Mathematical Biosciences and Engineering

MBE, 20 (12): 21611–21625.
DOI: 10.3934/mbe.2023956
Received: 06 October 2023
Revised: 04 November 2023
Accepted: 07 November 2023
Published: 06 December 2023

http://www.aimspress.com/journal/MBE

Research article

R-CNN and YOLOV4 based Deep Learning Model for intelligent detection of weaponries in real time video

K. P. Vijayakumar [1,*], K. Pradeep [1], A. Balasundaram[1,2] and A. Dhande[1]

[1] School of Computer Science and Engineering, Vellore Institute of Technology, Chennai, India
[2] Center for Cyber Physical Systems, Vellore Institute of Technology, Chennai, India

* **Correspondence:** Email: vijayakumar.kp@vit.ac.in.

Abstract: The security of civilians and high-profile officials is of the utmost importance and is often challenging during continuous surveillance carried out by security professionals. Humans have limitations like attention span, distraction, and memory of events which are vulnerabilities of any security system. An automated model that can perform intelligent real-time weapon detection is essential to ensure that such vulnerabilities are prevented from creeping into the system. This will continuously monitor the specified area and alert the security personnel in case of security breaches like the presence of unauthorized armed people. The objective of the proposed system is to detect the presence of a weapon, identify the type of weapon, and capture the image of the attackers which will be useful for further investigation. A custom weapons dataset has been constructed, consisting of five different weapons, such as an axe, knife, pistol, rifle, and sword. Using this dataset, the proposed system is employed and compared with the faster Region Based Convolution Neural Network (R-CNN) and YOLOv4. The YOLOv4 model provided a 96.04% mAP score and frames per second (FPS) of 19 on GPU (GEFORCE MX250) with an average accuracy of 73%. The R-CNN model provided an average accuracy of 71%. The result of the proposed system shows that the YOLOv4 model achieves a higher mAP score on GPU (GEFORCE MX250) for weapon detection in surveillance video cameras.

Keywords: security; weapon detection; CNN; YOLOv4; deep learning

1. Introduction

For a long time, humans were responsible for security (either through security personnel, or police). Although they are appointed to oversee a checkpoint, there are still security breaches. In addition, after the security breaches, the offender was not caught in the act immediately. The forensic team had to investigate the situation to catch the offender, which took a lot of time. Prevention is better than cure, i.e. if we can catch this offender before he/she makes a move, we can prevent loss of life. This method requires high deployment of security personnel and continuous monitoring of the area. This is not efficient as it can be costly and the attacker can make a move whenever there is a change in security personnel shifts.

Surveillance cameras are installed in many places to prevent crimes, and security personnel are needed to keep an eye on all cameras. Once a crime happens, security personnel check the recorded video to gather the essential evidence for further investigation. Nowadays, criminal activities are done by using handheld weapons. It is evident from several studies [1–6] that handheld weapons are widely used during criminal activities such as theft, robbery, kidnapping, terrorist attack, assassination, etc. Deployment of surveillance video cameras or control cameras are the foremost solutions for taking suitable actions at an early stage [7,8]. Thus, it is essential to build a system that will learn to detect frightening entities.

Deep learning is a subset of machine learning that improves the performance of tasks in security control systems [9]. Artificial intelligence and computer vision have facilitated the detection and classification of entities based on application specifications. There are several applications, such as security feeds, autonomous vehicles, etc. In India, crimes such as theft, burglary and kidnapping are committed not only by using handheld weapons (guns or knives), but also by using other kinds of weapons such as axes, rifles, swords, iron rods, etc.

The objective is to build a weapon detection system to classify different types of weapons in real-time to decrease the aforementioned incidents, and these incidents can be controlled using an early alarm system by alerting the security personnel to take immediate action. The proposed system uses a live video feed from CCTV cameras to classify five types of weapons (Axe, Knife, Pistol, Rifle, and Sword) by employing deep learning approaches. The proposed system is employed with faster R-CNN and YOLOv4 models to assess and compare the classification performance of proposed models.

The rest of the paper is organized as follows: Section 2 describes the related work. The proposed system is presented in Section 3. The results of YOLOv4 and Faster R-CNN are discussed and compared in section 4. Section 5 concludes this work.

2. Literature survey

The authors in [10] used thermal/IR images with the conventional RGB image (or HSV) to capture an image of the area to be monitored, and used a canny detection algorithm to identify hidden objects. The authors claimed that the fusion of both types of images helped in image noise reduction and retention of critical features of the image. The system was developed with the YOLOv4 algorithm to detect the weapons [11–13] and the authors claimed that in their implementation, they achieved 70% accuracy in low quality videos, whereas the system achieved 95% accuracy in high quality videos. The system was deployed with the help of the IoT (Internet of

Things). In [14], authors implemented automatic gun or weapon detection using a convolutional neural network (CNN) based single shot multi box detector (SSD) and Faster R-CNN algorithms. Two datasets were used in their implementation. One dataset has pre-labeled images and the other dataset consists of manually labeled images. It was observed that both SSD and RCNN algorithms achieve good accuracy, but in real-life implementations, the choice of algorithm is based on the trade-off between accuracy and speed.

