Real-Time Image Based Weapon Detection Using YOLO Algorithms

Manoj Gali[1], Sunita Dhavale[1](✉), and Suresh Kumar[2]

[1] Defence Institute of Advanced Technology (DIAT), Pune, India
sunitadhavale@gmail.com

[2] Defence Institute of Psychological Research (DIPR), Delhi, India

Abstract. From last few years country faces major challenge in maintaining security standards particularly in public and highly sensitive places such as airports, movie theatres, stadiums, and national parks, etc. The offsite and onsite planers use many tactics to resist authority and disrupt and to add turmoil in order to achieve their goals and objectives. These tactics can be planned or unplanned. Many crowd management experts suggest that the availability of suspicious objects such as Camera, Handgun, Rifles, Dagger, Sword and Sticks at sight before or during the event can be an indication of upcoming threat or any unlawful activities and their identification may help security forces in their proactive management and control of any destructive activities. In this research work, we generated a novel dataset "DIAT-Weapon" Dataset for weapon object detection using web scraping techniques. DIAT-Weapon Dataset consists of 2712 images divided into six categories mainly: Camera, Handgun, Rifles, Dagger, Sword and Sticks. We customized and fine-tuned YOLOv4 models to classify and position the six types of harmful objects i.e. Camera, Handgun, Rifles, Dagger, Sword, and Sticks in real time. To achieve real-time faster performance and better detection accuracy, YOLOv4 is fine-tuned, and the preset anchors trained on DIAT-Weapon annotated dataset. Using a series of YOLOv4 object detection algorithms, we demonstrated experiments on our dataset, achieving 0.63 mAP. To our best knowledge, this is the first work that utilizes customized YOLOv4 model for real-time localization and classification of weapon objects into six different categories. To our best knowledge, this is the first work that utilizes customized YOLOv4 model for real time localization and classification of weapon objects into six different categories.

Keywords: Weapon detection · YOLOv4 · Deep learning · Real time object detection

1 Introduction

A minor security breach in identification of suspicious elements who carrying such harmful or restricted weapons/objects at such sensitive locations might have serious consequences in terms of national security. This becomes more challengeable particularly in public spaces such as airports, protests, movie theatres, stadiums, and national parks where maintaining security standards is necessary to ensure safety of infostructure and

M. Singh et al. (Eds.): ICACDS 2022, CCIS 1614, pp. 173–185, 2022.
https://doi.org/10.1007/978-3-031-12641-3_15

national themes including human life. DIPR studies [31, 32] suggest that the availability of violent material like riffle, churra, sword, sticks, handgun, grandees, bomb, etc. at location or even showing the violent material can escalate aggression and violent behavior in the people. The early detection of these material or the person who carrying these vulnerable materials may help security forces to apply their best crowd management strategies. Although the human visual framework has performed admirably in terms of monitoring, humans can be slow, expensive, and corruptible in the long run, can expose people on the ground to danger. With the escalation of technologies in hardware has given an opportunity to study and monitor the situations physically using videos and CCTV live footages but the amount of data generated was enormous, and studying each footage was a daunting process for humans. With the advancement of computational powers and availability of intensive dataset, training large deep learning networks for computer vision applications is possible. In the domain of object detection, CNN-based architecture algorithms like R-CNN (Regions with CNN features), Fast R-CNN, Faster R-CNN, SSD (Single Shot Multi-Box Detector), YOLO (You Only Look Once), and its versions, perform exceptionally well on real-time object detection, which in turn paved the way for machine-based surveillance systems (Fig. 1).

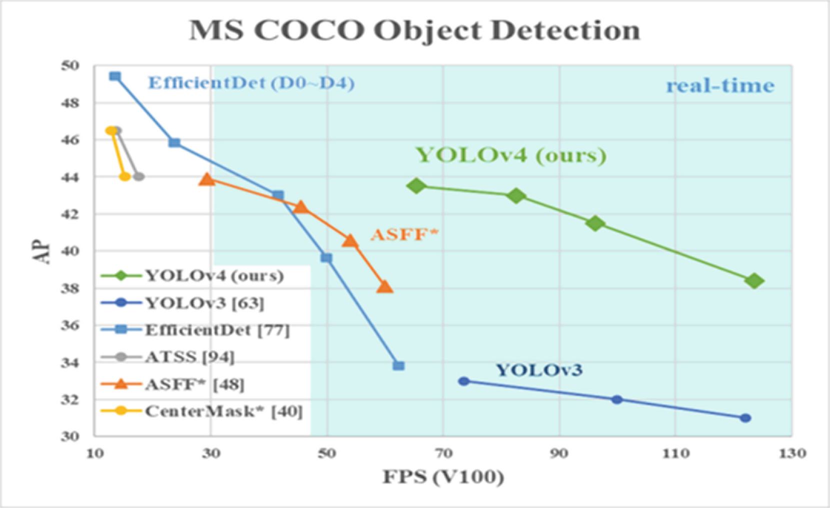

Fig. 1. Comparison of YOLO with other algorithms [11]

