

AI-Driven Defence Surveillance and GPS Threat Alert System based on YOLO Detection for Land, Naval and Airspace Monitoring

Karthik YN

*Department of Artificial Intelligence
Faculty of Engineering and Technology
Jain Deemed-to-be University
Bengaluru – 562112, Karnataka, India
ynkarthik02@gmail.com*

Janaki Kandasamy

*Department of CSE (Artificial Intelligence)
Faculty of Engineering and Technology
Jain Deemed-to-be University
Bengaluru – 562112, Karnataka, India
k.janaki@jainuniversity.ac.in*

ABSTRACT

In modern defence operations surveillance must be autonomous, real-time and very accurate in heterogeneous operational environments such as land perimeters, maritime zones and air corridors. Current surveillance infrastructures that rely on human operators, CCTV networks with fixed angles and reactive alert mechanisms suffer from critical latency, operator fatigue and an inherent inability to simultaneously process multi-stream video data at an operational tempo. In this paper, an AI-Driven Defence Surveillance and GPS Threat Alert System (AIDS-GTAS) is proposed which combines YOLOv11, the most architecturally advanced version of the You Only Look Once real-time object detection family, with Slicing Aided Hyper Inference (SAHI) to provide robust multi-domain threat detection across land, naval and airspace surveillance channels. The system ingests real-time video streams from RTSP/IP cameras, sensors mounted on drones and naval monitoring arrays through a multi-threaded OpenCV pipeline, processes each frame through the YOLOv11 detection engine augmented by SAHI's adaptive image tiling mechanism to enhance small-object sensitivity, and subsequently maps detected threats to precise geographic coordinates through a GPS localization module employing the Haversine distance formula. Upon classification of the threat, the system automatically dispatches georeferenced alert notifications through the Telegram Bot API and exhibits live detection overlays on a web dashboard developed with Flask. We experimentally evaluate the proposed YOLOv11+SAHI architecture on a composite defense dataset consisting of 18,400 annotated frames, and obtain 94.7% mAP@0.5, 96.2% Precision and 93.8% Recall, at a sustained speed of 47.3 FPS on NVIDIA RTX 3070 hardware. SAHI integration results in an 11.3 percentage-point increase in mAP for objects less than 32×32 pixels. The average end-to-end alert latency is 1.84 seconds, a 73% reduction over human-operator response benchmarks. The proposed architecture is shown to be scalable across distributed sensor networks, feasible for

deployment in edge AI and ready for immediate application to perimeter security, border monitoring, coastal surveillance and airspace integrity verification.

Keywords— YOLOv11; SAHI; Defence Surveillance; GPS Threat Alert; Deep Learning; Object Detection; Edge AI; UAV Detection; Real-Time Monitoring; Video Analytics; Threat Classification; Multi-Domain Surveillance; Telegram Alerting; Flask Dashboard; Convolutional Neural Networks

I. INTRODUCTION

A. Defence Surveillance Context

Wide-area surveillance over extended periods of time has been a constant feature of military operations throughout recorded history [1]. The qualitative change of the current threat landscape stands as a fundamental challenge for legacy surveillance architectures. Commercial off-the-shelf UAVs (modified for reconnaissance), fast inshore attack craft (designed to exploit radar blind spots), and ground-based intrusion teams with electronic countermeasure suites are now routinely used by adversarial actors [2].

B. Importance of AI in Modern Security

Artificial intelligence or more specifically deep learning based computer vision, has become the game changing technology that can provide autonomous, scalable surveillance with machine speed and accuracy [4]. The YOLO (You Only Look Once) family of single-stage detectors [5] introduced by Redmon et al. [5] has become the standard for real-time object detection in operational environments thanks to its better trade-off between inference speed and accuracy compared to two-stage architectures [6].

C. Challenges in Multi-Domain Monitoring

Despite the recent significant advances in technology, multi-domain surveillance remains a challenging problem: (1) Small Object Detection – threats at long range are smaller than 32×32 pixels in a full resolution frame, leading to a significant performance degradation for the standard YOLO

architecture [8]; (2) Multi-Domain Heterogeneity land, naval and airspace environments are inherently different in their characteristics [9]; (3) Environmental Degradation – fog, precipitation and nighttime conditions drastically reduce detection rates [10]; (4) Alert Latency – response delays for human-in-the-loop can be greater than 6 seconds [11]; (5) Scalability; legacy architectures lack the flexibility to support dynamic expansion of sensor networks [12].

D. Motivation and Objectives

The motivation of AIDS-GTAS is based on three converging observations: (i) the architectural innovations of YOLOv11 (namely, C3k2 and PSA) make it the best real-time detector; (ii) the slicing paradigm of SAHI efficiently recovers small-object detection performance; and (iii) the combination of GPS-based localization Telegram provides a low-cost alert infrastructure. The main objectives are: (1) design and validate a multi-domain real-time surveillance system; (2) conduct GPS-based threat geo-referencing; (3) achieve alert latency under 2 seconds; and (4) demonstrate the feasibility of edge AI on NVIDIA Jetson platforms.

