International Journal of Computational Research in Science and Technology
E-ISSN: 3108-0871, Volume 2 Issue 5
Sep-Oct 2026, PP 01-6

AI-Based Smart Weapon Detection and Threat Assessment in CCTV Surveillance: A Review

Disha Narayan Kale, Prof. Bangar A.P, Prof. Bhosale S.B., Dr. A. A. Khatri

Dept. of Computer Engineering,

Jaihind collage of Engineering, Pune,India

Abstract. Closed-circuit television (CCTV) networks now produce far more footage than human operators can watch, and delayed recognition of an armed person can cost lives. This paper reviews AI-based approaches to weapon detection and threat assessment in CCTV surveillance. Four recent studies are examined in detail: an edge-deployable weapon and threat detector built on a lightweight CNN and Mask R-CNN, an edge–cloud collaborative framework for real-time object detection, a hierarchical context-tree approach to anomaly detection, and a review of deep-learning benchmarks for video surveillance anomaly detection. The studies are compared on architecture, datasets, hardware, evaluation metrics and reported results. The review shows that object detectors reach high accuracy on curated weapon classes, but small and imbalanced datasets, occlusion, false alarms, limited edge compute and the lack of a context-aware risk-scoring layer still prevent dependable deployment. A conceptual layered pipeline that joins edge detection, location-aware anomaly scoring and cloud-assisted model updates is outlined as a direction for future work.

Keywords: Weapon Detection, CCTV surveillance, threat assessment, deep learning, anomaly detection, edge computing, YOLO.

I. Introduction

Public places such as schools, malls, stadiums and transport hubs are covered by large numbers of surveillance cameras, yet most of the footage is only reviewed after an incident. Gun violence illustrates the cost of this delay: the Gun Violence Archive records several hundred mass shootings in a single year in the United States alone [1]. Manual monitoring suffers from fatigue, lost concentration and the sheer number of screens one operator must follow. Automated systems that can recognise weapons and flag dangerous behaviour as it happens would turn passive cameras into active observers.

Advances in deep learning, embedded accelerators and Internet of Things (IoT) devices have made such systems realistic. Object detectors can localise handguns, rifles and knives in a frame, and anomaly detectors can flag behaviour that departs from the normal pattern of a scene. However, a detector that only labels objects does not tell a security team how serious a situation is. Threat assessment, which combines what is seen, where it is seen and how people behave, is the step that decides whether an alert is raised.

This paper reviews four recent studies that address parts of this problem [2]–[5]. Hawk-Eye [2] builds a weapon and threat detector that runs on an edge camera and in the cloud. Chen et al. [3] propose an edge–cloud collaboration framework for real-time object detection. Pelvan et al. [4] present a hierarchical ensemble for locational anomaly detection. Duja et al. [5] survey deep-learning benchmarks for video surveillance anomaly detection. The contributions of this review are: (i) a structured summary of the techniques used in each study, (ii) a side-by-side comparison of datasets, hardware and results, (iii) a discussion of open challenges, and (iv) a conceptual pipeline that links detection with threat assessment.

The remainder of the paper is organised as follows. Section II gives background and evaluation metrics. Section III covers related techniques. Section IV reviews the selected studies. Section V compares them. Section VI outlines a conceptual threat-assessment pipeline. Section VII discusses challenges and future directions, and Section VIII concludes.

Background

Weapon Detection and Anomaly Detection

Two related tasks appear in the literature. Weapon detection is a supervised object-detection or classification problem: the system is trained on labelled images of guns, knives and similar objects and must find them in new frames. Anomaly detection is usually trained mainly on normal behaviour and must flag events that deviate from it. Weapon detection is precise about what is present but blind to intent, while anomaly detection captures unusual behaviour but may not name the cause. A practical threat assessment system needs both.

Types of Anomaly

Duja et al. [5] divide surveillance anomalies into global and local. A global anomaly involves a group of people or the whole frame, for example panic movement of a crowd or an explosion. A local anomaly involves one or two individuals in a specific region while the rest of the scene looks normal, for example snatching, loitering, shoplifting, pickpocketing or an armed person. Weapon brandishing is a local anomaly, and it may later trigger a global one when a crowd starts to run. Pelvan et al. [4] further note that a local anomaly can depend on location: a pedestrian on a footpath is normal, but the same pedestrian on a motorway is not.

Evaluation Metrics

The reviewed studies use standard measures. For classification, precision is TP/(TP+FP), recall is TP/(TP+FN) and accuracy is (TP+TN)/(TP+FP+FN+TN), where TP, FP, FN and TN are true positives, false positives, false negatives and true negatives [2]. Detection quality is reported as mean average precision (mAP), and ranking quality for anomaly scores as the area under the ROC curve (AUC). Because real-time use is the goal, speed matters equally: prediction time per image and frames per second (FPS) are reported in [2] and [3].