In [15], the system was implemented using the YOLOv3 object detection model by training it on their customized dataset. From their result, it was noticed that YOLOv3 outperforms well than YOLOv2 and traditional convolutional neural networks. The authors also mentioned that their implementation did not require high-end GPUs as they used transfer learning for training their models. The system implemented in the YOLOv3 algorithm detects guns with a mean average precision of 95%. In [16], the proposed method combines multiple sensors by hybrid fusion of sigmoidal Hadamard wavelet transform and PCA basis functions. For weapon recognition and detection, the proposed system implements image segmentation and K-means support vector machines. The authors in [17] developed a new algorithm to fuse a color visual image and a corresponding IR image for such a concealed weapon detection application. The fused image obtained by the developed model maintains the high resolution of the visual image incorporates any concealed weapons detected by the IR sensor, and keeps the natural color of the visual image.

In [7], the system was implemented using Faster R-CNN to detect handguns in video, and this system was integrated with an alarm. The study shows that the developed system triggers an alarm after five successful true positives in less than 0.2 seconds, in 27 out of 30 scenes. The alarm activation time per interval (AATpI) metric was used to assess the performance of detection. Weapon detection in luggage in airports using linear and non-linear pseudo-coloring maps and a single high energy X-Ray system was implemented in [18]. For the input to the color mapping schemes, various enhanced images, grey-scaled images, and segmented scenes were used.

The authors provided a study on various machine and deep learning algorithms for detecting weapons such as knives, guns, and rifles [19], and also presented a comparative study of performances on machine and deep learning algorithms. In [20], authors developed a 350 GHz imaging system to detect concealed weapons. Due to the system's wideband operation, the object can be visualized in three dimensions and provide ranging information. The authors of the paper [21] developed a weapon detection system to detect cold steel weapons using a CNN. The authors claim that their implementation helps in detecting weapons whose outer body surfaces blur the image capture due to surface reflectance. The authors also integrated the system with an automatic alarm system. The paper [22] discusses the implementation of a weapon detection system using F-RCNN. The author used two approaches viz.: GoogleNet and SqueezeNet using a CNN as a base. The author concluded that the SqueezNet implementation performed better than GoogleNet.

The work carried out in [23] proposed a digital twin model for predicting wild fire by applying reduced order modeling, convolutional auto encoding, recurrent neural networks and latent data assimilation. The data provided by JULES INFERNO was used as input to the proposed model for foreseeing wildfires. The suggested digital twin model ran five hundred times faster for online forecasting without needing high performance computing clusters. The authors proposed a wildfire prediction model by applying machine learning and reduced order modeling techniques [24]. The forward and the inverse modeling were tested on two recent large wildfire events in California and achieved more accurate future forecast. A learning-based method to examine the precipitate area and size distribution in Cr-superalloys was developed [25]. Authors proposed a two-stage end to end, DT-SegNet approach to accomplish object detection and segmentation for electron microscopy imaging.

The research gap observed from the analysis of existing literature is given as follows: (i) many researchers have focused on detecting weapons like guns and knives from the video input, (ii) no standard weapons dataset was available for a weapon detection system, and (iii) several algorithms employed in weapon detection systems used various labeling and preprocessing procedures. Thus, labeled datasets used in one approach may not be suitable for other approaches. To the best of our knowledge, none of the existing literature had considered various types of weapons such as axes, swords, sickles, bill hooks, etc. To fulfill the research gap, the proposed system is employed with Faster R-CNN and YOLOv4 algorithms on a custom dataset for detecting various types of weapon classes such as axes, swords, guns, knives and pistols.

3. Proposed system

The proposed weapon detection system is implemented using two different algorithms (Faster R-CNN and YOLOv4) to detect five weapons (pistols, automatic rifles, axes, swords, and knives). The system identifies the weapons from the video fed into the system and saves a snapshot of the video frame where the weapon was identified. In this paper, the performances of both algorithms are compared by using evaluation metrics such as mean average precision (mAP), precision, recall, and F1 score.

3.1. Data acquisition

The dataset plays a vital role in machine and deep learning applications since there is no standard dataset to detect weapons. Consequently, 609 images were downloaded with the help of the internet. Table 1 describes the dataset used for training both algorithms. The dataset is classified into six classes: first one is the none class (without any weapon), second is the axe class, third is the knife class, fourth is the pistol class, fifth is the rifle class and sixth is the sword class. As part of this work, 80% of the images across each class was used for training and 20% of the images across each class was used for testing. The number of images used for training and testing for each class are shown in Table 1. Among the images used for training and testing, 60% of images in each class contained objects only while the remaining 40% contained objects in real time background.

Table 1. Dataset description**.**

Weapon	Number of images	Training Images	Testing Images
Axe	129	103	26
Knife	118	94	24
Pistol	120	96	24
Rifle	135	108	27
Sword	107	85	22
Total	609	486	123

In this dataset, for every weapon, approximately 50% of images contain only weapons that have no background or plain color background as shown in Figure 1, and the other 50% of images contains weapons in a real background, like a person holding a knife, a person firing a rifle, etc., as shown in Figure 2.

