

Edge-AI-Enabled UAV System for Rapid Visual Confirmation of Illegal Logging in Remote Forest Environments

Henry Muchiri

Strathmore University

Simon Mburu

`simon.mburu@strathmore.edu`

Strathmore University

Research Article

Keywords: BoT-SORT, domain randomization, edge computing, forest conservation, illegal logging detection, UAV surveillance, object detection, YOLOv8s

Posted Date: September 30th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-10091974/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

Edge-AI-Enabled UAV System for Rapid Visual Confirmation of Illegal Logging in Remote Forest Environments

Henry Muchiri

Strathmore University, School of Computing and
Engineering Sciences
Nairobi, Kenya
<https://orcid.org/0000-0003-1773-6674>

Simon Mburu*

Strathmore University, School of Computing and
Engineering Sciences
Nairobi, Kenya
simon.mburu@strathmore.edu

Abstract

Illegal logging poses a critical environmental and economic threat, with Kenya's Mau Forest Complex losing over 19% of its cover between 2001 and 2022. Traditional monitoring, such as human patrols and satellite imagery, suffers from terrain inaccessibility and data latency of up to 1.5 months, creating a gap in rapid intervention. Existing UAV-based detection systems have largely relied on offboard cloud inference or manually annotated datasets, limiting their autonomy in remote environments and scalability to niche scenarios. This paper presents a proactive, edge-based UAV system for real-time visual confirmation of illegal logging in open and moderately canopied forests (15–65% canopy density), addressing these gaps through two novel contributions. First, a domain-randomized synthetic data pipeline using AirSim semantic segmentation automates YOLO-format annotation, eliminating manual labeling overhead when augmenting a real-world-trained baseline detector. Second, a fully onboard detection pipeline integrates YOLOv8s with BoT-SORT tracking and a temporal AlertFilter, enabling autonomous threat confirmation without offboard processing dependency. A multi-channel dataset was assembled from web-scraped imagery, video extraction, open-source repositories, and AirSim-generated synthetic data. A domain randomization pipeline comprising weather variation, camera angle diversity, and color jittering improved model robustness against organic forest clutter. The final model, evaluated on a 659-image holdout test set of exclusively real-world imagery, achieved mAP@0.5 of 96.80%, mAP@0.5:0.95 of 74.86%, precision of 96.92%, and recall of 93.01%. Chainsaw background false positives decreased from 63% to 48% following domain randomization. System integration testing in a live AirSim simulation validated end-to-end pipeline functionality, with the AlertFilter confirming threats after five consecutive tracked frames. Peak throughput reached 10.80 FPS, with projected deployment performance of 3.90–10.80 FPS.

Keywords: BoT-SORT, domain randomization, edge computing, forest conservation, illegal logging detection, UAV surveillance, object detection, YOLOv8s.

1 Introduction

Illegal logging constitutes the world's third-largest transnational crime, with an estimated annual economic value of USD 52–157 billion [1]. In Kenya, this crisis is particularly acute in vital water towers such as the Mau Forest Complex, which lost over 19% of its tree cover between 2001 and 2022 due to illicit harvesting and

agricultural encroachment [2]. These losses threaten hydrological regulation and rural livelihoods, yet enforcement remains reactive and resource-constrained [3].

Traditional monitoring methods exhibit structural limitations. Ground patrols by the Kenya Forest Service (KFS) cannot adequately cover vast, inaccessible terrain [4]. Satellite-based systems such as JJ-FAST update deforestation data only every 1.5 months [5], while cloud cover frequently obscures imagery and spatial resolution remains insufficient to detect small-scale selective logging events [6]. This creates a critical real-time gap between illegal activity and intervention.

Unmanned Aerial Vehicles (UAVs) have emerged as a compelling complement to existing monitoring frameworks, offering rapid deployment, high-resolution imagery, and operational flexibility in difficult terrain [7]. When combined with embedded artificial intelligence, UAVs can transition from passive data collectors to active, autonomous surveillance agents. Recent systems combine IoT sensor networks with UAV visual inference [8]; however, architectures that offload inference to cloud servers introduce latency and connectivity constraints that limit effectiveness in remote forest environments [9, 10].

This paper addresses these limitations by presenting a proactive, self-contained UAV edge node for the rapid visual confirmation of illegal logging activities. The system is designed as a responsive confirmation mechanism, deploying in response to precursor alerts from acoustic or environmental sensors rather than as a primary surveillance tool. By embedding the full detection pipeline onboard the UAV, it eliminates reliance on cloud connectivity, addressing the latency, operational cost, and data sovereignty concerns that limit existing off-board architectures in remote forest environments. Additionally, by enhancing a real-world-trained baseline detector with domain-randomized synthetic data generated via automated semantic segmentation, the system addresses the data scarcity challenge inherent in niche scenarios, where collecting and manually annotating diverse real-world imagery is both time-consuming and impractical at scale.

2 Related Work

2.1 UAV-Based Forest Monitoring

Forest monitoring has evolved from labor-intensive ground patrols to automated UAV-based systems. Manual monitoring is constrained by personnel shortages, terrain inaccessibility, and institutional vulnerabilities [4, 11]. Satellite systems such as NASA Landsat, Global Forest

Watch, and Kenya's Forest Alert system have improved large-scale situational awareness but remain limited by update latency and resolution constraints insufficient for detecting subtle selective logging [6, 12, 13].