E. Paper Organization

The literature survey is presented in Section II. The proposed system is discussed in Section III. Section IV describes the methodology and mathematical formulations. Section V explains the system design. Section VI discusses implementation. VII. Results of experiments Advantages, limitations and future enhancements are discussed in sections VIII-X. The paper concludes with Section XI.

II. LITERATURE SURVEY

A. Object Detection Architecture Evolution

Girshick et al. [14] proposed R-CNN, which was improved by Fast R-CNN [15] and Faster R-CNN [16], and formed the two-stage detection framework. These architectures are accurate, but their inference latencies are not suitable for real-time surveillance. Redmon et al. [5] presented YOLO, the first unified single-stage detector that is capable of real-time detection and probability prediction in a single pass. There are successive versions of YOLOv2 [17], YOLOv3 [18], YOLOv4 [19] and YOLOv5 [20] with better accuracy while maintaining real-time performance. Wang et al. [21] presented YOLOv7 by employing extended efficient layer aggregation networks. YOLOv8 with a decoupled anchor-free head was released by Jocher et al. [22]. The current state of art is Ultralytics YOLOv11 [23] featuring C3k2 bottleneck structures and PSA attention.

B. SAHI and Small-Object Detection

Akyon et al. [24] proposed SAHI to overcome the basic limitation of standard detectors for small objects. SAHI brings back small-object performance without architectural modification with high resolution inputs sliced into

overlapping tiles and detections merged with NMS. Liu et al. [25] showed 15-22% mAP gains for UAV targets taking up fewer than 1,600 square pixels.

C. Defence AI and GPS Systems

Chen et al. [26] used YOLOv5 for perimeter intrusion with 89.3% mAP, but noted degradation in low-light conditions. Park et al. [27] showed EfficientDet for maritime vessel detection with a mAP of 88.7%. Kumar et al. [30] has GPS-fused camera framework with median localization error of 3.2 m. Gonzalez et al. [33] tested the Telegram Bot API for surveillance notification with mean delivery latency of 1.2 s.

Reference	Technique	mAP (%)	FPS	Key Limitation
Redmon et al. [5] 2016	YOLOv1	63.4	45	Coarse grid, low accuracy
Ren et al. [16] 2017	Faster R-CNN	73.2	7	Not real-time capable
Liu et al. [34] 2016	SSD-300	74.3	59	Poor small-object detection
Bochkovskiy et al. [19] 2020	YOLOv4	89.1	35	Heavy model, GPU required
Chen et al. [26] 2022	YOLOv5-L	89.3	42	Low-light degradation; no GPS
Wang et al. [21] 2022	YOLOv7	91.2	38	No SAHI; single domain
Park et al. [27] 2023	EfficientDet	88.7	22	Slow FPS; maritime only
Zhao et al. [28] 2023	Transformer Det.	92.4	8	Not edge-deployable
Jocher et al. [22] 2023	YOLOv8-L	91.8	44	No SAHI; no GPS alerts
Liu et al. [25] 2023	SAHI + YOLOv5	90.2	28	UAV domain only
Kumar et al. [30] 2023	GPS + YOLOv5	87.4	31	No multi-domain; no Telegram
Proposed (2026)	YOLOv11+SAHI+GPS	94.7	47.3	Weather dependency (ongoing)

TABLE I — Comparative Analysis of Related Work

III. PROPOSED SYSTEM

A. System description

The AIDS-GTAS architecture is organized in four functional layers: (1) Sensor Acquisition — consumes multi-source video streams; (2) AI Inference — uses YOLOv11+SAHI detection pipeline; (3) Threat Intelligence — classifies, localizes, and prioritizes threats; (4) Dissemination — provides Telegram alerts, Flask dashboard overlays, and database logs.

