© May 2026 | IJIRT | Volume 12 Issue 12 | ISSN: 2349-6002

AI-Driven Real-Time Weapon Detection for Anti-Poaching Surveillance Using YOLOv8

Komal Yangal[1], Rutuja Khandagale[2], Saish Wader[3], Pratishtha Mane[4], Nikunj Wandile[5], Dr. P. M. Gavali[6]
[1,2,3,4,5]*Student, Dept. of Computer Science & Engineering, D.K.T. E Society's Textile and Engineering Institute, Ichalkaranji, India*
[6]*Professor, Dept. of Computer Science & Engineering, D.K.T.E Society's Textile and Engineering Institute, Ichalkaranji, India*

***Abstract*—Poaching is a significant threat to wildlife particularly in forests. The conventional methods of surveillance have the problem of low visibility, geographical coverage and slow human response. In this paper, an anti-poaching surveillance system of an AI based real time weapon detection system is presented using the YOLOv8 model. The system examines live video streams of CCTV and IP cameras, uses preprocessing (CLAHE enhancement, Gaussian denoising) to enhance the quality of images in low-light and covered areas, and detects weapons like guns and knives. When detected, real time alerts are sent through SMS/email and stored in PostgreSQL database. Experimental findings on a custom forest dataset (2,500 annotated images) show results of 80% accuracy, 83 percent precision, 84 percent recall and 91.2 mAP/0.5 on NVIDIA GTX 1080 Ti GPU. When compared to YOLOv5 and Faster R CNN, it can be determined that the best tradeoff between speed and accuracy in this application is with YOLOv8. The system lessens the use of manual patrolling and reduces the response time by the forest authority.**

***Index Terms*—Forest surveillance, weapon detection, YOLOv8, deep learning, computer vision, real time alerting.**

I. INTRODUCTION

Forest reserves are important in conservation of biodiversity and balance in the ecosystem. Still, poaching of endangered species is a significant issue - the World Wildlife Fund estimates that more than 20,000 African elephants are poached every year to obtain ivory. The conventional surveillance (manual patrols, CCTV, motion sensors, etc.) has certain limitations due to human reliance, environmental conditions, and the lack of speed in reaction. The poachers may work in groups wielding rifles and knives, hence hard to follow in a thick forest. The above challenges accentuate the necessity to have an automated surveillance system that is intelligent and capable of detecting threats in real time. The recent developments in artificial intelligence and computer vision have rendered automated analysis of live video feeds a possibility. Detectors like YOLO, SSD and Faster R CNN have been heavily used in real time detection applications [1,4,7]. Among them, YOLO based models are popular because of their high speeds and efficiency [1,2]. YOLOv8 latest has a better tradeoff between accuracy of detection and inference speed, so it is good in the case of real time surveillance [3]. Despite these developments, there is a lack of studies on weapon detection within a forest setting, and the visibility is low, vegetation is dense, and lighting is variable, creating distinct challenges. None of the existing systems is capable of weapon detection under these conditions at a rate of above 30 FPS with inbuilt alerting. This paper suggests an AI-based weapon detection system, which will combine YOLOv8 with real time video capture, an alerting system (SMS/email via Redis queue) and a monitoring dashboard to improve the efficiency of wildlife protection and response. Ethical statement: The system is solely meant to be used in lawful anti-poaching activities by the authorized forest departments and not to be used to spy on people. Real-world anomaly detection in surveillance videos has been explored by Sultani et al. [20], though not specifically for weapon detection in forests.

The key contributions of our work are as follows. First, we create a real-time weapon detection system tailored for the difficult forest environment. Second, we improve detection accuracy in darkness by combining YOLOv8 with efficient preprocessing methods such as Contrast Limited Adaptive Histogram Equalization (CLAHE) and Gaussian filtering. Third, we build a real-time alert system with backend technologies to facilitate immediate action in response to threats.

II. LITERATURE REVIEW

Modern surveillance systems use object detection. Initial techniques such as Viola Jonson and Histogram of oriented gradients (HOG) did not work well in dynamic and challenging environments [16,17]. The introduction of deep learning and convolutional neural networks (CNNs) enhanced object detection a great deal. AlexNet and VGGNet models were also shown to be effective in image classification [5,8]. Other region-based approaches such as Faster R CNN offered high detection but had low processing rates (about 5 10 FPS on GPU) and could not be used in real time systems. To address these shortcomings, stage detectors like SSD and YOLO have been created, which allows detecting in real time at high speed and efficiency [7,1]. YOLOv4 also enhanced feature extraction and accuracy of detection [2]. The most recent YOLOv8 is faster, more accurate and more robust in adverse environments [3]. A comprehensive survey of deep learning-based object detection methods is provided by Zhao et al. [19]. Big data sets such as ImageNet and Pascal VOC have played a crucial role in training object detection models [10,11]. Even more recent architectures, such as Feature Pyramid Networks, EfficientNet and Transformer based detectors (DETR) keep pushing performance limits [9,18]. Current anti-poaching technologies (e.g., TrailGuard AI, Air Shepherd) are currently mostly motion sensors or human piloted drones with no real time AI-based weapon detection. The paper addresses these gaps by introducing a real time weapon detection system that integrates YOLOv8 and the recent technologies of the backend to realize efficient monitoring and alerting. For a comprehensive background on deep learning, see [6].

