AIAA Aviation Forum and ASCEND co-located Conference Proceedings
21 - 25 July 2025, Las Vegas, Nevada
AIAA AVIATION FORUM AND ASCEND 2025

10.2514/6.2025-3546

Viability of NPU-Equipped SBCs for Real-Time Onboard Vision Autonomy on Lighter-Than-Air UAVs

Matthew Widjaja[*], Caeden Taylor[†], Andrew Meighan[‡], and Or D. Dantsker[§]
Indiana University, Bloomington, IN 47408

Abstract

UAVs face unique constraints in autonomous operation, including limited mass, power, and possible unreliable communication under degraded telemetry relay conditions. These challenges necessitate stable onboard inference, as off-board compute becomes undependable during real-time flight. In this work, we assess the viability of lightweight, Neural Processing Unit (NPU)-equipped single-board computers (SBCs) for onboard object detection and control and propose a configurable system design adaptable to a range of UAV types, including lightweight, high-speed, maneuverable platforms. We benchmark CPU-only and NPU-accelerated SBCs–specifically the Raspberry Pi Zero 2W and the Rockchip-based Radxa ZERO 3W and Khadas Edge 2, respectively–through stress tests to evaluate inference latency consistency. Following that, we present an onboard perception pipeline using a YOLOv8-nano model with minimal off-board telemetry. Our results demonstrate that modern NPUs sustain reliable real-time inference and control under the physical and environmental limitations of lighter-than-air UAVs, enabling robust autonomy without dependence on high-bandwidth ground links.

Nomenclature

DTR	=	Defend the Republic competition
$FLOP$	=	Floating-Point Operations Per Second
$GFLOP$	=	Billion Floating-Point Operations Per Second
LTA	=	Lighter-Than-Air
SBC	=	Single Board Computer
UAV	=	Unmanned Aerial Vehicle
$YOLO$	=	You Only Look Once
NPU	=	Neural Processing Unit
CV	=	Coefficient Variance
SoC	=	System-on-a-Chip
p_{50}	=	50th percentile of inference latency
p_{95}	=	95th percentile of inference latency
p_{95}	=	99th percentile of inference latency

[*] Undergraduate Student, Department of Intelligent Systems Engineering, AIAA Student Member. mawidj@iu.edu
[†] Graduate Student, Department of Intelligent Systems Engineering, AIAA Student Member. caedtayl@iu.edu
[‡] Graduate Student, Department of Intelligent Systems Engineering, AIAA Student Member. acmeighan@iu.edu
[§] Assistant Professor, Department of Intelligent Systems Engineering, AIAA Member. odantske@iu.edu

I. Introduction

Unmanned Aerial Vehicles (UAVs) are increasingly deployed across a wide spectrum of applications, including precision agriculture, industrial inspection, environmental surveillance, and disaster response. As these mission profiles grow in complexity, UAV systems must support real-time obstacle avoidance, sustained flight, and autonomous decision-making—all without relying on constant human supervision. Achieving such independence is especially challenging in environments that are adversarial, highly dynamic, and bound by limited onboard computation. Under those conditions, traditional methods that rely on GPS waypoint-based navigation, manual piloting, and reliable wireless communications become less viable.

Observing strict limits on mass, power, and processing capability, the Defend the Republic (DTR) collegiate competition serves as a rigorous testbed for these challenges. Participating teams design and test lighter-than-air (LTA) UAVs that must autonomously locate, collect, and deliver neutrally buoyant mylar balloons to various goal targets. Success is measured by the number of goals scored, highlighting critical research problems in aerial autonomy applicable not only to LTA UAVs but also to many lightweight and highly maneuverable aerial vehicles.

To address these constraints, LTA systems typically incorporate basic sensing tools such as IMUs, barometers, ultrasonic sensors, and cameras .[1–11] Early approaches employed simple vision algorithms (e.g., color-based blob detection), but these have proven inadequate under dynamic lighting and cluttered environments. Advanced techniques leveraging machine learning deliver greater robustness and accuracy, but their high computational demands have traditionally exceeded the capabilities of small embedded platforms. Server-based approaches that offload computation remotely[12] introduce reliability concerns. Because the entire system becomes dependent on an external connection, systems may fail under challenging environmental conditions.[13]

Recent advances in hardware acceleration, particularly the emergence of compact Neural Processing Units (NPUs)–specialized microprocessors that accelerate deep neural network inference with high energy efficiency–are beginning to shift this paradigm.[14] Such developments enable low-power embedded devices to execute real-time object detection directly onboard, facilitating true autonomous operation in edge-constrained environments.