UAVs bridge this gap by enabling on-demand, high-resolution aerial surveillance spanning forest mapping [14], tree health monitoring [15], fire detection [16], and conservation enforcement [8, 17]. For illegal logging specifically, Ramadan et al. [8] combined acoustic alerts with UAV visual verification, achieving targeted detection. Aman et al. [18] applied YOLO-NAS for direct illegal logger detection, emphasizing the need for robust continuous object tracking alongside detection. Both systems, however, relied on offboard inference or lacked the temporal confirmation mechanism needed to suppress false alerts in cluttered forest scenes.

2.2 Object Detection and Tracking Architectures

Deep learning has supplanted hand-crafted feature methods for object detection. R-CNN [19] initiated CNN-based detection, followed by Faster R-CNN [20], SSD [21], and the YOLO family [22, 23]. YOLO variants are consistently preferred for UAV applications due to their real-time single-stage inference [24]. Song et al. [25] specifically evaluated YOLOv8 variants for onboard UAV deployment, recommending the Small (YOLOv8s) architecture based on its accuracy-FPS balance on constrained hardware.

For multi-object tracking, SORT [26] and DeepSORT [27] combine motion estimation with appearance features. BoT-SORT [28] extends this with camera motion compensation and appearance feature extraction to maintain target identification even when objects are temporarily obscured.

2.3 Simulation and Domain Randomization

Photorealistic simulation has become a critical testbed for UAV computer vision, reducing deployment costs and safety risks [29, 30]. AirSim, integrated with Unreal Engine, provides high-fidelity physics and visual simulation [31]. A key challenge is the sim-to-real gap: purely photorealistic approaches may not generalize to real-world deployment. Domain Randomization (DR) addresses this by systematically varying simulation parameters such as lighting, texture, viewpoint, object color to compel the model to learn robust, transferable features rather than simulation-specific cues [29, 32]. Hybrid strategies combining synthetic pre-training with real-world fine-tuning have achieved performance comparable to fully real-world-trained models [29].

3 Methodology

3.1 System Architecture

The proposed system is designed as an edge computing node co-located with the UAV's onboard processor. The software architecture follows Object-Oriented Analysis and Design principles to ensure modularity and maintainability. As illustrated in Fig. 1, five core components constitute the pipeline: (1) VisualSensor abstracts the AirSim camera API, acquiring RGB frames at 30 Hz; (2) DetectorModel wraps YOLOv8s inference, returning bounding boxes and class labels; (3) TrackingModule implements BoT-SORT to assign and maintain persistent object IDs across frames; (4) AlertFilter applies temporal persistence logic, triggering a confirmed threat only after five consecutive detections of the same tracked ID; (5) DatabaseLogger records events asynchronously to SQLite without blocking the inference thread. A Flask web application exposes MJPEG video streams and JSON telemetry endpoints to dual-interface dashboards for technical operators and conservation personnel respectively.

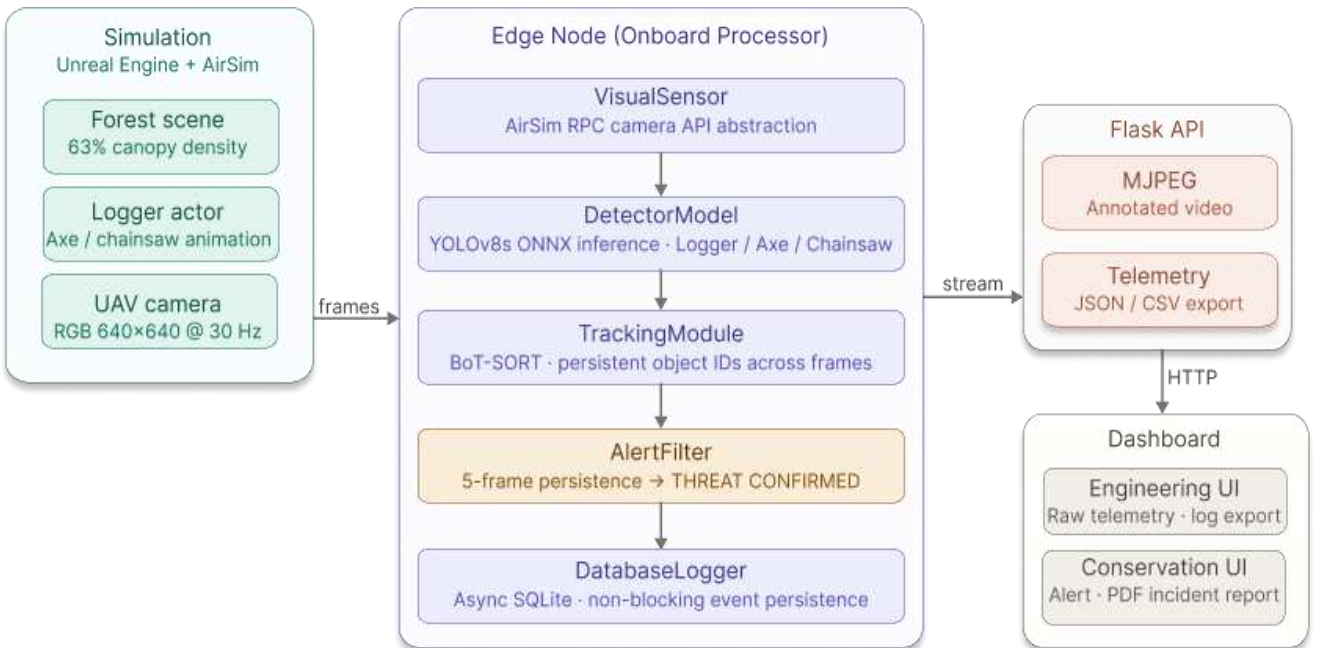


Fig. 1 System architecture: five-module edge detection pipeline with dual-interface Flask dashboard.