Building a customized weapon object detection algorithm poses major problems like, 1) lack of freely available weapon datasets, 2) necessity for a domain expert to decide harmful objects classes, and 3) real-time accurate classification and localization these objects from given video surveillance footage. In this work, we recognised various markers/harmful objects, such as cameras that can hamper privacy of sensitive locations like army base, and some hazardous weapons, such as Sticks, Daggers, Swords, Handguns, and Riffles, that can cause significant physical human injury. Further, we generated a novel dataset “DIAT-Weapon” Dataset along with annotations for threat object detection. The dataset consists of 2712 images divided into six categories. We customized one stage model – YOLO for the first time to detect weapons as it is outperforming other algorithms in terms of FPS, AP score and being size invariant, for getting real-time performance as any event like triggering gun in public places can happen in any moment. We fine-tuned YOLO algorithm to get better trade-off between accurate detection/localization and real-time performance.

2 Literature Review

Many object detection techniques based on deep learning have been proposed since 2018. These techniques can be based on 1) two-stage models, such as R-CNN [28], Fast R-CNN [25], Faster R-CNN [26], mask-RCNN [27], etc. mostly consisting, the region proposal network (RPN) network to select the approximate region of possible objects; followed by the object detection network to classify the candidate regions with accurate bounding; or 2) one-stage model, such as YOLO series [7, 8, 11] SSD [13], etc. where object detection is framed as a regression problem offering faster speed, with slightly lower accuracy.

Murugan et al. [1], has presented different object identification, object classification, and object tracking algorithms in the literature and has given methods for video summarization. Hu et al. [2], proposed a novel unified method for recognizing vehicle number plates and automobiles, gathering high energy frequency portions of images from digital camera imaging sensors with their proposed algorithm. For video surveillance Raghunandan et al. [3], has enhanced algorithms for various object detection techniques such as face detection, skin detection, color detection, shape detection, and target detection.

Elhoseny et al. [4], proposed a machine learning model for multi-object recognition and tracking that use an optimal Kalman filter [5] to track objects. Ahmad et al. [6], developed a framework for monitoring students during virtual tests, which employs YOLOv2 [7] and YOLOv3 [8] to detect things such as cell phones, laptops, iPads, and notebooks. Thoudoju et al. [9], uses YOLOv3 to detect objects in aerial images and satellite images. Kumar et al. [10], Detects vehicle classes such as automobile, truck, two-wheeler, and people using YOLOv3 and YOLOv4 [11]. Jose et al. [12], Detects things such as firearms and knives in suspicious regions using YOLO architecture to determine the likelihood of domestic violence. Though there are many variants of convolutional neural networks like SSD [13], R-FCN [14] performs well in terms of accuracy compared to YOLO, but their FPS (frame per second) is major drawback.

Though there are many object detection algorithms none has used to identify threat objects to ensure security standards. This study aims to build a real-time surveillance system that can detect dangerous objects in live CCTV feeds and, in the future, support these systems in surveillance robots. The paper is organized as follows: Sect. 3 describes about the dataset; Sect. 4 describes about the YOLO algorithms used and Sect. 5 consists of results and experiments.

3 DIAT Weapon Dataset

We chose stick, riffle, sword, handgun, camera, and dagger as our markers based on the severity of the damage that may be caused by utilizing these objects in prohibited locations, and it will offer the institution a perspective of what countermeasures they should do. Many weapons were brought under the umbrella of these classes, and subclass specifications were given in Table 1.

Table 1. Subclasses

Class	Sub classes
Sticks	Banton, lathi, hockey sticks, baseball bat
Riffle	Rifles, shotguns, muzzle loading firearms
Handgun	Pistols and revolvers
Camera	Surveillance cameras, digital cinema cameras, point and shoot cameras, DSLR e.tc
Daggers	Dagger, kitchen knives
Swords	Swords

The data was gathered from a variety of open sources, including the OIDV4 toolkit [15] and web scraping the images available in Internet ensuring diversity of collected data with respect to various conditions including different color, different shapes, different backgrounds, different time periods, variety of weather conditions, different occlusions, multiple perspective etc. Roboflow software is used to construct bounding boxes for each image to support for training the models in Darknet, TensorFlow and PyTorch. These classes are difficult to collect as there are few unique images for class in open domain, yet we managed 2,712 photos in total, divided into six classifications, with an average of 1.6 annotations per image. The sample images with annotation from generated DIAT-Weapon dataset are shown in Fig. 2. Those who are interested to get educational access to the DIAT-Weapon dataset, please send an e-mail request to "sunitadhavale@diat.ac.in" mentioning the subject: "DIAT-Weapon Image Dataset Educational Access Request" from their institutional e-mail id. This dataset will also be made publicly available at https://www.diat.ac.in/view-profile/?id=98.