This paper presents a fully onboard autonomy solution specifically designed for lighter-than-air UAVs, based on a systematic evaluation of single-board computers (SBCs) ranging from CPU-only baselines to NPU-accelerated platforms. Each board was benchmarked to determine inference latency, stability, and thermal behavior under representative vision workloads to identify the optimal balance of performance and board mass. Leveraging the selected NPU platform, a YOLOv8-nano object detection model processes camera inputs, which are fused with sensor data in real-time to drive low-level flight controls, entirely eliminating dependency on off-board servers. The resulting system significantly reduces inference latency down to 17 ms on the strongest NPU platform, and by 60 ms on the platform chosen based on LTA UAV constraints. This ensures reliable performance, even in environments with limited connectivity. By co-locating vision processing, sensor fusion, and control on a single embedded platform, the design cuts out any reliance on remote servers or communication links, paving the way for deployment in high-speed fixed-wing, agile rotary-wing, multi-rotor, and hybrid UAVs operating under tight size, weight, and power constraints.

The rest of this paper is organized as follows. Section II provides background on the DTR competition, vehicle constraints, and related research that motivates this work. Then Section III outlines the methodology used to benchmark and analyze the SBCs. Next, Section IV provides an analysis of the SBCs' inference latency, stability, and thermal behavior, highlighting their interdependence. Afterwards, Section V details the implementation of our onboard computing system based on the findings from Section IV. Finally, Section VI summarizes our contributions and outlines directions for future work.

II. Background

A. DTR Overview and Constraints

At the Defend The Republic (DTR) competition,[15,16] collegiate teams field autonomous lighter-than-air vehicle (LTA) vehicles (i.e., autonomous blimps) in a Robotic Quidditch-style match to autonomously capture helium balloons and score them into the opponent's goals. Fig. 1 shows an example of the Indiana University's LTA vehicles navigating the game space during a DTR match. Specifically, the goal is for the autonomous LTA vehicles to autonomously capture green and purple helium balloons and score them into the opponent's fluorescent yellow or orange goals. The scope of the challenge includes the physical architecture, sensor payload and design, software implementation, and cyber-physical development of the entire system.

Each game consists of two 30 minute halves, subdivided into intervals of manual and fully autonomous flight. Points are awarded to reward autonomy: manual scoring yields 1 pt, autonomous scoring with manual assistance 3 pts, and uninterrupted autonomy 10 pts–strongly incentivizing autonomy. The playing field consists of two sets of three retro-reflective plywood goals—one circle (36.5 in interior, 44.5 in exterior diameter), one square (38 in inside leg, 46 in outside), and one inverted equilateral triangle (55 in height)—are suspended from the ceiling at each end of the field in alternating yellow and orange. Fig. 2 shows a 2D birds-eye view of this arrangement.

Each LTA must follow a strict set of rules, constraining helium usage, overall weight, and capture mechanisms. The team is allotted 200 cubic feet of helium to inflate their entire fleet for the week. Each vehicle while at rest must weigh no more than 100 grams, including helium, which may not be more than 50 cubic feet. Since designs are constrained by total buoyancy, this limits complexity, sensing capabilities, and overall computational power. Typical implementations include camera-based computer vision navigation with brushless motor propulsion. Basic nets are used for ball capturing and mylar balloons with helium provide the lift for the vehicles. Therefore, due to the limited weight budget, optimal design of an LTA vehicle requires careful optimization of all subsystems of a vehicle, i.e. structure, propulsion,[17] and computation.