Figure 1. Weapon images with plain background.

Figure 2. Weapons in real life scenario images.

3.2. Dataset preprocessing

The images collected may not always be appropriate for training. For example, the image size may be too weird (either stretched or flat), and the weapon might not even be visible in the image due to a lot of clustering of other objects around it, watermarks on the images, poor quality of image etc. It is made sure that the dataset is balanced as much as possible for both Faster R-CNN and YOLOv4. Faster R-CNN and YOLOv4 algorithms require images (both train and test dataset) to be annotated. Annotation refers to drawing bounding boxes around the object to be detected. Figure 3 shows the usage of the annotation tool [21] used for this implementation.

The annotation tool is based on the JavaScript tool. Initially, it is necessary to select the images to be annotated and then the names of the selected images will appear in the upper-left box. Next, select a labels.txt file which contains the names of the classes (that is, names of weapons) with every name on a new line. The list of classes is then displayed in the middle-left box. After setting up the files, the image to be annotated must be selected from the list, as well as the class of the object. Now, with the help of a mouse cursor, a box has to be drawn around the object to be detected. The same process of file selection has to be repeated for all the images. After annotating the image, the coordinates must be saved with any one of the following options: COCO, YOLO, and PASCAL-VOC format. In the proposed system, two types of annotation formats are used, namely the PASCAL-VOC format and the YOLO format for Faster R-CNN and YOLOv4 respectively.

PASCAL-VOC is an abbreviation for Pattern Analysis, Statistical Modeling, and Computational Learning Visual Object Classes. This type of annotation format uses the XML format to save the details. The details of the format are given as follows: a) name of the folder in which the image is stored, b) name of the file (along with its extension), c) width of the image, d) Height of image, e) class or label of the image, and f) coordinates of the bounding boxes drawn (xmin, ymin, xmax, ymax). To save the details of multiple bounding boxes in the same image, steps (e) and (f) are repeated for every bounding box.

The YOLO format saves the following details of the image and the bounding boxes in the text (.txt) file: a) class (or label) number, b) normalized X-coordinate of the center of the bounding box, c) normalized Y-coordinate of the center of the bounding box, d) normalized width of the image and e) normalized height of the image. For multiple bounding boxes in the same image, the details from (a) to (e) from above are repeated.

Figure 3. Annotator tool for fixing bounding boxes.

3.3. Model selection

In the proposed system, Faster R-CNN and YOLOv4 are considered as weapon detection models from the analysis of existing literatures that these achieve better performance in terms of mAP.

3.3.1. Faster R-CNN

Faster region based convolutional neural network (Faster R-CNN) is an improved version of R-CNN. Faster R-CNN takes two inputs: the whole image and the proposed object to be detected [26]. A convolution feature map is created by processing the entire image by several convolution layers and max pooling layers, followed by extraction of the feature vector from the previously created feature map by region of interest pooling layer for every proposed object. The feature vector so created is fed to the fully connected layer which branches into two output layers: one classifies the object and the other creates coordinates of bounding boxes to be drawn on the image to show the detected object. Figure 4 shows the architecture of Faster R-CNN.

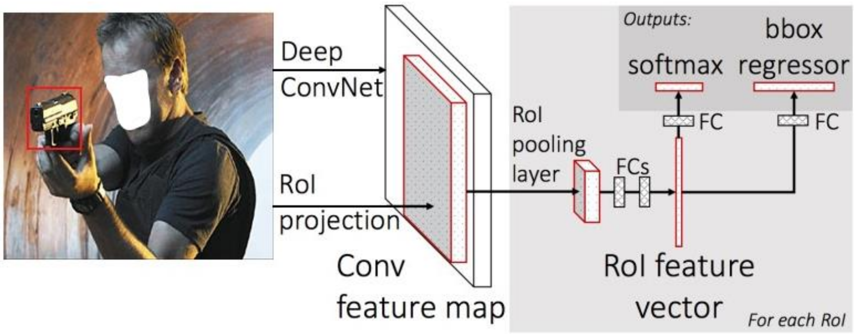

Figure 4. Architecture of Faster R-CNN.

3.3.2. YOLOv4

You Look Only Once version 4 (YOLOv4) consists of 4 stages of networks: input, backbone, neck and head. The input consists of an Image, Patches, and an Image Pyramid [18]. The backbone consists of 4 networks: VGG16, ResNet-50, ResNeXt-101 and Darknet53. The neck consists of the following networks: Feature Pyramid Network (FPN), Path Aggregation Network (PAN), spatial pyramid pooling (SPP), atrous spatial pyramid pooling (ASPP), RFB, SAM, BiFPN, NAS-FPN, FCFPN, ASFF and SFAM. The head consists of RPN, SSD, YOLO, RetinaNet, CornerNet, CenterNet, MatrixNet, FCOS, Faster R-CNN, R-FCN, Mask R-CNN, RepPoints. The architecture and flow diagram of YOLOv4 is shown in Figure 5.