II. Related Techniques

1. Classical Methods

Early anomaly detectors relied on hand-crafted features: trajectories, optical flow histograms and statistical models of normal motion. Duja et al. [5] describe approaches built on trajectory analysis, sparse reconstruction, Hidden Markov Models and one-class classifiers. They work in moderately crowded scenes but struggle with occlusion and with scenes unlike those used to design the features. Most recent work has therefore moved to deep networks, which learn features directly from data.

2. Deep Learning Detectors

Convolutional neural networks (CNNs) are the base of modern detection. One-stage detectors such as You Only Look Once (YOLO) [6] and its successors such as YOLOv3 [7] treat detection as a single regression step and run in real time on a GPU. Two-stage detectors such as Faster R-CNN [8] and Mask R-CNN first propose regions and then classify them; Mask R-CNN also produces an instance mask for each object but is slower. For temporal information, the surveyed literature [5] combines CNNs with recurrent units (LSTM, GRU), uses 3D convolutions, applies vision transformers with self-attention, and uses autoencoders for reconstruction-based anomaly scoring.

3. Weapon-Specific Systems

Lim et al. [10] collected an annotated set of about 5,500 gun images from 250 CCTV videos and trained a single-stage detector with a feature pyramid network; the collected data raised precision by 18% over two existing datasets. Grega et al. [11] proposed automatic firearm detection for CCTV, using 1,000 positive and 3,500 negative examples, and reported that poor image quality and low resolution reduced accuracy. These works, as summarised in [2], were evaluated on recorded video processed offline on servers, which does not meet the delay requirements of live security.

International Journal of Statistics and Mathematical Research Trends
E-ISSN: 3108-0871, Volume 2 Issue 5
Sep-Oct 2026, PP 01-6

IV. Review of Selected Studies

1. Hawk-Eye: Edge Threat Detector

Ahmed and Echi [2] present Hawk-Eye, a prototype split between the camera and the cloud. A custom dataset of 10,097 images covers nine classes: grenade, machine gun, pistol, rocket-propelled grenade (RPG), masked face, motorcycle, car, knife and bat, with the vehicle classes included because armed offenders often use them. On the camera side, a CNN with four convolutional layers (about 6.6 million parameters) runs on a Raspberry Pi 3 with an Intel Neural Compute Stick 2, fed by a Logitech C920 webcam. The model is converted with the OpenVINO toolkit for the stick. On the cloud side, a Mask R-CNN (about 63.8 million parameters) draws segmentation masks around each object. A motion-detection module based on Gaussian-mixture background subtraction captures images only when something moves, and a Flask web interface lets users upload or capture images.

The authors report an average classification accuracy of 94.20% for the CNN and 93.09% for Mask R-CNN. The CNN needs about 5.1 ms per image, while Mask R-CNN needs about 4.18 s, so only the CNN suits the edge. The CNN scored 100% on machine guns and above 92% on pistols and knives, but confused RPGs with machine guns and cars with motorcycles. Fig. 2 plots the per-class accuracies as given in [2].

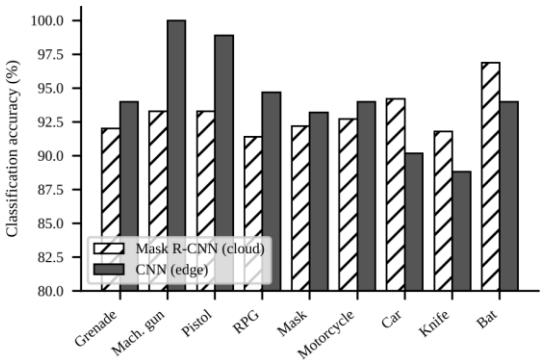


Fig. 2. Per-class accuracy of the edge CNN and cloud Mask R-CNN, as reported in [2]

2. Edge–Cloud Collaborative Detection

Chen et al. [3] argue that sending all video to the cloud wastes bandwidth and adds delay. Their framework has three tiers: end devices (cameras), edge devices that run detection locally, and a cloud that retrains models and shares the result with every edge. An edge device runs YOLOv3 and also performs image matting by frame differencing, binarisation, erosion and dilation, and connected-component labelling. The foreground area that YOLOv3 failed to cover gives a residual-area ratio

$$Ra = \Sigma ai / (W \times H) \quad (1)$$

where Ai are the white-pixel areas of the matted image and W and H are the image width and height. If ra exceeds a threshold α (1.0% in the experiments) in β consecutive cases (β = 10) and the network is not congested, the frames are sent to the cloud, which retrains the model with a human-supplied label and pushes the update back to the edge.

