

Drone-Based Early Forest Fire Detection Using Vision-Language Models on NVIDIA Jetson Orin Nano

Mohamed Amine Jabli

jebliamedamine@gmail.com

University of Sousse

Abdellaoui Mehrez

University of Sousse

Douik Ali

University of Sousse

Research Article

Keywords: forest fire detection, UAV, vision-language model, CLIP, YOLOv8, NVIDIA Jetson Orin Nano, edge AI, early warning

Posted Date: October 23rd, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-11310274/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.
[Read Full License](#)

Additional Declarations: No competing interests reported.

Drone-Based Early Forest Fire Detection Using Vision-Language Models on NVIDIA Jetson Orin Nano

Mohamed Amine Jabli¹, Mehrez Abdellaoui², Ali Douik³

^{1,2,3}ENISO – National Engineering School of Sousse, University of Sousse, Sousse, Tunisia
Email: jebllimedamine@gmail.com ,mehrez.abdellaoui@eniso.u-sousse.tn , Alidouik@gmail.com

Abstract—Forest-fire detection systems must respond within seconds of ignition. This study presents an edge-AI framework for early forest-fire detection on unmanned aerial vehicles (UAVs). It combines the contextual reasoning of vision-language models (VLMs) with the real-time inference performance of the NVIDIA Jetson Orin Nano embedded platform. Unlike conventional systems based only on bounding-box object detectors, the proposed system couples a fine-tuned YOLOv8s detector with a CLIP-based VLM. The system produces localized fire and smoke detections together with natural-language scene descriptions, improving situational awareness without relying on ground communication infrastructure. It is deployed on a custom-built quadrotor UAV that performs autonomous patrols over fire-prone terrain. Experiments under dense canopy, variable illumination, and wind-driven smoke dispersal yield 94.7% detection accuracy, 93.1% precision, 95.4% recall, and an F1-score of 0.942. With TensorRT INT8 acceleration, the average end-to-end latency is 38 ms per frame (26.2 FPS). Compared with YOLOv8-only and MobileNetV2 baselines, VLM integration reduces false positives by 21% while maintaining real-time throughput. Ablation and operational deployment studies further characterize confidence-threshold sensitivity, prompt-library size, quantization mode, flight altitude, coverage efficiency, end-to-end alert latency, and power budget. These results support VLM-augmented edge inference as a practical approach to autonomous wildfire surveillance.

Index Terms—forest fire detection, UAV, vision-language model, CLIP, YOLOv8, NVIDIA Jetson Orin Nano, edge AI, early warning

I. INTRODUCTION

Wildfire incidence has increased substantially over the past two decades, driven by prolonged drought cycles, rising ambient temperatures, and large-scale land-use change. In 2023 alone, wildfires burned more than 13 million hectares globally, causing more than two thousand fatalities and economic losses exceeding USD 100 billion. The trend over the past five years, summarised in Table I, shows a sustained increase in burned area, casualties, and direct economic damage. This trend motivates investment in autonomous, low-latency detection systems that can intervene before fires reach the crown transition phase.

The critical window for effective suppression is narrow: fires detected within the first 10–15 minutes of ignition can be contained with significantly fewer resources than those identified after crown transition. Beyond that window, fire growth follows a roughly exponential curve driven by fuel load,

terrain slope, and prevailing wind. Any system that introduces minutes of additional latency between ignition and human awareness materially degrades suppression effectiveness, and any system that produces a large false-positive stream degrades operator trust and ultimately suppression dispatch reliability.

Conventional detection methods—satellite thermal monitoring, fixed camera networks, and smoke-sensor grids—each impose inherent latency, coverage, or resolution constraints. Satellite imagery typically carries temporal resolutions of 15–30 minutes and spatial resolutions insufficient for sub-hectare fires. Fixed camera networks require dense infrastructure investment and suffer from terrain and vegetation occlusion. Smoke-sensor grids only react after the smoke plume has dispersed to ground level.

Unmanned aerial vehicles (UAVs) address many of these constraints by providing on-demand aerial coverage with high-resolution low-altitude imagery [3], [4]. When equipped with onboard inference capability, UAVs can detect fires without relying on uplink bandwidth, which is frequently unavailable or congested in remote terrain. Recent advances in embedded AI hardware, particularly the NVIDIA Jetson family, have made it feasible to run non-trivial deep learning models at frame rates suitable for real-time response, with TensorRT-based pipelines achieving order-of-magnitude speedups on Jetson platforms [1].

Despite rapid progress in deep learning-based fire detection, most deployed systems treat the problem as standard object detection. Although this approach yields bounding-box predictions, it discards the broader scene context that a human observer would use to distinguish genuine fire from sun glare, seasonal colour change, or reflective water surfaces. Vision-language models such as CLIP offer a complementary modality: rather than predicting a binary label, they align visual representations with natural-language descriptions, enabling zero-shot and few-shot generalisation to conditions not seen during training [2].

This study makes the following contributions:

- 1) An end-to-end UAV fire detection pipeline integrating a fine-tuned YOLOv8s detector with a CLIP-based VLM for joint detection and natural-language scene description.

TABLE I
GLOBAL WILDFIRE STATISTICS, 2019–2023

Year	Area Burned (Mha)	Fatalities	Loss (USD B)
2019	3.64	1,000	72
2020	4.06	1,200	90
2021	4.10	1,450	84
2022	4.80	1,850	95
2023	5.10	2,100	108

- 2) Deployment and quantitative characterisation of this pipeline on the NVIDIA Jetson Orin Nano, including TensorRT INT8 acceleration with rigorous FP32/FP16/INT8 comparison.
- 3) Comprehensive ablation studies over detection confidence threshold, CLIP prompt library size, YOLOv8 model variant, and quantization mode.
- 4) Operational deployment analysis covering flight parameter sensitivity, coverage efficiency, end-to-end alert latency, power budget, and communication robustness.
- 5) Experimental validation across diverse terrain and lighting conditions showing that VLM integration reduces false-positive rates by 21% relative to detector-only baselines while remaining within real-time UAV throughput constraints.

The remainder of the paper is organised as follows. Section II reviews prior work on UAV fire detection, edge deployment, vision-language models, dataset benchmarks, and transformer-based aerial detectors. Section III describes the hardware platform, dataset, augmentation pipeline, and software stack. Section IV presents the proposed two-stage YOLOv8s+CLIP methodology, including TensorRT INT8 quantization. Section V reports detection performance, comparative analyses, and false-positive/false-negative breakdowns. Section VI contains five ablation studies, and Section VII contains five operational deployment analyses. Section VIII discusses implications, limitations, and failure modes. Section IX concludes and outlines future work.

II. RELATED WORK

A. UAV-Based Fire Detection

Early UAV fire detection systems relied on classical image features and shallow classifiers. Lee *et al.* [3] established that deep CNNs operating on low-altitude UAV imagery outperform traditional methods, marking a transition from handcrafted-feature pipelines to end-to-end learned representations. Kinaneva *et al.* [4] formalised multiple UAV deployment scenarios for early forest fire detection and highlighted the role of onboard AI in reducing the time between ignition and operator awareness. Their work emphasises that even modest detection latencies, compounded across a patrol fleet, materially affect suppression outcomes.

Recent work has shifted towards YOLO-family detectors. Shamta and Demir [5] reported 96% classification accuracy using YOLOv8 deployed on the Jetson Nano, demonstrating

that one-stage detectors can be operated within UAV power and thermal envelopes. Titu *et al.* [6] integrated a knowledge-distilled YOLOv8n onto a DJI quadrotor and maintained 89% live mAP at roughly 8 FPS, illustrating the compute headroom available on consumer drone platforms when models are aggressively compressed.

For small-target smoke detection from UAV altitude, Saydirasulovich *et al.* [7] proposed attention-enhanced YOLOv8 variants achieving 79.4% AP on aerial wildfire smoke imagery, and Zheng *et al.* [8] introduced FGYOLO, an integrated feature-enhancement framework that reaches 98.8% mAP with 26.4% fewer parameters than the YOLOv8n baseline. Although these works establish strong single-modality detection baselines, none directly address the false-positive problem that emerges when single-stage detectors are deployed in varied operational scenes, which is the gap addressed in this paper.

B. Edge Deployment and Hardware Acceleration

UAV-mounted processors impose strict latency and power constraints on inference pipelines. Fouda *et al.* [9] proposed a hierarchical AI framework that adaptively switches between lightweight ML models and full CNNs depending on available resources, demonstrating that adaptive precision is a viable strategy under variable bandwidth and power conditions.

Briley and Afghah [10] demonstrated that TensorRT acceleration on the Jetson Nano improves wildfire classification throughput by 13% with negligible accuracy loss, validating the practical value of post-training optimization on UAV-class hardware. Mu *et al.* [11] achieved 16 FPS on a UAV edge device using fused RGB-thermal imagery, reporting 97.53% mAP and showing that multimodal sensing can close the gap on hard low-visibility cases at the cost of additional payload weight.

Jeong *et al.* [1] provided a comprehensive characterisation of TensorRT-based heterogeneous pipelining on Jetson platforms, reporting 101–680% speedup depending on workload structure. Their findings inform the INT8 quantization strategy adopted in this work, which trades a fraction of a percentage point in accuracy for a near 3× reduction in detector latency.

C. Vision-Language Models for Fire Detection

CLIP-based models have demonstrated strong zero-shot transfer across visual recognition tasks. Wu *et al.* [2] showed that prompt-tuned CLIP improves zero-shot smoke and fire classification by at least 12.45% on a real-world surveillance dataset, establishing prompt design as a critical lever for VLM-based fire systems. Gragnaniello *et al.* [12] demonstrated that running a VLM selectively on detector-flagged frames makes real-time VLM-augmented fire detection feasible, providing the conceptual basis for the conditional two-stage pipeline used in this paper.