Histogram plot of number objects for each class and number of annotations per image has given in Fig. 3 and 4. Data augmentation techniques are used to handle data imbalance problems. The Purpose of the data augmentation is to make model much robust towards the data. We Primarily focused on Photometric distortions such as random noise, Hue and Exposure. The image quality statistics were collected using Roboflow software, the average size of collected images were 0.7 mp, ranging from 0.01 to 20.90 mp and the median ratio of images is 1024 × 685. The aspect ratio histogram plot was given in Fig. 5, where majority of the images fell into the category of wider images. We conducted the experiments on images and resized to the size 416 × 416 and we added random noise of 5%, Hue and Exposure are between $-25°$ to $+25°$ as an augmentation technique.

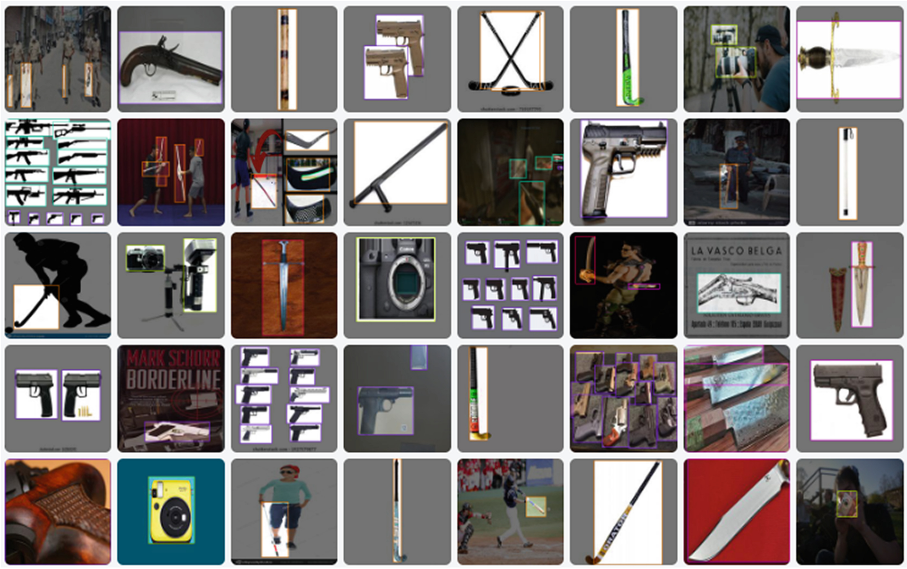

Fig. 2. Sample images

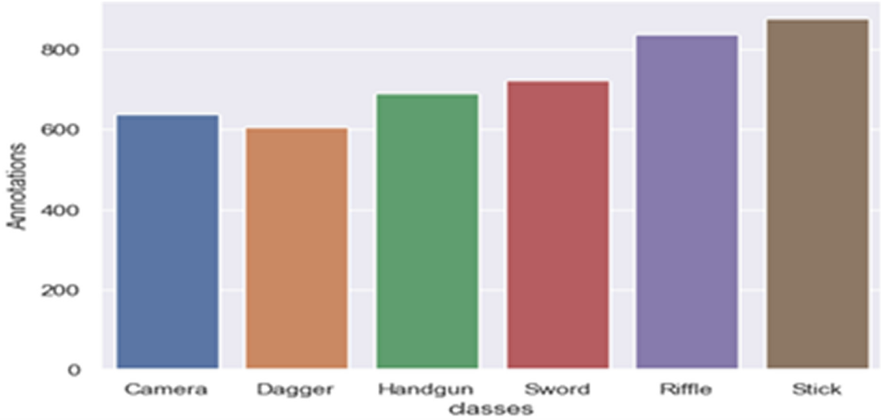

Fig. 3. Histogram of class balance

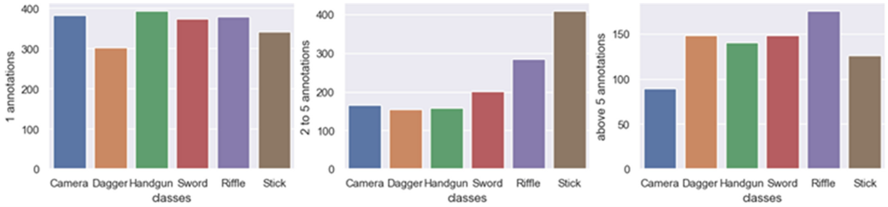

Fig. 4. Annotations distribution of classes

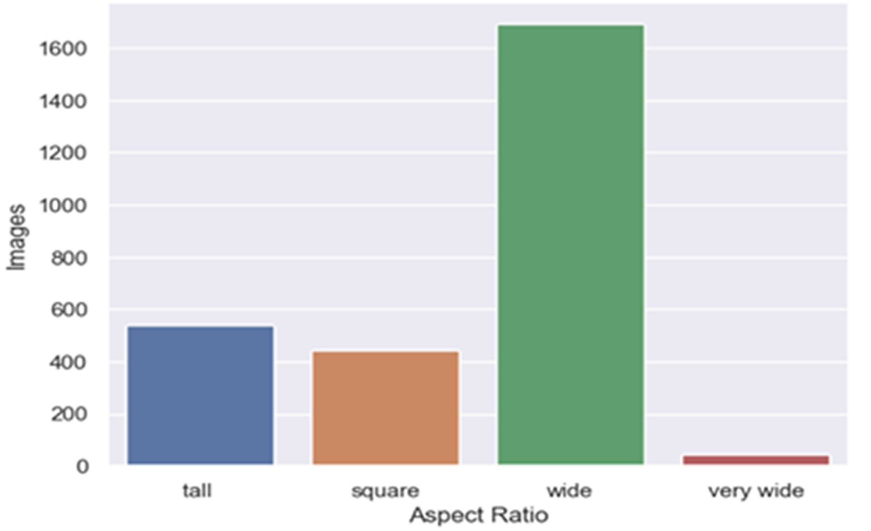

Fig. 5. Aspect ratio of images

4 YOLO Architecture

YOLOv1 [29] a single stage object detector where it localizes and classifies at the same time. Input size of YOLO v1 is 448 * 448, and it is divided into s * s grids where each grid is responsible for detection of one object, however it can use multiple bounding boxes, but it only gives the