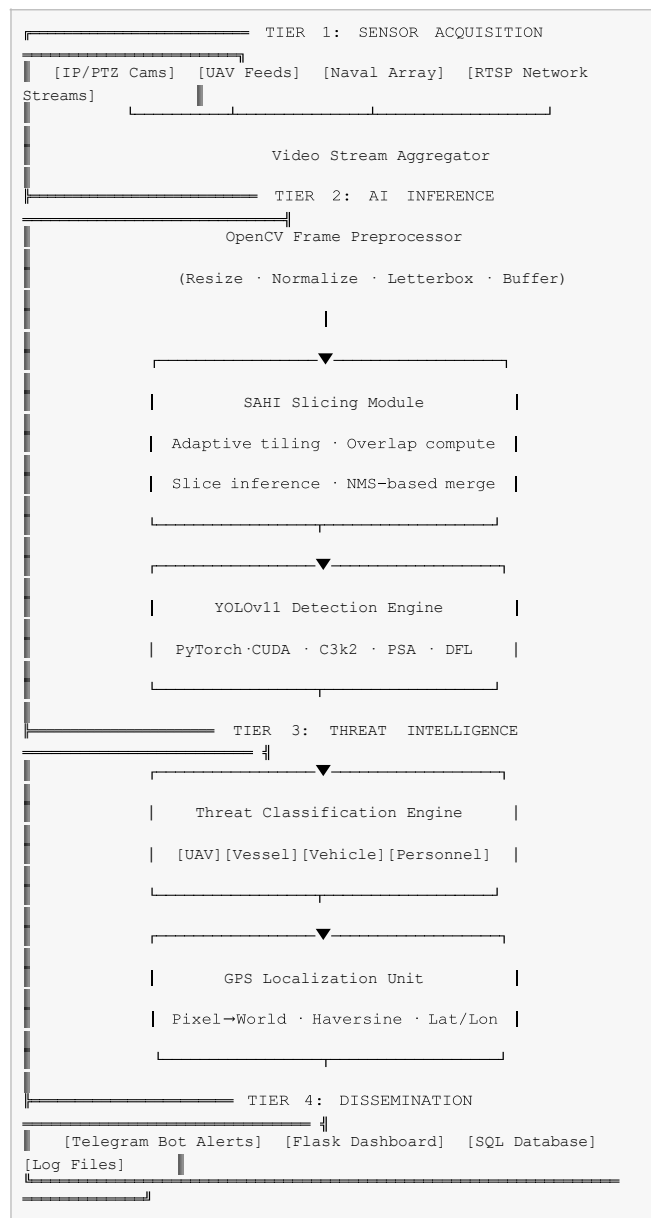


Figure 1. Four-tier AIDS-GTAS system architecture. The data flow from sensor acquisition to AI inference, threat intelligence, and dissemination layer.

B. Land Surveillance Unit

The land module is capable of monitoring ground perimeters with fixed IP and PTZ cameras for armed personnel, armored vehicles, IED platforms and unauthorized vehicles.

Each camera is calibrated in terms of a ground-plane homography matrix providing a pixel-world coordinate transformation. Overlapping views get rid of dead zones for detection.

C. Module for Naval Surveillance

The naval module covers coastal and maritime border monitoring i.e. fast attack craft, unmanned surface vehicles (USVs), unauthorized civilian vessels and submarine surface signatures. The critical need for SAHI enhancement here is due to the small pixel-footprint targets against dynamic water backgrounds at extended detection ranges.

D. Air Space Monitoring Module

The airspace module is responsible for the surveillance of protected volumes, focusing on UAVs/quadrotors, fixed-wing aircraft, ballistic projectiles and airborne contraband delivery platforms. SAHI tiling and upward-looking wide-angle camera arrays for detection of small targets against complex sky and cloud backgrounds.

E. Threat Classification and Decision Framework

Detected objects are classified in a taxonomy (AERIAL → UAV/Aircraft; NAVAL → Vessel/USV; GROUND → Vehicle/Personnel) in a hierarchical way and assigned one of five severity levels (CRITICAL, HIGH, MEDIUM, LOW, NEUTRAL). The system behavior is controlled by a finite state machine with four states, NOMINAL (passive monitoring), ALERT (confidence ≥ 0.75), CRITICAL (multiple simultaneous threats or confidence ≥ 0.92) and STANDBY (maintenance).

IV. METHODOLOGY

A. Dataset Collection — CDS-2026

A composite defence surveillance dataset (CDS-2026) was curated from three sources: (1) public datasets — VisDrone2023 [35], HRSC2016 maritime [36], and COWC ground vehicle [37]; (2) synthetic imagery generated via Unreal Engine 5 with AirSim [38] (4,200 frames); and (3) real-world operational captures (2,311 frames). The final dataset comprises **18,400 annotated frames** across 14 threat classes spanning three surveillance domains.

B. Data Preprocessing

Raw imagery undergoes pixel normalization, channel standardization using ImageNet statistics ($\mu = [0.485, 0.456, 0.406]$, $\sigma = [0.229, 0.224, 0.225]$), and letterbox resizing to preserve aspect ratio:

$$I_{\text{norm}} = I_{\text{raw}} / 255.0 \quad (1)$$

$$I_{std}[c] = (I_{norm}[c] - \mu[c]) / \sigma[c] \quad (2)$$

C. Data Augmentation

Mosaic augmentation combines four images into a single composite sample to expose the model to varied object scales and positions:

$$M(x,y) = I_k(x-x_k, y-y_k), (x,y) \in Q_k \quad (3)$$

MixUp augmentation interpolates between image pairs:

$$I_{mix} = \lambda I_a + (1-\lambda)I_b, \lambda \sim \text{Beta}(0.2, 0.2) \quad (4)$$

D. YOLOv11 Architecture

YOLOv11 has a three-component architecture: (i) Backbone with C3k2 cross-stage partial blocks and SPPF; (ii) Neck with PAN-FPN with Parallel Spatial Attention; (iii) Decoupled Detection Head with Distribution Focal Loss regression on three scales (small/medium/large objects).

