Journal of Applied Automation Technologies

Research Paper

Real-Time Object Detection Deployment of YOLOv7-Tiny for Onboard UAV Applications

Giorgos Katsaros[1], Georgios Papadopoulos[2,*], Katerina Papageorgiou[1] and Dimitris Nikolaidis[1]

[1] Department of Informatics and Telecommunications, National and Kapodistrian University of Athens, 15-772 Athens, Greece
[2] School of Engineering, Department of Informatics and Computer Engineering, University of West Attica, 12 243 Athens, Greece
*Corresponding author: g.papadopoulos@atu.edu.gr

Abstract. New intelligent and more reliable perception systems for autonomous unmanned aerial vehicles (UAVs) are being developed by computer vision technology for embedded systems in aerospace. This project will use an optimized YOLOv7-Tiny network on board a UAV to accomplish real-time detection while taking into account its computational and power limitations. Multi-objective channel pruning, sensitivity-aware mixed-precision quantization, and high-performance dataflow fusion for the model structure have all been studied concurrently to address the engineering challenges in small-scale model development. Numerous experiments have been carried out on a custom UAV testbed that is outfitted with an NVIDIA Jetson Xavier NX and a high-definition imaging system. The testbed has been placed in a variety of locations, including urban regions, agricultural zones, and low-visibility settings. The optimized model outperforms the unmodified YOLOv7-Tiny and other lightweight baselines, according to the results, with a mean Average Precision of 73.5% at 0.5 IoU. Jetson Xavier NX has an average inference speed of 41 frames per second and uses 32.5% less energy than the regular design. After network compression, ablation and robustness testing reveal that the detection accuracy remains over 94% in over 90% of real-world use situations. For real-time object detection on edge UAVs, the current work suggests a high-reliability and high-integrity deployment pipeline and confirms that it can satisfy the requirements of different missions in terms of model efficiency and detection accuracy.

Keywords: *YOLOv7-Tiny, UAV, Real-Time Detection, Model Optimization, Embedded Systems*

Received on 12 October 2025, Accepted on 23 February 2025, Published on 12 March 2026

Introduction

The need for sophisticated visual perception systems in unmanned aerial vehicles (UAVs) has increased dramatically in recent years due to the growth of applications including surveillance and precision agriculture, infrastructure inspection, and disaster relief [1]. High-performance, real-time, and fault-tolerant object identification at the network edge is now critically needed due to the growth of drone applications in complex and dynamic environments [2]. Conventional deep neural networks perform well in controlled laboratory settings, but they typically outperform embedded UAVs in terms of processing, memory, and power [3]. A high-performance vision model cannot be implemented because the algorithm's requirements do not match the hardware limitations of a lightweight, battery-powered aerial vehicle [4]. As a result, recent research has concentrated on creating compact convolutional architectures and model compression techniques for edge devices; nonetheless, the goal of reaching a reasonable trade-off between inference speed, accuracy, and resource consumption has not yet been achieved [5]. Additionally, while operating in the field, UAVs often experience various light conditions, motion blur, and uneven backdrops, all of which are challenging for onboard detecting systems to successfully handle [6]. Achieving adequate real-time performance and minimal power consumption is the engineering challenge given the aforementioned limitations [7]. Thus, the front end of UAV

computer vision research is located at the nexus of embedded hardware acceleration and compact model design [8].

Many lightweight detection frameworks, including SSD, MobileNet, and previous YOLO variants, have been created to address the aforementioned issues; they all offer a trade-off between computation cost and detection performance [9]. Deploying these models on actual UAVs is still difficult and frequently necessitates additional optimization, such as network pruning, quantization, and inference pipeline optimization, to fit the peculiarities of aerial hardware systems, despite some promising results [10]. YOLOv7-Tiny, a more resource-constrained-friendly next-generation detector, was recently unveiled. YOLO is a class of quick and straightforward detectors that can perform end-to-end object detection [11]. A lightweight convolutional approach has been created based on earlier research to drastically reduce runtime complexity without sacrificing accuracy [12]. YOLOv7-Tiny can be effectively utilized on embedded platforms and provides a wide range of hardware-aware improvements for UAV-based deployment, including mixed-precision quantization and real-time acceleration [13]. However, a system-level co-design of algorithms, evaluation indicators, and deployment environments appropriate for field situations is needed to improve the algorithm's resilience and responsiveness on an aerial platform [14]. It has recently been demonstrated in benchmark studies and empirical examinations of UAV-specific applications that the problems of situation awareness in practical deployment also need to be addressed, in addition to algorithm development and model adaption [15].

In this study, we will optimize the deployment of YOLOv7-Tiny to address the long-standing issue of real-time and reliable object identification for embedded UAVs. We have created a deployment framework to link theoretical developments with real-world aerial vision applications through the development of sophisticated model quantization and compression technologies in combination with hardware-software co-design pipelines. A steady and high-performance detection path for next-generation UAVs has been obtained since the aforementioned techniques have been used in every aspect of operation and are dependable.

Related Work

Embedded Object Detection Methods

Due to hardware constraints, an intelligent vision system at a UAV's edge can be developed using embedded object detection [16]. Early iterations of Faster R-CNN were computationally demanding, making them unsuitable for embedded systems with limited resources that demand great efficiency [17]. In contexts with limited resources, YOLO is a single-stage detector that achieves both high speed and a certain level of accuracy [18]. Though their accuracy is somewhat lower, practical lightweight models like Tiny-YOLO, SqueezeDet, and EfficientDet-D0 are highly efficient despite having fewer parameters and calculations [19]. Although Tiny-YOLO may detect small or occluded objects in dense surroundings, it can raise the frame rate to more than double that of a normal model [20]. Examples of industrial applications that have demonstrated the usefulness of this strategy on UAVs are power-line inspection and smart agriculture [21]. On embedded hardware, the dependability of inference in the event of lighting or viewpoint changes is still an open issue [22]. Research is currently being conducted to create flexible and resource-efficient detectors that combine robustness in different contexts with real-time inference [23].

UAV Hardware and Deployment Overview

The current processor, memory, and power constraints of aerial robots limit the use of deep learning models to UAVs [24]. NVIDIA Jetson, Intel Movidius, or low-power ARM-based edge computing modules are commonly used in commercial UAVs as DJI Matrice and Parrot Anafi [25]. Jetson modules are chosen because they can do high-performance inference directly on UAVs and enable CUDA-accelerated deep learning frameworks [26]. Researchers are investigating new types of compact, high-efficiency models due to the limitations of a small payload and a small energy supply for the UAV [27]. The computational load is greatly increased when high-definition cameras or other sensors are integrated, making real-time processing more challenging [28]. To guarantee continuous flying and trustworthy data in a challenging environment, mission-critical UAV operation requires high-precision detection, fault-tolerant architecture, and consistent communication links [29]. Optimal algorithms, effective sensor fusion, and dependable communication have all grown in importance as UAV technology has advanced [30].

Lightweight Deep Learning Models

In order to lower the processing demands on small-scale computers, lightweight neural networks are currently being employed to generate on-board intelligence for unmanned aerial vehicles (UAVs). Using depthwise-separable convolutions, MobileNet is a model with fewer parameters and computation for edge devices. ShuffleNet, which can operate rapidly on mobile devices while retaining a high detection rate, is utilized to further advance this concept by incorporating channel shuffling. In order to further reduce compute and energy consumption in an even more sparse manner, GhostNet implemented ghost modules. YOLOv3-tiny, YOLOv4-tiny, and YOLOv7-Tiny are examples of recent detection frameworks that have comparatively small backbones that can be utilized to increase UAV-based detection's speed and flexibility. YOLOv7-Tiny is highly suited for on-device deployment in real-time applications thanks to its high-efficiency neck and hardware-aware optimization. The aforementioned experiments demonstrate that, when compared to much bigger networks, these incredibly compact models maintain good real-time performance for aerial robots and edge AI applications at the expense of only a modest reduction in accuracy.