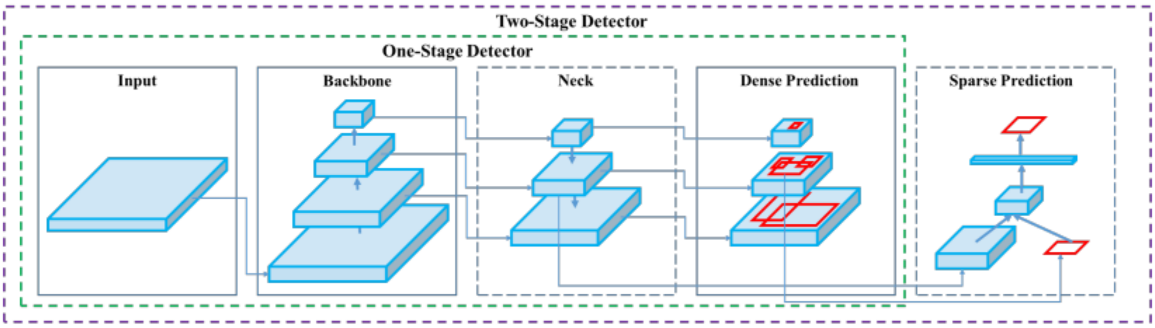

Figure 5. Architecture of YOLOv4.

4. Results and discussions

In the proposed system, models such as Faster R-CNN and YOLOv4 models are deployed by using several libraries, namely Pytorch, Numpy, Pandas, Sklearn, Os, albumenation, matplotlib, tqem and darknet framework. The models were trained on a computer equipped with an Intel Core i5 processor, Nvidia GeForce MX250 graphics card, and 8GB RAM.

In the proposed system, experiments were done for five different weapon classes, and the performance of the proposed Fast R-CNN and YOLOv4 models were evaluated using the test dataset (20% of the custom dataset) and compared. The implementation of Faster R-CNN uses Python libraries such as the Pytorch library (GPU enabled) and the OpenCV library. The algorithm was trained with the configurations as shown in Table 2.

Table 2. Configuration details of Fast R-CNN.

Configuration	Value
Training Device	GPU - Nvidia GEFORCE MX250
Batch size	1
Image resolution	192 x 192
Epochs	54
Learning Rate	0.001
Momentum	0.9
Weight Decay	0.0005
Train-Test Split	80% : 20%

Apart from the above mentioned configurations, a pre-trained fasterrcnn_resnet50_fpn model was used as transfer learning to speed up the learning process which in turn increases its accuracy. First, a Faster R-CNN model is created with the hyperparameters as mentioned in Table 2. Next, the Python code loads train and test images in memory, extracts the label, and retrieves the coordinates of the bounding boxes from the .xml files of the corresponding images. The size of the image and the bounding box coordinates are resized to the given parameter and then fed it into the model. The accuracy and loss graph of Faster R-CNN is given in Figure 6 and Figure 7. From Figure 8, it is observed that the model has started to learn after the second epoch and reached a success accuracy of 80%. Likewise, for loss it is also noticed from Figure 9 that the loss is reduced approximately below 0.1%.

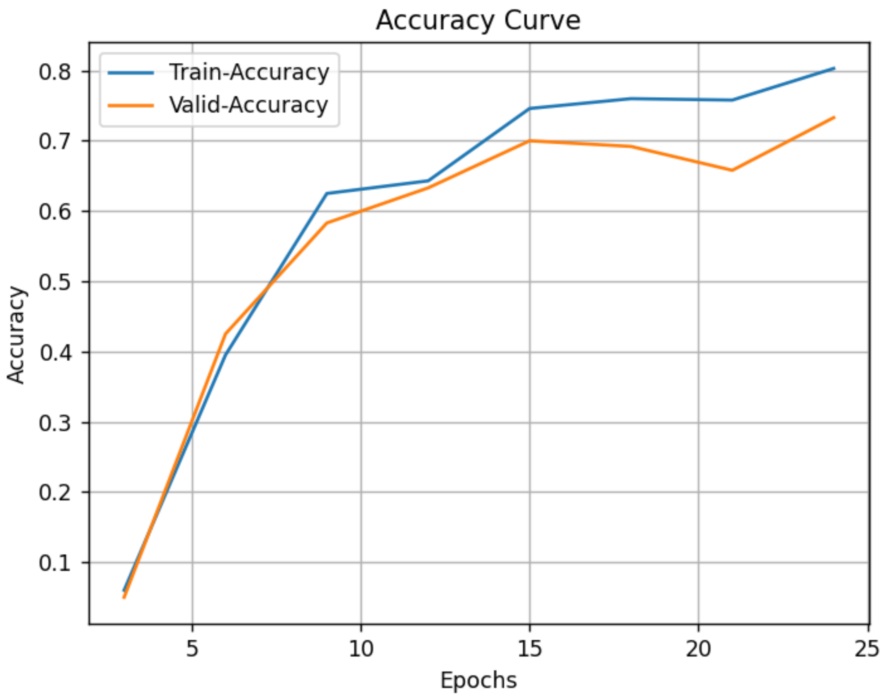

Figure 6. Faster R-CNN training and validation accuracy graph.

