating OBB. There are 10,973 pictures of firearms and rifles in the weapon dataset presented by the ITUF (Iqbal et al., 2021). The ITUF dataset was used to examine the OAOD technique to other state-of-the-art classification techniques, such as fully supervised oriented object detectors. The overall obtained mAP on AABB was 88.30% and the mAP on OBB was 77.50%. However, because the model was computationally expensive and the mAP was quite low, it could not be used for real-time gun detection.

González et al. (2020) used Faster R-CNN to utilize FPN with ResNet-50 on a new dataset collected from a genuine CCTV installed in a university campus. Further they developed synthetic images to be employed in quasi real-time CCTV. The FPN architecture achieved an accuracy score of 88.12%. However, the developed model could not be utilized for training or testing purposes because the created synthetic dataset was not providing satisfactory results.

Kaya et al. (2021) presented a novel model based on deep learning that utilizes VGG-16, ResNet-101, ResNet-50, and a suggested CNN model with seven layers to detect seven distinct weapons. Assault rifles, knives, bazookas, hunting rifles, pistols, grenades, and revolvers are among the 5214 weapon illustrations split into seven categories. The system was developed with a total of 3128 images in training and 1043 images in validation and testing. Consequently, the proposed model was found to be 98.40% accurate. However, that device was capable

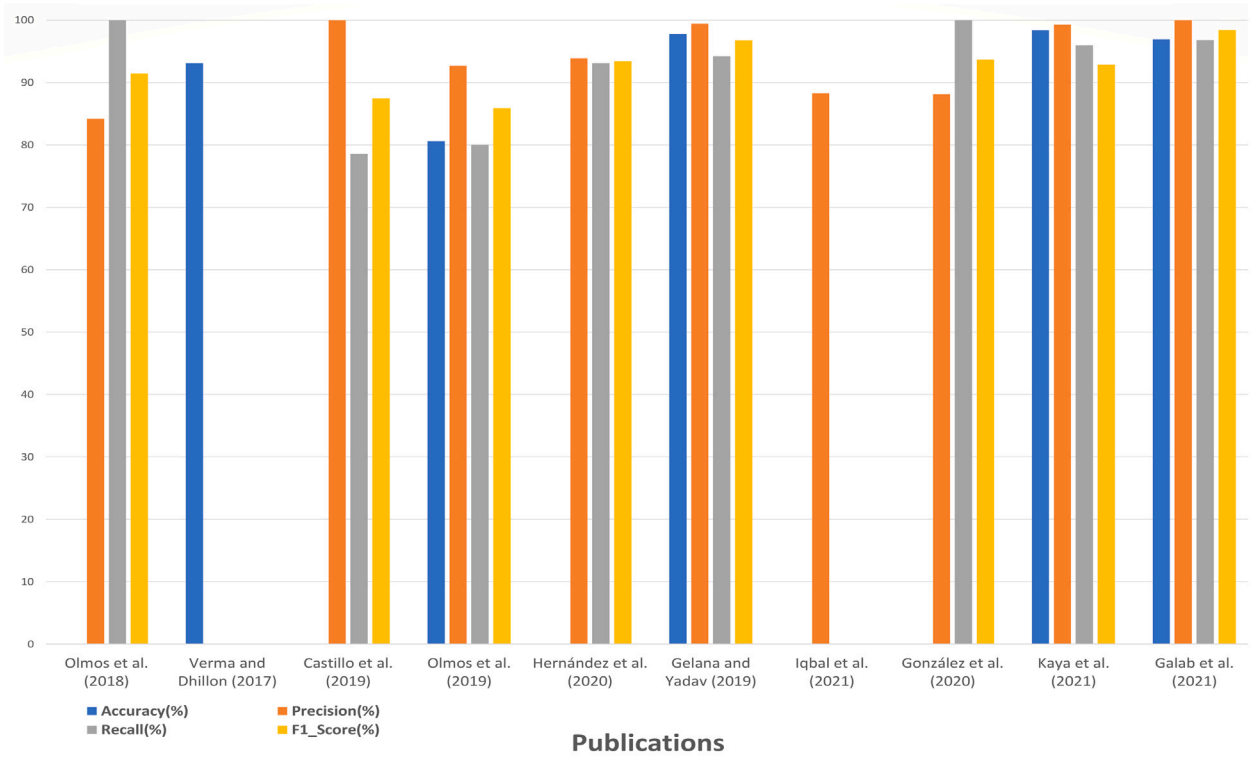

Fig. 19. Comparative analysis of performance of two-stage deep learning methods.

of identifying a few types of weapons only. The intended method was not very complicated, but it was very slow when it came to computing.

Galab et al. (2021) demonstrated how to improve the brightness of knife detection in surveillance systems using an adaptive method. Based on the preprocessing Brightness Handler procedure (BHp), they compared four CNN architectures: AlexNet, VGGNet, GoogLeNet, and ResNet. AlexNet with BHp produced excellent outcomes with a 96.95% accuracy. AlexNet was an early CNN with six convolutional layers and performed quite slowly in contrast to current CNN models. That model took images of the size of 227 × 227 pixels, indicating that the weapon must cover the majority of the image.

Fig. 19 shows the performance analysis of two-stage deep learning methods in terms of accuracy, precision, recall, and F1_score. It may be implied from the graph that the model developed by Kaya et al. (2021) observed the best accuracy with 98.40% compared to all other methods, whereas Galab et al. (2021) secured the highest F1_score as 98.42%. The models developed by Castillo et al. (2019) and Galab et al. (2021) have the highest precision values. On the other hand, other models developed by González et al. (2020) and Olmos et al. (2018) observed highest recall values.

Table 5 shows a detailed comparative analysis based on two-stage deep learning methods' outcomes in terms of accuracy, precision, recall and F1_score.

5.3. Weapon detection using one-stage deep learning methods

Narejo et al. (2021) created a smart surveillance security system that identifies weapons, especially firearms. They used backbone Darknet-53 to train the YOLOv3 classification model for that purpose. They collected a large number of photos from Google manually and approximately 50 pictures for each weapon class. The overall accuracy of the model was 98.89%, but precision and F1_score were not measured. Also, the number of images in the collection was not provided.