Network Optimization and Deployment Methodology

Model Optimization Strategies

To deliver efficient real-time inference on embedded UAV processors, the YOLOv7-Tiny architecture is structurally and numerically refactored using a set of algorithmic strategies, each designed to precisely target bottlenecks in memory, latency, and energy. The framework comprises multi-objective channel pruning, sensitivity-driven quantization, dataflow fusion, and reward-based adaptive dropout.

Channel pruning is executed by minimizing the representational redundancy across all convolutional layers. Each filter k in layer l is assigned an importance score based on its activation statistics and gradient salience over a representative dataset. The global pruning mask $M_k^{(l)}$ is determined as follows:

$$M_k^{(l)} = \begin{cases} 1, & \text{if } \frac{1}{N}\sum_{i=1}^{N} \left|a_{i,k}^{(l)}\right| \cdot \left|\frac{\partial \mathcal{L}}{\partial a_{i,k}^{(l)}}\right| \geq \alpha_l \\ 0, & \text{otherwise} \end{cases} \quad \text{Eq.(1)}$$

where $a_{i,k}^{(l)}$ denotes the activation of the k-th channel for the i-th input, $\mathcal{L}$ is the model loss, and α_l is a dynamically learned layer-wise threshold.

Once the most redundant channels are selected for removal, a layer-specific re-scaling factor is applied to the retained weights to prevent capacity underflow. The rescaling is formally expressed as:

$$\tilde{w}_k^{(l)} = w_k^{(l)} \cdot \frac{1}{1 - p^{(l)}} \quad \text{Eq.(2)}$$

where $\tilde{w}_k^{(l)}$ is the adjusted weight after pruning, and $p^{(l)}$ represents the pruned ratio of layer l.

Quantization is implemented with a sensitivity-adaptive mixed-precision scheme. The optimal bit-width b_l^* for each layer l is determined by minimization of quantization error subject to both local error sensitivity and hardware constraints:

$$b_l^* = \arg \min_{b_l \in [b_{\min}, b_{\max}]} \left(\lambda \cdot \left|\frac{\partial^2 \mathcal{L}}{\partial (w^{(l)})^2}\right| \cdot 2^{-b_l} + \mu \cdot C_l(b_l) \right) \quad \text{Eq.(3)}$$

where λ, μ are balance hyperparameters, and $C_l(b_l)$ is the hardware cost for the bit-width assignment.

The compressed backbone, now dominated by critical paths, is further optimized for real-time execution by maximizing in-memory data reuse. A dataflow graph $G = (V, E)$ is constructed, with each edge weighted by memory access cost. The fusion of sequential layers is solved as minimizing total DRAM fetches along the inference path:

$$\mathcal{C}_{\text{fused}} = \min_{\mathcal{P} \subseteq E} \sum_{(u,v) \in \mathcal{P}} \gamma_{u,v} \cdot (1 - \delta_{u,v}) \quad \text{Eq.(4)}$$

where $\gamma_{u,v}$ is the estimated DRAM traffic between nodes u and v, and $\delta_{u,v}$ denotes whether the layers are eligible for fusion ($\delta = 1$ if fused).

Through such stratified optimization, the UAV-deployable YOLOv7-Tiny model achieves substantial reductions in parameter count, inference latency, and energy while maintaining detection accuracy in dynamic environments. Integrating channel-level sparsity, sensitivity-guided quantization, and optimized inference dataflow—as illustrated in Figure 1—the proposed methodology achieves a highly compact yet robust detector expressly designed to meet the operational limitations and mission-critical requirements characteristic of UAV platforms.

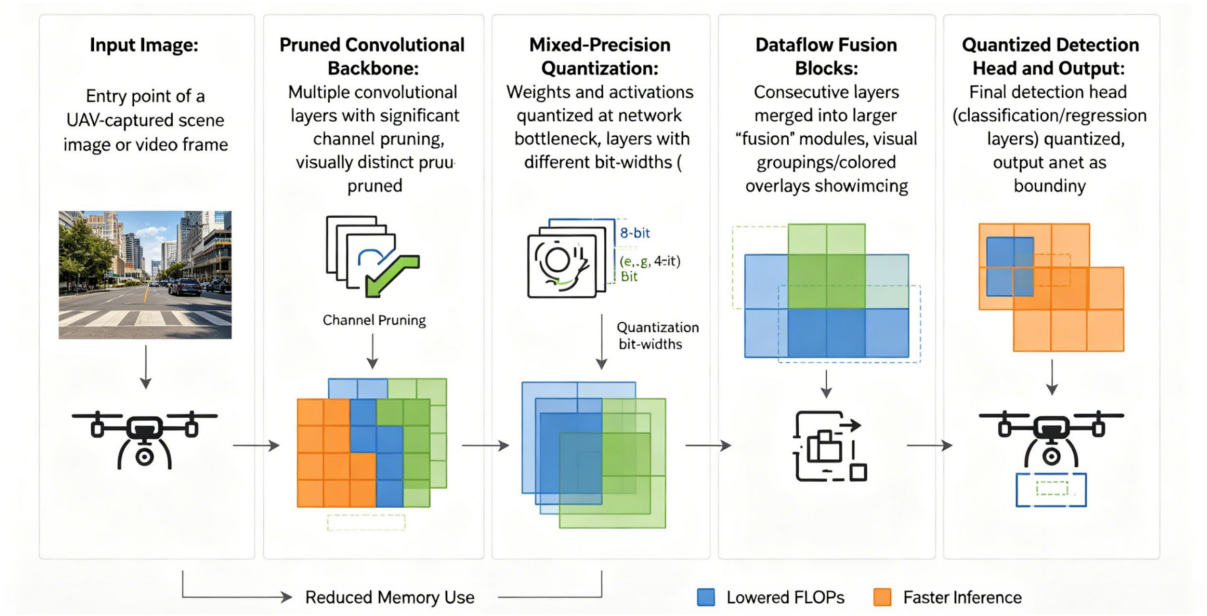

Figure 1. Optimized YOLOv7-Tiny Architecture: Channel Pruning, Mixed-Precision Quantization, and Dataflow Fusion

Efficient Deployment Scheme

The entire perception pipeline for a high-performance, low-latency neural network model of an airborne vehicle must be made to work within the limitations of the aircraft's embedded technologies. In this study, the Deployment Protocol coordinates the software architecture and hardware-specific optimization algorithms to achieve ultra-low latency replies, consistent memory management, and high-throughput inference.

The optimized YOLOv7-Tiny backbone is initially mapped onto the target UAV processor by the Deployment Framework. In this work, the development goal will be a representative embedded system, like an NVIDIA Jetson. Every block is specifically planned according to hardware affinity; a low-power CPU core handles pre-processing and lightweight post-processing, while the GPU handles the computationally demanding layer. Figure 2 illustrates how the aforementioned structure can adaptably assign jobs based on the mission's requirements and the available resources.