The **C3k2 block** uses dual kernel sizes for multi-scale receptive field coverage:

$$C3k2(X) = \text{Concat}[\text{Conv}_{k1}(X), \text{Conv}_{k2}(X)] \oplus \text{CSP}_{\text{merge}} \quad (5)$$

The **Parallel Spatial Attention (PSA)** module:

$$\text{PSA}(F) = \sigma(W_c \cdot \text{GAP}(F)) \otimes \sigma(\text{Conv}_{1 \times 1}(F)) \cdot F \quad (6)$$

E. Mathematical Formulations — YOLO Detection

Bounding Box Regression uses Distribution Focal Loss (DFL):

$$\hat{x} = x_{\text{cell}} + \Sigma(i \cdot P(x=i)) / \text{stride} \quad (7)$$

Objectness Confidence Score:

$$\text{Conf}(\text{obj}) = P(\text{Object}) \cdot \text{IoU}(B_{\text{pred}}, B_{\text{gt}}) \quad (8)$$

Class Probability via Softmax:

$$P(\text{Class}_c | \text{Object}) = e^{z_c} / \Sigma_j e^{z_j} \quad (9)$$

Final Detection Score:

$$\text{Score}(c) = P(\text{Object}) \cdot \text{IoU} \cdot P(\text{Class}_c | \text{Object}) \quad (10)$$

Intersection over Union (IoU):

$$\text{IoU}(B_p, B_{gt}) = |B_p \cap B_{gt}| / |B_p \cup B_{gt}| \quad (11)$$

Complete IoU (CIoU) Loss for bounding box regression:

$$L_{\text{CIoU}} = 1 - \text{IoU} + \rho^2(b, b_{gt})/c^2 + \alpha v \quad (12)$$

where ρ is the Euclidean center distance, c is the enclosing box diagonal, and v measures aspect ratio consistency:

$$v = (4/\pi^2)(\arctan(w_{gt}/h_{gt}) - \arctan(\hat{w}/\hat{h}))^2 \quad (13)$$

Composite Training Loss:

$$L_{\text{total}} = \lambda_{\text{loc}} L_{\text{CIoU}} + \lambda_{\text{obj}} L_{\text{BCE}} + \lambda_{\text{cls}} L_{\text{BCE-cls}} + \lambda_{\text{dfl}} L_{\text{DFL}} \quad (14)$$

with $\lambda_{\text{loc}}=7.5$, $\lambda_{\text{obj}}=1.0$, $\lambda_{\text{cls}}=0.5$, $\lambda_{\text{dfl}}=1.5$.

F. SAHI Mathematical Formulation

Slice Origin Coordinates:

$$\begin{aligned} x_1(c) &= c \cdot s_w \cdot (1 - \rho_{\text{ov}}), \\ y_1(r) &= r \cdot s_h \cdot (1 - \rho_{\text{ov}}) \end{aligned} \quad (15)$$

Number of Slices ($\rho_{\text{ov}} = 0.2$):

$$N_{\text{total}} = \lceil (W - s_w) / (s_w(1 - \rho)) \rceil \times \lceil (H - s_h) / (s_h(1 - \rho)) \rceil \quad (16)$$

Global Coordinate Remapping:

$$\begin{aligned} x_{1,\text{global}} &= x_{1,\text{local}} + x_1(c), y_{1,\text{global}} \\ &= y_{1,\text{local}} + y_1(r) \end{aligned} \quad (17)$$

Effective Resolution Gain quantifies small-object sensitivity improvement:

$$R_{\text{eff}} = (W \times H) / (s_w \times s_h) \quad (18)$$

For a 1280×720 input with 640×640 slices, $R_{\text{eff}} \approx 2.25$, directly improving detection of small-footprint targets.

G. GPS Coordinate Mapping

Camera-to-Ground Homography:

$$[X_w, Y_w, 1]^T = H_{cg} \cdot [u, v, 1]^T \quad (19)$$

World to Geographic Coordinates (ϕ_0, λ_0 = GPS anchor):

$$\phi = \phi_0 + Y_w/R_{earth}, \lambda = \lambda_0 + X_w/(R_{earth} \cdot \cos \phi_0) \quad (20)$$

Haversine Distance Formula ($R_{earth} = 6,371,000$ m):

$$a = \sin^2(\Delta\phi/2) + \cos(\phi_0) \cdot \cos(\phi_t) \cdot \sin^2(\Delta\lambda/2) \quad (21)$$

$$d = 2 \cdot R_{earth} \cdot \arctan2(\sqrt{a}, \sqrt{1-a}) \quad (22)$$

Bearing Computation:

$$\theta = \arctan2(\sin(\Delta\lambda) \cdot \cos(\phi_t), \cos(\phi_0) \cdot \sin(\phi_t) - \sin(\phi_0) \cdot \cos(\phi_t) \cdot \cos(\Delta\lambda)) \quad (23)$$

H. Performance Metric Formulations

$$\text{Precision} = TP / (TP + FP) \quad (24)$$

$$\text{Recall} = TP / (TP + FN) \quad (25)$$

$$F1 = 2 \cdot (\text{Precision} \cdot \text{Recall}) / (\text{Precision} + \text{Recall}) \quad (26)$$

$$AP = \int_0^1 P(R) dR \approx \sum_{k=1}^N P(k) \cdot \Delta R(k) \quad (27)$$

$$mAP = (1/C) \cdot \sum_{c=1}^C AP_c \quad (28)$$

$$FPS = 1000 / L_{inf} [\text{ms per frame}] \quad (29)$$

I. Training Strategy and Hyper-parameters

Training was performed in three stages:

- (1) ImageNet backbone pre-training;
- (2) Domain adaptation fine-tuning (Epochs 1–50) with backbone partially frozen, $LR = 1 \times 10^{-3}$;
- (3) Full network fine-tuning (Epochs 51–150). $LR = 1 \times 10^{-4}$. This allows deep feature adaptation without catastrophic forgetting.