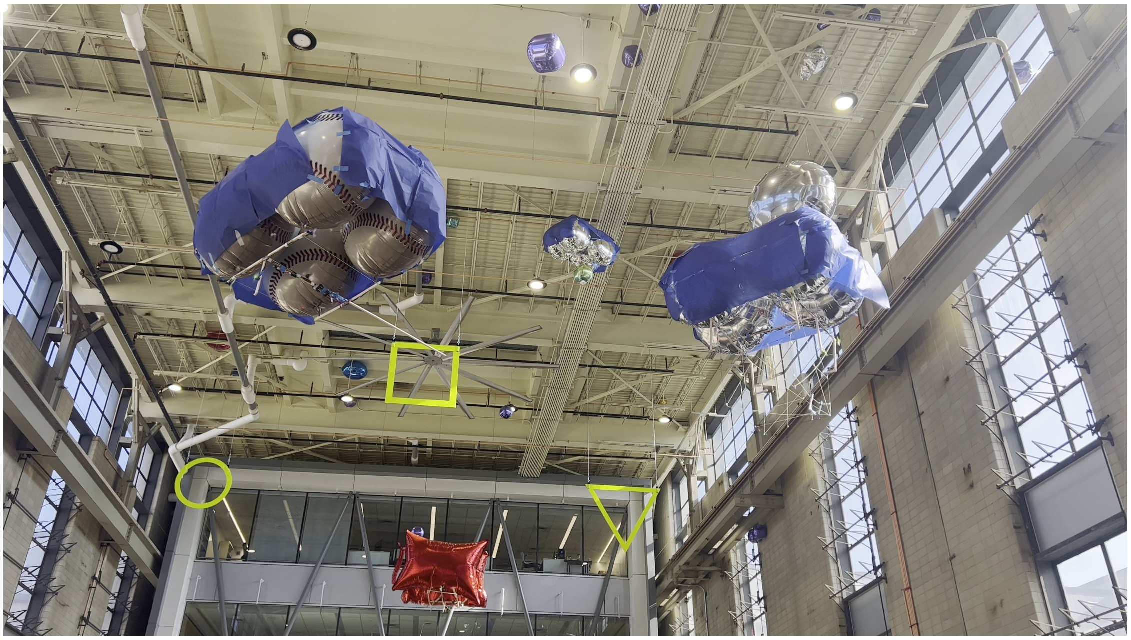

Figure 1: LTA vehicles navigating the game space during an autonomous period.

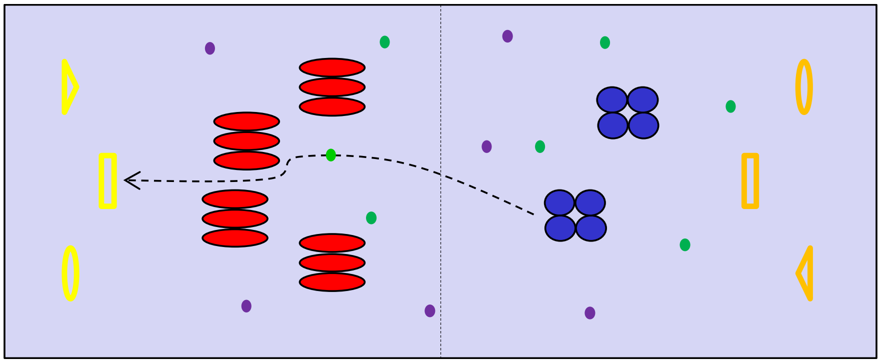

Figure 2: Birds-eye view of the DTR match layout.

B. Indiana University LTA UAV Design and Constraints

To establish clear benchmarks and system constraints, we utilized the Indiana University team's scoring LTA UAV, named "Maverick", as our baseline vehicle.[8] Maverick, which is shown in Fig 3, uses a composite structure consisting of four helium-filled balloons mounted atop a lightweight frame, which houses essential hardware components such as motors, propellers, and the onboard computing platform. The propulsion system, consisting of 4 brushless motors attached directly to the frame, enables precise maneuverability within the advesarial competition environment.

The real-time control demands of Maverick impose strict latency requirements on the onboard computational resources. Experimentally, we identified a minimum viable flight control rate of 10 Hz, corresponding to a maximum acceptable latency of 100 ms per inference and control cycle. This threshold ensures responsive and stable vehicle dynamics, crucial for accurate navigation and successful target engagement during competition scoring tasks.

The baseline architecture of Maverick integrated a Raspberry Pi Zero 2 W single-board computer (SBC), as depicted in the hardware diagram[8] shown in Fig. 4. This SBC processes incoming data from a Raspberry Pi Camera Module and a sensor array that includes an inertial measurement unit (IMU), barometer, and magnetometers. The processed sensor fusion and visual data feed directly into the vehicle's flight control algorithms.

In this effort, several candidate SBC platforms are benchmarked—including NPU-accelerated solutions—to assess their suitability as drop-in replacements for the existing Raspberry Pi Zero 2 W, with the aim of achieving consistent onboard inference times within the critical 100 ms latency constraint.