bounding box with maximum Intersection-over-Union (IOU) as output. The output will be position information of this bounding box (centre point coordinates x, y, width w, height h), and prediction confidence. YOLO v1 has certain drawbacks like constant image size and able to predict one object per grid. These drawbacks were improved in later versions. The architecture of YOLO v1 is given Fig. 6 where it contains 24 convolutional layers followed by 2 fully connected layers, in later algorithms these fully connected layers are with anchor boxes for predicting bounding boxes.

In YOLOv2 [7], YOLOv2 has added several amazing concepts such as 1) pre-trained anchor boxes, which were built using K-means clustering, to solve the problems of imprecise bounding box detections and relatively low recall caused by fully connected layers in YOLOv1. 2) Batch Normalization has improved 2% mAP score compared to YOLOv1. 3) Multi scale Training, to improve the model's stability across multiple image sizes, images were resized to 416 * 416 and for every 10 epochs, the image dimension was varied at random by multiples of 32 from 320 to 608 as the YOLOv2 model down samples by a factor of 2. The total number of detections in YOLOv2 is of 13 * 13 * number of anchor boxes. Darknet 19 was used as backbone architecture, to detect things faster, and it includes of 19 convolutional layers and 5 max pool layers.

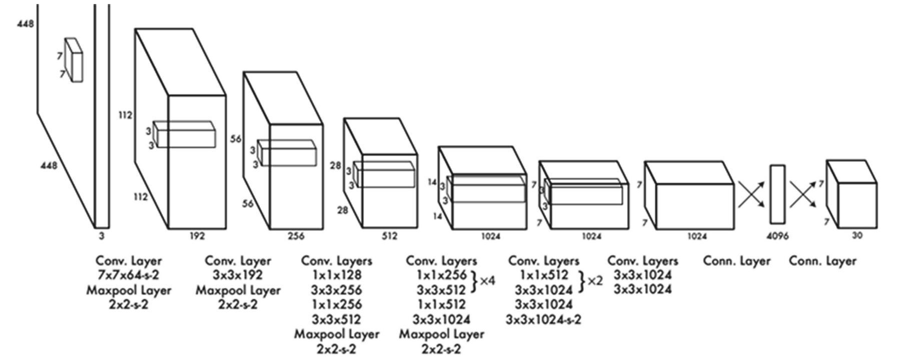

Fig. 6. YOLOv1 architecture [29]