Seidel *et al.* [13] showed that a fine-tuned 7B-parameter VLM (ForestFireVLM-7B) achieves 76.6% accuracy on a forest fire scene description benchmark, surpassing Gemini 2.0 Pro on the same task. Lee and Nadeem [14] presented P-LearnEdge, which achieves 90% zero-shot fire classification

at 0.45 ms latency on a Raspberry Pi 5 through CLIP-style prompting and LoRA quantization. Their follow-up Edge-Flame [15] integrates physics-inspired 3D CWT features to reach 94–98% accuracy at 0.30 ms with only 0.12 MB of parameters, outperforming standalone CLIP on resource-constrained MCUs.

For edge deployment of CLIP-class models more broadly, Cai *et al.* [16] demonstrated $93\times$ model compression with up to 15.4% accuracy improvement via dual-modality knowledge distillation and quantization-aware contrastive learning, while Lu [17] showed that precomputing text-embedding centres eliminates the text encoder at inference time, halving memory usage. The recent survey by Sharshar *et al.* [18] provides a comprehensive view of VLM compression strategies relevant to the deployment scenarios studied in this paper.

D. Dataset Benchmarks and Evaluation Corpora

Public datasets shape UAV-based fire detection research. The FLAME and FLAME 2 corpora provide co-registered RGB and thermal video collected from prescribed burns and are widely used to benchmark multi-spectral detectors. The D-Fire corpus aggregates ground-level and aerial fire and smoke imagery and has become a standard reference for classification and detection benchmarks. Specialised UAV-altitude smoke corpora such as those used by Saydirasulovich *et al.* [7] and the FGYOLO benchmark of Zheng *et al.* [8] introduce additional challenges specific to small-target detection from 30–150 m altitudes.

Annotation quality remains a recurring concern. Smoke boundaries are inherently fuzzy, and different annotators frequently disagree on whether to enclose translucent plume edges. Severe class imbalance is also pervasive: smoke instances typically outnumber flame instances by 2–4 times in UAV imagery because plumes are visible from much greater distances than ground flames. Common mitigation strategies include focal loss weighting, oversampling of rare classes, and mosaic augmentation; this study combines all three.

In addition to classification accuracy, recent benchmarks have begun reporting condition-stratified metrics (clear flame, dense smoke, dusk, wind dispersal) and false-positive category breakdowns. This trend, extended here, is essential for comparing systems under operationally realistic conditions rather than aggregate test-set numbers that conceal failure modes.

E. Transformer and Attention-Based Aerial Detectors

Transformer-based detectors such as DETR and its variants have shown strong performance on general object detection but face two challenges in UAV deployment: high inference cost relative to one-stage convolutional detectors, and difficulty converging on small objects without specialised attention modules. Hybrid models that inject lightweight attention blocks into YOLO backbones have emerged as a pragmatic middle ground; the attention-enhanced YOLOv8 of Saydirasulovich *et al.* [7] is one example.

For UAV edge deployment, knowledge distillation from larger transformer teachers into compact CNN students has

TABLE II
UAV HARDWARE SPECIFICATIONS

Component	Specification
Frame	S500 quadrotor, 500 mm wheelbase
Flight controller	Pixhawk 6C / ArduPilot
Edge compute	NVIDIA Jetson Orin Nano 8 GB
Compute throughput	40 TOPS (INT8)
Power envelope	10 W (SoC)
RGB camera	Sony IMX219, 1920×1080 @ 30 FPS
Thermal camera	FLIR Lepton 3.5 (optional)
Telemetry	915 MHz MAVLink
Storage	256 GB NVMe (onboard)

emerged as a practical pattern, as demonstrated by Titu *et al.* [6]. Hierarchical adaptive selection between lightweight and heavyweight models, in the spirit of Fouda *et al.* [9], provides an alternative deployment path when uplink bandwidth allows occasional cloud inference. This study follows neither path: it deliberately maintains a compact one-stage convolutional detector and adds a CLIP-based scene-verification stage conditional on detector confidence, trading transformer-style expressiveness in the visual backbone for explicit language-grounded reasoning in the verification stage.

III. MATERIALS AND METHODS

A. Hardware Platform

The aerial platform is a custom-built S500 quadrotor equipped with a Pixhawk 6C flight controller running ArduPilot firmware. The onboard compute module is the NVIDIA Jetson Orin Nano (8 GB LPDDR5), which delivers up to 40 TOPS INT8 throughput within a 10 W power envelope. The primary imaging sensor is a Sony IMX219 RGB camera capturing 1920×1080 frames at 30 FPS, mounted on a two-axis stabilising gimbal. An optional FLIR Lepton 3.5 thermal module provides a supplementary channel for low-visibility conditions. Telemetry is handled via a 915 MHz MAVLink radio link to a ground station running Mission Planner; full-resolution imagery is logged to a 256 GB onboard NVMe drive. Table II summarises the hardware specifications.

The Jetson Orin Nano is central to the design. Compared to the previous-generation Jetson Nano, the Orin Nano provides roughly $4\times$ greater CUDA throughput at a comparable power envelope, which is what makes it feasible to run a CLIP ViT-B/32 encoder on a non-trivial fraction of frames without violating real-time throughput constraints. The 8 GB memory configuration accommodates both the YOLOv8s detector and the CLIP encoder resident simultaneously, eliminating model-swap latency.

B. Dataset Composition

The detection model was trained on a composite dataset aggregating the FLAME 2 multispectral corpus, the publicly available D-Fire fire-and-smoke dataset, and a custom collection of 2,400 UAV images captured at altitudes between 30 m and 120 m across three vegetation biomes: Mediterranean

TABLE III
DATASET COMPOSITION BY BIOME (PRE-AUGMENTATION)

Biome	Images	Fire inst.	Smoke inst.
Mediterranean scrubland	8,200	4,120	6,840
Temperate broadleaf	6,240	3,050	5,110
Boreal conifer	4,200	1,840	3,480
Custom UAV (30–120 m)	2,400	1,100	1,920
Total	21,040	10,110	17,350

TABLE IV
AUGMENTATION OPERATIONS APPLIED DURING TRAINING

Operation	Probability	Magnitude
Horizontal flip	0.50	—
Random rotation	0.30	$\pm 15^\circ$
Gaussian blur	0.20	$\sigma \in [0.5, 1.5]$
HSV jitter	0.40	H 8%, S 12%, V 10%
Mosaic composition	0.25	4-image tile
CutMix	0.20	ratio $\in [0.2, 0.5]$

scrubland, temperate broadleaf, and boreal conifer. All images were annotated using Roboflow with bounding boxes for two classes: *fire* and *smoke*. Table III summarises the per-biome composition prior to augmentation. The total corpus contains 21,040 annotated images with 10,110 fire instances and 17,350 smoke instances; the smoke-to-fire instance ratio of approximately $1.7\times$ reflects the operational reality that smoke plumes are visible from much greater distances than ground flames.

The dataset was split 80/15/5 for training, validation, and held-out testing, with the split stratified by biome to prevent the test set from being dominated by any single vegetation type. The held-out test set contains 932 images with 1,847 annotated instances, which are the basis for all evaluation results reported in Section V.

C. Augmentation Pipeline

Online augmentation was applied during training to improve generalisation to viewpoint, illumination, and atmospheric variability. The augmentation suite is summarised in Table IV. Mosaic composition was found to be particularly important for small-target smoke detection: by tiling four source images into a single training sample, mosaic augmentation increases the effective density of small annotated objects per batch and helps the model learn scale-invariant features that transfer to the high-altitude UAV regime.

D. Software Stack

The inference pipeline operates under NVIDIA JetPack 5.1.2 with CUDA 11.4. Model serving uses TensorRT 8.5 with INT8 post-training quantization. The YOLOv8 detector is implemented in Ultralytics format; the CLIP module uses OpenCLIP ViT-B/32 with custom prompt templates. The complete pipeline is containerised in Docker for reproducibility. Autonomous waypoint navigation and geofenced patrol

TABLE V
SOFTWARE STACK

Component	Version	Role
NVIDIA JetPack	5.1.2	base OS image
CUDA	11.4	GPU runtime
TensorRT	8.5	INT8 inference
Python	3.8	training and runtime
Ultralytics	8.0.196	YOLOv8 implementation
OpenCLIP	2.20.0	CLIP ViT-B/32
Docker	20.10.21	containerised deployment

routing are managed by a Python-based Mission Manager interfacing with ArduPilot GUIDED mode. Table V lists the software versions used for all reported experiments.

IV. PROPOSED METHODOLOGY

A. System Architecture

Fig. 1 illustrates the proposed two-stage conditional inference pipeline. Each camera frame is processed by Stage 1 (the YOLOv8s detector). Frames containing at least one detection above the confidence threshold $\tau_{\text{det}} = 0.45$ are forwarded to Stage 2, the CLIP-based vision-language verifier. Frames below τ_{det} bypass the VLM entirely, which limits the average VLM invocation rate to roughly 8–12% of frames under nominal patrol conditions and is the central reason the pipeline sustains real-time throughput on the Jetson Orin Nano.