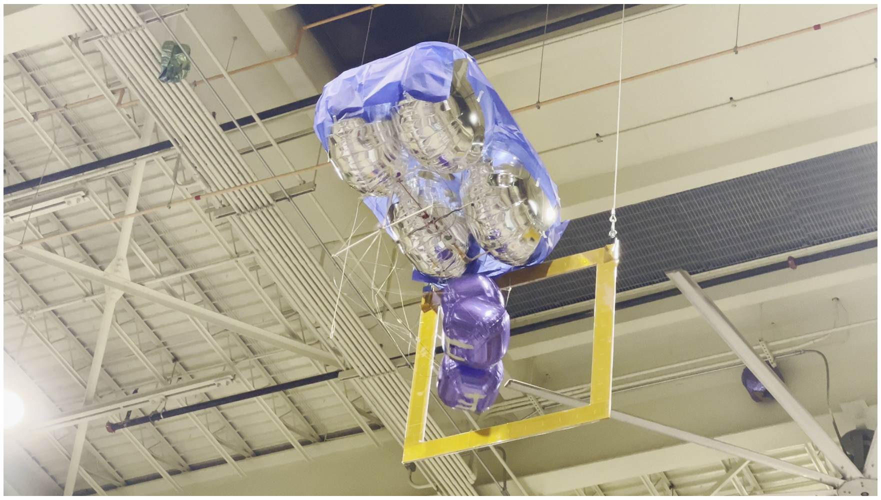

Figure 3: The IU Maverick vehicle shown scoring neutrally boyant goal balloons in a square goal.

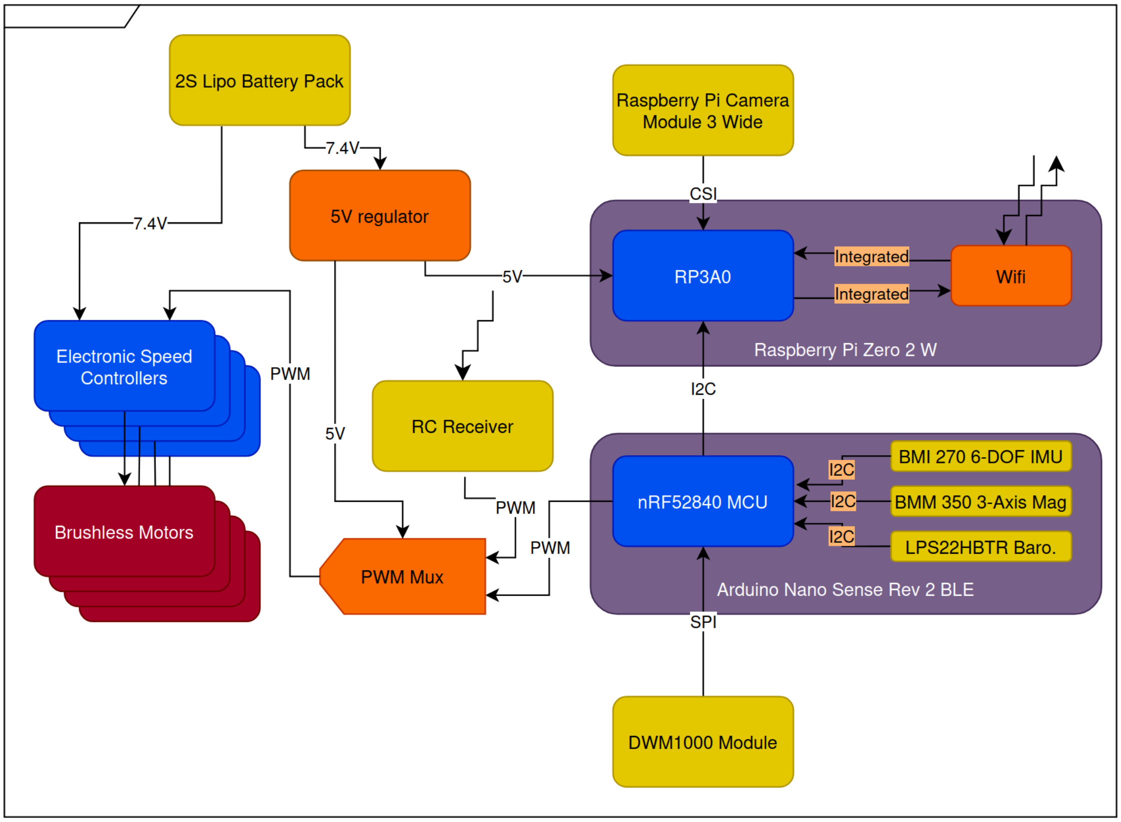

Figure 4: The hardware system diagram for the IU Maverick scoring vehicle.