Romero and Salamea (2019) developed the system to resolve existing issues and divided the operation of system into two halves. The first front end was in charge of limiting the area of interest, while the second back end was in charge of detecting the weapon from the front end. The authors created a database comprising 17,684 images from various movies, with firearms (class A) and without firearms (class B). Using various approaches (rotating and flipping), the image dataset was additionally expanded by 2,29,892 (from 17,568 to 2,47,576). As mentioned before, the system was made up of two parts; namely the front end and the rear end. The authors employed YOLO for real-time object recognition and localization in the front end. YOLO was trained on the COCO dataset to recognize people while ignoring the rest of the image, which reduces the complexity of the system and, hence, the probability of false positives. The VGG-Net and ZFNet models were used to identify weapons. Grayscale pictures were also useful in enhancing the efficiency of the system. If the individual in the bounding box does not have a weapon, the bounding box will be eliminated, narrowing the area of concern. The overall performance of the system was observed as 86% recall and 90.80% accuracy. Fig. 20 depicts the operation of a weapon-detecting system proposed by Romero and Salamea (2019).

Cardoso et al. (2019) used YOLO object detector to detect guns in images using CNN. The idea was tested on a database of 608 images, including 304 weapons. Experiments observed an accuracy of 89.15%. However, the number of images in the dataset was very low, hence it was found infeasible for real-time detection

Jain et al. (2020) used SSD and Faster R-CNN algorithms to develop automated weapon identification using CNN. Faster R-CNN observed better accuracy as 84.60%. On the other hand, SSD achieved an accuracy of 73.80%, which was low compared to the Faster R-CNN. Due to the higher speed, SSD provided real-time detection, but Faster R-CNN observed higher accuracy. In a fully automated system, a person in charge double-checks every gun detection alert with @0.73 fps (SSD) and @1.606 fps (Faster R-CNN), which are too slow for real-time detection.

Salido et al. (2021) compared three CNN models for automatic identification of pistols in video surveillance. The goal was to see if integrating posture information with the way firearms were held in

Table 5
Comparison of detection results based on two-stage deep learning methods.

Authors	Data specifications	Detection results			
		Accuracy (%)	Precision (%)	Recall (%)	F1_Score (%)
Verma and Dhillon (2017)	–	93.10	–	–	–
Olmos et al. (2018)	3000	–	84.21	100	91.43
Castillo et al. (2019)	19,379	–	100	78.55	87.44
Gelana and Yadav (2018)	5869	97.78	99.45	94.21	96.76
Olmos et al. (2019)	3000	80.62	92.68	80	85.88
Pérez-Hernández et al. (2020)	5680	–	93.87	93.09	93.43
González et al. (2020)	7649	–	88.12	100	93.68
Kaya et al. (2021)	–	98.40	99.28	95.97	92.89
Iqbal et al. (2021)	10,973	–	88.30	–	–
Galab et al. (2021)	12,899	96.95	100	96.80	98.42

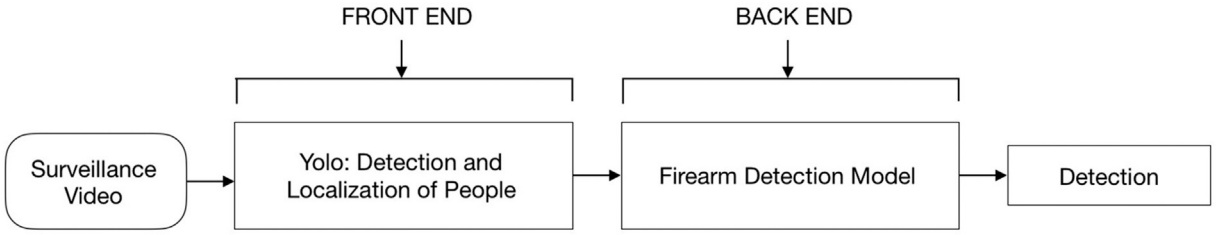

Fig. 20. System architecture of Romero and Salamea (2019).

the training dataset would reduce false positives. The findings showed that RetinaNet fine-tuned with the unfrozen ResNet-50 backbone had the greatest average precision of 96.36% and recall of 97.23%, while YOLOv3 had the highest accuracy of 96.23% and F1_score values (93.36%) when trained on the dataset with posture information. Using YOLOv3, the number of false positives and false negatives was 8 and 21, respectively, which was quite high for the tiny dataset and poor resolution images.

Singh et al. (2021) presented a computer vision-based method for identifying firearms using YOLOv4. The images of knives, swords, pistols, machine guns, shotguns, and other weapons were included in the dataset used to train the model. They combined them into a single weapon class. The model employed had a mean Average Precision (mAP) of 77.75% and an average loss of 1.314. However, mAP was an insufficient factor for measuring the real-time weapon identification performance.

Sliding window and region proposal/object detection were two methodologies used by Bhatti et al. (2021). Some of the algorithms employed were VGG16, Faster R-CNN, Inception-ResnetV2, SSD, MobileNetV1, Inception-V3, Inception-ResnetV2, YOLOv3, and YOLOv4. A total of 8327 images comprising pistols and non-pistol classes were used, which were collected from various sources. A total of 7328 images were utilized for the training and another 999 for testing. YOLOv4 outperformed all other algorithms, receiving an F1_score of 91% and a mean average accuracy of 91.73%. The number of false positives and false negatives were still relatively high as 54 and 52, respectively.

Lamas et al. (2022) presented a reproducible and traceable top-down weapon detection over pose estimation methodology that exploits the human presence in scenarios where a person carries a weapon, firearm, or knife. The two types of detection architectures were used among the four selected detection models for evaluating the approaches. Faster R-CNN, a two-stage detector based on ResNet101 and various one-stage detectors such as SSD, based on ResNet50, EfficientDet (Tompson et al., 2015) based on D3, and CenterNet (Duan et al., 2019). All deep learning architectures were trained on the Sohas weapon dataset with a precision score of 94.4%, in which EfficientDet outperformed others. However, this method was able to detect human-handled weapons only.