In YOLOv3 [8], the Independent SoftMax layers was replaced with Independent logistic classifier for multi label classification to address the overlapping labels like Women and Person. YOLO v3 predicts in 3 feature scales like feature pyramid networks [30] as to improve prediction levels at large, medium, and small targets. At each scale it uses 3 boxes, and the shape of tensor is N * N * (3 * (4 + 1 + C)), where C is number of classes, 4 bounding box offsets and 1 objectiveness score. These feature maps are up sampled to concatenate with previous layer outputs. The backbone architecture Darknet-19 was modified with darknet-53 architecture because darknet-53 is size invariant. The convolutional layer of stride 2 is use instead of max pooling operation. YOLO v3 tiny is variation is YOLO v3 architecture where its backbone network consists of 7 convolutional layers and 6 max pooling layers, and it predicts in 2 scales. YOLO v3 tiny has compromised accuracy but it has a faster detection time.

Bochkovskiy et al. [11], proposed YOLOv4 architecture for object detection shown in Fig. 7, this architecture was implemented in Darknet framework. The YOLOv4 architecture was divided into 4 categories. 1) Input, it contains images, patches and video stream etc. 2) Back Bone, these convolutional Neural Network architectures were trained on Imagenet [18] Dataset and the author considers these networks CSPDarknet53 [19], CSPResNext50 [19] and EfficientNet-B3 [17] and finalizes CSPDarknet53 [19] as backbone network. 3) Neck, at neck combinations of various levels of backbone features are mixed and the author uses SPP [20] and PAN [21] as neck.4) Head, as a head the architecture uses YOLOv3 [8] for detection of objects.

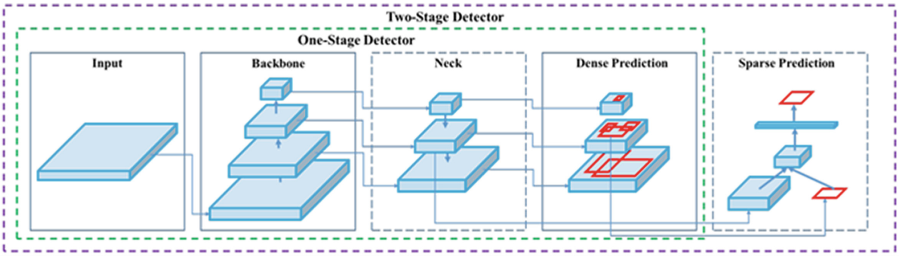

Fig. 7. Object detection algorithm architecture [11]

Jiang et al. [16], the proposed YOLOv4 small architecture was adapted from the original YOLOv4 algorithm with a few tweaks to conduct real-time prediction with trade-off accuracy. The major changes in the architecture are, total convolutional layers were compromised to 29 layers and YOLO layers were reduced to two instead of three and uses CSPDarknet53 [19] as its backbone architecture. With the introduction of YOLOv4 [11], ultralytics has announced YOLOv5 with open-source code [22], many believe YOLOv5 is further modification of YOLOv4, and it is implemented in PyTorch.

Wangh et al. [24], proposed a scaled YOLOv4 architecture implemented in PyTorch framework. Developed a network scaling strategy in YOLOv4 architecture using CSP approach that scales the network in both directions while maintaining standards in accuracy and optimal speed. This scaling modifies the depth, width, resolution, and structure of the network. The YOLOv4 large is one network in scaled in YOLOv4 networks, designed for cloud-based GPU to achieve high accuracy and it has variations like YOLOv4-P5, YOLOv4-P6 and YOLOv4-P7 where it detects objects in the scale of 3, 4, 5.

5 Experimental Results

In this section, the outcomes of the models were explained, as well as the metrics used to evaluate the models and the models' outputs. The DIAT Weapon image dataset is divided into train set, and test set in the ratio of 80:20. All these images are manually labelled using Roboflow software. All images are resized to 416 × 416 size. Due to small sized dataset, we used data augmentation like random horizontal translation, image flipping, and image distortion. The same transformation is performed on corresponding bounding boxes. All experiments are carried out on NVIDIA RTX-6000 GPU powered high end Tyrone workstation with 2 Intel Xeon processors, 256 GB RAM, 4 TB HDD configuration. For software stack, Python 3.7.2, CUDA 10.0, cuDNN 7.6.5, PyTorch 3.7.2, Darknet used.