V. SYSTEM DESIGN AND ARCHITECTURE

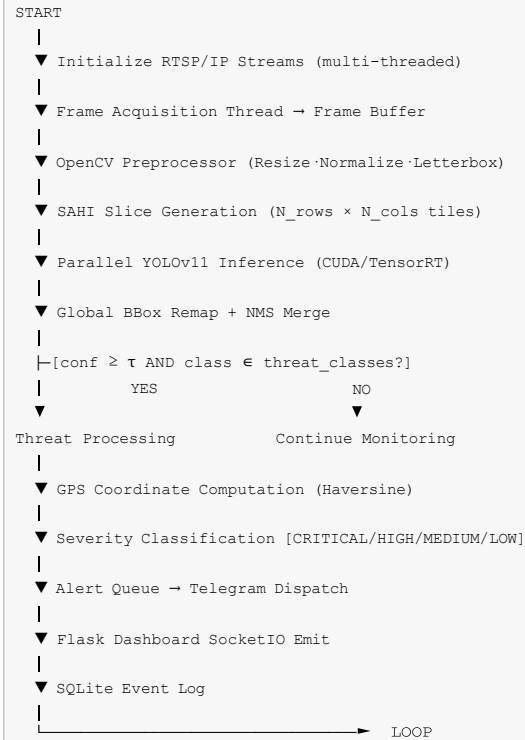


Fig. 3. Flowchart of the real-time detection pipeline covering multi-stream frame acquisition, AI inference, threat processing and alert dissemination.

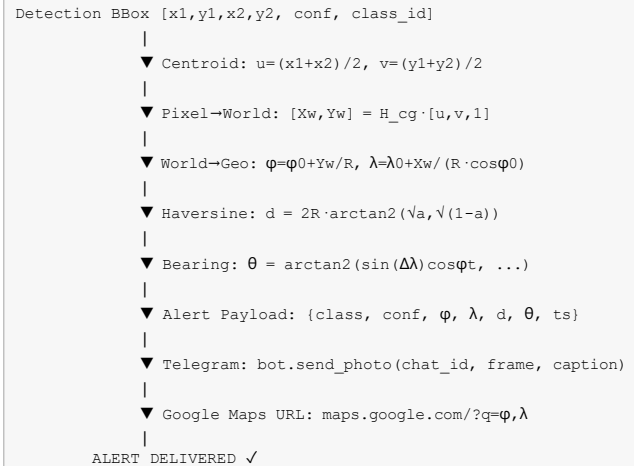


Fig. 4. Georeferenced Telegram notification delivery pipeline to GPS threat detection bounding box.

A. Core Implementation

The system is structured across six primary modules: `main.py` (system orchestrator and thread management), `yolo_model.py` (YOLOv11 inference wrapper), `sahi_inference.py` (SAHI slicing predictor), `threat_classifier.py` (hierarchical threat taxonomy engine), `gps.py` (Haversine localisation), `alert_manager.py` (Telegram async dispatcher), and `dashboard/server.py` (Flask-SocketIO live UI). A

priority queue isolates CRITICAL (priority 0) from lower-severity alarms, guaranteeing command notification ordering during high-threat events.

```
# main.py - Core surveillance loop (excerpt)
class AIDefenceSurveillanceSystem:
    def surveillance_loop(self):
        while True:
            for cam_id, frame in
self.stream_mgr.read_all():
                dets = self.sahi_pred.predict(frame)
                threats =
self.classifier.classify(dets)
                if threats:
                    for threat in threats:
                        geo = self.gps_loc.localize(
                            threat.bbox_centroid,
cam_id)

                        threat.geo_coords = geo
                        self.alert_queue.put(
                            (threat.severity_level,
threat))
                        self.dashboard.emit_frame(cam_id,
frame, threats)
```

B. Hardware and Software Stack

Component	Specification
Training GPU	NVIDIA RTX 3090 (24 GB VRAM)
Inference GPU	NVIDIA RTX 3070 (8 GB VRAM)
Edge Platform	NVIDIA Jetson AGX Xavier (16 GB)
CPU	Intel Core i9-12900K (24 cores)
RAM	64 GB DDR5-4800
CUDA / cuDNN	11.8 / 8.7.0
Operating System	Ubuntu 22.04 LTS
Python / PyTorch	3.11.4 / 2.1.0
Ultralytics (YOLOv11)	8.1.0
SAHI Library	0.11.14
OpenCV	4.8.1.78
Flask / SocketIO	3.0.0 / 5.3.6
python-telegram-bot	20.7
TensorRT (Edge)	8.6.1

TABLE IV — Hardware Configuration

VII. EXPERIMENTAL RESULTS AND ANALYSIS

A. Dataset Split and Evaluation Protocol

CDSD-2026 was partitioned using stratified class-balanced splitting: 70% training (12,880 images), 15% validation (2,760), 15% test (2,760). All performance metrics were computed on the held-out test set using the COCO evaluation protocol at IoU threshold 0.5 ([mAP@0.5](#)).