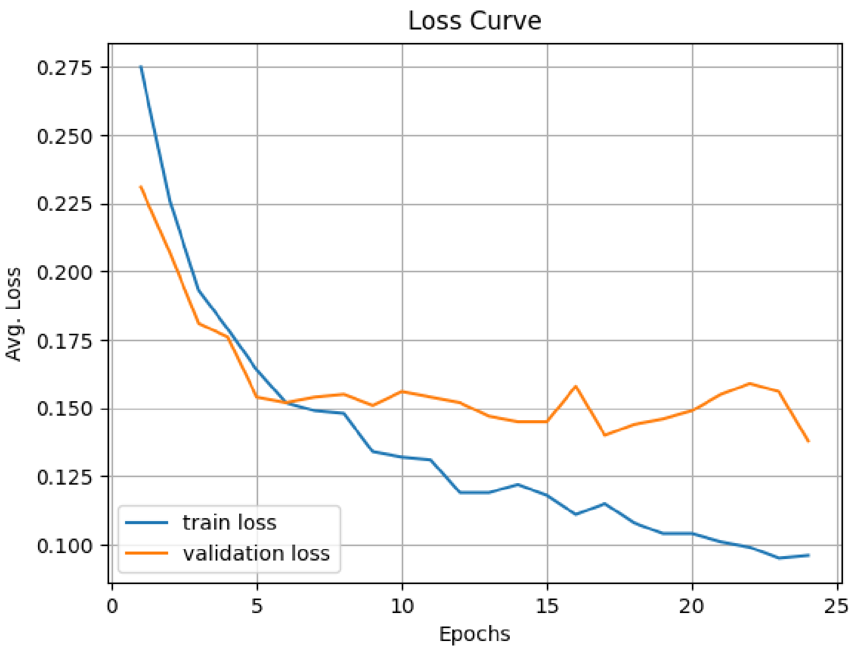

Figure 7. Faster R-CNN training and validation loss graph.

Next, the YOLOv4 model is implemented using a darknet framework which includes all the Python codes and dynamic library files required for training testing, and detection. The algorithm was trained with the configuration given in Table 3.

Table 3. Configuration details of YOLOv4.

Configuration	Value
Training Device	GPU - Nvidia GEFORCE MX250
Batch size	64
Subdivision	64
Image resolution	192 x 192
Channels	3
Max_Batches	10000
Learning Rate	0.001
Momentum	0.949
Decay	0.0005
Train-Test Split	80% : 20%

Table 4. Success rate of YOLOv4 model.

Weapon Classes	No.of Images	Average Precision in %	mAP in %
Axe	129	94.57	
Knife	118	96.36	
Pistol	120	98.94	96.04mAP@0.5IoU
Rifle	135	96.48	
Sword	107	93.85	

To evaluate the weapon images in the dataset, the mAP metric is used as shown in Table 4. The YOLOv4 model with a subdivision of 64 and image resolution of 192 x 192 with the default hyperparameters setting (as given in Table 3) illustrating mAP of 96.04 and an average loss of 1.2 is shown in Figure 8. It is observed from the outcome that the proposed model obtained an mAP of 96.04 than the model used in the existing system for detecting only pistols [10].

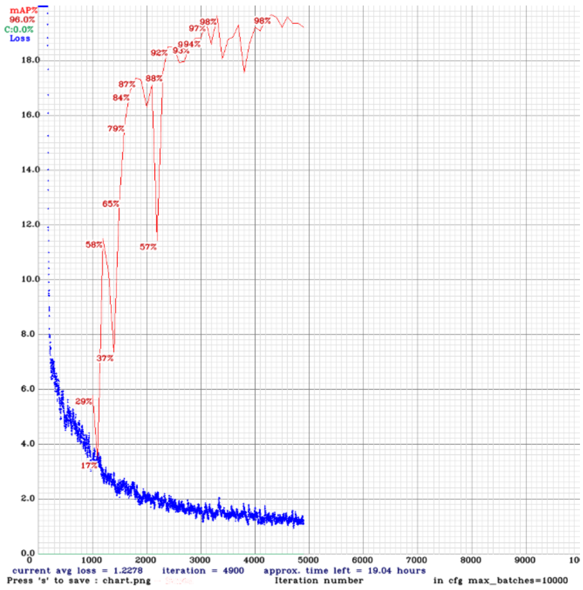

Figure 8. Model with subdivision of 64 and image resolution 192 x 192 with default hyperparameters setting illustrating mAP of 96.04 and an average loss of 1.2.

The proposed models Faster R-CNN and YOLOv4 are compared using the metrics precision, recall, F1 score, and average precision as given in Table 5 and Figure 9. From Table 5 and Figure 9, it is evident that the average precision of YOLOv4 is better at detecting axes, knives, pistol and swords, whereas Faster R-CNN achieved 100% average precision in detecting rifle.