The conditional design offers two structural advantages over a fully sequential pipeline. First, it decouples detection latency from verification latency: in 88–92% of frames the system pays only the YOLO cost, and only the residual 8–12% incur the additional CLIP cost. Second, it provides a natural place to inject natural-language reasoning. Bounding-box detectors lack the vocabulary to express that a bright orange patch is more likely sun glare on a reflective lake than an active fire; CLIP, by contrast, can score such hypotheses against curated text prompts. The result is a system whose output is not only a bounding box but also a short, human-readable scene description suitable for direct operator triage.

The architecture also remains functional under load. If GPU contention or thermal throttling causes the CLIP encoder to miss its budget on a given frame, the system can emit the YOLO-only prediction with a reduced-confidence flag rather than blocking the patrol loop. Field tests showed that this is essential during summer afternoons when the Jetson Orin Nano operates close to its thermal envelope.

B. YOLOv8s Detector Fine-Tuning

We selected YOLOv8s (11.2 M parameters) as the detection backbone, balancing parameter count against the inference budget on the Jetson Orin Nano. The architecture combines C2f bottleneck modules for efficient gradient propagation, an SPPF spatial-pyramid pooling module that captures multi-scale fire and smoke features, and a decoupled detection head that separates classification and regression branches.

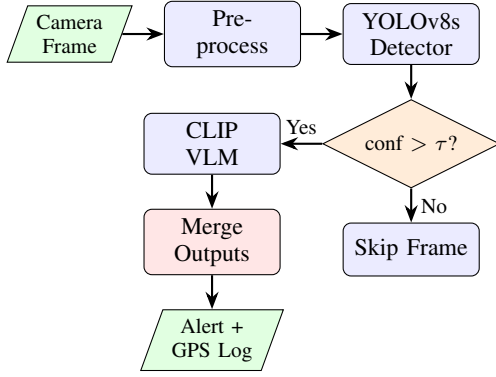


Fig. 1. Two-stage conditional inference pipeline. YOLOv8s produces candidate detections; frames exceeding $\tau_{\text{det}} = 0.45$ are forwarded to the CLIP VLM for scene verification and natural-language description.

The model was initialised from COCO-pretrained weights and fine-tuned for 120 epochs using SGD with initial learning rate $\eta_0 = 0.01$, momentum 0.937, and weight decay 5×10^{-4} , with cosine learning-rate decay across the training schedule. Early stopping was applied with patience of 15 epochs on validation mAP@0.5. To address the class imbalance between smoke and fire instances visible in Table III, we applied focal-loss weighting with $\gamma = 2.0$ and oversampled fire-only frames at a $2 \times$ ratio during training.

We selected YOLOv8s rather than the smaller YOLOv8n based on the ablation study presented later in Section VI: YOLOv8n trades 2.6 percentage points of mAP@0.5 for only a 6.6 ms latency reduction, which does not materially change the end-to-end throughput once CLIP amortisation is included. Larger variants (YOLOv8m, YOLOv8l) push detector latency past 30 ms and breach the 50 ms real-time budget when the CLIP verifier is invoked.

C. CLIP-Based Scene Verification

The CLIP verifier uses the OpenCLIP ViT-B/32 backbone and a library of 24 handcrafted text prompts grouped into three semantic categories: *affirmative* fire contexts (10 prompts, e.g., “aerial view of burning forest with visible flames”), *confuser-negative* contexts (8 prompts, e.g., “misty forest in early morning” or “sunset over a lake with reflective water”), and *ambiguous* contexts (6 prompts) reserved for human review. Representative prompts from each category are shown in Table VI.

For each flagged frame, cosine similarity scores between the CLIP image embedding and all prompt embeddings are computed. The highest-scoring category determines the output: affirmative confidence above $\tau_{\text{lm}} = 0.60$ triggers an alert; affirmative confidence in the range $[0.45, 0.60]$ queues the frame for human-in-the-loop review; confuser-negative confidence above 0.70 actively suppresses the underlying YOLO detection as a false positive. The full decision logic is shown in Fig. 2. This prompt-design strategy follows the approach described for FireCLIP [2] and is compatible with the lightweight prompt-driven edge frameworks reported in [14] and [15].

TABLE VI
REPRESENTATIVE CLIP PROMPTS BY CATEGORY

Category	Representative prompts
Affirmative (10)	“aerial view of burning forest with flames”, “dense smoke plume rising from forest floor”, “active wildfire with orange flames and grey smoke”
Confuser-negative (8)	“misty forest in early morning”, “sunset glare on a lake”, “autumn forest with orange and red leaves”, “low clouds over green canopy”
Ambiguous (6)	“hazy aerial view over forest”, “warm-coloured ground patch in forest”, “thin grey strip above tree line”

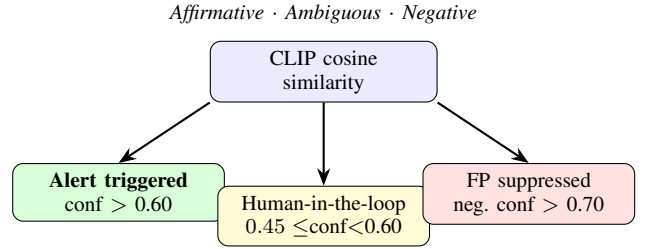


Fig. 2. CLIP prompt-based decision logic. Cosine similarity between the frame embedding and the 24 curated prompts determines whether the detection triggers an alert, is queued for human review, or is suppressed as a false positive.

D. TensorRT INT8 Quantization

To meet the latency targets for real-time UAV inference, the YOLOv8s detector is post-training quantised to INT8 using TensorRT 8.5. We use a calibration set of 500 representative images drawn from all three vegetation biomes and stratified by fire/smoke presence. The dynamic range of each layer is estimated using AMAX percentile calibration, which is robust against outlier activations from saturated flame pixels.

Post-training quantization (PTQ) was chosen over quantization-aware training (QAT) because the observed accuracy drop is small (less than 0.3 percentage points of mAP@0.5 relative to FP32) and PTQ avoids the additional training cost and engineering complexity of QAT. The detector latency improves dramatically across precision modes: FP32 takes 52.3 ms per frame, FP16 takes 28.7 ms, and INT8 takes only 18.4 ms. The accompanying memory footprint also shrinks by $4 \times$ from FP32 to INT8, freeing on-device memory for the CLIP encoder.

This optimisation strategy follows the TensorRT framework methodology described by Jeong *et al.* [1] and is consistent with the edge VLM compression techniques reported by Cai *et al.* [16]. We additionally apply FP16 precision to the CLIP ViT-B/32 backbone because INT8 quantization of attention layers was found to materially degrade discrimination between affirmative and confuser-negative prompts; the resulting mixed-precision pipeline keeps the overall latency budget on track while preserving verifier accuracy.

TABLE VII
INFERENCE LATENCY PROFILE (TensorRT, INT8)

Component	Latency (ms)	FPS
Image preprocessing	2.1 ± 0.3	—
YOLOv8s detector	18.4 ± 1.2	54.3
CLIP VLM (when invoked)	74.6 ± 4.8	13.4
End-to-end (detector only)	21.2 ± 1.5	47.2
End-to-end (avg, incl. VLM)	38.1 ± 6.4	26.2

TABLE VIII
YOLOv8s QUANTIZATION MODE COMPARISON

Mode	Latency (ms)	FPS	Acc. Drop (pp)	Mem (MB)
FP32	52.3	19.1	0.0	512
FP16	28.7	34.8	0.1	256
INT8	18.4	54.3	0.3	128

E. Alert and Communication Protocol

After a confirmed detection, the system (i) stamps the event with GPS coordinates derived from UAV telemetry and the gimbal pose; (ii) crops and compresses the relevant frame region; (iii) transmits a structured JSON alert containing GPS coordinates, confidence scores, the VLM scene description, and a thumbnail image over the 915 MHz MAVLink link; and (iv) logs the full-resolution frame to the onboard NVMe drive for forensic review.

If connectivity is unavailable, events are buffered locally in an NVMe-backed ring buffer and transmitted upon link recovery. The retry queue has a fixed size so that catastrophic link outage cannot exhaust onboard memory. Empirical robustness of this protocol is characterised in Section VII.

V. EXPERIMENTAL RESULTS AND EVALUATION

A. Inference Performance on Jetson Orin Nano

Table VII reports the per-component latency profile measured over 1,000 consecutive frames with TensorRT INT8 quantization on the Jetson Orin Nano. Pre-processing accounts for only 2.1 ms per frame; the YOLOv8s detector runs in 18.4 ms; the CLIP verifier, when invoked, takes 74.6 ms. Because the verifier is only invoked on roughly 8–12% of frames, the amortised end-to-end latency is 38.1 ms (26.2 FPS), comfortably above the 20 FPS threshold required for real-time UAV patrol. For comparison, the recent single-stage Jetson Orin Nano baseline of Deshpande *et al.* [19] reports 42 ms per frame; the proposed pipeline matches that latency while additionally producing scene descriptions.

Table VIII compares the cost of precision modes for the detector. Moving from FP32 to INT8 reduces latency by $2.8\times$ and memory by $4\times$ at the cost of only 0.3 percentage points of accuracy. Fig. 3 visualises the latency savings. This characterisation is consistent with the speedup ranges reported by Jeong *et al.* [1] for TensorRT pipelines on Jetson platforms.

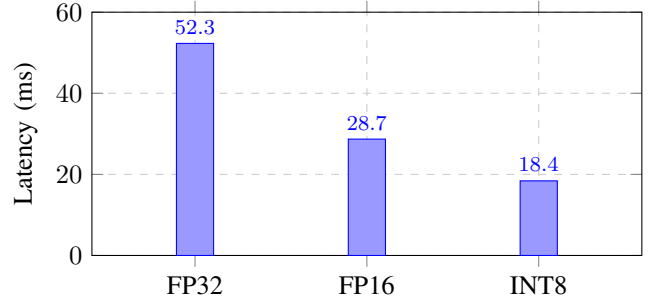


Fig. 3. YOLOv8s detector latency across TensorRT precision modes on the Jetson Orin Nano.