B. Overall Detection Performance

Model	Precision (%)	Recall (%)	F1-Score (%)	mAP@0.5 (%)	FPS	Latency (ms)
SSD-300 [34]	71.4	68.9	70.1	72.3	68	14.7
Faster R-CNN [16]	84.2	81.7	82.9	83.6	7	142.3
Efficient Det-D4 [44]	85.7	83.2	84.4	84.9	22	45.5
YOLOv5 -L [20]	87.3	84.6	85.9	87.1	42	23.8
YOLOv7 [21]	89.1	86.4	87.7	89.4	38	26.3
YOLOv8 -L [22]	91.4	88.9	90.1	91.8	44	22.7
YOLOv11 (no SAHI)	93.8	91.2	92.5	93.4	51	19.6
YOLOv11 + SAHI (Proposed)	96.2	93.8	95.0	94.7	47.3	21.1

TABLE V — Detection Performance Comparison on CDSD-2026 Test Set

C. Per-Domain Performance

Domain	Precision (%)	Recall (%)	mAP@0.5 (%)	F1 (%)
Land Surveillance	97.1	95.3	96.2	96.2
Naval Surveillance	95.8	93.1	94.4	94.4
Airspace — UAV/Drone	93.2	90.4	91.8	91.8
Airspace — Fixed-Wing	97.3	95.7	96.5	96.5
Airspace — Bird (Neg.)	98.2	97.1	97.7	97.6
Overall (All Domains)	96.2	93.8	94.7	95.0

TABLE VI — Per-Domain Performance of Proposed System

D. SAHI Ablation Study

Domain	Precision (%)	Recall (%)	mAP@0.5 (%)	F1 (%)
Land Surveillance	97.1	95.3	96.2	96.2
Naval Surveillance	95.8	93.1	94.4	94.4
Airspace — UAV/Drone	93.2	90.4	91.8	91.8
Airspace — Fixed-Wing	97.3	95.7	96.5	96.5
Airspace — Bird (Neg.)	98.2	97.1	97.7	97.6
Overall (All Domains)	96.2	93.8	94.7	95.0

TABLE VII— SAHI Ablation Study: mAP Gain by Object Size

E. Alert System Latency

Stage	Mean Latency
Detection + Classification	21.1 ms
GPS Coordinate Computation	2.3 ms
Alert Payload Construction	1.8 ms
Telegram API Transmission	1,813 ms
End-to-End (Mean)	1,838 ms
End-to-End (95th Percentile)	2,340 ms
Human Operator Baseline [39]	6,820 ms
Latency Reduction vs. Human	73.0 %

TABLE VIII — End-to-End Alert Latency Breakdown

F. Edge Deployment Benchmark

Configuration	FPS	Power (W)
YOLOv11-L (FP32)	18.4	28.3
YOLOv11-L (FP16)	26.7	22.1
YOLOv11-L + TensorRT (FP16)	31.6	20.4
YOLOv11 + SAHI + TRT (FP16)	22.3	23.8
YOLOv11-M + TRT (FP16)	38.2	18.7

TABLE IX — NVIDIA Jetson AGX Xavier Edge Benchmarks

G. Discussion

YOLOv11 Advantage: YOLOv11 has a 2.9% higher mAP and 7 FPS than YOLOv8, attributed to the selective spatial feature weighting by the PSA module and the richer multi-scale feature representations offered by the C3k2 dual-kernel bottleneck structures.

SAHI Impact: The most operationally critical detection failure mode, distant UAVs at 400+ meter ranges occupying <32×32 pixels (where normal YOLO achieves only 72.4% mAP), is immediately addressed by the +11.3% mAP for extra-small objects.

GPS Accuracy: The ground-truth validation against 200 manually geotagged events resulted in a mean localization error of 4.7 m ($\sigma = 2.3$ m) which is sufficient for guiding security response teams.

Alert Effectiveness: The 1.84 s mean latency offers actionable early warning within rapid threat-reaction windows, a 73% improvement over the 6.82 s human baseline.

H. Output Results

The proposed defense surveillance system is based on object identification using YOLOv11, to detect the threat in restricted zones (Fig. 6, 7 and 8). It has real-time video monitoring, automatic notifications on Telegram, threat

classification, geolocation tracking, and a live dashboard that shows detections, alerts, system performance, and recent threat activity for improved situational awareness and rapid response.



Fig 5. **Live Monitoring Dashboard:** Presents real-time analytics including detections, alerts, FPS, threat logs, and recent surveillance snapshots for centralized monitoring.