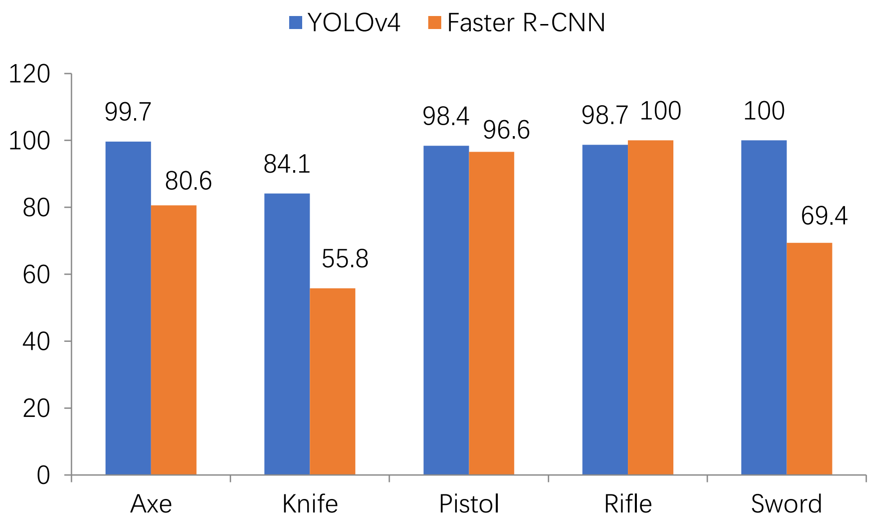

Figure 9. Comparison of YOLOv4 with Faster R-CNN using Average Precision.

Table 5. Average precision of faster R-CNN and YOLOv4 models.

Weapon Classes	YOLOv4 in %	Faster R-CNN in %
Axe	99.7	80.6
Knife	84.1	55.8
Pistol	98.4	96.6
Rifle	98.7	100
Sword	100	69.4

Figure 10 and Figure 11 refer to the precision, recall, and F1 score of YOLOv4 and Faster R-CNN. A video input downloaded from YouTube was given to the YOLOv4 model and was able to give 19 FPS on Nvidia GEFORCE MX250 GPU and able to detect weapons clearly as shown in Figure 10. The same video input was given to the Faster R-CNN model frame-by-frame, but the performance was terrible. The highest performance of only 2 FPS (frames per second) was recorded during testing on a test video as shown in Figure 11. The Gmean results for fast RCNN and YOLOv4 models are provided in Table 6.

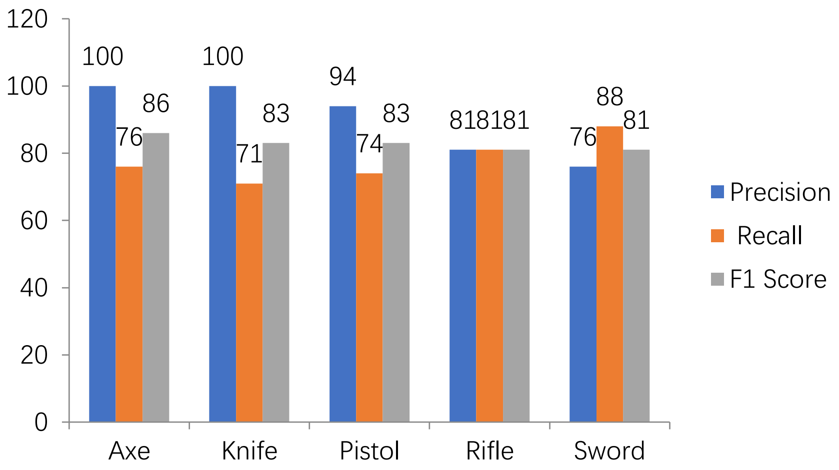

Figure 10. Precision, Recall and F1 Score of YOLOv4.

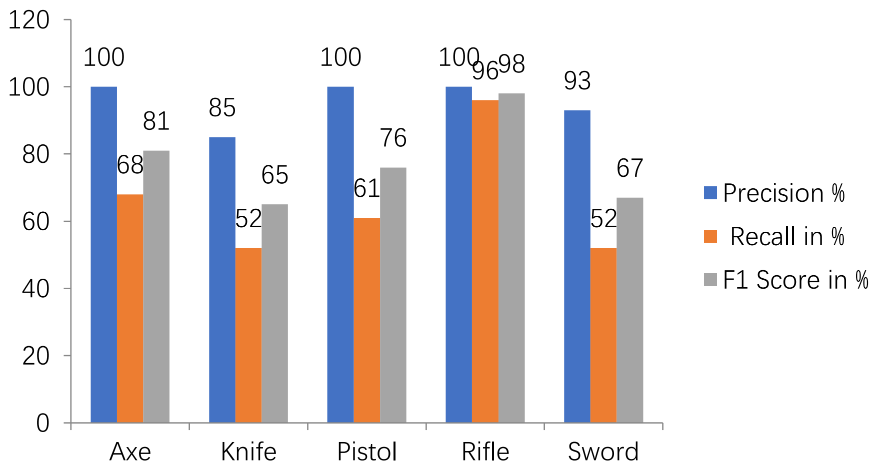

Figure 11. Precision, Recall and F1 Score of Faster R-CNN.

Table 6. GMean of Faster RCNN and YOLOV4.