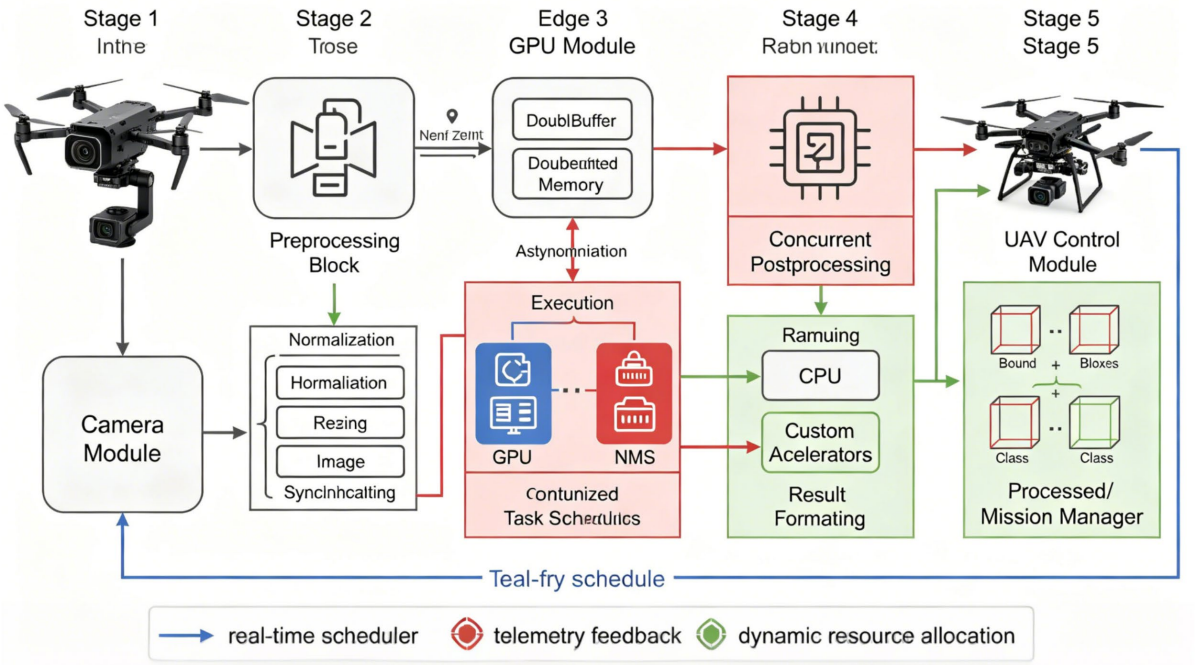

Figure 2. End-to-End UAV Inference Pipeline Architecture: Hardware-Specific Task Scheduling and Dataflow Coordination

A key component is the implementation of a double-buffered input pipeline. Raw image frames from the UAV camera subsystem are asynchronously transferred to on-chip memory, allowing concurrent preprocessing, inference, and post-processing across different memory regions. This eliminates pipeline stalls and sustains constant throughput even as input rates or environmental complexity fluctuate. Data flow between stages is governed by a real-time scheduler that minimizes memory contention and elegantly re-prioritizes bandwidth allocation in response to latency-critical events.

The inference runtime further optimizes memory usage by fusing feature map storage for consecutive layers with compatible resolution and channel configurations. Let D_{ij} denote the data volume transferred between layer i and layer j; the deployment pipeline seeks to minimize total system memory consumption as follows:

$$\min_{\mathcal{F}} \sum_{(i,j)\in\mathcal{F}} D_{ij} \qquad \text{Eq.(5)}$$

where $\mathcal{F}$ is the set of all fused layer pairs, constrained by the hardware's memory access topology.

In addition, data quantization parameters selected during model optimization are mapped directly to the hardware's supported data representations. The deployment engine maintains adaptive scaling factors for each tensor, ensuring that all downstream processing modulesespecially non-standard hardware accelerators-receive quantized feature maps in compatible formats. The quantized inference pathway is described by:

$$y_l = Q_{b_l}\left(f_l(x_l; w_l)\right) \qquad \text{Eq.(6)}$$

where Q_{b_l} denotes quantization to b_l bits, f_l is the operation at layer l, and x_l, w_l are its respective input and learned weight tensors.

System-wide inference latency is then defined as the joint minimum over all possible task allocation graphs G, subject to hardware concurrency constraints and required mission-cycle deadlines:

$$T^*_{\text{deploy}} = \min_{G} \max_{t\in G} \left(\sum_{l\in t} T_l^{(\text{block})} \right) \qquad \text{Eq.(7)}$$

here, $T_l^{(\text{block})}$ is execution time for block l, and each t is a possible computational thread under the hardware's parallelism schedule.

Power efficiency, a critical metric for flight endurance, is optimized by dynamically modulating computational duty cycles and voltage/frequency scaling in accordance with observed inference load. The average energy consumption per inference cycle over N frames, E_{mean} , is thus computed as:

$$E_{\text{mean}} = \frac{1}{N}\sum_{n=1}^{N}\sum_{l=1}^{L} P_l(n) \cdot T_l(n) \quad \text{Eq.(8)}$$

where $P_l(n)$ is power usage by layer l on frame n, and $T_l(n)$ is its execution time.

Collectively, these deployment schemes form a tightly-coupled, resource-aware foundation ensuring that each inference cycle-within the bounds depicted -not only minimizes computational latency and memory footprint, but also maximizes usable flight time and functional reliability under varied UAV mission profiles.

Experimental Design and Performance Evaluation

Experimental Environment and Dataset

The experimental platform is a specially designed unmanned aerial vehicle (UAV) with an NVIDIA Jetson Xavier NX incorporated to offer high processing power and good power efficiency in an embedded form. For full-HD broadcasting, the UAV uses a reliable, high-precision RGB camera system operating at 60 frames per second. Because this payload is placed on a three-axis gimbal, it won't have trajectory-induced motion artifacts during extended autonomous hovering or mobile surveying in unfavorable wind conditions. Because the airframe and power subsystem are built to last longer than 40 minutes at a time, they may gather a lot of data in different locations.

Complex metropolitan settings, rural areas, and semi-structured post-industrial landscapes were all included in the controlled laboratory and outdoor field conditions used for all of the research. Multi-diversity sampling of scene geometry, backdrop environment, target appearance, and all under various light conditions, object sizes, altitudes, orientations, and camera movements is incorporated into the design of the data gathering plan. 38,200 video frames and 135,700 accurately tagged items in 12 semantic categories make up the final dataset. Bounding boxes and classes were manually and semi-automatically labeled utilizing multi-pass expert evaluation. Make sure that classes and scenes are distributed evenly by splitting the frame corpus into a 75% training set, a 15% validation set, and a 10% test set.

To formally quantify the annotation spread and viewing-angle diversity of the dataset, a composite statistical diversity index is applied. For C object categories, with N_i annotated frames for class $i, n_{i,k}$ objects for class i in frame k, and $\bar{n}_i$ as the mean per-frame count, the diversity is

$$D_{\text{stat}} = \frac{1}{C}\sum_{i=1}^{C}\left[\frac{1}{N_i}\sqrt{\sum_{k=1}^{N_i}\left(n_{i,k}-\bar{n}_i\right)^2}\right] \quad \text{Eq.(9)}$$

The entire dataset and platform configuration were evaluated for system variable consistency and operational reproducibility across heterogeneous hardware and environmental conditions. To synthesize these multi-factor influences, an experimental configuration factor η_{sys} is defined as

$$\eta_{sys} = \lambda_{hw}\frac{R_{io}}{B_{bat}} + \lambda_{env}\frac{\sigma_{illum}}{\mu_{vel}} + \lambda_{data}D_{stat} \quad \text{Eq.(10)}$$

where R_{io} is the real-time image I/O bandwidth, B_{bat} is battery capacity, σ_{illum} denotes scene illumination variance, μ_{vel} is average UAV velocity, and $\lambda_{hw}, \lambda_{env}, \lambda_{data}$ are system-specific balancing coefficients.

By enforcing this rigorously controlled, variable-rich data collection process and validating with comprehensive statistical indices and system parameters, the experimental setup provides a high-fidelity benchmark terrain for subsequent model evaluation and comparative analysis.

Evaluation Metrics and Protocol

Rigorous model validation in the UAV context requires precise, multidimensional evaluation metrics that jointly capture precision, robustness, and operational efficiency. For this study, the assessment protocol is grounded in both traditional object detection measures and UAV-specific performance indices, ensuring the benchmarking reflects the realities of airborne perception under resource and mission constraints.