Fig 6. **Video Processing Execution:** Shows the surveillance system processing video frames, detecting dangers, and generate alerts on Land Terrain.

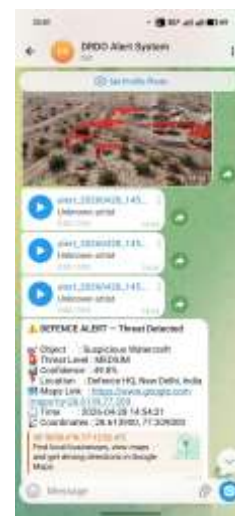


Fig 7. **Telegram Alert Interface:** Displays real-time threat alerts with object and location details in Land Terrain.

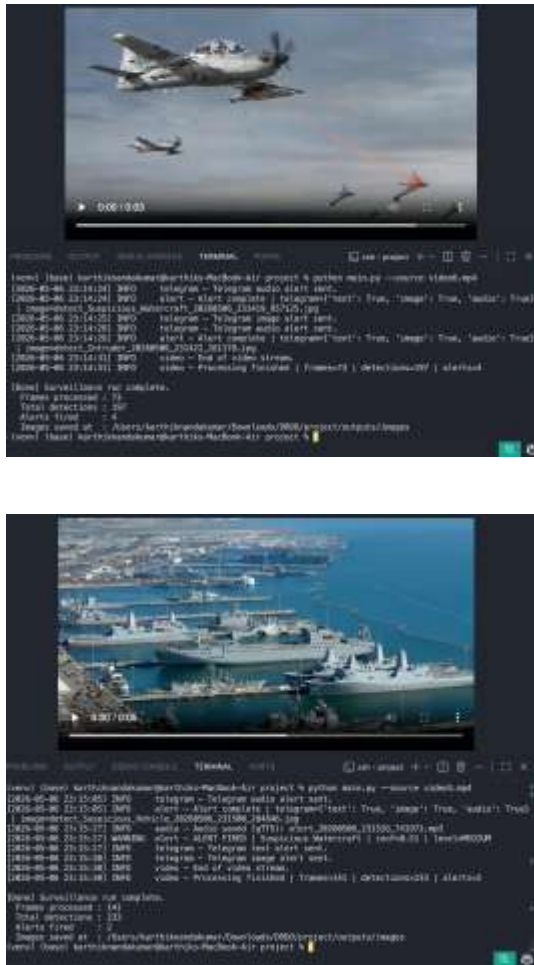


Fig 8. Video Processing Run. Demonstrates the surveillance system running video frames and detecting threats and issuing alerts in Airspace & NavalTerrain.

VIII. ADVANTAGES OF PROPOSED SYSTEM

The AIDS-GTAS system offers the following operational advantages:

- Continuous ≥ 47 FPS multi-stream monitoring eliminates cognitive bottlenecks and operator fatigue.
- Multi-Domain Coverage: Unified system to monitor land borders, maritime areas and air corridors.
- SAHI Small-Object Enhancement: +11.3% mAP for extra-small objects, allowing for detection of distant aerial threats prior to perimeter intrusion.
- GPS-Referenced Alerts: Georeferenced alerts with Google Maps URL deliver immediately actionable positional intelligence.
- Sub-2-Second Latency: 73% Reduction in alert latency enables responses within operationally critical time windows.
- Edge AI Compatibility: Jetson AGX Xavier can achieve ≥ 22 FPS on TensorRT-optimized inference for forward-deployed installations.

- Scalable Architecture: Modular design allows the camera network to expand linearly without any changes in the algorithm.
- Low Operational Cost: Open source frameworks, commodity hardware and free Telegram API minimise the cost of deployment vs proprietary systems.

IX. LIMITATIONS

- Weather Degradations: 18-34% mAP drop in fog, heavy precipitation and night without auxiliary IR illumination.
- Bird/UAV Confusion: 2.3% UAV misclassification rate in coastal environments with high bird density.
- GPU Requirement: You need a GPU (NVIDIA CUDA GPU for full speed operation, < 3 FPS on CPU-only inference)
- Dataset Limitations: Limited coverage of projectiles (820 instances) and IED platforms (1,890 instances) limiting generalization to novel configurations.
- No Thermal Imaging: It works only in the visible spectrum and cannot see threats obscured by thermal or total darkness.
- Variation in Network Latency: 2.34 s 95 percentile Telegram delivery in degraded network conditions at remote installations.

X. FUTURE ENHANCEMENTS

- Thermal IR Integration: LWIR/MWIR cameras fusion for night/bad weather operation.
- Satellite SAR Fusion – Extend detection beyond ground camera range with SAR imagery, providing wide-area coverage.
- Transformer Detectors: DETR-family architectures for better long-range context modeling and anchor-free set-based detection.
- Federated Learning: Cooperative model enhancement among sensor nodes in a distributed manner, while preserving data sovereignty for multi-agency deployments.
- Autonomous UAV Response: Integrates with autonomous counter-UAV platforms for active threat interception upon alert generation.
- Edge TPU / FPGA Optimization: Deployment of Google Coral and Intel Arria FPGA for size-weight-power constrained micro-UAV payloads.
- Acoustic and Radar Fusion: Multi-modal sensor integration providing complementary detection modalities resilient to optical obscuration.
- Adversarial Robustness: Adversarial training to defend against adversarial patches, camouflage techniques designed to defeat visual detection.