3.2 Dataset Construction

The training dataset targeted three classes: Logger (person engaged in logging activity), Axe, and Chainsaw, the most prevalent manual tree-felling methods. Data was collected across four channels:

Real-world imagery: Automated web scraping via the Bing Image Search API and frame extraction from open-source logging videos at 2–3 second intervals. Source provenance was catalogued in a CSV registry for reproducibility.

Open-source datasets: Curated datasets from Roboflow Universe, identified through keyword searches ("axe

chainsaw", "lumberjack"). Specifically, images were sourced from the *Axe Chainsaw* dataset [35]. All inherited annotations were manually reconciled against the three target classes. Extraneous annotations removed and missing ones added.

Synthetic data with automated annotation: Simulated loggers were captured in Unreal Engine via AirSim's Computer Vision mode. Pixel-perfect semantic segmentation masks were acquired for each frame, and a custom OpenCV script automatically extracted YOLO-format bounding box coordinates, eliminating manual labeling for all synthetic images (Fig. 2).

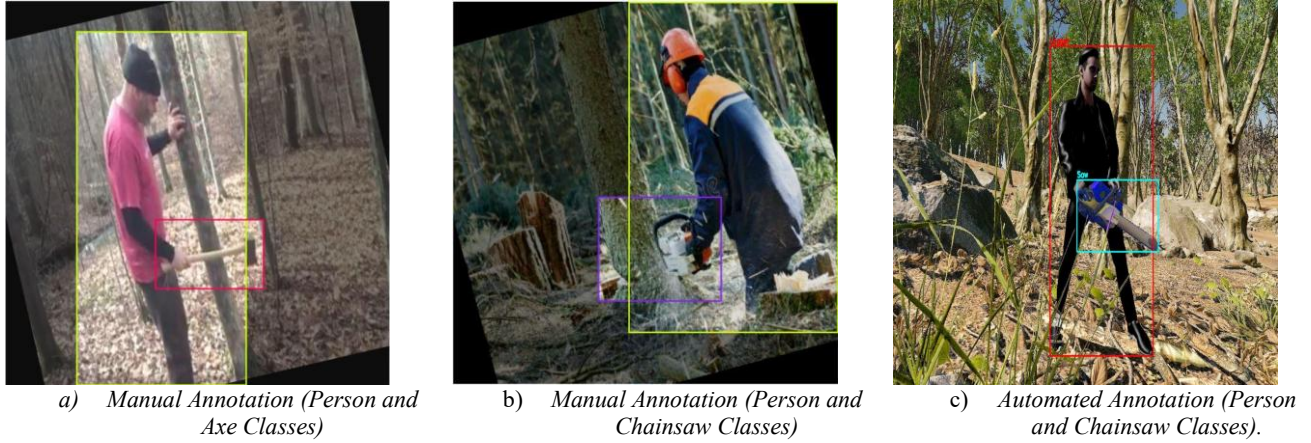


Fig. 2 Representative annotation examples: manual (a,b) and automated via AirSim semantic segmentation (c).



Fig. 3 Weather and camera angle randomization: eye-level foggy (a) vs. nadir sunny (b) perspectives.

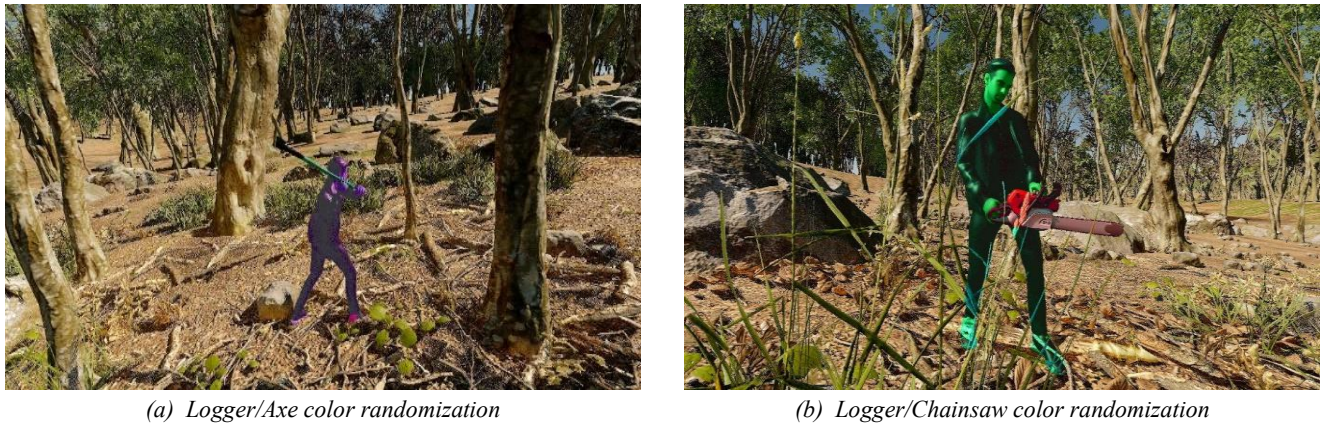


Fig. 4 Color randomization applied to all three target classes, forcing the model to learn structural morphology over surface-level color cues.