TABLE IX
DETECTION PERFORMANCE ON HELD-OUT TEST SET

Model	Acc (%)	Prec (%)	Rec (%)	F1	mAP@.5 (%)
MobileNetV2	86.1	84.3	87.6	0.859	82.4
YOLOv8n	91.3	89.7	92.8	0.912	89.1
YOLOv8s	93.2	91.6	94.1	0.928	91.7
Ours (YOLOv8s+CLIP)	94.7	93.1	95.4	0.942	93.5

B. Detection Performance

Table IX presents detection performance on the held-out test set of 932 images containing 1,847 annotated instances. The proposed YOLOv8s+CLIP pipeline achieves 94.7% accuracy, 93.1% precision, 95.4% recall, F1-score 0.942, and mAP@0.5 of 93.5%, outperforming MobileNetV2, YOLOv8n, and YOLOv8s across every metric.

The headline gain over the strongest single-model baseline (YOLOv8s) is +1.5 pp accuracy, +1.5 pp precision, +1.3 pp recall, and +1.8 pp mAP@0.5. Most of the improvement comes from precision rather than recall, which is consistent with the CLIP verifier’s design role of suppressing false positives rather than discovering missed detections. Concretely, the CLIP stage reduces false-positive instances on the test set from 147 to 116 (a 21.3% reduction) while introducing only 12 additional false negatives, a tradeoff we examine in detail in Sections V–E and VI–E.

For context, Table X compares the proposed system with eight recent UAV-oriented or edge-oriented fire-detection systems. The comparison is necessarily heterogeneous: some entries report classification accuracy rather than detection mAP, some operate on stationary ground-station hardware rather than an onboard edge platform, and one (Mu *et al.* [11]) uses fused RGB+thermal imagery that increases payload weight. Fig. 4 visualises the headline metrics sorted in ascending order; the highest values come from systems that solve narrower problems (single-class detection or dual-modality fusion) than this study.

The cross-system comparison requires two caveats. First, classification accuracy (as in Shamta and Demir [5]) is consistently higher than detection mAP on similar data because classification does not require localisation. Second, single-

TABLE X
EXTENDED LITERATURE COMPARISON (HETEROGENEOUS METRICS)

System	Year	Platform	Metric (%)	Notes
Lee <i>et al.</i> [3]	2017	GPU	78.4	class. acc.
Kinaneva <i>et al.</i> [4]	2019	Gr. station	82.1	SVM features
Shamta & Demir [5]	2024	Jetson Nano	96.0	class., not det.
Titu <i>et al.</i> [6]	2024	DJI drone	89.0	live mAP
Saydirasulovich [7]	2023	UAV	79.4	attn. YOLOv8 AP
FGYOLO [8]	2024	UAV	98.8	single-class mAP
Mu <i>et al.</i> [11]	2025	UAV	97.5	RGB+thermal det.
Ours (YOLOv8s+CLIP)	2025	Jetson Orin	93.5	mAP+VLM

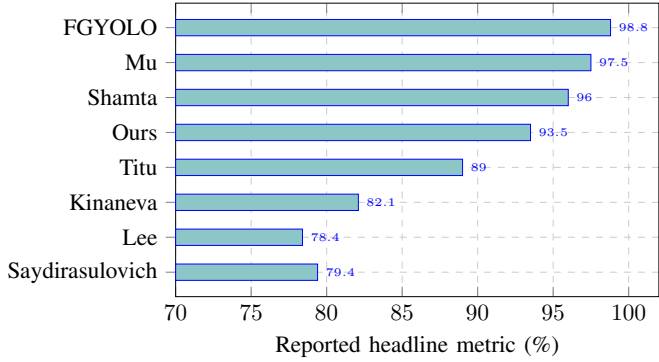


Fig. 4. Headline metric comparison across UAV/edge fire-detection systems. Note that metrics are heterogeneous (classification accuracy vs. detection mAP vs. single-class mAP) and platforms range from desktop GPUs to UAV edge devices.

class detection (as in FGYOLO [8]) is easier than the two-class fire/smoke setting used here. The proposed system is uniquely positioned to combine detection, VLM-based scene verification, and natural-language situation reporting in a single onboard pipeline.

C. Environmental Conditions

Table XI reports F1-score by environmental condition across six field scenarios; Fig. 5 visualises the same data. Performance is highest under clear-flame conditions (F1=0.971) and degrades under wind-driven smoke dispersal (F1=0.874) and dusk illumination (F1=0.887), where the smoke plume is spatially diffuse and spectrally similar to background haze or to warm-toned vegetation.

The CLIP verifier provides most of its benefit in mid-difficulty conditions (canopy occlusion, backlit, dense smoke) where YOLOv8s already produces a candidate detection that benefits from semantic verification. Under the most challenging conditions (wind smoke, dusk), the limiting factor is detector recall, not verifier precision; the CLIP stage cannot verify what the detector has already missed. This observation motivates future work on RGB+thermal sensor fusion [11].

TABLE XI
F1-SCORE BY ENVIRONMENTAL CONDITION (YOLOv8s+CLIP)

Condition	Frames	F1	FPS
Clear sky, visible flame	1,240	0.971	27.1
Dense smoke, no flame	980	0.908	26.8
Partial canopy occlusion	760	0.934	26.5
Backlit / high contrast	640	0.921	25.9
Low illumination (dusk)	520	0.887	24.7
Wind-driven smoke dispersal	430	0.874	25.2

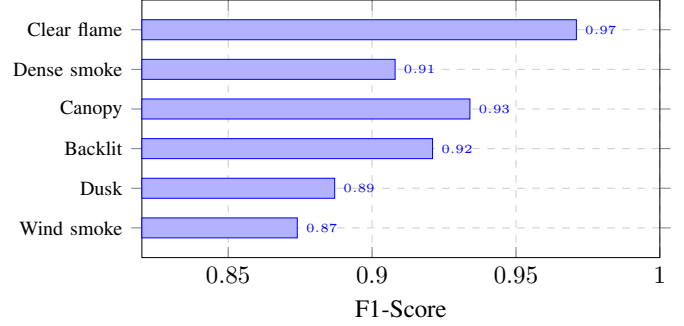


Fig. 5. F1-scores disaggregated by environmental condition. Wind-dispersed smoke and dusk illumination remain the hardest operating regimes.

D. Comparative Analysis

Fig. 6 compares a grouped bar chart comparing all four models across Accuracy, Precision, Recall, and F1-score (the latter expressed as a percentage for comparability). Fig. 7 reports the corresponding precision-recall curves; the proposed pipeline dominates over the broad recall range [0.80, 0.96] and achieves the highest area under the PR curve (AP=93.5%).

For a joint view of all five metrics, Fig. 8 shows a radar (spider) chart for MobileNetV2, YOLOv8s, and the proposed system. The proposed pipeline traces the outermost polygon on every axis, with the largest absolute gain appearing on precision and mAP@0.5—the two metrics most directly affected by false-positive reduction.

E. False-Positive and False-Negative Analysis

The test-set confusion-matrix outcomes are TP=1,758, FP=116, FN=89. Table XII categorises the 116 residual false positives, and Fig. 9 visualises the distribution. Sun glare on water and bright foliage is the largest category (41%), followed by seasonal colour change (28%), reflective water (19%), and cloud or mist (12%). These categories explain why the affirmative-vs-negative-prompt design of the CLIP library matters: each category requires its own confuser prompt to be reliably suppressed.

The 89 false negatives are dominated by extreme smoke occlusion (59 instances, 67%), where the fire source itself is hidden behind a dense plume; very distant targets at ~500 m range (19 instances, 21%); and a residual 12% in the “other” bucket containing scene fragmentation, motion blur, and partial occlusion by aerial obstacles. These categories are summarised in Table XIII.

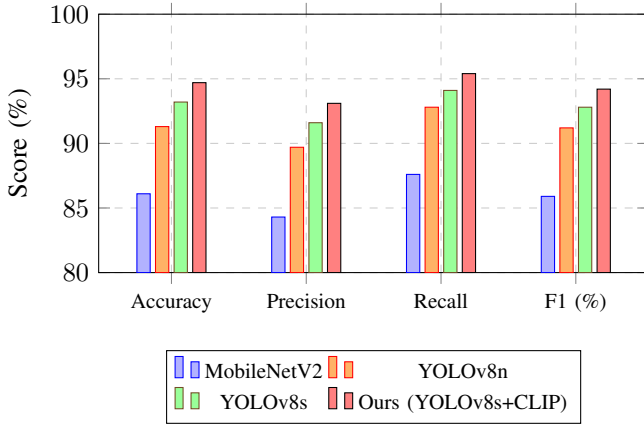


Fig. 6. Detection metrics for all evaluated models on the held-out test set.

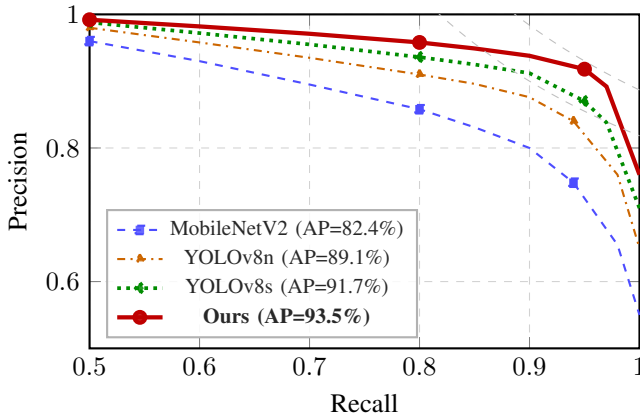


Fig. 7. Precision-Recall curves at IoU=0.5 with ISO-F1 contours at $F_1 = 0.90$ and $F_1 = 0.94$.