Fig. 21 shows the performance analysis of one-stage deep learning methods in terms of accuracy, precision, recall, and F1_score. According to Fig. 21, the model created by Narejo et al. (2021) observed the greatest accuracy of 98.89%, while the model developed by Salido et al. (2021) achieved the best precision score of 96.23%.

Fig. 22 shows the overall performance analysis of detection using deep learning methods in terms of accuracy and precision parameters.

Table 6 presents a thorough comparison of one-stage deep learning models developed by different researchers in terms of accuracy, precision, recall, and F1_score.

6. Conclusion

In the area of security and surveillance, weapon detection is of significant use in computer vision. An automatic weapon detection system that responds quickly in situations that could be dangerous is good for public safety. This literature attempts to showcase several conventional weapon detection systems using machine learning and the most advanced deep learning techniques. The journey began with a manually operated system and progressed to completely automated and sophisticated technologies. In light of this, numerous conventional weapon detection techniques have already been developed, viz. HIPD, AAMs, SIFT, SURF, FREAK, and many more, wherein the AAMs have emerged to be the preeminent among these. Although the multitudinous applications of these conventional techniques have been reviewed in the past, none has so far emerged as an effective technique owing to the imprecision in detection of tiny objects due to their complex background and partial occlusion. Classical methods require manual intervention for extracting features, and thus, they are not very precise for weapon recognition (Krizhevsky et al., 2012). This opens a window for the development of deep learning architectures capable of automatically discovering higher level features from input images that offer speed, accuracy, and real-time applications viz. self-driving cars (Maqueda et al., 2018), natural language processing (Worsham & Kalita, 2020), face detection (Zhan et al., 2016), speech recognition (Nassif et al., 2019), text recognition (Roy et al., 2017), and disease diagnosis (Hu et al., 2018; Ma et al., 2021) etc., of this technology in the field. Additionally, a wide literature in the domain of DCNN and transfer learning methods incorporating multiple models (one-stage and two-stage) like Faster R-CNN, VGG-Net, ZFNet, and YOLOv3 is available. In the case of one-stage deep learning methods, YOLOv3 has higher precision and shows better performance in comparison to others.

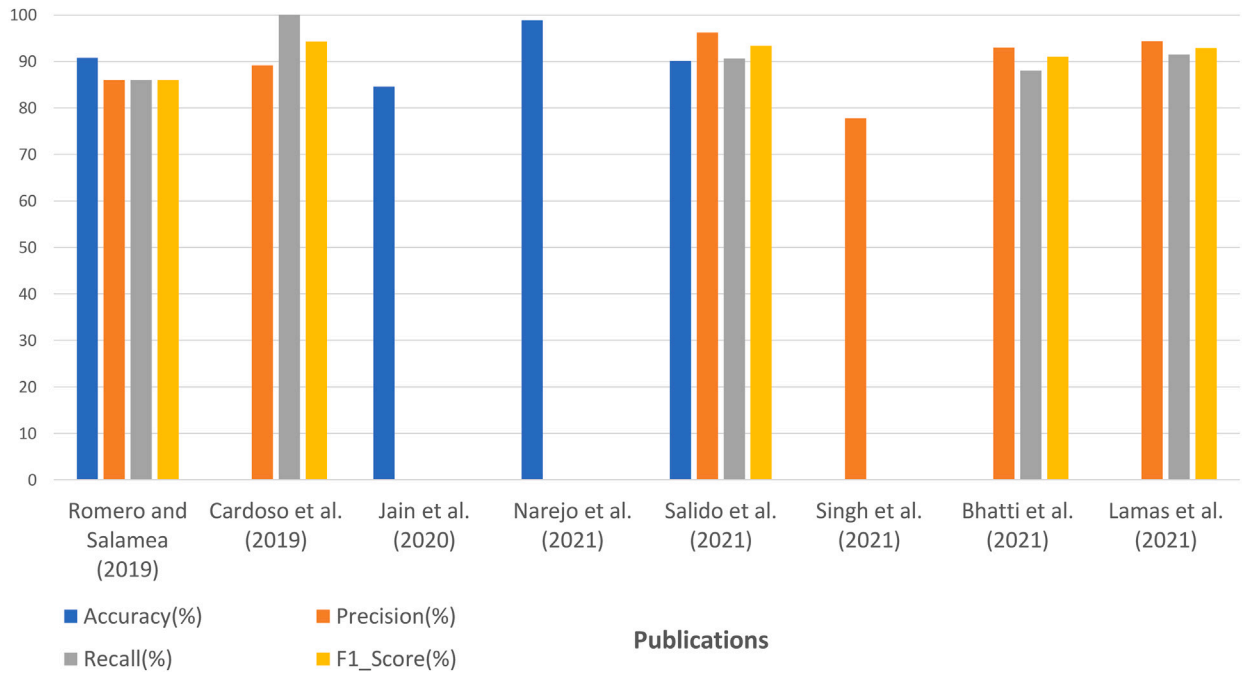

Fig. 21. Analysis of performance of one-stage deep learning methods in terms of accuracy, precision, recall and F1_score.