In object detection models, we calculate precision based on the Intersection over Union metric [23]. Mean average precision (mAP) is the standard evaluation metric used for any object detection algorithms. Here we used mAP along with precision (P) and recall (R). Precision (P) ranged between [0, 1] and refers to the proportion of the correctly predicted 'True' labels in all the predicted 'True' labels. Recall (R) ranged between [0, 1] and represents the proportion of correctly predicted 'True' labels in the total number of actual 'True' labels. F1 Score is used to comprehensively measure the quality of algorithm in terms of both P and R as given in Eq. 1. For a category, average Precision (AP) refers to the area under the curve drawn according to P and R is given in Eq. 2. For multi-classification tasks, mAP of multiple categories is calculated as the average mAP score of all classes, and it is given in Eq. 3. Higher mAP means better model. For real-time classifications, frames per second (FPS) is used to measure real-time performance of the model.

$$\mathrm{F_1} = 2 * \frac{\mathrm{PR}}{\mathrm{P + R}} \in [0, 1] \tag{1}$$

$$\mathrm{AP_i} = \int_0^1 \mathrm{P_i(R_i)dR_i} \tag{2}$$

$$\mathrm{mAP} = \frac{\sum_{1}^{n} \mathrm{AP_i}}{n} \in [0, 1] \quad (3)$$

The metrics of the model are given in Table 2. All YOLO models pertained with MS-COCO benchmark dataset are fine-tuned to our weapon dataset. Whereas YOLOv4 outperformed other models in terms of mAP, Precision, and F1-score as 0.63, 0.77, 0.65 YOLOv4-csp outperformed other models in terms of recall with 0.67. The mAP value plots per batch for YOLOv4 and YOLOv4 tiny, trained in Darknet framework given Fig. 8 where x-axis represents number of batches and y-axis represent loss. For YOLOv5 and scaled YOLOv4 CSP, trained using PyTorch were given in Fig. 9 where x-axis represents epochs and y-axis for mAP score. Figures 10 and 11 show the results of YOLOv4 detecting single objects per image and multiple objects per image of six classes, where each object is identified by a bounding box and tagged with its class. In real time frames are extracted from videos using OpenCV and detection has done per image basis.

Table 2. Model results

Metric	YOLOv4-tiny	**YOLOv4**	YOLOv4-csp	YOLOv5
mAP@0.50	0.42	**0.63**	0.61	0.61
Precision	0.59	**0.77**	0.47	0.69
Recall	0.4	**0.56**	0.67	0.58
F1-Score	0.48	**0.65**	0.55	0.63

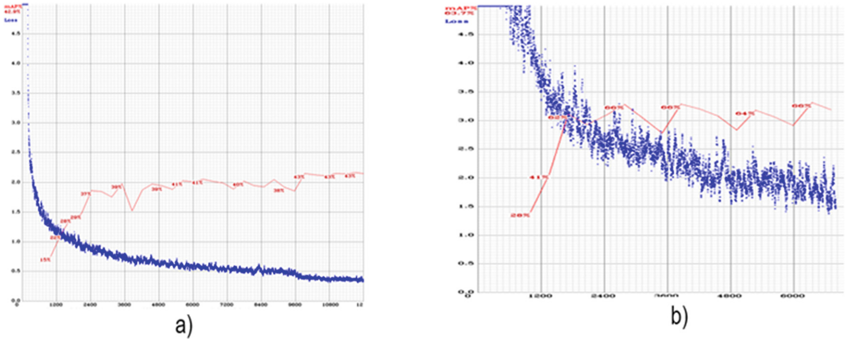

Fig. 8. Plot of mAP a) YOLOv4tiny b) YOLOv4

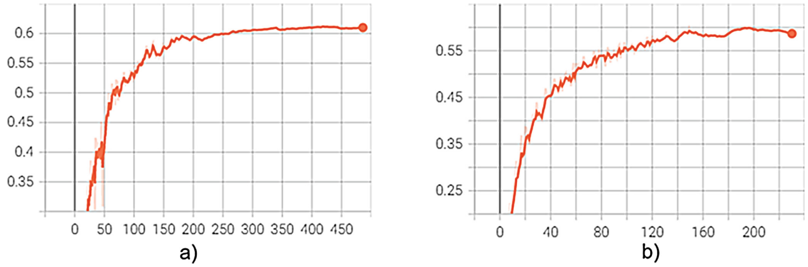

Fig. 9. Plot of mAP a) YOLOv5 b) scaled YOLOv4 CSP

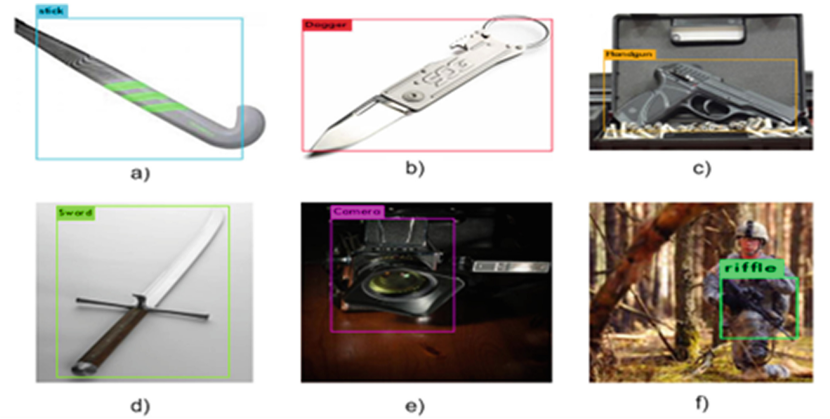

Fig. 10. Detection of single objects a) stick b) dagger c) handgun d) sword e) camera f) rifle