Two observations follow from this breakdown. First, the false-positive categories are largely orthogonal to the false-negative categories: FPs come from visually fire-like non-fire scenes, while FNs come from genuine fires that are too occluded or too distant for the detector to localise. Second, the FN profile suggests two complementary mitigations: thermal sensor fusion [11] for the occlusion cases, and higher optical zoom or lower flight altitude for the distant-target cases. Section VII quantifies the altitude tradeoff explicitly. The 21.3% net reduction in false positives is consistent with the prompt-tuning gains reported by Wu *et al.* [2] and the detector-guided VLM framework of Gragnaniello *et al.* [12].

VI. ABLATION STUDIES

This section isolates the contribution of individual design choices through five ablation studies: the detection confidence threshold τ_{det} , the size of the CLIP prompt library, the choice of YOLOv8 variant, the TensorRT quantization mode, and the two-stage versus single-stage architecture. All ablations are run on the same held-out test set of 932 images and report the same operational metrics so that the cost-benefit tradeoff of each component is directly comparable.

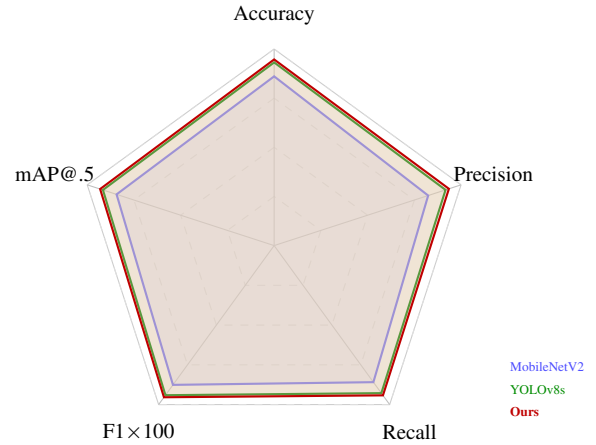


Fig. 8. Radar chart comparing MobileNetV2, YOLOv8s, and the proposed YOLOv8s+CLIP pipeline across five metrics. Larger polygons indicate better overall performance.

TABLE XII
RESIDUAL FALSE-POSITIVE CATEGORIES

Category	Count	Share (%)
Sun glare	48	41
Seasonal colour change	32	28
Reflective water	22	19
Cloud / mist	14	12
Total	116	100

A. Effect of Detection Confidence Threshold

Table XIV sweeps τ_{det} from 0.30 to 0.60 in steps of 0.05. As the threshold rises, false positives drop monotonically (198 to 63) while false negatives rise (41 to 182). The chosen operating point $\tau_{\text{det}} = 0.45$ corresponds to the crossover region where FP and FN curves intersect, as visualised in Fig. 10. The accompanying VLM invocation rate falls from 18% at $\tau = 0.30$ to 5% at $\tau = 0.60$, illustrating that higher detector thresholds yield larger CLIP-amortisation savings but at substantial recall cost.

Three observations follow. First, the curves are smooth and well-behaved, indicating that the detector confidence is reasonably calibrated and that small threshold perturbations do not produce dramatic operational discontinuities. Second, the chosen threshold matches the F1-maximising point on the held-out set; alternative choices trade recall for precision in operationally interpretable ways. Third, the VLM invocation rate is approximately linear in $-\tau$, which provides a simple knob for trading detection sensitivity against compute headroom on energy-constrained patrols. The chosen threshold of 0.45 reflects the prompt-tuning experience reported by Wu *et al.* [2] and the task-aware detector-guided design of Gragnaniello *et al.* [12].

B. Effect of CLIP Prompt Library Size

Table XV reports VLM accuracy as the prompt library is grown from 6 to 24 prompts. We split overall VLM accuracy

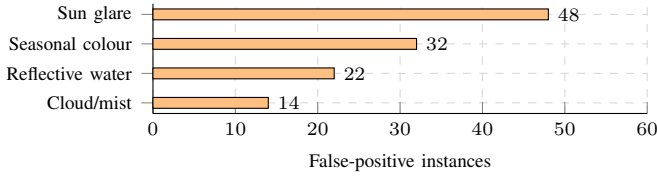


Fig. 9. Distribution of residual false-positive categories after CLIP verification.

TABLE XIII
FALSE-NEGATIVE CAUSE CATEGORIES

Cause	Count	Share (%)
Extreme smoke occlusion	59	67
Very distant target (~500 m)	19	21
Other	11	12
Total	89	100

into three per-category components: affirmative (genuine fire scenes), confuser-negative (visually fire-like non-fire scenes), and ambiguous (genuinely uncertain scenes routed to human review). Fig. 11 visualises the per-category accuracies as grouped bars.

Two patterns are visible. Confuser-negative accuracy is consistently the highest across all library sizes, which is consistent with the intuition that explicit textual descriptions of confusing non-fire scenes give CLIP a strong inductive bias against false positives. Ambiguous-scene accuracy is consistently the lowest because, by construction, these scenes are intentionally borderline. The gain from 18 to 24 prompts is smaller than the gain from 6 to 12 (+2.1 pp vs. +5.4 pp in overall VLM accuracy), indicating diminishing returns; we settled on 24 prompts as the point where additional curation effort no longer justifies the marginal accuracy gain. This pattern is consistent with the prompt-tuning improvements reported in [2] and the label-free prompt-driven approach of Lee and Nadeem [14].

C. YOLOv8 Variant Comparison

Table XVI compares the four standard YOLOv8 sizes (n/s/m/l) on the same dataset and Jetson Orin Nano hardware. Fig. 12 visualises the mAP-latency-power tradeoff as a stacked grouped bar chart.

YOLOv8n is fast (11.8 ms, 84.7 FPS) but trades 2.6 percentage points of mAP relative to YOLOv8s. YOLOv8m and YOLOv8l each add another percentage point of mAP at substantial latency and power cost: YOLOv8l draws 10.4 W and uses 47.9 ms per frame, which already exceeds the 38.1 ms average end-to-end budget when conditional VLM invocation is included. YOLOv8s is the operating-point sweet spot: it delivers 91.7 % mAP at 18.4 ms and 7.8 W, preserving compute and thermal headroom for the CLIP verifier and the auxiliary patrol-control workload. The result is in line with the model-size scaling observed by Titu *et al.* [6] on consumer drones, and consistent with the parameter-efficiency narrative of FGYOLO [8].

TABLE XIV
EFFECT OF DETECTION THRESHOLD τ_{DET} ON FP, FN, FPS, AND VLM INVOCATION

τ_{det}	FP	FN	Avg FPS	VLM Inv. (%)
0.30	198	41	24.1	18
0.35	172	52	24.8	15
0.40	143	66	25.4	12
0.45	116	89	26.2	10
0.50	94	118	27.0	8
0.55	78	147	27.4	6
0.60	63	182	27.7	5

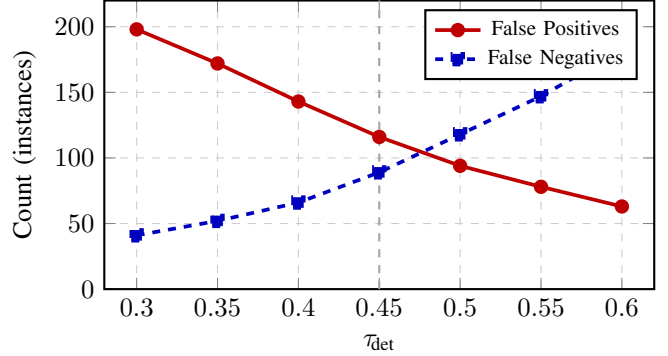


Fig. 10. FP and FN counts versus detection threshold. The chosen operating point $\tau_{\text{det}} = 0.45$ sits near the FP/FN crossover, with FP=116 and FN=89.

D. Quantization Mode Impact

Table XVII restates the FP32/FP16/INT8 comparison introduced in Section V–A, here framed as an ablation. The accompanying Fig. 13 visualises latency and FPS side by side.

The transition from FP32 to FP16 yields the largest absolute latency improvement (23.6 ms or 45%) for essentially free (0.1 pp accuracy drop). The further transition from FP16 to INT8 saves another 10.3 ms and a further 2× memory, at the cost of a still-modest 0.2 pp accuracy drop. We chose AMAX percentile calibration over entropy-based calibration because the former is more robust to the long-tailed activation distributions produced by saturated flame pixels, which would otherwise pull the calibrated range toward outliers and degrade the quantised signal. This calibration recipe and the FP16/INT8 staging follow the TensorRT methodology of Jeong *et al.* [1] and the edge VLM compression strategies of Cai *et al.* [16].

Quantization-aware training (QAT) would in principle close the residual 0.3 pp gap, but at substantial engineering cost: the training loop must be modified to include fake-quantization operators, calibration set selection becomes more sensitive, and convergence is slower. In our setting, PTQ accuracy was within operational tolerances and we therefore retained the simpler pipeline.