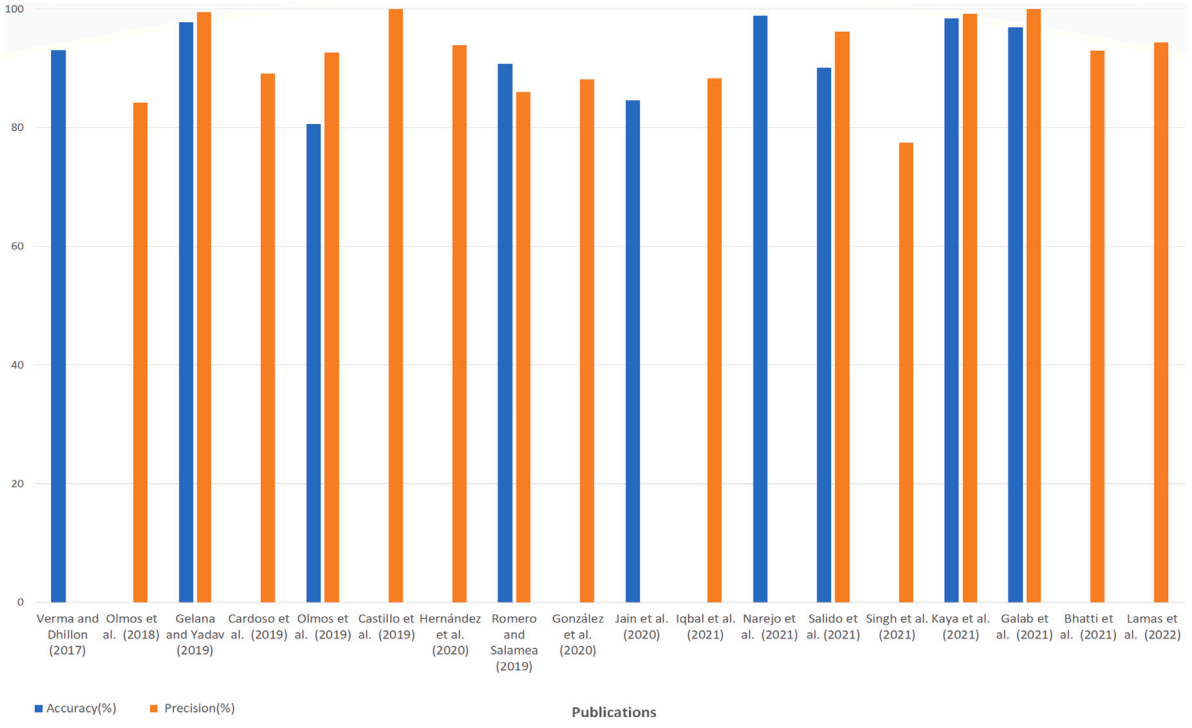

Fig. 22. Analysis of performance of detection using deep learning methods in terms of accuracy and precision.

Faster R-CNN architecture observed the highest precision compared to other methods in two-stage methods.

In Table 7, we provide the important findings of the study that were discovered. The following information are included in the table: (a) The name of the method used, (b) The strength of method, (c) Problems encountered in weapon detection, and (d) The respective publication and comments.

The two-stage detectors exhibit better accuracy in comparison to single-stage detectors, which is evidenced by their vast real-time applications, but the latter are more cost-effective than the former. One-stage detectors are usually faster than two-stage ones because they use lightweight backbone networks, eliminate preprocessing algorithms, and consider fewer candidate regions for prediction. However, two-stage detectors can run in real time with the introduction of similar

Table 6
Comparison of weapon detection results based on one-stage deep learning methods.

Authors	Data specifications		Detection results			
	Positive images	Negative images	Accuracy (%)	Precision (%)	Recall (%)	F1_Score (%)
Romero and Salamea (2019)	8843	8841	90.80	86.00	86.00	86.00
Cardoso et al. (2019)	3000	6857	–	89.15	100	94.26
Jain et al. (2020)	–	–	84.60	–	–	–
Narejo et al. (2021)	–	–	98.89	–	–	–
Salido et al. (2021)	1220	–	90.09	96.23	90.67	93.36
Singh et al. (2021)	–	–	–	77.75	–	–
Bhatti et al. (2021)	3073	5254	–	93.00	88.00	91.00
Lamas et al. (2022)	3000	14,684	–	94.40	91.50	92.90

Table 7
An overview of the survey's major results using the classical machine learning approach and deep learning approach.

Methods	Strengths	Issues	Publications and remarks
Haar Cascades	The accuracy of the cascade improves with the increased number of positive and negative samples images.	The findings are unsatisfactory due to the low true positive rate obtained.	Żywicki et al. (2011) observed that increasing the Haar Scale coefficient reduced the frequency of incorrectly detected knives. As previously stated, the obtained results for this cascade are unsatisfactory.
AAMs	According to the test results, this technique outperforms over other classical machine learning algorithms with a TRP of 92.50%.	This method is not rotation invariant. The technique works only if the knife tip is visible in the images.	Glowacz et al. (2013) present AAMs as a weapon detection tool. Later on, Kmieć et al. (2012) enhanced their findings by using the Harris corner detection approach in their own work.
HIPD and FREAK	The K-means clustering method along with HIPD and FREAK is applied to utilized the color-based segmentation which results in higher accuracy.	However, this is only useful if the gun is entirely visible in the scene. The approach fails in the case of a partially visibility of gun or in a blurred images.	This approach was adopted by Tiwari and Verma (2015a). Additionally, similarity score surpasses 50% after the alert mechanism applied.
SURF	The findings from previously used methodologies are improved in terms of accuracy.	The computational time is higher in view of real-time detection.	Tiwari and Verma (2015b) used SURF methods for the classification task and improved the better efficiency.
SVM with VGG-16	For feature extraction, VGG-16 is very effective and commonly used deep learning architecture. It improves the extraction of high-level features from images.	SVM is a classification method that is slow as well as complexed. However, CNN techniques produce better results than SVM method.	Gelana and Yadav (2018) used this method with a variety of techniques such as Canny edge detection, enhanced Gaussian mixture model and others to achieve results. However, the approach is slow and complicated which makes it unsuitable for real-time detection.
Faster R-CNN	This approach improves the precision for small weapon detection using two-stage deep learning approach.	Despite of higher precision, it is a time-consuming and complex method and computationally expensive.	Various studies employed Faster R-CNN with the variety of backbone networks, including VGG-16 Olmos et al. (2018), Inception ResNetv2 Castillo et al. (2019), FPN with ResNet-50 González et al. (2020) and others to obtain better results. Instead of using a selective search method, it suggests using RPN to generate region suggestions which makes it much faster than R-CNN and Fast R-CNN.
SSD based CNN	Localization and classification tasks are completed in a single forward pass across the network which results in much faster detection.	It can analyze a video at the rate of 0.75 frames per second. However, it does not perform well with small objects like a handguns, because it uses the first convolution layers to create high-level feature maps.	This approach along with Faster R-CNN was suggested by Jain et al. (2020). SSDs deliver much faster performance but achieves less accuracy.
YOLO and its versions	YOLOv3 and YOLOv4 are faster than any of the deep learning architectures currently available. This method generates high-level feature maps by using both previous and subsequent layers, resulting in more accuracy than SSD. It is the most effective approach for detecting firearms in real-time.	YOLO and YOLOv2 are more effective for the identification of large objects. Importantly, it performs poor while dealing with small objects.	Several researchers (Cardoso et al., 2019; Narejo et al., 2021; Romero & Salamea, 2019) use the YOLO architecture for weapon detection. YOLOv4 (Bhatti et al., 2021; Singh et al., 2021) outperforms over other enhanced versions of YOLO.

techniques. One-stage frameworks' performance is poorer than two-stage architectures like Faster RCNN in the detection of small objects, which gives fair competition in the detection of large objects.