Domain randomization data: Domain randomization data: Seven scenario categories were generated by varying camera angles (eye-level vs. nadir) and weather conditions

(sunny, cloudy, foggy) (Fig. 3). A secondary color randomization pass programmatically altered the color

encodings of all three classes, preventing the model from learning surface-level color cues (Fig. 4).

All data was centralized and processed using the Roboflow platform. Augmentation (horizontal flipping, random cropping, saturation and brightness adjustments) was applied at a 1:2 ratio. Both Stage 1 and Stage 2 datasets used a 70/20/10 train/validation/test split. Critically, both test sets comprised exclusively real-world manual images. No synthetic data was included in either test partition ensuring uncontaminated benchmark evaluation at both stages. The Stage 1 dataset comprised approximately 2,623 base images, yielding a 394-image test set. The Stage 2 dataset, augmented with domain-randomized synthetic data, comprised 6,592 images (4,615 train / 1,318 valid / 659 test). The increase in test set size reflects the larger dataset while maintaining the 10% holdout ratio, not a change in composition policy. Canopy density of the simulated environment was measured using Monte Carlo ray tracing with 10,000 rays, yielding a density of 0.627 (~63%), consistent with the moderately canopied operational scope.

3.3 Model Selection and Training

YOLOv8s was selected for its optimal balance between mAP and computational efficiency for edge deployment [25]. BoT-SORT is natively integrated into the YOLOv8s architecture. Training was conducted on an NVIDIA Tesla T4 GPU using a progressive two-stage pipeline:

Stage 1 (Baseline): Trained for 50 epochs (batch size 16, imgsz 640) on real-world data only, using COCO-pretrained weights. The class label 'Person' was used at this stage to reflect the real-world training data sourcing.

Stage 2 (Domain Randomization): Trained for 150 epochs on the full dataset incorporating weather, angle, and color randomized synthetic imagery. Re-initialized from COCO-pretrained weights (not Stage 1 weights) to ensure a clean generalization baseline. A patience threshold of 50 epochs was applied. The class label was updated to 'Logger' to more precisely reflect the detection target in the augmented training context.

3.4 Evaluation Protocol

Model accuracy was assessed using mAP@0.5, mAP@0.5:0.95, Precision, Recall, F1-score, and normalized confusion matrices. System performance was evaluated across six detection runs (n=3,856 events total), with a Z-score analysis (threshold=3.0) applied to isolate hardware-induced latency spikes from algorithmic performance. Integration testing validated the end-to-end pipeline within a live AirSim simulation, with the AlertFilter's five-frame confirmation mechanism serving as the deterministic functional benchmark.

4 Experimental Results

4.1 Stage 1: Baseline Model Performance

The Stage 1 model, evaluated on a 394-image holdout test set of real-world manual images, established a strong baseline with an aggregate mAP@0.5 of 93.45% and F1-score of 0.9072 (Table 1). The Person class achieved near-perfect detection (mAP@0.5=99.4%), which is operationally critical as the alert logic requires co-detection of a person with a tool. The Chainsaw class exhibited the

lowest metrics (mAP@0.5=87.4%), attributable to its visual complexity and geometric similarity to fallen forest debris (Table 2).

Table 1 Aggregated performance metrics — Stage 1 Baseline

Parameter	Value
mAP@0.5	0.9345
mAP@0.5:0.95	0.6560
Precision	0.9243
Recall	0.8908
F1-score	0.9072

Table 2 Per-class metrics — Stage 1 Baseline

Class	Instances	Precision	Recall	mAP@0.5
Person	393	0.975	0.990	0.994
Axe	217	0.939	0.853	0.935
Chainsaw	187	0.859	0.829	0.874

Confusion matrix analysis (Fig. 5a) revealed significant background confusion: Chainsaw exhibited a 63% false positive rate against background, and Axe 24%. False negative rates were 11% for both Axe and Chainsaw. These failure modes, attributable to the model's sensitivity to organic forest textures geometrically similar to man-made tools, justified the transition to domain randomization.

4.2 Stage 2: Domain Randomization Model Performance

The Stage 2 model was evaluated on a 659-image holdout test set, larger than Stage 1 due to the expanded dataset maintaining the 10% holdout ratio, but equally composed of exclusively real-world manual images. The model demonstrated substantial improvements across all metrics (Table 3). The aggregate mAP@0.5 improved from 93.45% to 96.80%, and mAP@0.5:0.95 increased from 65.60% to 74.86%, indicating tighter, more accurate bounding boxes. Per-class results are detailed in Table 4.