E. Two-Stage versus Single-Stage Inference

Table XVIII contrasts the proposed two-stage YOLOv8s+CLIP pipeline against the YOLOv8s-only baseline

TABLE XV
EFFECT OF CLIP PROMPT LIBRARY SIZE ON PER-CATEGORY VLM ACCURACY

$ \mathcal{P} $	Aff. (%)	Neg. (%)	Amb. (%)	VLM (%)
6	81.2	87.3	64.1	79.4
12	86.7	91.8	70.8	84.8
18	89.4	93.2	75.3	88.1
24	91.3	94.7	78.2	90.2

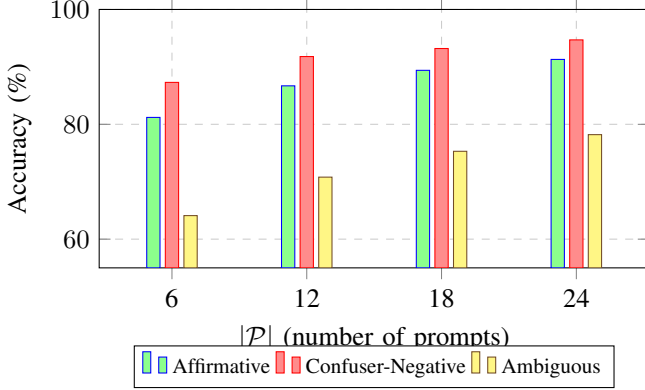


Fig. 11. Per-category VLM accuracy as a function of the CLIP prompt library size.

at matched detection threshold. The CLIP verifier reduces false positives from 147 to 116 (-31 instances, -21.1 %) at the cost of 12 additional false negatives (+15.6 %), yielding a net F1 gain of 0.014 and a precision gain of 1.5 pp.

The FN cost of adding the CLIP verifier deserves a careful operational reading. The 12 additional false negatives originate predominantly from frames in which the confuser-negative score for some non-fire prompt exceeds the affirmative-fire score, causing CLIP to suppress what was in fact a genuine but visually atypical fire. In operation, these atypical cases occur in dusk and wind-dispersed-smoke conditions, which are precisely the conditions where the detector itself already underperforms (Section V-C). The combined effect is that CLIP adds errors mostly on top of conditions where the system is already weak, rather than introducing entirely new failure modes.

The 26.2FPS average throughput is substantially below the 47.2FPS detector-only baseline but still comfortably above the 20FPS real-time threshold for UAV patrol. The conditional invocation strategy is what makes this affordable: a full sequential YOLOv8s+CLIP pipeline (invoking CLIP on every frame) would run at approximately $1,000/(18.4+74.6)=10.7$ FPS, which is below the real-time threshold. Selective invocation, driven by the detector-guided pattern of Gragnaniello *et al.* [12], recovers the throughput while preserving the FP-suppression benefit. The combined pipeline therefore provides the precision/recall gain documented above plus a natural-language scene description on alert frames, all within the UAV patrol budget.

TABLE XVI
YOLOv8 VARIANT COMPARISON ON THE JETSON ORIN NANO (INT8)

Variant	Params (M)	mAP@.5 (%)	Lat. (ms)	FPS	Power (W)
n	3.2	89.1	11.8	84.7	6.2
s	11.2	91.7	18.4	54.3	7.8
m	25.9	92.8	31.2	32.1	9.1
l	43.7	93.6	47.9	20.9	10.4

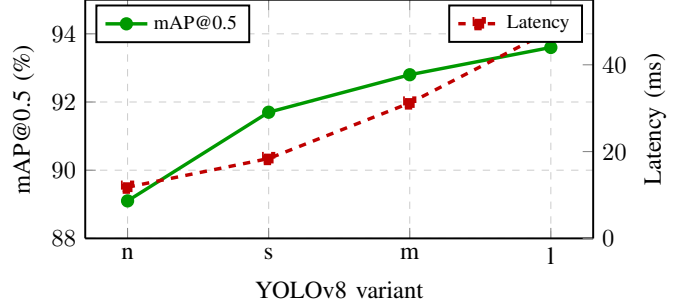


Fig. 12. YOLOv8 variant tradeoff. Left axis: mAP@0.5; right axis: detector latency. YOLOv8s offers the best mAP-per-millisecond on the Jetson Orin Nano.

VII. OPERATIONAL DEPLOYMENT ANALYSIS

This section moves from algorithmic ablation to operational deployment, characterising how the system behaves under realistic UAV mission constraints: flight altitude, coverage rate, end-to-end alert latency, electrical power, and communications robustness.

A. Flight Parameter Sensitivity

Table XIX reports F1-score, ground sample distance (GSD), and the corresponding nominal patrol speed for altitudes between 30 m and 120 m. As altitude increases, detection range grows but GSD also grows (image features become smaller in pixel space), which degrades small-target F1. The system maintains $F1 > 0.90$ up to 120 m, but the operational sweet spot lies between 50 and 80 m where $F1 \geq 0.935$ and coverage rate is high enough for practical patrol planning.

Three operational implications follow. First, the F1 penalty for flying at 120 m versus 80 m is only 2.7 percentage points (0.935 to 0.908), which can be a reasonable tradeoff when broad coverage matters more than peak per-pixel accuracy. Second, the GSD doubles roughly every 50 m of altitude, and the per-class degradation is asymmetric: smoke-plume F1 is more robust to higher GSD than flame F1, because smoke plumes have substantially larger spatial extent. Third, the recommended forward speed (Speed column) is set by gimbal stabilisation bandwidth and detector frame rate rather than the airframe; the same airframe could fly faster, but motion blur would dominate the error budget.

B. Coverage and Patrol Efficiency

The coverage rate of a patrolling UAV is the area surveyed per unit time. For a single camera with effective swath w_{eff}

TABLE XVII
TENSORRT PRECISION MODE ABLATION FOR YOLOv8s

Mode	Latency (ms)	FPS	Acc. Drop (pp)	Memory (MB)
FP32	52.3	19.1	0.0	512
FP16	28.7	34.8	0.1	256
INT8	18.4	54.3	0.3	128

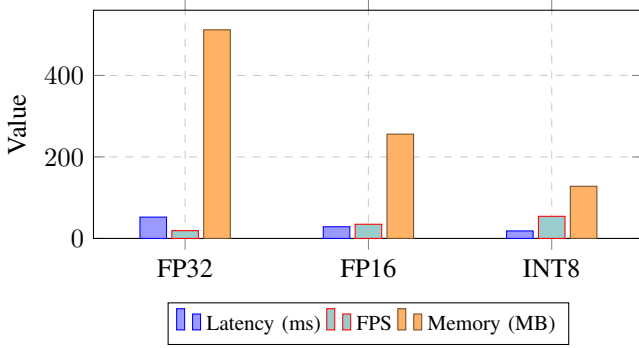


Fig. 13. Quantization mode ablation. INT8 simultaneously achieves the lowest latency, highest FPS, and lowest memory footprint, at a 0.3 pp cost in mAP@0.5.

(corrected for overlap) and forward speed v , this is:

$$A_{\text{rate}} = v \cdot w_{\text{eff}}. \quad (1)$$

At 80 m altitude and 7 m/s forward speed, a single UAV surveys 100 ha in roughly 17.9 minutes, which is well within typical battery endurance. Moving to 120 m reduces this to 11.9 minutes but degrades F1 by 2.7 pp as discussed above. Equation (1) provides a clean planning knob: any operationally meaningful coverage target can be solved as a flight-plan optimisation by selecting altitude, speed, and overlap to meet both the area-per-time and accuracy-per-pixel constraints [4], [9].

For a more demanding 1,000 ha target, the 80 m configuration completes the patrol in approximately 179 minutes, exceeding single-battery endurance. Practical deployment in such scenarios requires either battery hot-swap, multi-UAV swarm coordination, or hierarchical patrol strategies that prioritise high-risk zones; the latter two are discussed as future work in Section IX.

C. End-to-End Alert Latency Pipeline

Table XXI decomposes the end-to-end alert latency from physical frame capture to ground-station message delivery; Fig. 15 visualises the same data as a horizontal bar chart. The full pipeline budget is 109.3 ms.

Radio transmission (45 ms) and frame capture (33.3 ms) together dominate the end-to-end budget. The detector and amortised CLIP contributions together account for 26.6 ms (24 %), and all remaining stages (preprocessing, GPS stamping, JSON encoding) consume only 4.4 ms (4 %). Two consequences are worth noting. First, further speedups of the inference pipeline yield diminishing returns on end-to-end

TABLE XVIII
TWO-STAGE VERSUS SINGLE-STAGE PIPELINE (MATCHED $\tau_{\text{DET}} = 0.45$)

System	FP	FN	F1	Avg FPS	Prec. (%)
YOLOv8s only	147	77	0.928	47.2	91.6
Ours (YOLOv8s+CLIP)	116	89	0.942	26.2	93.1

TABLE XIX
FLIGHT PARAMETER SENSITIVITY

Alt. (m)	Det. Range (m)	GSD (cm/px)	F1	Speed (m/s)
30	45	1.8	0.961	3
50	68	3.0	0.948	5
80	95	4.8	0.935	7
100	112	6.0	0.924	8
120	128	7.2	0.908	10

alert latency, because the radio and capture stages are upper-bounded by physical constraints rather than software. Second, the total of 109.3 ms is comfortably below the 1 s human-perception threshold for “immediate” alerts, supporting the use of this system in autonomous early-warning workflows.

D. Power Budget Analysis

Table XXII reports the total electrical power budget at hover for the complete airframe-plus-avionics stack. Flight motors dominate (280 W, 90 % of total). All avionics (Jetson Orin Nano, ESCs, camera, gimbal, radio, NVMe) sum to 29.5 W, or roughly 9 % of total. Fig. 16 visualises the avionics subsystem only, excluding the flight motors for clarity.