There are still certain challenges in the field of weapons detection that need to be addressed, such as a lack of datasets, the detection of weapons in a variety of lighting conditions, and others. Table 8 provides a more in-depth discussion of these concerns.

Despite the fact that datasets have recently emerged, the lack of large and well-balanced datasets limits the development of deep learning algorithms that are generalizable enough to be employed in automatic weapon detection systems. As the public datasets originate from a range of machines with different inherent architectures, domain adaption techniques might help. Deep learning techniques can provide very productive outputs considering all the above-mentioned features, but models based on these techniques for real-time applications are still not at the forefront. The reason behind this is the complex nature of the performed simulations, as a large dataset is required for computing the output. Moreover, detectors based on deep learning generally contain a high number of parameters and are consequently data-hungry, requiring a powerful computing system for the training of the developed model.

Table 8
Several issues of weapon detection systems.

Issues	Comments
The unavailability of real-time datasets	The datasets that have been presented in a number of research papers are gathered from the internet sources. There are just a few datasets González et al. (2020) available which are obtained from closed-circuit television cameras.
Multiple weapon detection system	There is a still a requirement for multiple weapon detection. This specific problem has only been addressed in a few researches like González et al. (2020), Salido et al. (2021) and Verma and Dhillon (2017).
The partial appearance of the weapon	Only a few researches have tackled the subject of partial occlusion of weapon. Nonetheless, these are important difficulties take place for weapon detection.
Weapon detection of different kinds	This is a significant problem to take into consideration. Only a few works (González et al., 2020; Iqbal et al., 2021; Olmos et al., 2018) are capable of detecting distinct sorts of guns.

Additionally, device may be constructed utilizing these algorithms for automatic weapon detection, which notifies security staff when it detects a weapon. The companies and organizations that supply security and surveillance systems would benefit from the implementation of an automated weapon detection system on internet of things (IoT) devices, such as a smartphone, laptop, etc. Human resource management and the creation of new products or applications are being transformed by machine learning and deep learning. This creates an environment appropriate for deep learning in open innovation and small and medium enterprises (SMEs) (Alam & Ansari, 2020; Malo-Perisé & Merseguer, 2022). According to the findings of the research (Baierle et al., 2020), open innovation characteristics have a significant impact on the competitiveness of manufacturing SMEs in a Southern Brazilian area.

7. Future scope

There is still a long way to go before developing a single robust deep learning technique. The following future scopes are offered based on the thorough survey as discussed in the paper. The following points are inferred for the automated identification of firearms:

i. **Requirement of real-time dataset:** The specific dataset for weapon detection is unavailable. At the time, only a few real-time datasets were available. Usually, the datasets are gathered from virtual sources such as movies, games, and others, which raises an issue about the reliability of the data due to varying surrounding conditions, viz. illumination conditions, viewing angle, and resolution of images. The scarcity of real-time datasets emerges as a major obstacle in the development of automatic weapon detection systems.
ii. **A heterogeneous model:** The existing methods are not entirely capable of detecting weapons of the same class, various shapes, colors, and complex backgrounds. For example, different sensors are used to capture the images of guns or knives, resulting in different intensity distributions for a single image. The same weapon image is mapped to different pixel resolutions with different imaging parameters such as the size of the weapon. Thus, the development of a heterogeneous method for a reliable automatic weapon detection system is indispensable.
iii. **Constructive use of contextual information:** Objects in the visual world have complex relationships, and precise context is essential for comprehending them. Insufficient consideration has been devoted to the use of contextual information appropriately in the object detection field. A guidebook about the precise and successful utilization of this information might be a potential future avenue for visual software development.
iv. **Detection of small objects:** One more significant challenge in object detection system studies is the identification of small objects such as weapons or knives, which is one of the shortcomings of existing methods using deep learning architecture. As a result, there is a potential scope for developing techniques for small-sized objects.
v. **Need for low-computing network:** These networks comprise hundreds of millions of parameters, demanding large amounts of data as well as high-performance graphical processing units (GPUs) for training. This fascinated the researchers, who were building small and lightweight networks to decrease or eliminate network redundancy. The developed model can be operated effectively on tiny devices like smart phones and can be employed with the IoT.

Funding information

This research did not receive any specific grants from funding agencies in the public, commercial, or non-profit sectors.

CRediT authorship contribution statement

Pavinder Yadav: Data curation, Writing – original draft, Methodology, Writing – review & editing. **Nidhi Gupta:** Methodology, Writing – review & editing, Visualization, Investigation, Project administration. **Pawan Kumar Sharma:** Supervision.

Declaration of competing interest

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.

Data availability

No data was used for the research described in the article.

Acknowledgment

The present work has been carried out in the computer laboratory of the Department of Mathematics and Scientific Computing at the National Institute of Technology, Hamirpur, Himachal Pradesh, India.