Table 3 Aggregated performance metrics — Stage 2 Final Model

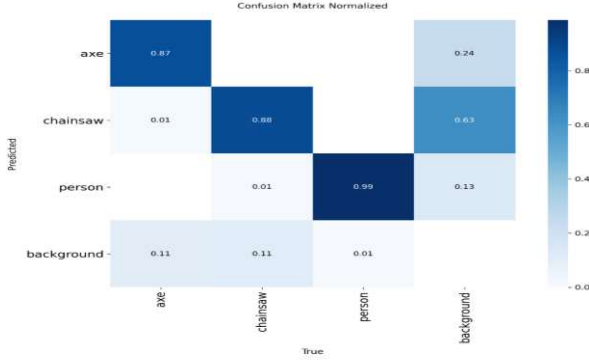
Parameter	Value
mAP@0.5	0.9680
mAP@0.5:0.95	0.7486
Precision	0.9692
Recall	0.9301
F1-score	0.9492

Table 4 Per-class metrics — Stage 2 Final Model

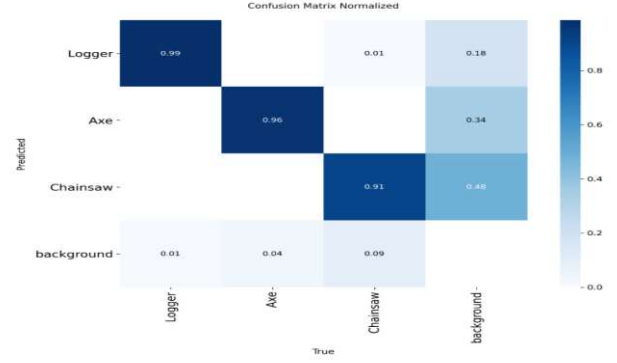
Class	Instances	Precision	Recall	mAP@0.5
Logger	651	0.991	0.983	0.993
Axe	363	0.969	0.934	0.978
Chainsaw	311	0.948	0.873	0.933

Confusion matrix analysis (Fig. 5b) confirmed meaningful progress in background discrimination. True positive diagonal values were Logger: 99%, Axe: 96%, Chainsaw: 91%. Background false positive rates decreased to 18% (Logger), 34% (Axe), and 48% (Chainsaw), a 15-percentage-point reduction for Chainsaw compared to

Stage 1. False negative rates improved to 1% (Logger), 4% (Axe), and 9% (Chainsaw), down from 11% for both Axe and Chainsaw in Stage 1. Cross-class confusion was minimal, with only 1% of Logger detections misclassified as Chainsaw and all other off-diagonal values at zero.



(a) Stage 1 Baseline



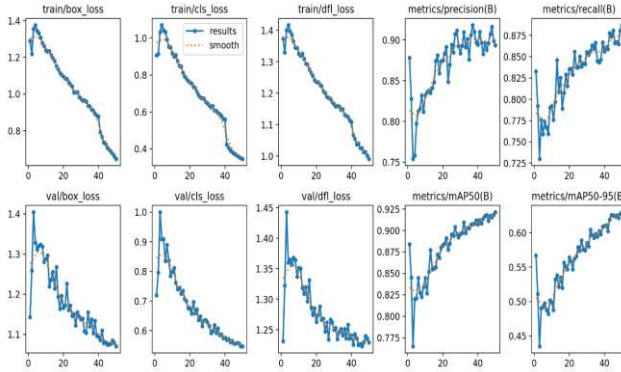
(b) Stage 2 Final Model

Fig. 5 Normalized confusion matrices for Stage 1 (a) and Stage 2 (b). Note the reduction in background false positives for Chainsaw from 63% to 48%.

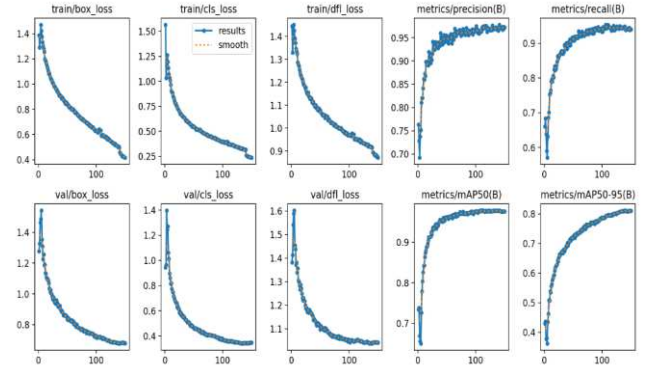
4.3 Training Dynamics

Training curves are shown in Fig. 6. Stage 1 (50 epochs) showed continuous improvement without plateau: classification loss decreased from 0.906 to 0.346, and mAP@0.5 rose from 88.38% to a final value indicating further gains were possible, motivating the extended Stage 2 run. Stage 2 (150 epochs) began at a higher initial loss of

1.566 and mAP@0.5 of 73.47%, reflecting the increased dataset complexity. Peak mAP@0.5 of 97.80% was achieved at epoch 109, with peak precision of 97.79% at epoch 141. Performance stabilized through epoch 150, concluding at precision 97.20%, recall 93.96%, mAP@0.5 97.60%, and mAP@0.5:0.95 80.99%, confirming no further gains warranted additional training.



(a) Stage 1: 50 epochs



(b) Stage 2: 150 epochs

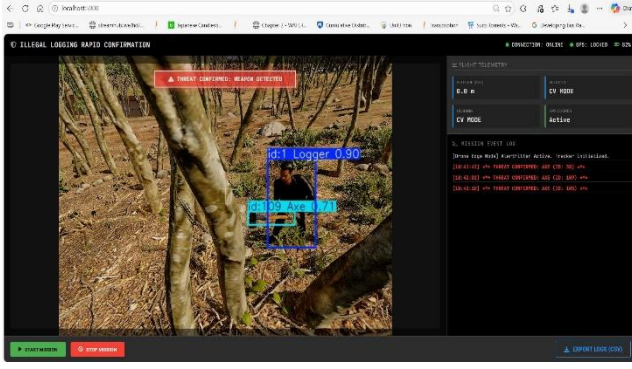
Fig. 6 Training dynamics: Stage 1 baseline (a) showing continuous improvement suggesting benefit from further training; Stage 2 (b) showing peak performance plateau around epoch 109 confirming training completion.