Models	Class	GMean
Faster RCNN	None	0.55703
	Axe	0.83057
	Knife	0.88298
	Pistol	0.932978
	Rifle	0.92608
	Sword	0.90787
YOLOV4	None	0.83454
	Axe	0.88465
	Knife	0.89062
	Pistol	0.97563
	Rifle	0.97563
	Sword	0.83452

In the proposed system, Faster R-CNN and YOLOv4 models were developed to determine whether a person has a weapon and furthermore classify which kind of the 5 various weapons (Axe, Knife, Pistol, Rifle and Sword). From the experiments, it was evident that the YOLOv4 achieved higher performance in detecting various types of weapons compared to Faster R-CNN. Further, the proposed models for detecting various types of weapons are compared with the existing literature as shown in Table 7.

Table 7. Comparison of proposed model with existing systems.

Models	Types of Weapons	Algorithms	Dataset Used	Results
Bhatti M.T. et.al [10]	Pistols	VGG16, Inception-V3, Inception-ResnetV2, SSDMobileNetV1, FRIRv2, YOLOv3 and YOLOv4	Custom Dataset	mAP of 91.73% F1-score 91%
Jain A et. al [11]	Guns: Machine Gun, Submachine Gun, Assault Rifle, Pistol	Haar Cascade Classifier	Custom Dataset	Accuracy of 95 %
Singh A et. al [14]	Knife, Guns	YOLOv4	Kaggle Dataset and Google images	Accuracy of 95%
JainHA et. a [15]	Guns	CNN based SSD, Faster R-CNN	Custom Dataset	mAP of 74%
Sanam N et.al [16]	Guns	YOLOv3	Custom Dataset	mAP of 95%
Olmos, R et. al [19]	Pistol	Faster R-CNN	Custom dataset	Not mentioned
Castillo A et.al [21]	Cold steel weapons	CNN R-FCN(ResNet101)	Custom Dataset	F1 –Score 93%
Proposed System	Axe, Knife, Pistol, Rifle and Sword	Faster R-CNN and YOLOv4	Custom Dataset	mAP of 96.04%

5. Conclusions

It is essential to deploy an instinctive weapon detection system in houses, apartments, and public places to evade criminal activities before they happen. The proposed works presented two models Faster R-CNN and YOLOv4 for detecting various kinds of weapons such as axe, knife, pistol, rifle and sword and alerting the security personnel. The models were assessed by using a custom dataset. The YOLOv4 model provided elevated performance and achieved 96.04 mAP with 19 FPS. YOLOv4 performed better than the Faster R-CNN in terms of average precision for detecting the axe, knife, pistol and sword weapons. This underlines that the proposed system outperforms the contemporary systems. In the future, the system can be applied to CCTV cameras for detecting weapons, and the system can be implemented using high performance GPU with higher FPS.

Also, the future scope will be directed towards enhancing the dataset with additional types of weapons and objects that could pose threats and challenging conditions such as low light conditions, rainy environment, etc. based frame sequences. Also, the proposed system can be extended to other types of weapons.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

The authors wish to thank the management of VIT, Chennai for their support in carrying out this research work. The authors received no specific funding for this work.

Conflict of interest

The authors declare there is no conflict of interest.

References

1. G. Raturi, P. Rani, S. Madan and S. Dosanjh, ADoCW: An automated method for detection of concealed weapon, in *Proc. International Conference on Image Information Processing (*ICIIP*)*, Shimla, India, (2019), 181–186. https://dx.doi.org/10.1109/ICIIP47207.2019.8985972
2. J. Salido, V. Lomas, J. Ruiz-Santaquiteria, O. Deniz, Automatic handgun detection with deep learning in video surveillance images, *Appl. Sci.*, **11** (2021), 1–17. http://dx.doi.org/10.3390/app11136085
3. J. Lim, M. I. Al Jobayer, V. M. Baskaran, J. M. Lim, K. Wong, et al., Gun detection in surveillance videos using deep neural networks, in *Proc. APSIPA ASC*, Lanzhou, China, (2019), 1998–2002. http://dx.doi.org/10.1109/APSIPAASC47483.2019.9023182
4. J. Yuan, C. Guo, A deep learning method for detection of dangerous equipment, in *Proc. ICIST, Cordoba, Granada, and Seville*, Spain, (2018), 159–164. http://dx.doi.org/10.1109/ICIST.2018.8426165
5. G. K. Verma, A. Dhillon, A handheld gun detection using faster R-CNN deep learning, *in Proc. ICCT*, Allahabad, India, (2017), 84–88. http://dx. doi.org/10.1145/3154979.3154988