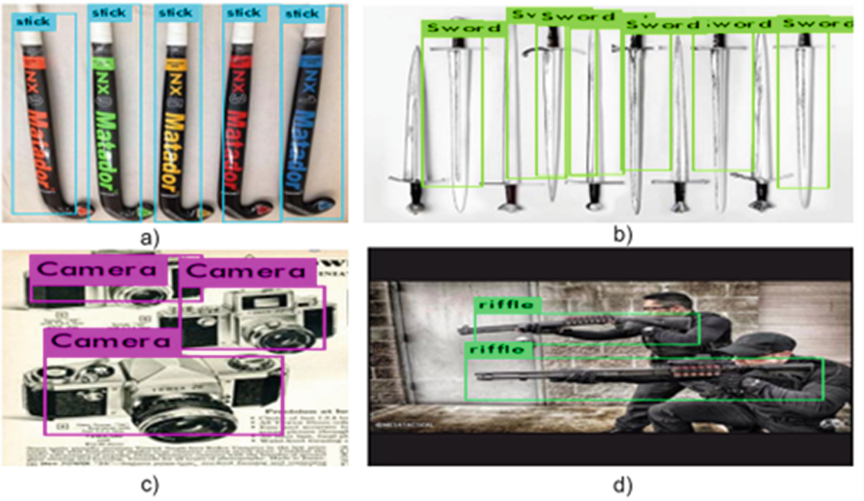

Fig. 11. Detection of multiple objects a) stick b) sword c) camera d) riffle

6 Conclusion and Future Scope

In this paper, we customized YOLO object detector for real time weapon detection. We introduced DIAT Weapon Dataset having 6 different classes of weapons i.e. Handgun, Sword, Camera, Riffle, Stick and Dagger. This dataset and learnt YOLO Model will be useful for harmful object detection to address the concerns of national security. During experimental analysis, it is found that YOLOv4 has achieved significant results in real time demonstration using videos as test dataset with more than 30 fps using OpenCV DNN module. Although YOLO v5 is lightweight than YOLO v4, accuracy of YOLO v4 was found good in real time performance. In future, we have two objectives, 1) we will add some more images and categories of weapon in our proposed dataset and will try to enhance prediction accuracy, generalization capability along with faster detection. To expand the data set, we will use Generative Adversarial Networks (GAN) network. 2) To integrate the trained models to work on real time CCTV feeds along with robots.

Acknowledgements. This research is supported by the Life Sciences Research Board (LSRB) in association with Defence Institute of Psychological Research (DIPR) sanction letter no. LSRB/o1/15001/M/LSRB-381/PEE& BS/2020, dated 15.03.2021. The authors want to thank NVIDIA for the academic GPU research grant. The authors wish to thank all the students who supported us during the data collection.

References

1. Senthil Murugan, A., Suganya Devi, K., Sivaranjani, A., Srinivasan, P.: A study on various methods used for video summarization and moving object detection for video surveillance applications. Multimedia Tools Appl. **77**(18), 23273–23290 (2018). https://doi.org/10.1007/s11042-018-5671-8
2. Hu, L., Ni, Q.: IoT-driven automated object detection algorithm for urban surveillance systems in smart cities. IEEE Internet Things J. **5**(2), 747–754 (2018). https://doi.org/10.1109/JIOT.2017.2705560
3. Raghunandan, A., Raghav, P., Ravish Aradhya, H.V.: Object detection algorithms for video surveillance applications. In: 2018 International Conference on Communication and Signal Processing (ICCSP). IEEE (2018)
4. Elhoseny, M.: Multi-object detection and tracking (MODT) machine learning model for real-time video surveillance systems. Circuits Syst. Signal Process. **39**(2), 611–630 (2020)
5. Welch, G., Bishop, G.: An introduction to the Kalman filter, pp. 127–132 (1995)
6. Ahmad, I.: A novel deep learning-based online proctoring system using face recognition, eye blinking, and object detection techniques. System **12**(10) (2021)
7. Redmon, J., Farhadi, A.: YOLO9000: better, faster, stronger. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6517–6525 (2017). https://doi.org/10.1109/CVPR.2017.690
8. Redmon, J., Farhadi, A.: YOLOv3: an incremental improvement. arXiv preprint arXiv:1804.02767 (2018)
9. Thoudoju, A.K.: Detection of aircraft, vehicles and ships in aerial and satellite imagery using evolutionary deep learning. Dissertation (2021). http://urn.kb.se/resolve?urn=urn:nbn:se:bth-22310