4.4 System Integration and Operational Performance

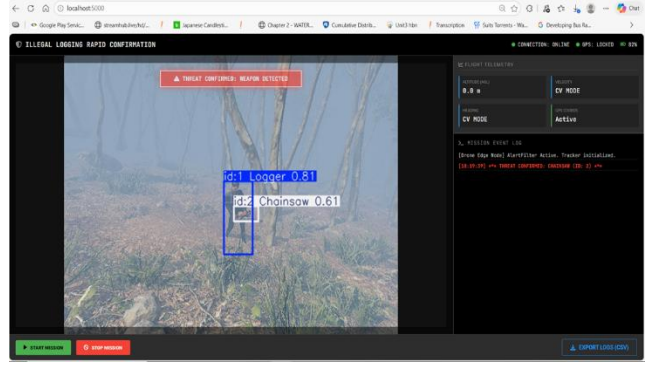
Integration testing within a live AirSim simulation confirmed end-to-end pipeline cohesion across all five modules as shown in Fig. 7. The AirSim interface delivered continuous (640×640×3) RGB frame arrays; the inference engine produced accurate bounding boxes with correct class mapping; the AlertFilter correctly reset its counter on a dropped frame and escalated to "THREAT CONFIRMED" precisely at the five-frame threshold, validating the temporal persistence logic; the database manager persisted detection events asynchronously without blocking the video thread; and the Telemetry API

returned HTTP 200 responses with correct JSON and CSV payloads.

Non-functional performance testing across six detection runs (n=3,856 events) revealed high latency variance: raw average 271.22 ms, range 92–4,173 ms. Z-score analysis (threshold=3.0) identified these spikes as hardware resource contention artefacts arising from simultaneous 3D rendering and neural network execution in the shared simulation environment. The adjusted baseline, excluding outliers, yielded an operational average of 256.18 ms (3.90 FPS). Peak performance reached 92.57 ms (10.80 FPS). Results are summarised in Table 5.



(a) Testing under clear conditions(axe).



(b) Testing under foggy conditions(chainsaw).

Figure 5.7: Integration testing within a live AirSim simulation confirming end-to-end pipeline cohesion across all five modules.

Table 5 Raw vs. adjusted operational performance (Z-score outlier mitigation, threshold = 3.0)

Metric	Raw aggregate (unfiltered)	Adjusted baseline (Z < 3)
Total Detection Events	3,856	3,822 (Excl. 34 outliers)
Average latency	271.22 ms	256.18 ms
Mean throughput	3.69 FPS	3.90 FPS
Peak performance	92.57 ms / 10.80 FPS	92.57 ms / 10.80 FPS
Threat confirm time	—	1.28 s (mean); 0.46 s (peak)

Under the 5-frame AlertFilter at 3.90 FPS, threat confirmation latency is 1.28 s; at peak 10.80 FPS, 0.46 s. Both values fall well below the 2.0-second rapid-response threshold specified for the system. In field deployment on an isolated Edge AI processor, the rendering-induced contention observed here would be eliminated, projecting consistent performance within the 3.90–10.80 FPS range.

5 Discussion

The results demonstrate that progressive domain randomization effectively addresses the core challenge of background confusion in forest surveillance. The 15-percentage-point reduction in Chainsaw false positives demonstrates that exposing the model to synthetic atmospheric extremes and color jittering compels it to develop morphological descriptors invariant to surface-level visual noise. The improvement in $mAP@0.5:0.95$ from 65.60% to 74.86% is particularly significant: this stricter metric penalizes imprecise localizations, indicating that randomization improved not just detection rates but bounding box accuracy required for reliable GPS-tagged evidence generation.

The near-perfect Logger detection (99.32% $mAP@0.5$) is operationally critical. The AlertFilter's threat confirmation logic requires a Logger to be co-detected with a tool, so missed Logger detections are a failure contributor to false negatives at the system level. The 1% false negative rate for Logger effectively minimizes this risk.

The low false negative rates: 1%, 4%, and 9% for Logger, Axe, and Chainsaw respectively are especially important given the operational context. A false negative in illegal logging detection represents an undetected threat with potentially irreversible environmental consequences. The progressive training strategy elevated aggregate Recall from 0.8908 (Stage 1) to 0.9301 (Stage 2), confirming that enhanced environmental variety improved the model's geometric understanding of tool shapes without compromising precision.

While direct metric comparison to prior work such as Ramadan et al. [8] and Aman et al. [18] is not feasible due to differing class definitions, evaluation datasets, and operational contexts, the achieved $mAP@0.5$ of 96.80% on real-world imagery compares favourably to reported YOLO-based UAV detection results in analogous surveillance domains, typically ranging from 88% to 95% $mAP@0.5$ [9, 24, 25]. The temporal AlertFilter and edge-only inference architecture represent differentiating contributions beyond raw detection accuracy. Ramadan et al. [8] remains the most architecturally comparable system, also confirming illegal logging alerts via UAV visual inference, but it offloads the detection decision to a commercial cloud service and reports no performance metrics for that component. By contrast, the present system embeds a custom-trained, fully documented detector onboard the UAV, eliminating cloud dependency while providing a transparent, quantified accuracy benchmark absent from that prior work.