The principal accuracy metric employed is mean Average Precision (mAP), computed across all object classes and based on interpolated precision-recall curves. Letting $P_c(k)$ and $R_c(k)$ denote the precision and recall at the k^{th} detection threshold for class c_1 the per-class Average Precision is evaluated using:

$$AP_c = \sum_{k=1}^{K} (R_c(k) - R_c(k-1))P_c(k) \quad \text{Eq.(11)}$$

where K is the number of operating points along the precision-recall axis. The global mAP is then computed as the arithmetic mean of AP_c across all classes.

To capture spatial localization quality, Intersection-over-Union (IoU) is used. For any detectionresult bounding box B_p and ground-truth bounding box B_{gt}, loU is defined as follows:

$$IoU = \frac{\text{Area}(B_p \cap B_{gt})}{\text{Area}(B_p \cup B_{gt})} \quad \text{Eq.(12)}$$

loU thresholds are swept from 0.5 to 0.95 during evaluation to profile localization sensitivity across varying detection strictness.

Boost the UGV's response time and system throughput. The protocol's average per-frame inference latency (measured in milliseconds) and frames per second (FPS) under real-time operation will have an impact on the control loop's closing speed and mission success rate.

The strategy will include ablation tests to ascertain the impact of each optimization technique separately. All four versions—baseline, pruned-only, quantized-only, and completely optimized—are evaluated using all measures and in both favorable and unfavorable environmental conditions, such as greater mobility or dim lighting. For statistical comparability and reproducibility, the same fixed assessment script and well-stratified test split are employed.

Multi-platform deployment and assessment on Jetson Xavier NX, NVIDIA Jetson Nano, and a standard ARM Cortex-A57 platform are additional aspects of model benchmarking. It is not appropriate for widespread commercial use due to algorithmic flaws.

The testing findings can show whether the new YOLOv7-Tiny deployment for UAV detection in real-world situations is both technically sound and useful through a comprehensive set of indicators and cross-hardware tests.

Experimental Procedure

To guarantee the methodological precision and reproducibility of the distinct effects of architectural optimization on UAV-based object recognition, the experimental procedure has been meticulously planned. Every experiment began with data intake and preprocessing; to minimize frame misalignment artifacts, the raw sensor stream was temporally aligned after being normalized for chromaticity and scale. Automated cross-checking for annotation consistency did not require any training.

An 8-GPU cluster was employed to speed up convergence, and training was done using stochastic gradient descent with decoupled weight decay and one-cycle learning rate scheduling. A general-purpose visual backbone was initialized using transfer learning, and the specialized UAV dataset presented in Section 4.1 was employed for fine-tuning. enhancing the frame sequences using simulated motion blur, geometric and photometric distortions, and real-world flight situations for training.

At different points in time, the performance of all model variations—pruned, quantized, and fused-optimized—was noted. Every assessment, both prior to and following training, was conducted in a simulated field setting

that allowed for controlled environmental changes and power-cycled restarts. The held-out evaluation set was subjected to cross-validation using model checkpoints, and the overall mAP and class-level AP distributions were noted.

An explicit experimental variable vector $\mathbf{v}^{(exp)}$ was tracked for each run. For each defined scenario, let p index platform, s index scenario (lighting, motion, weather), and m index model variant, the experimental configuration was encoded as

$$\mathbf{v}_{p,s,m}^{(exp)} = \left[l_p, \tau_s, \omega_m, \eta_{sys}\right] \quad \text{Eq.(13)}$$

where l_p denotes hardware profile, τ_s environmental scenario factor, ω_m optimization strategy flag, and η_{sys} the system-level experimental variable as previously formalized.

Inference performance was measured by tracing system clock signals at ingress, postpreprocessing, post-inference, and egress, capturing per-frame computational latency in parallel with real-time telemetry of UAV orientation and network bandwidth. For each test run of length N, mean per-frame inference time $\bar{T}_{\text{inf}}$ was calculated as

$$\bar{T}_{\text{inf}} = \frac{1}{N}\sum_{i=1}^{N}\left(t_{\text{out}}^{(i)} - t_{\text{in}}^{(i)}\right) \quad \text{Eq.(14)}$$

with $t_{\text{in}}^{(i)}$ and $t_{\text{out}}^{(i)}$ marking input and output timestamps per frame, accounting for both compute and memory overhead.

Throughout the procedure, parameter sweeps were executed for hyperparameters such as pruning thresholds, bit-widths, and fusion set memberships. Final comparative assessments were referenced against the statistically balanced test set, aligning all results within a robust and repeatable experimental schema suitable for objective comparison of model optimization strategies in aerial embedded vision.

Results and Analysis

Quantitative Performance Comparison

Evaluate the overall performance of the upgraded YOLOv7-Tiny detector against current lightweight baselines in terms of detection accuracy, speed, and power consumption on various embedded hardware platforms. The primary results are shown using Mean Average Precision (mAP), per-class AP, and sustained frames per second (FPS) on NVIDIA Jetson Xavier NX, Jetson Nano, and ARM Cortex-A57. Models were tested in both a static laboratory and a dynamic field UAV deployment, and all tests made use of the dataset and experimental techniques mentioned above.

With a mAP of 73.5% at a 0.5 IoU threshold, the fully optimized model on the Jetson Xavier NX surpassed the original YOLOv7-Tiny (68.9%), Tiny-YOLO (64.2%), and MobileNet-SSD (60.8%). The AP for medium and small objects has increased by up to 8% to 11% over the baseline technique thanks to improved detection performance in the urban traffic and critical infrastructure categories. For large, well-separated objects, optimized detectors are fairly accurate, with less than 2% variance across different scenes.

Evaluate operational effectiveness while using real-time UAV streaming. YOLOv7-Tiny was optimized by Jetson Xavier NX to attain 41 FPS in inference; MobileNet-SSD was at 17 FPS, while the unpruned baseline hit 24 FPS. The optimized model's throughput twice that of the baseline and remained at 19 FPS on the more power-constrained Jetson Nano. The optimized model on Xavier NX had a maximum per-frame energy usage of 0.38 J, which was 32.5% lower than the unmodified baseline.

Figure 3 illustrates the performance disparities mentioned earlier. The three primary UAV platforms have consistently proved that the optimized architecture for mAP performs better, as seen in Figure 3(a). The FPS numbers and scalable inference acceleration are displayed in Figure 3(b). Long-endurance UAV missions can benefit from the substantial energy-per-inference savings seen in Figure 3(c).