C. Related Work

1. CPU + GPU

Early demonstrations of airborne deep learning relied on general-purpose CPU + GPU system-on-modules, such as the NVIDIA Jetson family. These boards inherit mature software stacks, including CUDA, TensorRT, and ROS 2, which makes it easy to port laboratory vision pipelines to mobile robots.[1,2] However, their combined mass, power consumption, heat output, and airflow requirements exceed the tight weight, energy, and cooling budgets of lightweight multirotors and buoyant platforms.[18] Consequently, control loops that appear stable on a test bench may experience latency spikes or throttling during flight, revealing a persistent discrepancy between algorithmic feasibility and field reliability.

2. Offboard Computation

To overcome these power, heat, and weight issues, many researchers stream camera frames to ground station graphics processing units (GPUs), perform inference off-board, and transmit action commands back to the vehicle via Wi-Fi.[12] While this split-compute scheme is effective under ideal channel conditions, it unravels as soon as bandwidth fluctuates or packets drop. Even modest jitter can desynchronize perception and control, forcing operators to widen safety margins or relinquish autonomy. In short, off-board inference remains unreliable whenever the wireless channel departs from its best-case scenario.

3. NPU

Recent single-board computers equipped with dedicated neural processing units, such as the Rockchip RK3566/RK3588, now deliver several tera-operations per second within a sub-3 W budget.[14] By shifting the heaviest convolutional layers onto specialized silicon, these platforms reduce power consumption and latency variance. This allows quantized networks, such as YOLOv8-Nano, to run at real-time frame rates without active cooling [8]. Consequently, NPUs offer a practical middle ground; they preserve onboard autonomy, even when links degrade, yet they stay within the strict mass and energy constraints that previously drove researchers toward offloading. Our study builds on this trend by evaluating NPU-based vision systems under the same flight dynamics that compromise CPU and off-board alternatives. This approach closes the loop between algorithm design and deployable reliability.

III. Methodology

Building on the reliability concerns outlined in Section II, we investigate whether the frame rates reported in recent embedded-AI studies[14] remain stable when the same processor must also shoulder navigation, logging, and encryption workloads in flight. To focus squarely on the compute substrate, the camera, IMU, radio, and cooling system are held constant. Only two variables are manipulated:

(i) CPU-load profiles that emulate concurrent avionics tasks, and

(ii) the input-image resolution, which trades receptive-field detail against inference latency.

All other factors (detector weights, ambient temperature, and airflow) remain identical, so any performance differences can be attributed directly to the processor under test.

A. DTR Constraints

Based on control rate constraints stated in Section II, maximum round-trip delay for usable real-time control is 100 ms under Defend the Republic (DTR) field conditions.[12] An on-board computer therefore qualifies for lighter-than-air autonomy when, during a representative 60 s flight, it meets both criteria:

(i) p_{95} inference latency $\leq 100\,\mathrm{ms}$;[12] The 95th percentile latency (p_{95}) is the value at which 95% of the latency values are less than or equal to that number. The same applies to p_{50} and p_{99}, each being the value where 50% of the values are less than p_{50} and 99% of the values are less than p_{99}.

(ii) inference latency CV $\leq 10\,\%$ The coefficient of variation (CV) is defined as the ratio of the standard deviation to the mean inference latency, quantifying latency consistency and stability. A CV of $\leq 10\,\%$ ensures predictable inference processing times, essential for precise and reliable control.

Together, these two metrics comprehensively assess both worst-case inference performance and overall stability, establishing clear criteria for determining a system's suitability for real-time autonomous operation in DTR flight scenarios.

B. Boards Under Test

Table 1 summarizes the three SBCs evaluated in this study: a CPU-only baseline (Raspberry Pi Zero 2 W) and two NPU-equipped alternatives (Radxa Zero 3W and Khadas Edge 2). Key specifications relevant to on-board perception are reported, including CPU architecture, NPU TOPS, memory, size dimensions, and mass.