The benefit of augmenting a real-world-trained baseline with domain-randomized synthetic data is further contextualized by two studies on sim-to-real transfer for constrained-platform object detection [33, 34]. Bay et al. [33] found that mixed real-synthetic training outperformed synthetic-only training, but observed diminishing returns as the real-world proportion of the corpus grew, with gains flattening on a fixed 20-image real-world test set. The present study's real-dominant split (57% real to 43% synthetic) sits in a region of that spectrum where Bay et al.'s trend would predict minimal benefit. Yet, a 0.0926 $mAP@0.5:0.95$ improvement was observed, evaluated against a proportionally scaled, 659-image real-world holdout, suggesting that larger overall data volume and a larger evaluation set can offset the plateau effect reported in more constrained studies.

Peterson et al. [34], working in the orbital hardware-in-the-loop domain, achieved 0.87 $mAP@0.5$ and 0.73

mAP@0.5:0.95 using a 90% synthetic to 10% real split, reflecting that domain's practical inability to capture sufficient real imagery. The present system achieved a comparable 0.7486 mAP@0.5:0.95 and a substantially higher 0.9680 mAP@0.5 using less than half the proportional synthetic volume, indicating that for terrestrial forestry applications, where a substantial real-world baseline can be assembled, a minority, targeted synthetic injection is sufficient to meaningfully improve a real-world-trained detector.

The latency variance observed in integration testing reflects a known artefact of shared simulation hardware rather than an algorithmic constraint. In physical deployment, an isolated Edge AI processor such as the NVIDIA Jetson Nano would eliminate resource contention, projecting consistent performance within the 3.90–10.80 FPS range, sufficient for the 2.0-second confirmation threshold.

The automated synthetic annotation pipeline offers methodological value beyond this study. By leveraging AirSim semantic segmentation masks for automated YOLO-format annotation generation, the approach eliminates the primary bottleneck in training data preparation for niche scenarios where labelled datasets are scarce. This methodology is directly transferable to other conservation AI applications including poaching detection and wildlife monitoring.

Current limitations include: (i) the model is restricted to three classes, leaving heavy machinery and logging vehicles undetected; (ii) the sim-to-real gap remains to be validated through physical field deployment, where aerodynamic turbulence, motion blur, and sensor noise not present in simulation may degrade performance; (iii) the offline SQLite architecture lacks real-time multi-agency data sharing required for coordinated enforcement responses. These limitations define the primary directions for future work.

6 Conclusion

This paper presented a proactive UAV-based edge intelligence system for rapid visual confirmation of illegal logging activities in forest environments with up to 63% canopy density. Two primary contributions were demonstrated. First, domain-randomized synthetic data generated via AirSim and Unreal Engine with automated YOLO-format annotation constitutes an effective and scalable substitute for manual data collection when improving a real-world-trained baseline detector.

Second, the end-to-end detection pipeline was validated within a live photorealistic simulation, successfully streaming annotated video, confirming threats through a five-frame BoT-SORT persistence mechanism, and logging all events to the database, demonstrating full operational viability from frame acquisition to actionable alert generation for a YOLOv8s illegal logging rapid confirmation detector.

The immediate priority for future work is physical deployment on a dedicated Edge AI processor to validate sim-to-real transfer under authentic aerodynamic and environmental conditions. Longer-term directions include

multi-modal fusion with thermal and acoustic sensors, expanded class coverage to include logging machinery, and asynchronous cloud synchronization for multi-agency coordination.

Declarations

Author Contributions. This paper is based on research conducted as part of the requirements for the Master of Science in Computing and Information Systems degree at the School of Computing and Engineering Sciences, Strathmore University, Nairobi, Kenya.

S.M. conceived and implemented the system, conducted all experiments, analyzed the data, and wrote the manuscript. H.M. supervised the research, provided guidance on methodology and academic framing, and critically reviewed the manuscript.

Funding. This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Data Availability. The training datasets and trained model weights are publicly available via Google Drive.

Stage 1:

<https://drive.google.com/drive/folders/1bDmBbZluQgv06z1Aljc0pW51-aMe2glj?usp=sharing>

Stage 2:

<https://drive.google.com/drive/folders/145cvXRAceo6DBCxepCTzyScJALPermWL?usp=sharing>

Data Sources - Provenance Logs:

<https://drive.google.com/drive/folders/1x67slG9JaJurySs6riVi4IRruOCqVZju?usp=sharing>

Data Sources - Downloaded Videos:

https://drive.google.com/drive/folders/104hUfY_z2yWcK0b0xC7MED7P2lxXU4yt?usp=sharing

Competing Interests. The authors declare no competing interests.