III. METHODOLOGY

Dataset

We have a dataset of forest weapon images designed 2,500 in number: 1,800 images of guns/rifles • 700 images of knives They were taken using camera traps in forests, open-source repositories on the internet and in simulated environments. All images were manually annotated with bounding boxes. The data is divided into 80 percent training, 10 percent validation and 10 percent testing. The data is not made publicly accessible because of security reasons but is provided by the respective author, on a reasonable request. System Layers The system is structured into 6 collaborative layers:

1. Data Acquisition Layer:
RTSP is used to capture video streams received by CCTV and IP cameras placed in forest areas (1080p, 30 FPS).

2. Preprocessing Layer:
Frames are down sampled to 640×640 pixels, denoised with a Gaussian blur (kernel 5×5), and contrast enhanced with the CLAHE (clip limit 2.0, tile size 8×8). These were able to enhance recall by 5-8 percent when the conditions were in low light.

3. Detection Layer:
YOLOv8n (pretrained on COCO) is trained on our data with 100 epochs (batch size 16, learning rate 0.001, Adam optimizer). The augmentation applied to the data are random horizontal flips (with probability 0.5), mosaic augmentation, and HSV shift (hue +0.015, saturation +0.7, value +0.4). The alerts are activated at a confidence level of 0.5. All image preprocessing operations were implemented using OpenCV [12].

4. Backend Processing Layer:
FastAPI and JWT authentication records the detection to PostgreSQL (timestamp, weapon type, confidence, camera ID). The REST API was built with FastAPI [13], and detection logs were stored in a PostgreSQL database [14].

5. Notification and Alert System:
Redis message queue sends email (SMTP) and SMS (Twilio) alerts within 1.2 seconds of being detected. Redis [15] was used as a message queue for real-time alert handling.

6. Visualization Layer:
This is a React.js dashboard that displays live feeds, bounding boxes, and analytics.

IV. PROPOSED FRAMEWORK

Fig. 1 shows the system architecture. This presents an overview of the video capture, preprocessing, detection model application and processing by the backend to generate alerts. The distributed architecture incorporates video ingestion, YOLOv8 detection, FastAPI backend, PostgreSQL/Redis, and the alert/dashboard modules.

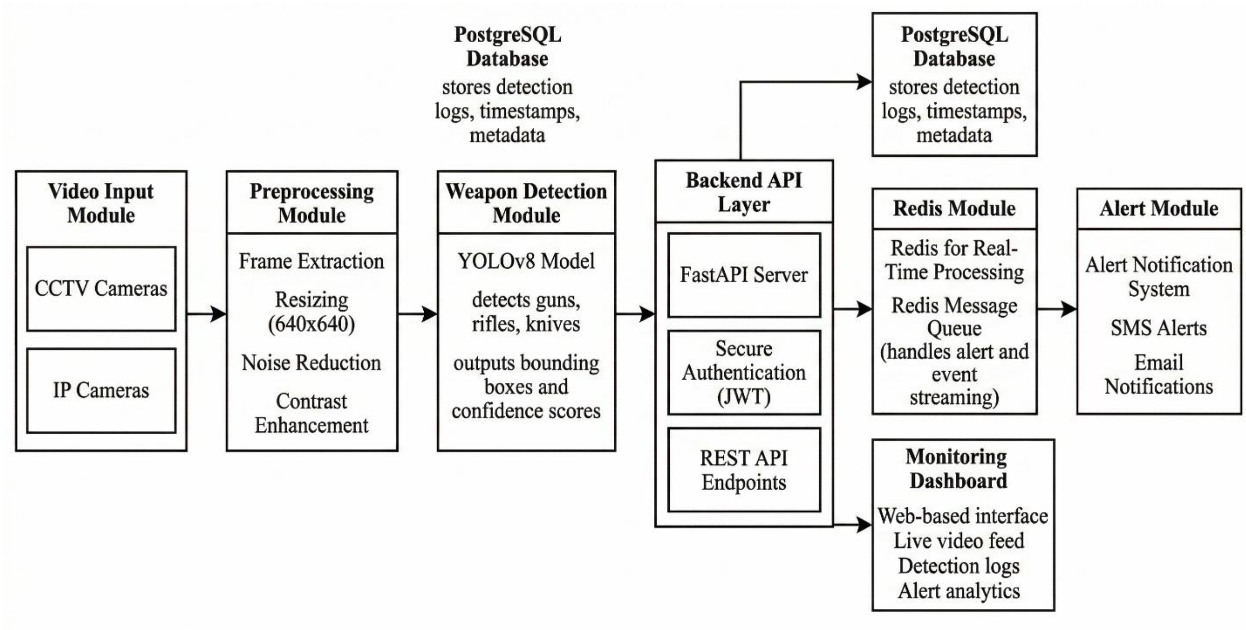

Fig. 1 System architecture

V. RESULTS AND DISCUSSION

Detection Performance

The test set (250 images, 450 weapon instances) evaluation metrics are given in Table 1. The high mAP 0.5 (91.2) means that there is great localization. Per frame accuracy (80%) is a bit less accurate since false positive (e.g., a branch identified as a rifle) hurts accuracy, but mAP@0.5 averages per class and IoU thresholds, so it is more resistant to false positives that occur occasionally. Some examples of incorrect detections were observed during experimentation. There were some cases where natural items (such as tree branches) were mistakenly detected as weapons, especially under dim or heavily obstructed conditions. We will address this issue in future research, perhaps by using temporal information or enhancing the training data.