C. YOLOv8 Model Configuration

A YOLOv8-nano detector ($\sim 3\,\mathrm{M}$ parameters) is fine-tuned on $3\,\mathrm{k}$ balloon-and-goal images, achieving $\mathrm{mAP}_{0.5} = 0.92$ and $\mathrm{mAP}_{0.5:0.95} = 0.70$ at the native training resolution. For benchmarking, we captured a few thousand test images

Table 1: Hardware characteristics of the evaluated SBCs.

Board	CPU Cores	NPU	RAM	Size (mm)	Mass (g)
Raspberry Pi Zero 2 W	4x Cortex-A53 @ 1.0 GHz	–	512 MB	65x30	13
Radxa Zero 3W	4x Cortex-A55 @ 1.8 GHz	RK3566 1 TOPS	2 GB	55x40	13
Khadas Edge 2	8x CortexB-A78AE @ 2.4 GHz	RK3588 6 TOPS	8 GB	82x57	27

during real flights; each frame is letter-boxed to $N \times N$ pixels, with $N \in \{256, 320, 640\}$, before being fed to the network.

D. Stress-Test Design

To mimic in-flight compute pressure, we combine five CPU-load levels with three image resolutions. CPU load is generated by worker threads that occupy 0 %, 25 %, 50 %, 75 %, or 100 % of a core, reflecting typical mixes of sensor fusion and motor control. Input resolutions of 256, 320, and 640 pixels were selected to capture the trade-off between small-object detectability and end-to-end latency. Among them, 256 pixels offers the lowest latency while preserving just enough detail for real-time control. Higher resolutions, while improving object visibility, increase processing time and thus compromise real-time responsiveness (real-time responsiveness defined in Section II). Table 2 outlines the factors; the 5×3 combinations yield 15 distinct test conditions. Each resolution is swept across all CPU loads before moving to the next.

Table 2: Stress-test factors and their levels.

Factor	Levels	Rationale
CPU load	0 %, 25 %, 50 %, 75 %, 100 %	Simulates concurrent sensors and navigation workloads
Input size	256, 320, 640 px	Small-feature vs. wide-context detection trade-off

IV. Results

Using the experimental protocol detailed in Section III, characterized per-frame latency, temporal stability, and thermal behavior for the three single-board computers (SBCs) summarized in Table 1. Throughout the analysis, we held two performance thresholds as stated in Section III A ($p_{95} \leq 100$ms, CV $\leq 10\%$) for stable lighter-than-air (LTA) flight.

Figure 5 introduces the latency distribution of the CPU-only baseline, while Figure 6 contrasts the two NPU-equipped boards Radxa Zero 3W and Khadas Edge 2. Table 3 gives percentile summaries, and Figures 7 and 8 show how synthetic CPU load and input resolution affect mean latency and SoC temperature.

A. CPU-only baseline: Raspberry Pi Zero 2 W

Across all three input sizes, the Raspberry Pi never approaches either limit (i) or (ii). With 256×256-pixel images, the median latency reaches 808 ms and the 95th percentile stretches to 1.5 s; enlarging the input to 640×640 pushes those figures to between 4.3 s and 6.4 s (Figure 5). The spread is wide, with CV values falling between 38-47%, and Figure 7 shows the latency increases linearly with injected CPU load, exceeding 5 s when the flight computer is 50 % busy with auxiliary tasks.

Although the board runs cool (50–56 C), Figure 8 reveals extreme thermal throttling: short boosts followed by idle dips, producing the long-latency tail[19] shown in Figure 5. Overall, the Pi violates both limits; an LTA UAV would respond seconds late, inducing limit-cycle oscillations.

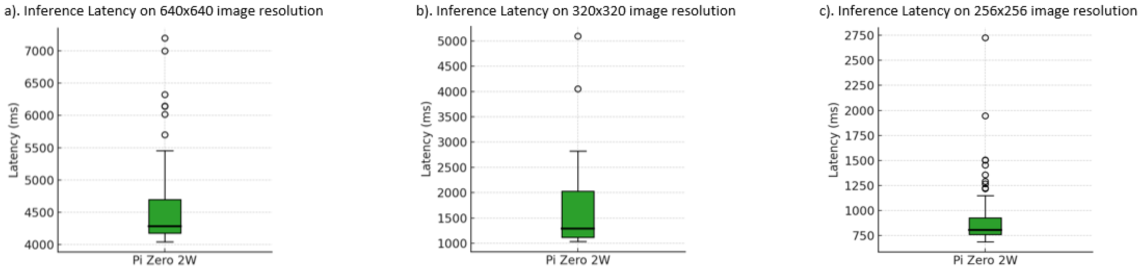

Figure 5: CDF of inference latency on the Raspberry Pi Zero 2 W at three input resolutions.