6. A. Warsi, M. Abdullah, M. N. Husen, M. Yahya, Automatic handgun and Knife detection algorithms: A review, in *Proc. IMCOM*, Taichung, Taiwan, (2020), 1–9. http://dx.doi.org/10.1109/IMCOM48794.2020.9001725
7. R. Olmos, S. Tabik, F. Herrera, Automatic handgun detection alarm in videos using deep learning, *Neurocomputing*, **275** (2018), 66–72. https://doi.org/10.1016/j.neucom.2017.05.012
8. M. Zahrawi,K. Shaalan, Improving video surveillance systems in banks using deep learning technique, *Sci. Rep.*, **13** (2023), 1–16. https://doi.org/10.1038/s41598-023-35190-9
9. L. Alzubaidi, J. Zhang, A. J. Humaidi, A. AI-Dujaili, Y. Duan, et al., Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions, *J. Big Data*, **8** (2021), 1–74. https://doi.org/10.1186/s40537-021-00444-8
10. M. T. Bhatti, M. G. Khan, M. Aslam, M. J. Fiaz, Weapon detection in real-time CCTV videos using deep learning, *IEEE Access*, **9** (2021), 34366–34382. https://doi.org/10.1109/ACCESS.2021.3059170
11. A. Jain, Aishwarya,G. Garg, Gun detection with model and type recognition using Haar Cascade classifier, in *Proc. ICSSIT*, Tirunelveli, India, (2020), 419–423. https://doi.org/10.1109/ICSSIT48917.2020.9214211
12. S. Gosain, A. Sonare, S. Wakodkar, Concealed weapon detection using image processing and machine learning, *IJRASET J. Res. Appl. Sci. Eng. Technol.*, **9** (2021), 1–13. https://doi.org/10.22214/ijraset.2021.39506
13. A. Singh, T. Anand, S. Sharma, P. Singh, IoT based weapons detection system for surveillance and security using YOLOV4, in *Proc. ICCES*, Coimbatre, India, (2021), 488–493. https://doi.org/10.1109/ICCES51350.2021.9489224
14. H. Jain, A. Vikram, Mohana, A. Kashyap, A. Jain, Weapon detection using artificial intelligence and deep learning for security applications, in *Proc. ICESC*, Coimbatore, India, (2020), 193–198. https://doi.org/10.1109/ICESC48915.2020.9155832
15. N. Sanam, P. Bishwajeet, E. V. Doris, C. Rodriguez, M. R. Anjum, Weapon detection using YOLO V3 for smart surveillance system, *Hindawi Math. Problems Eng.*, **2021** (2021), 1–9. https://doi.org/10.1155/2021/9975700
16. A. W. Altaher, S. K. Abbas, Image processing analysis of sigmoidal Hadamard wavelet with PCA to detect hidden object, *TELKOMNIKA Telecomm. Comput. Electron. Control*, **18** (2020), 1–8. http://doi.org/10.12928/telkomnika.v18i3.13541
17. Z. Y. Xue, R. S. Blum, Concealed weapon detection using color image fusion, in *Proc. ICIF*, Cairns, QLD, Australia, (2003), 622–627. https://doi.org/10.1109/ICIF.2003.177504
18. B. R. Abidi, Y. Zheng, A. V. Gribok, M. A. Abidi, Improving weapon detection in single energy X-ray images through Pseudocoloring, *IEEE Transact. Syst. Man Cybern. Part C Appl. Rev.*, **36** (2006), 784–796. https://doi.org/10.1109/TSMCC.2005.855523
19. P. Yadav, N. Gupta, P. K. Sharma, A comprehensive study towards high-level approaches for weapon detection using classical machine learning and deep learning methods, *Expert Syst. Appl.*, **212** (2022), 1–20. https://doi.org/10.1016/j.eswa.2022.118698
20. D. M. Sheen, T. E. Hall, R. H. Severtsen, D. L. McMakin, B. K. Hatchell, et al., Active wideband 350GHz imaging system for concealed-weapon detection, in *Proc. International Society for Optical Engineering (SPIE) Defense, Security and Sensing 2009*, Orlando, Florida, United States, 7309 (2009). https://doi.org /10.1117/12.817927

21. A. Castillo, S. Tabik, F. Pérez, R. Olmos, F. Herrera, Brightness guided preprocessing for automatic cold steel weapon detection in surveillance videos with deep learning, *Neurocomputing*, **330** (2019), 151–161. https://doi.org/10.1016/j.neucom.2018.10.076
22. M. M. Fernandez-Carrobles, O. Deniz, F. Maroto, Gun and Knife detection based on faster R-CNN for video surveillance, *Pattern Recogn. Image Anal.*, **11868** (2019), 441–452. https://doi.org/10.1007/978-3-030-31321-0_38
23. C. Zhong, S. Cheng, M. Kasoar, R. Arcucci, Reduced-order digital twin and latent data assimilation for global wildfire prediction, *Nat. Hazards Earth Syst. Sci.*, **23** (2023), 1755–1768. https://doi.org/10.5194/nhess-23-1755-2023
24. S. Cheng, Y. Jin, S. P. Harrison, C. Quilodrán-Casas, I. C. Prentice, Guo Y-K, et al., Parameter flexible wildfire prediction using machine learning techniques: Forward and inverse modelling, *Remote Sensing*, **14133228** (2022), 1–24. https://doi.org/10.3390/rs14133228
25. Z. Y. Xia, K. Ma, S. B. Cheng, T. Blackburn, Z. L. Peng, K. W. Zhu, et al., Accurate identification and measurement of the precipitate area by two-stage deep neural networks in novel chromium-based alloys, *Phys. Chem. Chem. Phys.*, **25** (2023), 15970–15987.
26. R. Girshick, Fast R-CNN, *in Proc. IEEE ICCV*, Santiago, Chile, (2015), pp. 1440–1448. https://doi.org/10.1109/ICCV.2015.169