The prototype used three NVIDIA Jetson Xavier NX edge nodes at a campus, an alley and a street, a cloud server with an RTX 2080 Ti GPU, and 5,005 labelled COCO 2017 images (4,004 for training and 1,001 for testing). After an update, ra for previously unseen objects returned to the normal level, and the framework reduced backbone-network bandwidth by about 94% compared with sending all video. The authors also compared YOLOv3 with YOLOv4 and YOLOR in terms of FPS, model

size and mAP. Weapons were not a target class in this study, but the update mechanism is directly relevant to weapon detection, where new object types appear after deployment.

3. Hierarchical Anomaly Detection

Pelvan et al. [4] address the bias–variance trade-off in anomaly detection. A single global model has low variance but high bias because it ignores location, while one model per region is location-aware but needs more data. Their context-tree-based anomaly detection (CTBAD) partitions the frame with a binary tree, giving a hierarchy from one coarse model at the root to many fine models at the leaves. Each node holds a local detector. Objects are found with YOLOv5, motion with FlowNet2 optical flow, and features come from convolutional autoencoders and flow statistics. The final decision is a weighted combination

$$f(xt) = \Sigma\, wj,t{-}1\, Mj,t(xt) \quad (2)$$

where the weights w are updated online in proportion to each partition model's performance. Early in a video stream, when data are scarce, coarse models dominate; as more data arrive, the weight shifts toward finer models. A tree of depth D contains an enormous number of possible partitions, but context-tree weighting reduces the computation to O(D). The authors report that the method detects locational anomalies and needs far fewer training samples to match a complex model, and it can wrap around existing detectors rather than replace them.

4. Review of Deep-Learning Benchmarks

Duja et al. [5] review deep-learning methods for video surveillance anomaly detection published between 2020 and 2024. They classify methods by learning model (CNN, sequential learning, transformers, YOLO, autoencoders), input size, dataset, accuracy and anomaly type, and list the limitations each reported. Their tables include weapon-related work, for example a pseudo-labelled detector with mAP of 71.32 on the UGR dataset and 58.56 on a YouTube gun dataset, and a YOLOv4 model trained on synthetic data that reached only 7.37 mAP. The review concludes that no framework is yet clearly ahead and highlights occlusion, intra-class similarity, background clutter, weak annotations, small datasets and false alarms from camera jitter as recurring problems.

Table 1: Comparison of the Reviewed Studies

Study	Approach	Data and hardware	Reported results	Limitations noted
Hawk-Eye [2]	Edge CNN plus cloud Mask R-CNN; motion trigger; web interface	10,097 images, 9 classes; Raspberry Pi 3 + NCS2; PC with CUDA GPU	CNN 94.20% accuracy, 5.1 ms; Mask R-CNN 93.09%, 4.18 s	Confusion of similar classes; small classes; masks too slow for edge
Chen et al. [3]	Edge–cloud collaboration; image matting + YOLOv3; residual-area trigger	COCO 2017 subset (4,004/1,001); 3 Jetson Xavier NX nodes; RTX 2080 Ti cloud	About 94% less backbone bandwidth; unseen objects recognised after update	Preliminary prototype; weapons not a target class; needs human labelling
Pelvan et al. [4]	Context-tree ensemble over image partitions; YOLOv5, FlowNet2, CAE features	Surveillance video benchmarks (e.g. UCSD Ped2 shown in the paper)	Detects locational anomalies; fewer training samples needed	Depends on quality of the underlying object detector and features
Duja et al. [5]	Review of CNN, sequential, transformer, YOLO and autoencoder methods	Many datasets (e.g. UCF-Crime, UCSD, Hockey Fight)	No clear leader; weapon mAP varies widely (7.37 to 71.32)	Occlusion, weak labels, small datasets, false alarms

International Journal of Statistics and Mathematical Research Trends
E-ISSN: 3108-0871, Volume 2 Issue 5
Sep-Oct 2026, PP 01-6

V. Comparative Analysis

Table I places the four studies side by side. Three observations follow. First, the studies sit at different layers of the same system: [2] is a complete weapon detector, [3] is an infrastructure for keeping edge detectors current, [4] is a decision layer that adds location awareness, and [5] maps the field. None combines all layers.

Second, accuracy and speed pull in opposite directions. In [2] the heavier Mask R-CNN gives masks but takes about 4 s per image, while the CNN takes about 5 ms. This is why a split deployment, with a light model on the edge and a heavy model in the cloud for verification, appears in both [2] and [3].

Third, the reported figures cannot be compared directly. Each study uses a different dataset, class set, hardware and metric, and the review in [5] shows that weapon-detection results range from very low to moderate mAP depending on data source. High accuracy on a curated set of nine classes in [2] does not guarantee the same result on noisy night-time CCTV footage.