Table 1. Detection metrics on forest weapon dataset

Metric	Value
Accuracy (per frame)	80.0%
Precision	83.0%
Recall	84.0%
F1-score	83.5%
mAP@0.5	91.2%
mAP@0.5:0.95	68.4%
Inference speed	45 FPS (22.2 ms per frame)

Comparison with Prior Models

We compared YOLOv8n to YOLOv5n and Faster R CNN (ResNet 50) using the same hardware (NVIDIA GTX 1080 Ti). Table 2 indicates that the YOLOv8n has the best trade off speed and accuracy in real time surveillance.

Table 2. Comparison of detection models

Model	mAP@0.5	FPS	Latency (ms)
YOLOv8n (ours)	91.2%	45	22.2
YOLOv5n	88.7%	52	19.2
Faster R-CNN	93.1%	8	125.0

Alert System Performance

The system handles 2,700 frames per minute at 45 FPS. The false alert rate using a confidence level of 0.5 was 2.3 per hour (mainly due to branches or animal antlers). It took 1.2 seconds (including Redis queuing) to send SMS/email alerts once detected. Preliminary suggestions by the forest rangers in a pilot implementation of a wildlife sanctuary (due to confidentiality reasons, the name is not disclosed) reported better situational awareness. Future work will be officially quantitatively evaluated.

VI. CONCLUSION

The suggested system has a high accuracy (91.2% mAP) and low latency (22 ms per frame) in real time weapon detection. The combination of the YOLOv8 with the scalable backend services (FastAPI, PostgreSQL, Redis) improve monitoring efficiency and response time. YOLOv8 has the best trade off to forest surveillance compared to YOLOv5 and Faster R

CNN. In the future, it can be deployed to edge devices (Jetson Xavier, Raspberry Pi); extended to thermal imaging to detect in the night; expanded to additional weapon types (machetes, bows) and unfavorable weather (rain, mist); and smoothed over time (majority vote across frames) to decrease false positives.

ACKNOWLEDGEMENT

The authors are quite appreciative of the management of the Textile and Engineering Institute of D.K.T.E. Society, Ichalkaranji, to provide facilities to carry out the experiments in the laboratory of the institution, and to colleagues to assist in preparing this paper. We also appreciate the input of the forest department authorities in our domain.

REFERENCES

[1] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2016.

[2] Bochkovskiy, C.-Y. Wang, and H.-Y. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," *arXiv preprint* arXiv:2004.10934, 2020.

[3] G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics YOLOv8," 2023. [Online]. Available: https://docs.ultralytics.com. [Accessed: Mar. 15, 2024].

[4] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2015.

[5] Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2012.

[6] Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[7] W. Liu *et al.*, "SSD: Single Shot MultiBox Detector," in *Proc. European Conf. Computer Vision (ECCV)*, 2016.

[8] Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *arXiv preprint* arXiv:1409.1556, 2014.

[9] T.-Y. Lin *et al.*, "Feature Pyramid Networks for Object Detection," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2017.

[10] M. Everingham *et al.*, "The Pascal Visual Object Classes (VOC) Challenge," *International Journal of Computer Vision (IJCV)*, vol. 88, pp. 303–338, 2010.

[11] Deng *et al.*, "ImageNet: A Large-Scale Hierarchical Image Database," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2009.

[12] OpenCV, "Open-Source Computer Vision Library." [Online]. Available: https://opencv.org. [Accessed: Mar. 10, 2024].

[13] FastAPI, "FastAPI Framework Documentation." [Online]. Available: https://fastapi.tiangolo.com. [Accessed: Mar. 10, 2024].

[14] PostgreSQL Global Development Group, "PostgreSQL Documentation." [Online]. Available: https://www.postgresql.org. [Accessed: Mar. 10, 2024].

[15] Redis Labs, "Redis Documentation." [Online]. Available: https://redis.io. [Accessed: Mar. 10, 2024].

[16] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2005.

[17] P. Viola and M. Jones, "Rapid Object Detection Using a Boosted Cascade of Simple Features," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2001.

[18] Tan and Q. Le, "EfficientNet: Rethinking Model Scaling for CNNs," in *Proc. Int. Conf. Machine Learning (ICML)*, 2019.

[19] Z. Zhao *et al.*, "Object Detection with Deep Learning: A Review," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 11, pp. 3212–3232, 2019.

[20] Sultani, C. Chen, and M. Shah, "Real-World Anomaly Detection in Surveillance Videos," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2018.