References

- [1] World Bank, "Illegal Logging, Fishing, and Wildlife Trade," World Bank, Washington, DC, 2019.
- [2] L. Kimbrough, "Deforestation continues in Kenya's largest water capturing forest, satellites show," *Mongabay Environ. News*, Nov. 2023.
- [3] J. G. Crafford and R. Strohmaier, "The role and contribution of montane forests and related ecosystem services to the Kenyan economy," UNEP, 2012.
- [4] I. Osewe et al., "Balancing conservation and community needs: Challenges, threats, and socioeconomic disparities in Kakamega Forest, Kenya," *Front. Forests Global Change*, vol. 8, p. 1554677, 2025.
- [5] JICA and JAXA, "JJ-FAST Technical Note," Version 10, JICA/JAXA, 2024.
- [6] A.-J. Welsink et al., "Towards the use of satellite-based tropical forest disturbance alerts to assess selective logging intensities," *Environ. Res. Lett.*, vol. 18, no. 5, p. 054023, 2023.
- [7] J. Paneque-Gálvez et al., "Small Drones for Community-Based Forest Monitoring," *Forests*, vol. 5, no. 6, pp. 1481–1507, 2014.
- [8] M. N. A. Ramadan et al., "AI-powered IoT and UAV systems for real-time detection and prevention of illegal logging," *Results Eng.*, vol. 24, p. 103277, 2024.
- [9] R. G. Sangeetha et al., "Unmanned Aerial Surveillance and Tracking System in Forest Areas for Poachers and Wildlife," *IEEE Access*, vol. 12, pp. 187572–187586, 2024.
- [10] Y. Yuan et al., "Edge-Cloud Collaborative UAV Object Detection," *IEEE Trans. Cloud Comput.*, vol. 12, no. 1, pp. 306–318, 2024.
- [11] M. Anagnostou et al., "An international scoping review of rangers' precarious employment conditions," *Environ. Syst. Decisions*, vol. 42, no. 4, pp. 479–503, 2022.
- [12] M. C. Hansen et al., "High-Resolution Global Maps of 21st-Century Forest Cover Change," *Science*, vol. 342, no. 6160, pp. 850–853, 2013.
- [13] J. F. Roberts et al., "Pyeo: A Python package for near-real-time forest cover change detection," *Comput. Geosci.*, vol. 167, p. 105192, 2022.
- [14] A. Belmonte et al., "UAV-derived estimates of forest structure to inform ponderosa pine forest restoration," *Remote Sens. Ecol. Conserv.*, vol. 6, no. 2, pp. 181–197, 2020.
- [15] S. Ecke et al., "UAV-Based Forest Health Monitoring: A Systematic Review," *Remote Sensing*, vol. 14, no. 13, p. 3205, 2022.
- [16] Z. Jiao et al., "A Deep Learning Based Forest Fire Detection Approach Using UAV and YOLOv3," in *Proc. 1st Int. Conf. Industrial AI (IAI)*, 2019, pp. 1–5.
- [17] J. Williams, "UAV survey mapping of illegal deforestation in Madagascar," *Plants People Planet*, vol. 6, no. 6, pp. 1413–1424, 2024.
- [18] I. Aman and J. Saini, "YOLO-NAS-powered Forest Monitoring System," in *Proc. 7th Int. Conf. Image Inf. Process. (ICIIP)*, 2023, pp. 312–315.
- [19] R. Girshick et al., "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," in *Proc. CVPR*, 2014, pp. 580–587.
- [20] S. Ren et al., "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, 2017.
- [21] W. Liu et al., "SSD: Single Shot MultiBox Detector," in *Proc. ECCV*, vol. 9905, pp. 21–37, 2016.
- [22] J. Redmon et al., "You Only Look Once: Unified, Real-Time Object Detection," in *Proc. CVPR*, 2016, pp. 779–788.
- [23] A. Bochkovskiy et al., "YOLOv4: Optimal Speed and Accuracy of Object Detection," *arXiv:2004.10934*, 2020.
- [24] Z. Wu et al., "Dense object detection methods in RAW UAV imagery based on YOLOv8," *Sci. Rep.*, vol. 14, p. 18019, 2024.
- [25] Z. Song et al., "An infrared dataset for partially occluded person detection in complex environment for search and rescue," *Sci. Data*, vol. 12, p. 300, 2025.
- [26] A. Bewley et al., "Simple Online and Realtime Tracking," in *Proc. ICIP*, 2016, pp. 3464–3468.
- [27] N. Wojke et al., "Simple Online and Realtime Tracking with a Deep Association Metric," *arXiv:1703.07402*, 2017.
- [28] N. Aharon et al., "BoT-SORT: Robust Associations Multi-Pedestrian Tracking," *arXiv:2206.14651*, 2022.
- [29] B. Alvey et al., "Simulated Photorealistic Deep Learning Framework and Workflows to Accelerate Computer Vision and UAV Research," in *Proc. ICCVW*, 2021, pp. 3882–3891.
- [30] A. R. Buck et al., "Unreal engine-based photorealistic aerial data generation and unit testing of AI algorithms," in *Geospatial Informatics XII*, 2022.
- [31] S. Shah et al., "AirSim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles," in *Field and Service Robotics*, 2017.
- [32] J. Tobin et al., "Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World," in *Proc. IROS*, 2017.
- [33] M. Bay, T. von Marcard, and D. Fazlija, "The Impact of Synthetic Data on Object Detection Model Performance: A Comparative Analysis with Real-World Data," *arXiv:2510.12208*, 2025.
- [34] M. Peterson, M. Du, B. Springle, and J. Black, "SpaceDrones 2.0—Hardware-in-the-Loop Simulation and Validation for Orbital and Deep Space Computer Vision and Machine Learning Tasking Using Free-Flying Drone Platforms," *Aerospace*, vol. 9, no. 5, p. 254, 2022.
- [35] Axe and Chainsaw, "Axe Chainsaw Dataset," *Roboflow Universe*, 2024. [Online]. Available: <https://universe.roboflow.com/axe-and-chainsaw/axe-chainsaw-0k6zh>