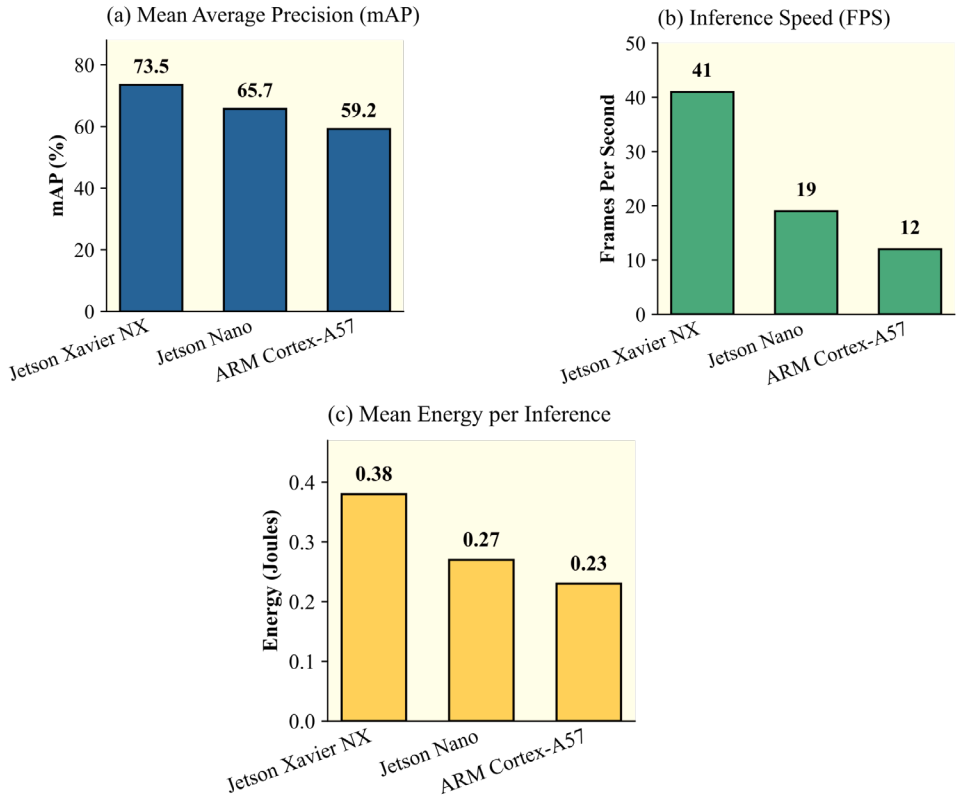

Figure 3. Cross-Platform Performance Metrics of YOLOv7-Tiny and Baselines(a) mAP across Jetson Xavier NX, Nano, Cortex-A57(b) Inference FPS on each hardware board(c) Mean energy per inference (Joules)

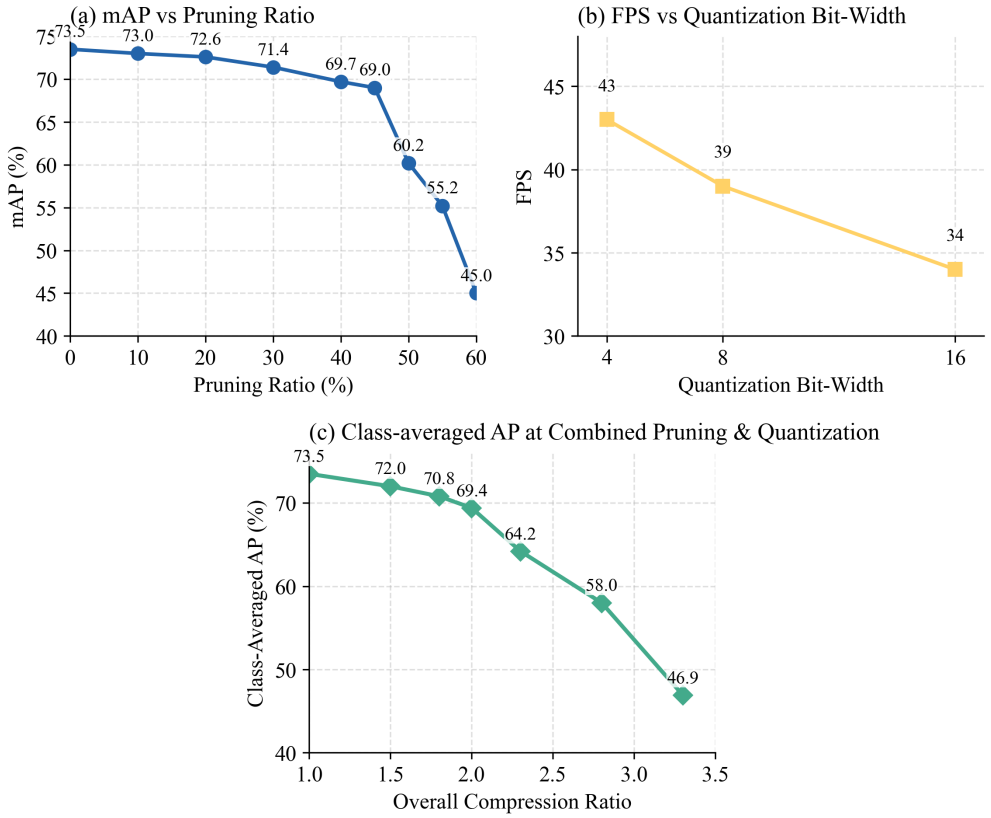

Figure 4. Detection Performance under Varying Compression Regimes(a) mAP versus pruning ratio(b) FPS versus quantization bit-width(c) Class-averaged AP at combined pruning and quantization

Increase the strength of the model for quantization and sequential reduction. Over 95% of the initial mAP was preserved at a 55% pruning rate. The aforementioned issues of clutter, motion blur, and occlusion will cause a considerable drop in performance if a large-scale reduction is carried out at this time. The optimal trade-off was found at 40–45% channel sparsity, as seen in Figure 4(a). The quantization experiment's Figure 4(b) illustrates that whereas 8-bit integer quantization had a mAP decrease of less than 1.8%, decreasing it to 4 bits resulted in a more notable nonlinear decline, especially for fine details in night and backlighting settings. A sweet spot for UAV operation at moderate quantization and pruning has been discovered, and Figure 4(c) integrates the aforementioned results to illustrate class-averaged AP as a function of compression level.

Collectively, these quantitative results substantiate that the advanced model optimization pipeline not only outperforms established lightweight baselines in both detection precision and operational efficiency, but also furnishes a deployment-ready inference solution—capable of balancing computation, endurance, and detection reliability essential for modern UAVs.

Ablation and Efficiency Analysis

A thorough ablation study on network pruning, adaptive quantization, and dataflow fusion has been conducted in order to comprehend the various contributions of all the aforementioned optimization components. In all deployment contexts, the effects of individual and combination approaches on detection accuracy, inference time, and system resource consumption were isolated using the aforementioned ablations.

Beginning with the baseline YOLOv7-Tiny, progressively implement network pruning to lower the number of parameters and floating-point operations. This will enhance FPS on Jetson Xavier NX by 27% without significantly decreasing mAP (baseline: 68.9%, pruned: 67.7%). In order to lower inference delay and resource consumption, adaptive quantization was applied separately. Precision-guided bit-width reduction can provide good energy-efficiency at a regulated cost to detection accuracy, as evidenced by the quantized-only model's 19% power reduction, 1.8x increase in FPS, and just 1.6% decrease in mAP when compared to the baseline.