References

Ainsworth, T. (2002). Buyer beware. *Security Oz, 19*, 18–26.
Alam, M. A., & Ansari, K. M. (2020). Open innovation ecosystems: Toward low-cost wind energy startups. *International Journal of Energy Sector Management, 14*(5), 853–869. http://dx.doi.org/10.1108/ijesm-07-2019-0010.
Baierle, I. C., Benitez, G. B., Nara, E. O., Schaefer, J. L., & Sellitto, M. A. (2020). Influence of open innovation variables on the competitive edge of small and medium enterprises. *Journal of Open Innovation: Technology, Market, and Complexity, 6*(4), 179. http://dx.doi.org/10.3390/joitmc6040179.
Barnich, O., & Van Droogenbroeck, M. (2011). ViBe: A universal background subtraction algorithm for video sequences. *IEEE Transactions on Image Processing, 20*(6), 1709–1724. http://dx.doi.org/10.1109/tip.2010.2101613.
Bay, H., Tuytelaars, T., & Van Gool, L. (2006). SURF: Speeded up robust features. In *Computer vision – ECCV 2006: Vol. 3951* (pp. 404–417). http://dx.doi.org/10.1007/11744023_32.
Bhatti, M. T., Khan, M. G., Aslam, M., & Fiaz, M. J. (2021). Weapon detection in real-time CCTV videos using deep learning. *IEEE Access, 9*, 34366–34382. http://dx.doi.org/10.1109/access.2021.3059170.

Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934.

Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI*, *8*(6), 679–698. http://dx.doi.org/10.1109/tpami.1986.4767851.

Cardoso, G. V., Ciarelli, P. M., & Vassallo, R. F. (2019). Use of deep learning for firearms detection in images. In *Anais do XV workshop de visão computacional* (pp. 109–114). http://dx.doi.org/10.5753/wvc.2019.7637.

Castillo, A., Tabik, S., Pérez, F., Olmos, R., & Herrera, F. (2019). Brightness guided preprocessing for automatic cold steel weapon detection in surveillance videos with deep learning. *Neurocomputing*, *330*, 151–161. http://dx.doi.org/10.1016/j.neucom.2018.10.076.

Cootes, T. F., Edwards, G. J., & Taylor, C. J. (1998). Active appearance models. In *European conference on computer vision*: *Vol. 1407*, (pp. 484–498). http://dx.doi.org/10.1007/BFb0054760.

Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition: Vol. 1* (pp. 886–893). http://dx.doi.org/10.1109/cvpr.2005.177.

Deng, J., Dong, W., Socher, R., Li, L., Li, Kai, & Fei-Fei, Li (2009). ImageNet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition* (pp. 248–255). http://dx.doi.org/10.1109/cvpr.2009.5206848.

Duan, K., Bai, S., Xie, L., Qi, H., Huang, Q., & Tian, Q. (2019). CenterNet: Keypoint triplets for object detection. In *2019 IEEE/CVF international conference on computer vision* (pp. 6568–6577). http://dx.doi.org/10.1109/iccv.2019.00667.

Freund, Y., & Schapire, R. E. (1996). Experiments with a new boosting algorithm. In *ICML: Vol. 96*, (pp. 148–156). http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.51.6252&rep=rep1&type=pdf.

Galab, M. K., Taha, A., & Zayed, H. H. (2021). Adaptive technique for brightness enhancement of automated knife detection in surveillance video with deep learning. *Arabian Journal for Science and Engineering*, *46*(4), 4049–4058. http://dx.doi.org/10.1007/s13369-021-05401-4.

Gelana, F., & Yadav, A. (2018). Firearm detection from surveillance cameras using image processing and machine learning techniques. *Smart Innovations in Communication and Computational Sciences*, *851*, 25–34. http://dx.doi.org/10.1007/978-981-13-2414-7_3.

Girshick, R. (2015). Fast R-CNN. In *2015 IEEE international conference on computer vision* (pp. 1440–1448). http://dx.doi.org/10.1109/iccv.2015.169.

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *2014 IEEE conference on computer vision and pattern recognition* (pp. 580–587). http://dx.doi.org/10.1109/cvpr.2014.81.

Glowacz, A., Kmieć, M., & Dziech, A. (2013). Visual detection of knives in security applications using active appearance models. *Multimedia Tools and Applications*, *74*(12), 4253–4267. http://dx.doi.org/10.1007/s11042-013-1537-2.

González, J. L. S., Zaccaro, C., Álvarez-García, J. A., Morillo, L. M. S., & Caparrini, F. S. (2020). Real-time gun detection in CCTV: An open problem. *Neural Networks*, *132*, 297–308. http://dx.doi.org/10.1016/j.neunet.2020.09.013.

Grega, M., Matiolański, A., Guzik, P., & Leszczuk, M. (2016). Automated detection of firearms and knives in a CCTV image. *Sensors*, *16*(1), 47. http://dx.doi.org/10.3390/s16010047.

Guo, Y., Liu, Y., Oerlemans, A., Lao, S., Wu, S., & Lew, M. S. (2016). Deep learning for visual understanding: A review. *Neurocomputing*, *187*, 27–48. http://dx.doi.org/10.1016/j.neucom.2015.09.116.

Haralick, R. M., & Shapiro, L. G. (1985). Image segmentation techniques. *Computer Vision, Graphics, and Image Processing*, *29*(1), 100–132. http://dx.doi.org/10.1016/s0734-189x(85)90153-7.

Harris, C., & Stephens, M. (1988). A combined corner and edge detector. In *Procedings of the alvey vision conference* (pp. 1510–5244). http://dx.doi.org/10.5244/c.2.23.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *2016 IEEE conference on computer vision and pattern recognition* (pp. 770–778). http://dx.doi.org/10.1109/cvpr.2016.90.

Hearst, M., Dumais, S., Osuna, E., Platt, J., & Scholkopf, B. (1998). Support vector machines. *IEEE Intelligent Systems and their Applications*, *13*(4), 18–28. http://dx.doi.org/10.1109/5254.708428.

Hu, Z., Tang, J., Wang, Z., Zhang, K., Zhang, L., & Sun, Q. (2018). Deep learning for image-based cancer detection and diagnosis - A survey. *Pattern Recognition*, *83*, 134–149. http://dx.doi.org/10.1016/j.patcog.2018.05.014.

Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In *2017 IEEE conference on computer vision and pattern recognition* (pp. 2261–2269). http://dx.doi.org/10.1109/cvpr.2017.243.

IMFDBs http://www.imfdb.org/wiki/Main_Page. (Online; Accessed 10 October 2021).