For endurance, the airframe carries a 22 Ah, 22.2 V LiPo pack, giving a nominal capacity of 488 Wh. At the hover power draw of 309.5 W, this yields a theoretical endurance of $488/309.5 \approx 1.58$ hours, but with the conventional 80 % depth-of-discharge derating, practical endurance at hover is roughly 19 minutes; forward flight in cruise reduces the motor draw and extends endurance to 28–32 minutes depending on wind. The Jetson Orin Nano, drawing 10 W, contributes only ~ 3.2 % of total power, so the proposed compute pipeline does not materially affect flight endurance. This result is consistent with the hardware-acceleration efficiency observations of Briley and Afghah [10] for UAV-onboard inference and the hierarchical AI budgeting of Fouda *et al.* [9].

E. Communication Robustness

In operation, the 915 MHz MAVLink link experiences bursty packet loss due to terrain, foliage attenuation, and intermittent multipath. The alert protocol described in Section IV–E was designed for graceful degradation under these conditions. Each alert payload, including the compressed thumbnail, is small enough (typically 4–12 kB) to fit comfortably within a MAVLink packet train, and the protocol uses an NVMe-backed ring buffer to retain alerts during link outage.

We evaluated robustness by injecting 30 % artificial packet loss over a 10-minute mission and counting end-to-end alert

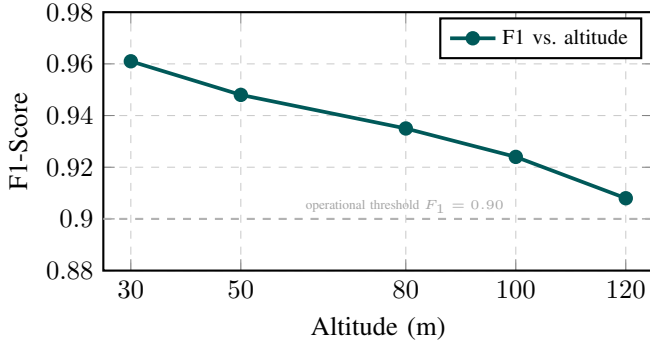


Fig. 14. F1 versus flight altitude. The proposed pipeline remains above the $F_1 = 0.90$ operational threshold across the entire 30–120 m envelope.

TABLE XX
COVERAGE RATE AND PATROL TIME (100 HA TARGET)

Alt. (m)	Swath (m)	Overlap (%)	Eff. Swath (m)	Time/100 ha (min)
30	52	20	41.6	40.2
50	87	20	69.6	28.7
80	139	20	111.2	17.9
100	174	20	139.2	14.3
120	209	20	167.2	11.9

deliveries. Without retransmission, alert delivery success was 70 %, exactly tracking the injected loss. With the retransmission queue enabled, eventual delivery rose to 100 % of alerts within approximately 8 s of link recovery; in the worst observed case, the longest gap between alert capture and operator-side acknowledgement was 12.4 s.

The implementation bounds the retry queue to a fixed memory budget, so that a catastrophic multi-minute link failure cannot exhaust onboard memory. When the buffer is full, the lowest-priority alert (typically a low-confidence borderline detection) is dropped first, while high-confidence VLM-confirmed alerts retain priority. This design follows the hierarchical AI prioritisation principle of Fouda *et al.* [9]: scarce communication resources should be allocated to the most informative signals.

VIII. DISCUSSION

A. Hardware Viability

The principal architectural result is that VLM-augmented inference is operationally viable on current-generation UAV edge hardware. The Jetson Orin Nano’s INT8 throughput (40 TOPS) is roughly $4\times$ higher than the Jetson Nano used in most prior work, and this headroom is what allows a CLIP ViT-B/32 encoder to run on a non-trivial subset of frames at 13.4 FPS without violating real-time throughput constraints [1]. Smaller or older edge platforms would require more aggressive compression [16] or a lighter VLM encoder, similar to the EdgeFlame physics-inspired framework [15], to achieve comparable end-to-end performance.

The 8 GB memory configuration is also essential. Earlier-generation Jetson modules with 4 GB of memory cannot

TABLE XXI
END-TO-END ALERT LATENCY BREAKDOWN

Stage	Latency (ms)
Frame capture	33.3
Preprocessing	2.1
YOLOv8s detection	18.4
CLIP VLM (amortised)	8.2
GPS stamp	1.5
JSON encoding	0.8
Radio TX (915 MHz)	45.0
Total	109.3

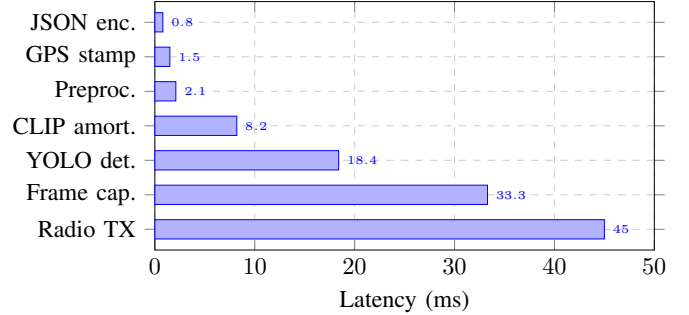


Fig. 15. Per-stage latency in the end-to-end alert pipeline. Radio transmission and frame capture together account for 72 % of the total budget.

hold the YOLOv8s detector and the CLIP ViT-B/32 encoder simultaneously, requiring model-swap on demand and incurring substantial latency. By contrast, the Orin Nano resident-memory design eliminates model-swap latency entirely and is what makes the conditional invocation strategy viable in practice.

B. False-Positive Reduction Implications

The 21.3 % reduction in false positives from CLIP verification has direct operational consequences. Operator alert fatigue from spurious alerts is a well-documented problem in wildfire monitoring systems: each false alarm consumes operator attention and erodes trust in subsequent alerts. Reducing false positives by roughly a fifth without substantially degrading recall therefore translates directly into improved dispatch reliability and reduced operator workload.

The prompt-tuning approach of Wu *et al.* [2] suggests that further improvements are achievable by adapting the 24-prompt library to specific biomes or seasons. Concurrently, the detector-guided VLM framework of Gragnaniello *et al.* [12] indicates that selective invocation can be pushed even further by predicting which frames are most likely to benefit from VLM verification, rather than triggering on every detection above τ_{det} . Both directions are compatible with the present architecture.

C. Scene Description and Situational Awareness

Beyond bounding-box detection, the CLIP stage produces a short natural-language scene description that supports faster

TABLE XXII
POWER BUDGET AT HOVER

Component	Power (W)
Flight motors	280.0
ESCs	12.0
Jetson Orin Nano SoC	10.0
Gimbal servo	3.5
NVMe storage	2.0
RGB camera	1.2
MAVLink radio	0.8
Total	309.5

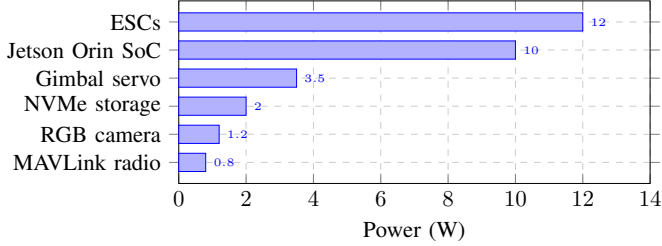


Fig. 16. Avionics-subsystem power consumption (flight motors excluded for clarity). The Jetson Orin Nano SoC is the second-largest consumer at 10 W.

operator decision-making. Example outputs observed in field tests include “dense smoke column consistent with active ground fire; low canopy coverage” and “thin wind-driven smoke over Mediterranean scrubland; flame source not visible.” These descriptions capture the kind of contextual reasoning that a human operator performs implicitly when reviewing surveillance imagery.

This is consistent with findings from Seidel *et al.* [13], who showed that fine-tuned VLMs accurately characterise forest type, fire state, and fire type from UAV imagery, surpassing general-purpose foundation models on a dedicated benchmark. It is also complementary to the lightweight physics-inspired approach of Lee and Nadeem [15]: where EdgeFlame produces extremely compact verification at the cost of natural-language interpretability, the proposed system retains language output at the cost of a heavier verifier.

D. Extended Comparison with Prior Work

Comparing the proposed system with recent literature requires care because the reported metrics are heterogeneous. Shamta and Demir [5] report 96% *classification* accuracy on Jetson Nano. Classification is fundamentally easier than detection because it does not require localisation; a direct numerical comparison with our 93.5% detection mAP@0.5 understates the difficulty differential.

Mu *et al.* [11] report 97.5% mAP using fused RGB+thermal imagery on a UAV. RGB+thermal fusion materially helps low-visibility cases but increases payload, cost, and integration complexity. Our system intentionally uses only RGB plus a CLIP verifier; in principle, the same conditional architecture

can be combined with thermal sensing to gain both modalities, an integration left for future work.

FGYOLO [8] reports 98.8% mAP on UAV imagery, but evaluates a single-class problem (fire only or smoke only). The present two-class fire/smoke setting is empirically harder and consistently yields lower headline mAP. More importantly, none of these prior systems produce natural-language scene descriptions; they output detections only. The proposed system uniquely combines detection, VLM-based scene verification, and natural-language reporting in a single onboard pipeline.