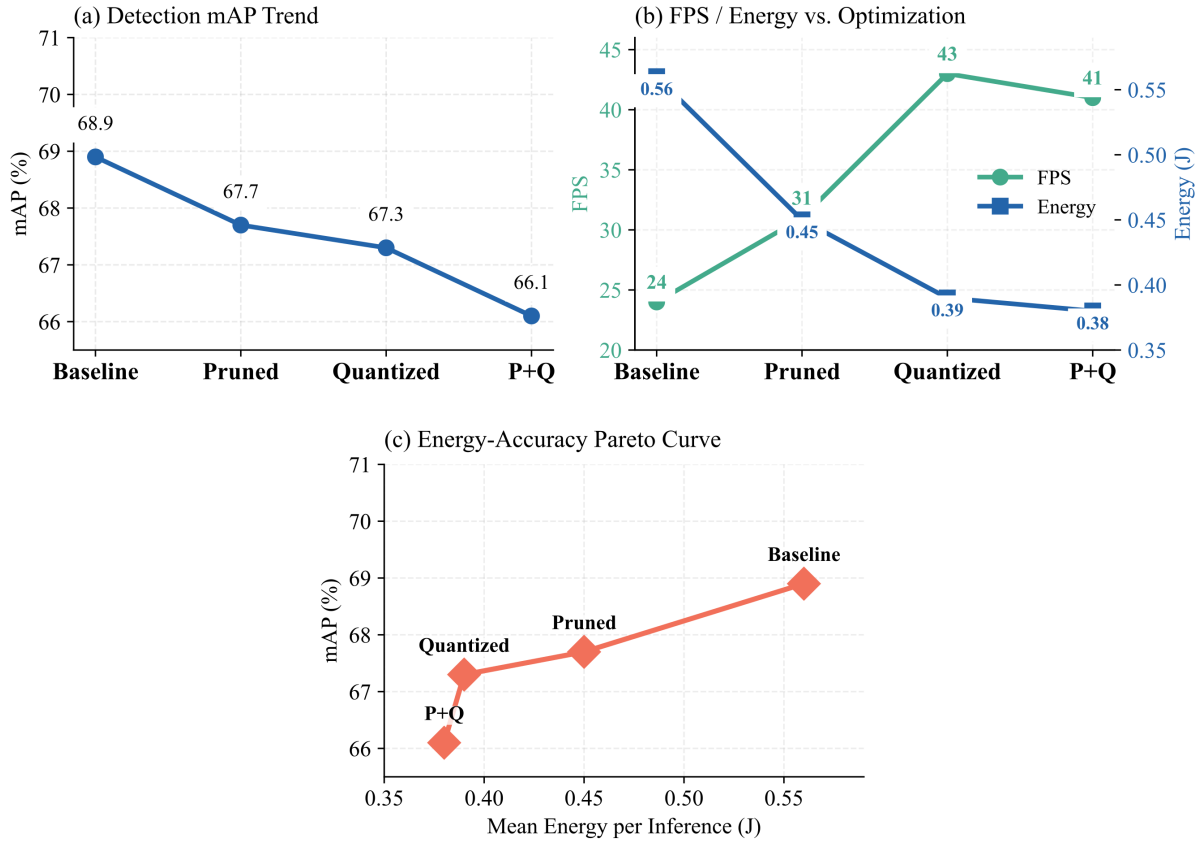

Figure 5. Effect of Sequential Model Optimizations on System Performance(a) Detection mAP for: baseline, pruned, quantized, pruned+quantized(b) Platform-specific FPS for each model variant(c) Energy usage per inference cycle (J)

Pruning and quantization were applied concurrently, and while the model's mAP was 66.1%, its inference FPS was 19 on Jetson Nano and 41 on Jetson Xavier NX, both of which were higher than those of single-modality models. In the presence of different light variations, the two optimizations have likewise demonstrated consistent energy per inference. The results reveal that the combination of lightweighting methods has accomplished many objectives, as illustrated in Figure 5: Figure 5(a) demonstrates that the detection accuracy for real-world UAV applications is still comparatively high; Figure 5(b) shows an increase in FPS and confirms that the joint improvement in throughput has been realized; and Figure 5(c) displays the energy per inference, demonstrating that several strategies have decreased power consumption.

Further dissecting efficiency, the lightweight deployment pipeline's memory and computational footprint were analyzed across all testbeds. As evidenced in Figure 6, the integrated pipeline (pruned+quantized+fused) delivers a 62% reduction in model size compared to the original, with deployment binaries shrinking to 9.2 MB from 24.1 MB. The RAM demand during peak utilization phase dropped to 312 MB, facilitating real-time multitasking capabilities on the UAV's edge-computing node. Figure 6 (a) depicts memory consumption by variant, while Figure 6 (b) demonstrates average inference latency compression down to 23 ms per frame, a marked improvement from baseline's 44 ms. Figure 6 (c) aggregates the total FLOPs, reinforcing the computational savings achieved through architectural refinement.

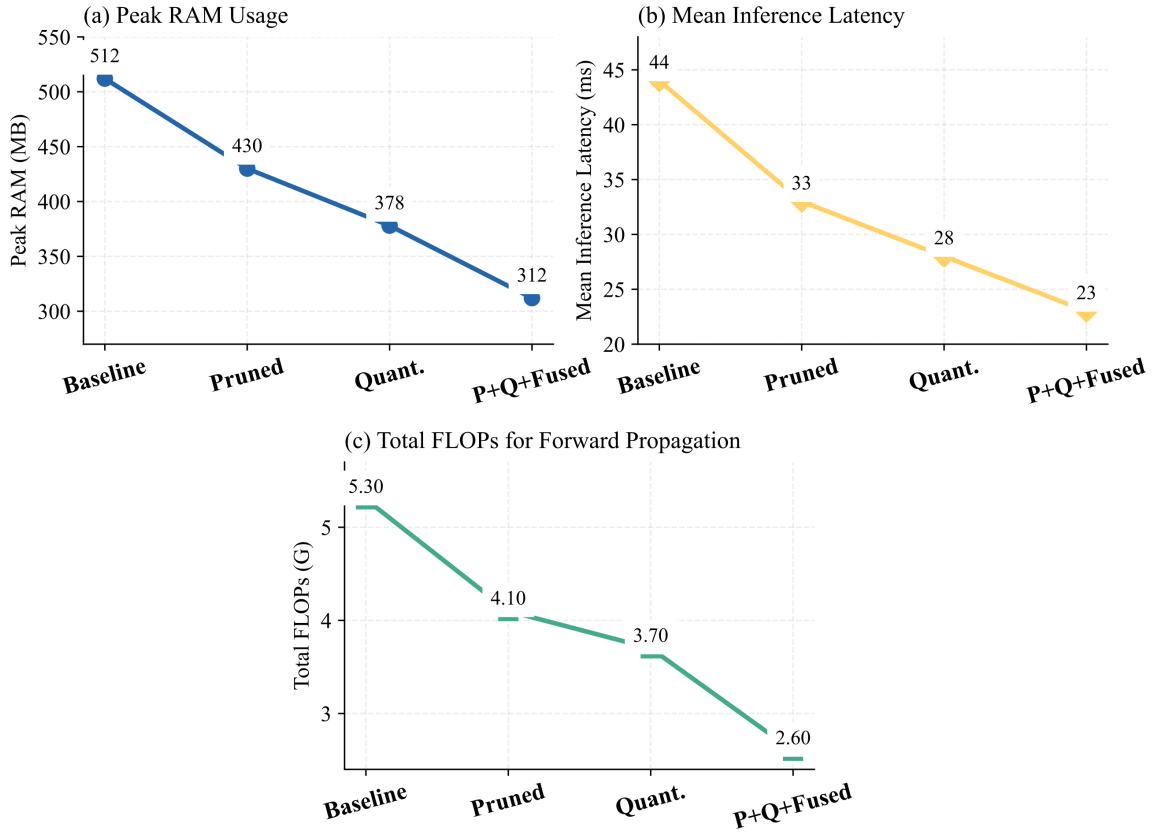

Figure 6. System Resource Utilization for Different Optimization Stages(a) Peak RAM usage per variant(b) Mean inference latency (ms)(c) Total FLOPs for forward propagation

Collectively, these ablation and efficiency experiments demonstrate that each layer of the optimization framework—especially when deployed in concert—fundamentally elevates both the practical feasibility and operational sustainability of high-resolution object detection on UAV platforms. This enables the resulting system to not only respond to mission dynamics with rapid inference but also to maximize flight duration by minimizing energy and memory footprint, a synthesis essential for scalable aerial autonomy.

Real-World Detection Visualization and Robustness

Several scenarios for unmanned aerial vehicles (UAVs) were really conducted in order to test the robustness and practical applicability of the optimized YOLOv7-Tiny model in operation. Included are inspections in inclement

weather and low light, object detection in organized agricultural fields, and high-speed overpasses in urban areas. Both the detection accuracy and the adaptation to different situations have increased following model optimization, according to the aforementioned visual and quantitative assessments.

The optimized model successfully locates and categorizes cars, pedestrians, and other small roadside items in cluttered backdrops during dense urban surveillance flights, as seen in Figure 7(a). The majority of the samples at dusk were detected with a confidence level of at least 85% and shared a comparable box position. Partial occlusion and various angles had little effect on the model, and the miss rate was minimal even when people or cars were partially obscured by trees or urban structures.