Iqbal, J., Munir, M. A., Mahmood, A., Ali, A. R., & Ali, M. (2021). Leveraging orientation for weakly supervised object detection with application to firearm localization. *Neurocomputing*, *440*, 310–320. http://dx.doi.org/10.1016/j.neucom.2021.01.075.

Jain, H., Vikram, A., Kashyap, A. Mohana, & Jain, A. (2020). Weapon detection using artificial intelligence and deep learning for security applications. In *2020 international conference on electronics and sustainable communication systems* (pp. 193–198). http://dx.doi.org/10.1109/icesc48915.2020.9155832.

Jeong, C. Y., Yang, H. S., & Moon, K. (2018). Fast horizon detection in maritime images using region-of-interest. *International Journal of Distributed Sensor Networks*, *14*(7), Article 155014771879075. http://dx.doi.org/10.1177/1550147718790753.

Jin, X., Zhang, Y., & Jin, Q. (2016). Pulmonary nodule detection based on CT images using convolution neural network. In *2016 9th International symposium on computational intelligence and design: Vol. 1* (pp. 202–204). http://dx.doi.org/10.1109/iscid.2016.1053.

Kaya, V., Tuncer, S., & Baran, A. (2021). Detection and classification of different weapon types using deep learning. *Applied Sciences*, *11*(16), 7535. http://dx.doi.org/10.3390/app11167535.

Kmieć, M., Głowacz, A., & Dziech, A. (2012). Towards robust visual knife detection in images: Active appearance models initialised with shape-specific interest points. *Communications in Computer and Information Science*, *287*, 148–158. http://dx.doi.org/10.1007/978-3-642-30721-8_15.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, *25*, 1097–1105. http://dx.doi.org/10.1145/3065386.

Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J., Krasin, I., & Pont-Tuset, J. (2020). The open images dataset V4. *International Journal of Computer Vision*, *128*(7), 1956–1981. http://dx.doi.org/10.1007/s11263-020-01316-z.

Lamas, A., Tabik, S., Montes, A. C., Pérez-Hernández, F., García, J., & Olmos, R. (2022). Human pose estimation for mitigating false negatives in weapon detection in video-surveillance. *Neurocomputing*, *489*, 488–503. http://dx.doi.org/10.1016/j.neucom.2021.12.059.

Li, J., Liang, X., Wei, Y., Xu, T., Feng, J., & Yan, S. (2017). Perceptual generative adversarial networks for small object detection. In *2017 IEEE conference on computer vision and pattern recognition* (pp. 1951–1959). http://dx.doi.org/10.1109/cvpr.2017.211.

Lin, T., Dollar, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature pyramid networks for object detection. In *2017 IEEE conference on computer vision and pattern recognition* (pp. 2117–2125). http://dx.doi.org/10.1109/cvpr.2017.106.

Lin, T., Maire, M., Belongie, S., Hays, J., Perona, P., & Ramanan, D. (2014). Microsoft COCO: Common objects in context. In *Computer vision – ECCV 2014* (pp. 740–755). http://dx.doi.org/10.1007/978-3-319-10602-1_48.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., & Fu, C. (2016). SSD: Single shot MultiBox detector. In *Computer vision – ECCV 2016: Vol. 9905* (pp. 21–37). http://dx.doi.org/10.1007/978-3-319-46448-0_2.

Lowe, D. (1999). Object recognition from local scale-invariant features. In *Proceedings of the seventh IEEE international conference on computer vision*: *Vol. 2*, (pp. 1150–1157). http://dx.doi.org/10.1109/iccv.1999.790410.

Ma, X., Niu, Y., Gu, L., Wang, Y., Zhao, Y., & Bailey, J. (2021). Understanding adversarial attacks on deep learning based medical image analysis systems. *Pattern Recognition*, *110*, Article 107332. http://dx.doi.org/10.1016/j.patcog.2020.107332.

Malo-Perisé, P., & Merseguer, J. (2022). The socialized architecture: A software engineering approach for a new cloud. *Sustainability*, *14*(4), 2020. http://dx.doi.org/10.3390/su14042020.

Maqueda, A. I., Loquercio, A., Gallego, G., Garcia, N., & Scaramuzza, D. (2018). Event-based vision meets deep learning on steering prediction for self-driving cars. In *2018 IEEE/CVF conference on computer vision and pattern recognition* (pp. 5419–5427). http://dx.doi.org/10.1109/cvpr.2018.00568.

Minaeian, S., Liu, J., & Son, Y. (2018). Effective and efficient detection of moving targets from a UAV's camera. *IEEE Transactions on Intelligent Transportation Systems*, *19*(2), 497–506. http://dx.doi.org/10.1109/tits.2017.2782790.

Narejo, S., Pandey, B., vargas, D. Esenarro, Rodriguez, C., & Anjum, M. R. (2021). Weapon detection using YOLO V3 for smart surveillance system. *Mathematical Problems in Engineering*, *2021*, 1–9. http://dx.doi.org/10.1155/2021/9975700.

Nassif, A. B., Shahin, I., Attili, I., Azzeh, M., & Shaalan, K. (2019). Speech recognition using deep neural networks: A systematic review. *IEEE Access*, *7*, 19143–19165. http://dx.doi.org/10.1109/access.2019.2896880.

Olmos, R., Tabik, S., & Herrera, F. (2018). Automatic handgun detection alarm in videos using deep learning. *Neurocomputing*, *275*, 66–72. http://dx.doi.org/10.1016/j.neucom.2017.05.012.

Olmos, R., Tabik, S., Lamas, A., Pérez-Hernández, F., & Herrera, F. (2019). A binocular image fusion approach for minimizing false positives in handgun detection with deep learning. *Information Fusion*, *49*, 271–280. http://dx.doi.org/10.1016/j.inffus.2018.11.015.

Opitz, D., & Maclin, R. (1999). Popular ensemble methods: An empirical study. *Journal of Artificial Intelligence Research*, *11*, 169–198. http://dx.doi.org/10.1613/jair.614.