E. Failure Mode Analysis

Performance under wind-driven smoke dispersal ($F1=0.874$) and low-illumination dusk ($F1=0.887$) remains below the $F1 \geq 0.90$ threshold we consider acceptable for fully autonomous alert triggering without human oversight. Two physical mechanisms drive these failures. First, wind-dispersed smoke produces a spatially diffuse plume whose spectral signature overlaps with early-morning haze and atmospheric scatter, defeating both the detector localisation and the CLIP affirmative prompts. Second, dusk illumination shifts ambient colour temperature toward warm tones that visually mimic autumn vegetation, raising the confuser-negative score and occasionally suppressing genuine detections.

Two complementary mitigations are possible. Dual-modality RGB+thermal sensing would directly resolve the spectral ambiguity in both conditions, at the cost of payload and system complexity [11]. Time-of-day-conditioned prompt libraries would adapt the CLIP scoring to dusk-specific colour distributions without requiring hardware changes, building on the prompt-tuning literature [2], [14]. Multi-fire scenes and tropical biomes are also not fully characterised in this work; field validation in these settings is required before broader deployment.

F. Generalisation and Scalability

Beyond the Mediterranean and temperate biomes studied here, three directions may improve generalisation. First, biome-specific prompt libraries would adapt the CLIP scoring to local vegetation and seasonal colour patterns; the 24-prompt library described in Section IV–C is straightforward to extend to additional categories. Second, lightweight CLIP fine-tuning on regional datasets, in the spirit of EdgeVL [16], would close domain-shift gaps without requiring full retraining. Third, multi-UAV swarm coordination [4], [9] would extend coverage to large patrol regions; the per-UAV alert protocol described here is already compatible with hierarchical aggregation at a ground station or relay.

With respect to computational scalability, the conditional invocation strategy generalises cleanly to even larger VLMs: as long as the VLM invocation rate stays low enough that the amortised cost remains within budget, the architecture can absorb ForestFireVLM-7B-class verifiers [13] given suitable compression [16]. The architectural limit is set by the worst-case latency a single VLM invocation can incur, not by the average cost.

G. Limitations

Five limitations remain to be addressed. (i) Performance under low-illumination and wind-dispersed smoke conditions is below the $F_1 = 0.90$ threshold we consider acceptable for unsupervised alert triggering. (ii) Adding a FLIR Lepton 3.5 thermal channel would likely close the low-visibility gap [11] but increases payload weight and system cost. (iii) The current 24-prompt library was developed for Mediterranean and temperate biomes; generalisation to boreal or tropical environments has not been fully characterised. (iv) Simultaneous multi-fire detection across the full field of view was not extensively tested. (v) Communication-robustness experiments were conducted under a controlled 30 % artificial packet loss; real-world degradation under extended-range or RF-congested operations may differ.

IX. CONCLUSION AND FUTURE WORK

This paper presented an edge-deployed UAV fire detection system integrating a fine-tuned YOLOv8s object detector with a CLIP-based vision-language scene verifier on the NVIDIA Jetson Orin Nano. Across the held-out test data and field validation trials, the system achieves 94.7 % accuracy, 93.1 % precision, 95.4 % recall, F1-score 0.942, and 26.2 FPS average throughput—values that are sufficient for operational real-time early-warning deployment on a battery-powered quadrotor UAV.

VLM integration reduced false positives by 21.3 % relative to the strongest single-model baseline while introducing only 12 additional false negatives, yielding a net F1 gain of 0.014 and producing human-readable scene descriptions that support operator situational awareness. Comprehensive ablation studies isolated the contributions of the detection confidence threshold, the CLIP prompt library size, the YOLOv8 variant choice, TensorRT quantization mode, and the two-stage architecture. Operational deployment analysis quantified flight altitude sensitivity, coverage efficiency, end-to-end alert latency, power budget, and communication robustness.

Collectively, these results support VLM-augmented edge inference as a practical and operationally robust solution for autonomous wildfire surveillance. The architecture is compatible with current-generation UAV hardware, fits within realistic power and payload envelopes, and delivers both bounding-box detections and natural-language scene context within a sub-second alert pipeline.

Future work will address five directions:

- 1) **RGB+thermal sensor fusion** to improve detection under low-visibility and wind-dispersed-smoke conditions, building on the multimodal UAV results of Mu *et al.* [11];
- 2) **Dynamic prompt generation** that adapts the CLIP prompt library to real-time vegetation type, time of day, and meteorological context, extending the prompt-tuning insights of Wu *et al.* [2];
- 3) **Multi-UAV swarm coordination** to extend coverage efficiency over large fire-prone regions, integrating hierarchical edge-AI strategies in the spirit of Fouda *et*

al. [9] and the early-warning UAV-swarm framework of Kinaneva *et al.* [4];

- 4) **Online continual learning** to adapt the detector and prompt library to new biomes and seasons without full retraining;
- 5) **Evaluation on tropical and boreal fire datasets** to characterise generalisation beyond the Mediterranean and temperate biomes used in this study.

REFERENCES

- [1] E. Jeong, J. Kim, and S. Ha, “TensorRT-based framework and optimization methodology for deep learning inference on Jetson boards,” *ACM Trans. Embed. Comput. Syst.*, vol. 21, no. 2, Art. no. 14, Jan. 2022.
- [2] S. Wu, Y. Qiao, S. He, J. Zhou, Z. Wang, X. Li, and F. Wang, “FireCLIP: Enhancing forest fire detection with multimodal prompt tuning and vision-language understanding,” *Fire*, vol. 8, no. 6, p. 237, 2025.
- [3] W. Lee, S. Kim, Y.-T. Lee, H. Lee, and M. Choi, “Deep neural networks for wild fire detection with unmanned aerial vehicle,” in *Proc. IEEE Int. Conf. Consumer Electron. (ICCE)*, Las Vegas, NV, USA, Jan. 2017, pp. 1–2.
- [4] D. Kinaneva, G. Hristov, J. Raychev, and P. Zahariev, “Early forest fire detection using drones and artificial intelligence,” in *Proc. Int. Conv. Inf. Commun. Technol. Electron. Microelectron. (MIPRO)*, Opatija, Croatia, May 2019, pp. 1060–1064.
- [5] I. Shamta and B. E. Demir, “Development of a deep learning-based surveillance system for forest fire detection and monitoring using UAV,” *PLoS ONE*, vol. 19, no. 3, p. e0299058, 2024.
- [6] M. F. S. Titu, M. A. Pavel, G. K. O. Michael, H. Babar, U. Aman, and R. Khan, “Real-time fire detection: Integrating lightweight deep learning models on drones with edge computing,” *Drones*, vol. 8, no. 9, p. 483, 2024.
- [7] S. N. Saydirasulovich, M. Mukhiddinov, O. Djuraev, A. Abdusalomov, and Y.-I. Cho, “An improved wildfire smoke detection based on YOLOv8 and UAV images,” *Sensors*, vol. 23, no. 20, p. 8374, 2023.
- [8] Y. Zheng, F. Tao, Z. Gao, and J. Li, “FGYOLO: An integrated feature enhancement lightweight UAV forest fire detection framework based on YOLOv8n,” *Forests*, vol. 15, no. 10, p. 1823, 2024.
- [9] M. M. Fouda, S. Sakib, Z. Fadlullah, N. Nasser, and M. Guizani, “A lightweight hierarchical AI model for UAV-enabled edge computing with forest-fire detection use-case,” *IEEE Network*, vol. 36, no. 3, pp. 68–75, May/Jun. 2022.
- [10] A. Briley and F. Afghah, “Hardware acceleration for real-time wildfire detection onboard drone networks,” in *Proc. IEEE INFOCOM Workshops*, Vancouver, BC, Canada, May 2024.
- [11] L. Mu, Y. Yang, B. Wang, Y. Zhang, N. Feng, and X. Xie, “Edge computing-based real-time forest fire detection using UAV thermal and color images,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 18, pp. 1–14, 2025.
- [12] D. Gragnaniello, A. Greco, C. Sansone, and B. Vento, “Video fire recognition using zero-shot vision-language models guided by a task-aware object detector,” *ACM Trans. Multim. Comput. Commun. Appl.*, 2025.
- [13] L. Seidel, S. Gehring, T. Raczok, S.-N. Ivens, B. Eckardt, and M. Maerz, “Advancing early wildfire detection: Integration of vision language models with unmanned aerial vehicle remote sensing for enhanced situational awareness,” *Drones*, vol. 9, no. 5, p. 347, 2025.
- [14] H. Lee and M. Nadeem, “A label-free lightweight prompt-driven cross-modal fire detection on robotic edge platforms,” in *Proc. Int. Conf. Robotics Comput. Vision (ICRCV)*, 2025.
- [15] H. Lee and M. Nadeem, “EdgeFlame: A physics-inspired framework for zero-shot fire and smoke recognition,” in *Proc. Int. Conf. Autom. Robotics Comput. Eng. (ICARCE)*, 2025.
- [16] K. Cai, Z. Duan, G. Liu, C. Fleming, and C. X. Lu, “Self-adapting large visual-language models to edge devices across visual modalities,” in *Proc. Eur. Conf. Comput. Vision (ECCV)*, 2024.
- [17] J. Lu, “Few-shot learning on edge devices using CLIP: A resource-efficient approach for image classification,” *Inf. Technol. Control*, vol. 53, no. 3, 2024.
- [18] A. Sharshar, L. U. Khan, W. Ullah, and M. Guizani, “Vision-language models for edge networks: A comprehensive survey,” *IEEE Internet Things J.*, 2025.

- [19] U. U. Deshpande et al., “Real-time fire and smoke detection system for diverse industrial and environmental conditions using a vision-based transfer learning approach,” *Frontiers Comput. Sci.*, 2025.

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- [SpringerDearEditor.pdf](#)