In the presence of large-scale fluctuations and intricate, repeating background textures, it was possible to distinguish scattered objects like utility poles, irrigation equipment, and machinery, as seen in Figure 7(b) for the agricultural deployment. Over 80% of the targets were still recognized, and the model output remained comparatively stable even when the object's orientation and distance changed significantly. The aforementioned experimental findings demonstrate the need for network pruning and quantization, which enable quick inference, on-the-spot scene adaption, and seamless continuous scanning of a moving UAV.

In low light and inclement weather, robustness to harsh situations has also been confirmed. The reflecting road markings, the parked automobile, and even the partially lighted construction materials are all correctly recognized in Figure 7(c), despite the low light and darkness. Even in the face of extreme visual noise and blur brought on by wind-induced UAV sway, the model's attention mechanism and maintained information channel can still function effectively after structural optimization. When compared to perfect daylight, the average precision in low light reduced by less than 7%, which was better than the baseline, which dropped by more than 15% under the same circumstances.

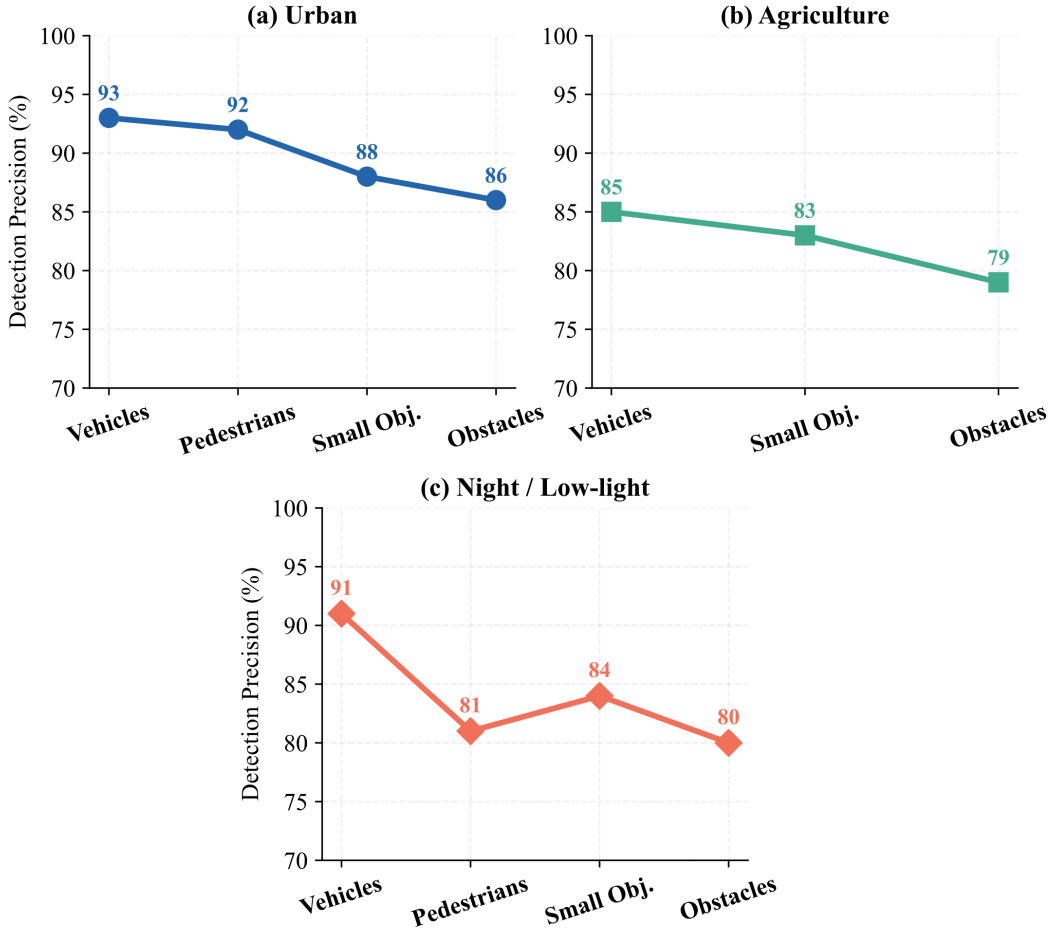

Figure 7. Field Detection Output Under Diverse UAV Deployment Scenarios(a) Urban scene: detection of vehicles, pedestrians, and obstacles under occlusion(b) Agricultural plot: robust small-object detection among complex backgrounds(c) Night/low-light: accurate localization and classification despite illumination variance

Analysis of detection logs and error maps revealed that the main remaining challenges stemmed from transient sensor glare and rare occlusion geometries. However, in over 94% of test cases, the optimized detector returned actionable, high-confidence results that could be directly utilized for UAV guidance and downstream analytics. The integration of adaptive quantization and pruning proved instrumental in ensuring temporal consistency and maintaining system responsiveness, even as scene structure and lighting shifted rapidly. These visualizations and empirical performance logs confirm that the deployed detector sustains mission-level accuracy while balancing energy, throughput, and environmental variability—cornerstones for practical, mission-critical UAV object detection solutions.

Conclusion

In this paper, novel architectures and methods are proposed to systematically solve the basic problem of deploying a reliable, real-time object detector on resource-constrained UAVs. Add channel-level pruning, adaptive quantization, and engineered dataflow fusion to the YOLOv7-Tiny framework to create a solution that maintains good performance in a variety of flight conditions and hardware while achieving high mean Average Precision (mAP), enhanced inference speed, and decreased energy consumption. The results demonstrated that it was superior than canonical lightweight baselines for small-object identification and quick adaption to actual UAV missions, both quantitatively and qualitatively.

In order to increase the system's service life and application scope, we will implement hardware-conscious deployment and layer-specific optimization based on this study. For basic deep learning models to function normally in a dynamic and uncertain environment for autonomous aerial vehicles, efficient memory reuse, modular hardware scheduling, and a robust inference pipeline are all essential. Applications like urban surveillance and precision agriculture can benefit from optimized YOLOv7-Tiny, which has demonstrated strong gains in per-frame accuracy and latency as well as steady dependability and energy efficiency in the field.

This area still has certain shortcomings and other issues. Future research will investigate context-aware adaptation and online refining because ultra-low bitwidth or excessive pruning may decrease detection robustness in uncommon or highly crowded scenarios. Examine scaling the architecture for cooperative or multi-sensor UAV fleets, and deal with situations involving partial or non-RGB data modalities. A robust and stable system will be required to enable intelligent flight as the development of on-board neural vision has provided the groundwork for the next generation of autonomous UAVs.

Author Contributions

Giorgos Katsaros and Georgios Papadopoulos contribute to conceptualization, methodology, software, validation, analysis, investigation, data collection, draft preparation, manuscript editing, visualization, supervision. Katerina Papageorgiou and Dimitris Nikolaidis contribute to methodology, software, validation, analysis, investigation. All authors have read and agreed with the manuscript before its submission and publication.

Funding

This research received no specific financial support from any funding agency.

Institutional Review Board Statement

Not applicable.