VI. Conceptual Threat-Assessment Pipeline

Drawing on the four studies, a layered pipeline is outlined in Fig. 1. This is the authors' synthesis of the reviewed ideas and has not been experimentally validated. In Stage 1, a lightweight detector on the edge device finds weapons, masks and persons, triggered by motion as in [2]. In Stage 2, a location-aware anomaly score in the manner of [4] asks whether the detected activity is unusual for that part of the scene. In Stage 3, weapon confidence, context and behaviour are fused into a risk level; for example, a knife in a kitchen-supply aisle and a pistol raised in a corridor should not produce the same alert. In Stage 4, an operator receives an alert with evidence. In parallel, uncertain frames are sent to a heavier cloud model for verification and retraining, and updated models are pushed back, following [3].

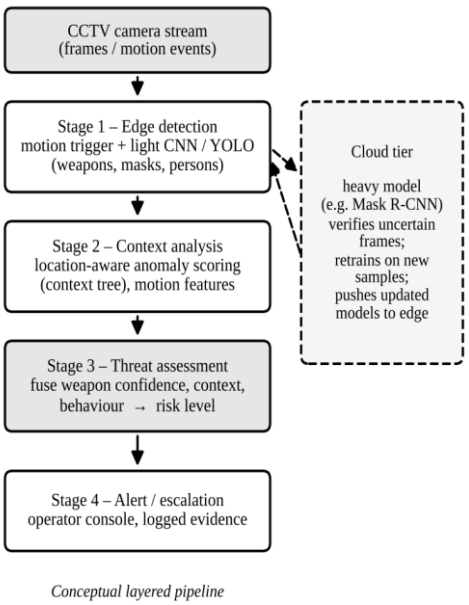


Fig. 1. Conceptual layered pipeline for weapon detection and threat assessment

The design keeps raw video at the edge, limits bandwidth use to uncertain frames, and gives the operator a graded risk level instead of a bare object label.

Challenges and Future Directions

Challenges

Data. Public weapon datasets are small, unevenly balanced across classes and often staged. The dataset in [2] has 375 bat images against 2,137 pistol images, and synthetic data gave very low mAP in a study listed in [5]. Occlusion and small objects. A weapon held close to the body or seen from a ceiling camera may occupy only a few pixels. False alarms. Tools, phones and toys look like guns, and every false alarm lowers operator trust. Edge limits. The accurate masks of [2] cannot run on the edge hardware used. Context. A detector in [2] or [3] does not use location or behaviour, and a context layer like [4] has not yet been tested with weapon classes. Privacy and ethics. Continuous analysis of public spaces raises questions on data retention, bias and responsibility for wrong alerts.

Future Directions

Promising directions include larger and more realistic CCTV weapon datasets with consistent labels; efficient transformer or YOLO-family detectors that fit embedded accelerators; fusing weapon detection with pose and motion cues to recognise brandishing and aiming; evaluating context-tree weighting [4] on weapon classes; applying the residual-area update trigger of [3] to weapons; and reporting results with common metrics, including false alarms per hour, so that systems can be compared fairly.

VII. Conclusion

This review examined four studies on AI-based surveillance. Hawk-Eye shows that weapon classes can be recognised in milliseconds on low-cost edge hardware; the edge–cloud framework shows how detectors can stay current while saving about 94% of bandwidth; the context-tree method shows how location awareness can improve anomaly detection with less data; and the benchmark review shows that reliable, general solutions remain out of reach. The main gap is integration: no reviewed system connects weapon detection, contextual anomaly scoring and continuous model updating into a single threat-assessment process. Closing this gap, with better datasets and common evaluation practice, is the main task for the next generation of smart CCTV systems.

Acknowledgment

The authors thank the Department of Computer Engineering, Jaihind College of Engineering, SPPU, Pune, for guidance and facilities.

References

1. Gun Violence Archive. Accessed: Oct. 2026. [Online]. Available: https://www.gunviolencearchive.org/
2. A. A. Ahmed and M. Echi, “Hawk-Eye: An AI-powered threat detector for intelligent surveillance cameras,” IEEE Access, vol. 9, pp. 63283–63293, 2021.
3. Y.-Y. Chen, Y.-H. Lin, Y.-C. Hu, C.-H. Hsia, Y.-A. Lian, and S.-Y. Jhong, “Distributed real-time object detection based on edge-cloud collaboration for smart video surveillance applications,” IEEE Access, vol. 10, pp. 93745–93759, 2022.
4. S. Ö. Pelvan, B. Can, and H. Ozkan, “A hierarchical approach for improved anomaly detection in video surveillance,” IEEE Access, vol. 11, pp. 101644–101665, 2023.
5. K. U. Duja, I. A. Khan, and M. Alsuhaibani, “Video surveillance anomaly detection: A review on deep learning benchmarks,” IEEE Access, vol. 12, pp. 164811–164842, 2024.
6. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 779–788.
7. J. Redmon and A. Farhadi, “YOLOv3: An incremental improvement,” arXiv:1804.02767, 2018.
8. S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 6, pp. 1137–1149, Jun. 2017.