10. Kumar, B.C., Punitha, R., Mohana, M.: YOLOv3 and YOLOv4: multiple object detection for surveillance applications. In: 2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT), pp. 1316–1321 (2020). https://doi.org/10.1109/ICSSIT48917.2020.9214094
11. Bochkovskiy, A., Wang, C.-Y., Liao, H.-Y.M.: YOLOv4: optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934 (2020)
12. Jose, D.: Deep learning based gender responsive smart device to combat domestic violence. SPAST Abstr. **1**(01) (2021). https://spast.org/techrep/article/view/2933
13. Liu, W., et al.: SSD: single shot multibox detector. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9905, pp. 21–37. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46448-0_2
14. Dai, J., Li, Y., He, K., Sun, J.: R-FCN: object detection via region-based fully convolutional networks. In: Advances in Neural Information Processing Systems, vol. 29. Curran Associates, Inc. (2016). https://proceedings.neurips.cc/paper/2016/file/577ef1154f3240ad5b9b413aa7346a1e-Paper.pdf
15. Vittorio, A.: OIDv4_ToolKit: toolkit to download and visualize single or multiple classes from the huge Open Images V4 dataset. GitHub repository (2018). https://github.com/EscVM/OIDv4_ToolKit. Accessed 04 Apr 2022
16. Jiang, Z., et al.: Real-time object detection method based on improved YOLOv4-tiny. arXiv preprint arXiv:2011.04244 (2020)
17. Tan, M., Le, Q.: EfficientNet: rethinking model scaling for convolutional neural networks. In: Proceedings of the 36th International Conference on Machine Learning. PMLR, vol. 97, pp. 6105–6114 (2019)
18. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems 25, pp. 1097–1105 (2012). https://doi.org/10.1145/3065386
19. Wang, C.-Y., Mark Liao, H.-Y., Wu, Y.-H., Chen, P.Y., Hsieh, J.W., Yeh, I.H.: CSPNet: a new backbone that can enhance learning capability of CNN. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 1571–1580 (2020). https://doi.org/10.1109/CVPRW50498.2020.00203
20. He, K., Zhang, X., Ren, S., Sun, J.: Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Trans. Pattern Anal. Mach. Intell. **37**(9), 1904–1916 (2015). https://doi.org/10.1109/TPAMI.2015.2389824
21. Liu, S., Qi, L., Qin, H., Shi, J., Jia, J.: Path aggregation network for instance segmentation. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8759–8768 (2018) https://doi.org/10.1109/CVPR.2018.00913
22. YOLOv5: Ultralytics open-source research into future vision AI methods. https://github.com/ultralytics/yolov5. Accessed 04 Apr 2022
23. Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., Savarese, S.: Generalized intersection over union: a metric and a loss for bounding box regression. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 658–666 (2019). https://doi.org/10.1109/CVPR.2019.00075
24. Wang, C., Bochkovskiy, A., Liao, H.: Scaled-YOLOv4: scaling cross stage partial network. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, pp. 13024–13033 (2021). https://doi.org/10.1109/CVPR46437.2021.01283
25. Girshick, R.: Fast R-CNN. In: 2015 IEEE International Conference on Computer Vision (ICCV), pp. 1440–1448 (2015). https://doi.org/10.1109/ICCV.2015.169
26. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans. Pattern Anal. Mach. Intell. **39**(06), 1137–1149 (2017). https://doi.org/10.1109/TPAMI.2016.2577031

27. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. In: 2017 IEEE International Conference on Computer Vision (ICCV), pp. 2980–2988 (2017). https://doi.org/10.1109/ICCV.2017.322
28. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition, pp. 580–587 (2014). https://doi.org/10.1109/CVPR.2014.81
29. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: unified, real-time object detection. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 779–788 (2016). https://doi.org/10.1109/CVPR.2016.91
30. Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 936–944 (2017). https://doi.org/10.1109/CVPR.2017.106
31. Suresh, K.: Detection, analysis and management of atypical behaviour of crowd and mob in LIC environment. ST/14/DIP-732, DIPR/Note/No./714 (2017)
32. Suresh, K.: Predicting the probability of stone pelting in crowd of J&K. ST/14/DIP-732, DIPR/Note/No./719 (2018)