Pérez-Hernández, F., Tabik, S., Lamas, A., Olmos, R., Fujita, H., & Herrera, F. (2020). Object detection binary classifiers methodology based on deep learning to identify small objects handled similarly: Application in video surveillance. *Knowledge-Based Systems*, *194*, Article 105590. http://dx.doi.org/10.1016/j.knosys.2020.105590.

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *2016 IEEE conference on computer vision and pattern recognition* (pp. 779–788). http://dx.doi.org/10.1109/cvpr.2016.91.

Redmon, J., & Farhadi, A. (2017). YOLO9000: Better, faster, stronger. In *2017 IEEE conference on computer vision and pattern recognition* (pp. 6517–6525). http://dx.doi.org/10.1109/cvpr.2017.690.

Redmon, J., & Farhadi, A. (2018). Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767.

Ren, S., He, K., Girshick, R., & Sun, J. (2017). Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *39*(6), 1137–1149. http://dx.doi.org/10.1109/tpami.2016.2577031.

Romero, D., & Salamea, C. (2019). Convolutional models for the detection of firearms in surveillance videos. *Applied Sciences*, *9*(15), 2965. http://dx.doi.org/10.3390/app9152965.

Roy, S., Das, N., Kundu, M., & Nasipuri, M. (2017). Handwritten isolated Bangla compound character recognition: A new benchmark using a novel deep learning approach. *Pattern Recognition Letters*, *90*, 15–21. http://dx.doi.org/10.1016/j.patrec.2017.03.004.

Salido, J., Lomas, V., Ruiz-Santaquiteria, J., & Deniz, O. (2021). Automatic handgun detection with deep learning in video surveillance images. *Applied Sciences*, *11*(13), 6085. http://dx.doi.org/10.3390/app11136085.

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

Singh, A., Anand, T., Sharma, S., & Singh, P. (2021). IoT based weapons detection system for surveillance and security using YOLOV4. In *2021 6th International Conference on Communication and Electronics Systems* (pp. 488–493). http://dx.doi.org/10.1109/icces51350.2021.9489224.

Singh, T., & Vishwakarma, D. K. (2019). Human activity recognition in video benchmarks: A survey. *Lecture Notes in Electrical Engineering*, *526*, 247–259. http://dx.doi.org/10.1007/978-981-13-2553-3_24.

Sommer, L. W., Schuchert, T., & Beyerer, J. (2017). Fast deep vehicle detection in aerial images. In *2017 IEEE winter conference on applications of computer vision* (pp. 311–319). http://dx.doi.org/10.1109/wacv.2017.41.

Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. (2017). Inception-v4, inception-resnet and the impact of residual connections on learning. In *Proceedings of the AAAI conference on artificial intelligence*: *Vol. 31*, (1), http://dx.doi.org/10.1609/aaai.v31i1.11231.

Szegedy, C., Liu, Wei, Jia, Yangqing, Sermanet, P., Reed, S., & Anguelov, D. (2015). Going deeper with convolutions. In *2015 IEEE conference on computer vision and pattern recognition* (pp. 1–9). http://dx.doi.org/10.1109/cvpr.2015.7298594.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *2016 IEEE conference on computer vision and pattern recognition* (pp. 2818–2826). http://dx.doi.org/10.1109/cvpr.2016.308.

Tiwari, R. K., & Verma, G. K. (2015). A computer vision based framework for visual gun detection using Harris interest point detector. *Procedia Computer Science*, *54*, 703–712. http://dx.doi.org/10.1016/j.procs.2015.06.083.

Tiwari, R. K., & Verma, G. K. (2015). A computer vision based framework for visual gun detection using SURF. In *2015 international conference on electrical, electronics, signals, communication and optimization* (pp. 1–5). http://dx.doi.org/10.1109/eesco.2015.7253863.

Tompson, J., Goroshin, R., Jain, A., LeCun, Y., & Bregler, C. (2015). Efficient object localization using convolutional networks. In *2015 IEEE conference on computer vision and pattern recognition* (pp. 648–656). http://dx.doi.org/10.1109/cvpr.2015.7298664.

Tong, K., Wu, Y., & Zhou, F. (2020). Recent advances in small object detection based on deep learning: A review. *Image and Vision Computing*, *97*, Article 103910. http://dx.doi.org/10.1016/j.imavis.2020.103910.

Velastin, S. A., Boghossian, B. A., & Vicencio-Silva, M. A. (2006). A motion-based image processing system for detecting potentially dangerous situations in underground railway stations. *Transportation Research Part C (Emerging Technologies)*, *14*(2), 96–113. http://dx.doi.org/10.1016/j.trc.2006.05.006.

Verma, G. K., & 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 - ICCCT-2017* (pp. 84–88). http://dx.doi.org/10.1145/3154979.3154988.

Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition: Vol. 1* (p. I). http://dx.doi.org/10.1109/cvpr.2001.990517.

Wilson, P. I., & Fernandez, J. (2006). Facial feature detection using Haar classifiers. *Journal of Computing Sciences in Colleges*, *21*(4), 127–133. http://dx.doi.org/10.5555/1127389.1127416.

Worsham, J., & Kalita, J. (2020). Multi-task learning for natural language processing in the 2020s: Where are we going? *Pattern Recognition Letters*, *136*, 120–126. http://dx.doi.org/10.1016/j.patrec.2020.05.031.

Zeiler, M. D., & Fergus, R. (2014). Visualizing and understanding convolutional networks. In *Computer vision – ECCV 2014* (pp. 818–833). http://dx.doi.org/10.1007/978-3-319-10590-1_53.

Zhan, S., Tao, Q., & Li, X. (2016). Face detection using representation learning. *Neurocomputing*, *187*, 19–26. http://dx.doi.org/10.1016/j.neucom.2015.07.130.

Zivkovic, Z. (2004). Improved adaptive gaussian mixture model for background subtraction. In *Proceedings of the 17th international conference on pattern recognition, 2004: Vol. 2* (pp. 28–31). http://dx.doi.org/10.1109/icpr.2004.1333992.

Żywicki, M., Matiolański, A., Orzechowski, T. M., & Dziech, A. (2011). Knife detection as a subset of object detection approach based on Haar cascades. In *In proceedings of 11th international conference pattern recognition and information processing* (pp. 139-142).