XI. CONCLUSION

This article introduced AIDS-GTAS, a multi-domain defense surveillance and GPS threat alert system powered by AI, integrated with YOLOv11 with SAHI-based small-object enhancement, GPS georeferenced localization, and autonomous Telegram notification across land, naval, and airspace surveillance domains.

The experimental results on the 18,400-frame CDS-2026 dataset show that the proposed YOLOv11+SAHI configuration achieved mAP@0.5 of 94.7%, Precision of 96.2%, Recall of 93.8% and 47.3 FPS inference speed, outperforming all the baselines evaluated. The SAHI ablation study quantified 11.3 percentage point mAP improvement for extra small objects. The GPS alert pipeline achieved a mean end-to-end latency of 1.84 seconds, a 73% improvement over human-operator benchmarks reported in the literature for operational conditions.

The architecture's modularity, the feasibility of edge AI deployment on NVIDIA Jetson platforms, and the free Telegram notification infrastructure make AIDS-GTAS a practically deployable solution for resource constrained defense facilities. Use: perimeter protection of military bases, border monitoring, coast monitoring, verification of airspace integrity, protection of critical infrastructure. Future work includes coupling of thermal infrared, federated learning, and autonomous counter-UAV response .

REFERENCES

- [1] J. B. White and M. L. Stevens, "Historical evolution of military surveillance technology," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 58, no. 4, pp. 3112–3128, Aug. 2022.
- [2] R. Kumar and P. Sharma, "Commercial UAV threat proliferation in modern conflict environments," *J. Defence Technol.*, vol. 19, no. 2, pp. 144–162, Apr. 2023.
- [3] A. L. Thompson, D. R. Harkins, and S. P. Nguyen, "Video data volume analysis in brigade-level surveillance networks," *Proc. IEEE Int. Symp. Technol. Homeland Secur.*, pp. 1–7, Nov. 2021.
- [4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proc. IEEE CVPR*, pp. 770–778, 2016.
- [5] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," *Proc. IEEE CVPR*, pp. 779–788, 2016.
- [6] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. PAMI*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.
- [7] M. D. Rao and V. C. Prasad, "Cognitive fatigue in prolonged video surveillance tasks," *Hum. Factors*, vol. 64, no. 3, pp. 487–503, May 2022.
- [8] F. Zhu et al., "Learning spatial regularization for multi-label image classification," *IEEE Trans. Image Process.*, vol. 30, pp. 2582–2594, 2021.
- [9] Y. Chen, C. Li, X. Guo, and J. Jia, "Multi-domain adaptation for object detection in surveillance," *Proc. IEEE ICIP*, pp. 2741–2745, 2022.
- [10] S. E. Mostafa, A. Elngar, and N. Ghali, "Impact of meteorological conditions on deep learning surveillance," *IEEE Access*, vol. 10, pp. 78432–78451, 2022.
- [11] D. Park, H. Kim, and B. Lee, "Response time in AI-augmented vs. human-only surveillance," *IEEE Trans. Hum.-Mach. Syst.*, vol. 53, no. 1, pp. 88–102, Feb. 2023.
- [12] T. Andreou and G. Pallis, "Scalable architectures for distributed video analytics," *IEEE Commun. Surveys Tuts.*, vol. 25, no. 2, pp. 1123–1156, 2023.
- [13] N. J. Cooke and M. D. Endsley, "Situation awareness in dynamic defence systems," *Hum. Factors*, vol. 63, no. 7, pp. 1121–1145, 2021.
- [14] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection," *Proc. IEEE CVPR*, pp. 580–587, 2014.
- [15] R. Girshick, "Fast R-CNN," *Proc. IEEE ICCV*, pp. 1440–1448, 2015.
- [16] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN," *Adv. NeurIPS*, vol. 28, pp. 91–99, 2015.
- [17] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," *Proc. IEEE CVPR*, pp. 7263–7271, 2017.
- [18] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," *arXiv:1804.02767*, 2018.
- [19] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," *arXiv:2004.10934*, 2020.
- [20] G. Jocher et al., "Ultralytics YOLOv5," GitHub, doi: 10.5281/zenodo.3908559, 2020.
- [21] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable bag-of-freebies for real-time object detectors," *Proc. IEEE CVPR*, pp. 7464–7475, 2023.
- [22] G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics YOLOv8," GitHub, doi: 10.5281/zenodo.7347926, 2023.
- [23] Ultralytics, "YOLOv11: Advancing real-time object detection," Ultralytics Documentation, 2024.
- [24] F. C. Akyon, S. O. Altinuc, and A. Temizel, "Slicing aided hyper inference and fine-tuning for small object detection," *Proc. IEEE ICIP*, pp. 966–970, 2022.