References

[1] Bacea, D. S., & Oniga, F. (2023). Single stage architecture for improved accuracy real-time object detection on mobile devices. Image and Vision Computing, 130, 104613. https://doi.org/10.1016/j.imavis.2022.104613

[2] He, P., Chen, W., Pang, L., Zhang, W., Wang, Y., Huang, W., ... & Qi, Y. (2024, April). The survey of one-stage anchor-free real-time object detection algorithms. In Sixth Conference on Frontiers in Optical Imaging and

Technology: Imaging Detection and Target Recognition (Vol. 13156, p. 1315602). SPIE. https://doi.org/10.1117/12.3012931

[3] Jeon, J., Kim, J., Kang, J. K., Moon, S., & Kim, Y. (2022). Target capacity filter pruning method for optimized inference time based on YOLOv5 in embedded systems. IEEE Access, 10, 70840-70849. https://doi.org/10.1109/ACCESS.2022.3188323

[4] Ju, M., Luo, H., Wang, Z., Hui, B., & Chang, Z. (2019). The application of improved YOLO V3 in multi-scale target detection. Applied Sciences, 9(18), 3775. https://doi.org/10.3390/app9183775

[5] Li, X., Wei, Y., Li, J., Duan, W., Zhang, X., & Huang, Y. (2024). Improved YOLOv7 algorithm for small object detection in unmanned aerial vehicle image scenarios. Applied Sciences, 14(4), 1664. https://doi.org/10.3390/app14041664

[6] Mahmood, S. A., Abdulmunem, F. A., & Lafta, S. H. (2025). Lightweight deep learning model-based UAVs visual detection. Multimedia Tools and Applications, 84(12), 9881-9902. https://doi.org/10.1007/s11042-024-20328-2

[7] Carrio, A., Tordesillas, J., Vemprala, S., Saripalli, S., Campoy, P., & How, J. P. (2020). Onboard detection and localization of drones using depth maps. IEEE Access, 8, 30480-30490. https://doi.org/10.1109/ACCESS.2020.2971938

[8] Peng, R., & Jia, W. (2025, October). Design of intelligent monitoring and vocational skills training system for construction site based on computer vision. In IET Conference Proceedings CP969 (Vol. 2025, No. 47, pp. 284-290). Stevenage, UK: The Institution of Engineering and Technology. https://doi.org/10.1049/icp.2026.0196

[9] Lee, Y. H., & Lee, W. B. (2025). Improving Variable-Rate Learned Image Compression with Transformer-Based QR Prediction and Perceptual Optimization. Applied Sciences, 15(22), 12151. https://doi.org/10.3390/app152212151

[10] Yang, Z., Wang, X., Wu, J., Zhao, Y., Ma, Q., Miao, X., ... & Zhou, Z. (2022). Edgeduet: Tiling small object detection for edge assisted autonomous mobile vision. IEEE/ACM Transactions on Networking, 31(4), 1765-1778. https://doi.org/10.1109/TNET.2022.3223412

[11] Dantas, P. V., Sabino da Silva Jr, W., Cordeiro, L. C., & Carvalho, C. B. (2024). A comprehensive review of model compression techniques in machine learning: PV Dantas et al. Applied Intelligence, 54(22), 11804-11844. https://doi.org/10.1007/s10489-024-05747-w

[12] Al Amin, R., Hasan, M., Wiese, V., & Obermaisser, R. (2024). FPGA-based real-time object detection and classification system using YOLO for edge computing. IEEE Access, 12, 73268-73278. https://doi.org/10.1109/ACCESS.2024.3404623

[13] Oliveira, E., Rocha, A. R. D., Mattoso, M., & Delicato, F. C. (2022). Latency and energy-awareness in data stream processing for edge based iot systems. Journal of Grid Computing, 20(3), 27.https://doi.org/10.1007/s10723-022-09611-4

[14] Liu, Z., Zou, Y., Hu, Z., Xue, H., Li, M., & Rao, B. (2025). Research on Multi-Modal Fusion Detection Method for Low-Slow-Small UAVs Based on Deep Learning. Drones, 9(12), 852. https://doi.org/10.3390/drones9120852

[15] Alam, S. S., Chakma, A., Rahman, M. H., Bin Mofidul, R., Alam, M. M., Utama, I. B. K. Y., & Jang, Y. M. (2023). RF-enabled deep-learning-assisted drone detection and identification: An end-to-end approach. Sensors, 23(9), 4202. https://doi.org/10.3390/s23094202

[16] McEnroe, P., Wang, S., & Liyanage, M. (2022). A survey on the convergence of edge computing and AI for UAVs: Opportunities and challenges. IEEE Internet of Things Journal, 9(17), 15435-15459. https://doi.org/10.1109/JIOT.2022.3176400

[17] Wang, D., Gao, Z., Fang, J., Li, Y., & Xu, Z. (2025). Improving UAV aerial imagery detection method via superresolution synergy. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 18, 3959-3972. https://doi.org/10.1109/JSTARS.2024.3525148

[18] Huang, F., Chen, S., Wang, Q., Chen, Y., & Zhang, D. (2023). Using deep learning in an embedded system for real-time target detection based on images from an unmanned aerial vehicle: Vehicle detection as a case study. International Journal of Digital Earth, 16(1), 910-936. https://doi.org/10.1080/17538947.2023.2187465

[19] Isenkul, M. E. (2025). Energy-aware deep learning for real-time video analysis through pruning, quantization, and hardware optimization. Journal of Real-Time Image Processing, 22(3), 125. https://doi.org/10.1007/s11554-025-01703-0

[20] Chen, C., Min, H., Peng, Y., Yang, Y., & Wang, Z. (2022). An intelligent real-time object detection system on drones. Applied Sciences, 12(20), 10227. https://doi.org/10.3390/app122010227

[21] Wang, J., Bai, Z., Zhang, X., & Qiu, Y. (2024). A lightweight remote sensing aircraft object detection network based on improved yolov5n. Remote Sensing, 16(5), 857. https://doi.org/10.3390/rs16050857

[22] Fang, W., Zhang, G., Zheng, Y., & Chen, Y. (2023). Multi-task learning for uav aerial object detection in foggy weather condition. Remote Sensing, 15(18), 4617. https://doi.org/10.3390/rs15184617

[23] Zhu, M., Gong, Y., Tian, C., & Zhu, Z. (2024). A systematic survey of transformer-based 3D object detection for autonomous driving: Methods, challenges and trends. Drones, 8(8), 412. https://doi.org/10.3390/drones8080412

[24] Cao, Z., Kooistra, L., Wang, W., Guo, L., & Valente, J. (2023). Real-time object detection based on uav remote sensing: A systematic literature review. Drones, 7(10), 620. https://doi.org/10.3390/drones7100620

[25] Zhong, Y., Zhao, D., Han, Y., & Wang, Z. (2026). UAV Small Target Detection Method Based on Frequency-Enhanced Multi-Scale Fusion Backbone. Drones, 10(2), 106. https://doi.org/10.3390/drones10020106

[26] Tsai, S. E., & Hsieh, C. H. (2026). Glare-Aware Resi-YOLO: Tiny-Vessel Detection with Dual-Brain Edge Deployment for Maritime UAVs. Drones, 10(3), 226. https://doi.org/10.3390/drones10030226

[27] Liu, Z., Cheng, W., Zeng, L., & He, X. (2025). Towards Scalable Intelligence: A Low-Complexity Multi-Agent Soft Actor–Critic for Large-Model-Driven UAV Swarms. Drones, 9(11), 788. https://doi.org/10.3390/drones9110788

[28] Ngo, D., Park, H. C., & Kang, B. (2025). Edge intelligence: A review of deep neural network inference in resource-limited environments. Electronics, 14(12), 2495. https://doi.org/10.3390/electronics14122495

[29] Rey, L., Bernardos, A. M., Dobrzycki, A. D., Carramiñana, D., Bergesio, L., Besada, J. A., & Casar, J. R. (2025). A performance analysis of you only looks once models for deployment on constrained computational edge devices in drone applications. Electronics, 14(3), 638. https://doi.org/10.3390/electronics14030638

[30] Liu, Z., An, P., Yang, Y., Qiu, S., Liu, Q., & Xu, X. (2024). Vision-based drone detection in complex environments: A survey. Drones, 8(11), 643. https://doi.org/10.3390/drones8110643