B. Embedded NPUs: Radxa Zero 3W vs. Khadas Edge 2

1. Nominal latency.

Khadas yields the lowest medians (17 ms at 256 × 256 and 20 ms at 320 × 320 (Figure 6)) well inside the 100 ms control budget. At 640 × 640 its 99th percentile reaches 170 ms, breaching the stability limit. Radxa is slower, with larger medians of 61 ms, 77 ms, and 227 ms for the three resolutions, yet still meets the ≤ 100 ms criterion. Its tail is shorter than that of Khadas at the same resolution, indicating more stable and predictable performance.

2. Temporal stability.

Latency on Radxa varies little, with CV values held between 6–7 % across all inputs and drifts only +6 ms over a 0–100 % CPU sweep (Figure 7). Khadas is three- to twelve-times spikier (CV = 54–76 %), and its p_{95}–p_{50} gap widens as the SoC warms (Figure 8).

3. Thermal margin.

Radxa completes every run near 55 C, which is well below its thermal-throttle threshold.[20] It even registers a slightly negative ΔT, indicating the NPU load is thermally invisible to the SoC. Khadas, by contrast, settles near 75 C and indicated heavy thermal throttling, explaining the 120–170 ms outliers visible in Figure 6. This causes the Khadas to require an additional weight budget for a heatsink in order to achieve maximum performance. Meanwhile, the Radxa performs consistently without one.

4. Net effect on LTA autonomy.

For 256–320 px inputs, Khadas offers unmatched raw speed but risks sporadic overruns at 640 px. Radxa responds more slowly yet remains predictably under 100 ms at the lower resolutions; its longer but consistent latency is better suited to maintaining stable control loops.

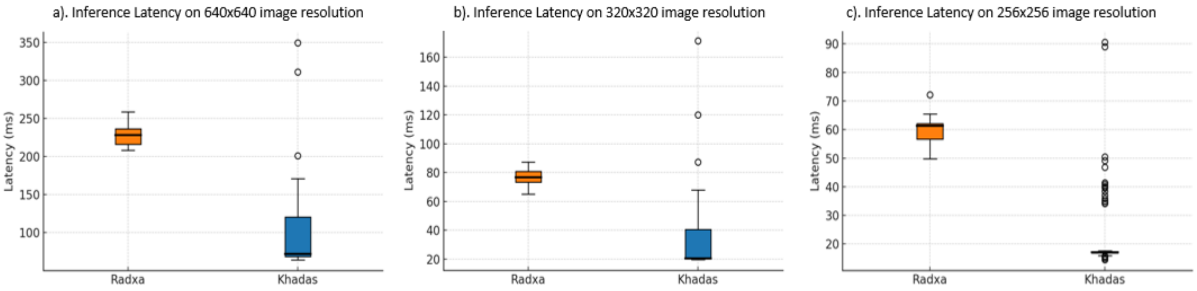

Figure 6: Latency distribution on the NPU-equipped boards (Radxa Zero 3W and Khadas Edge 2).

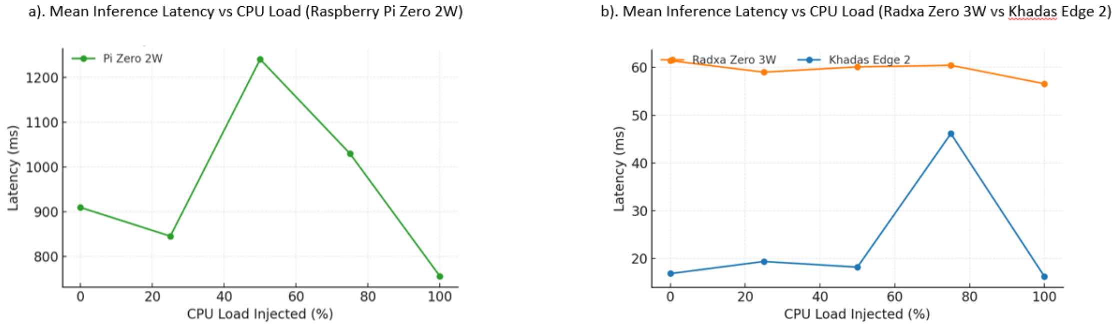

Figure 7: Mean inference latency under synthetic CPU load.

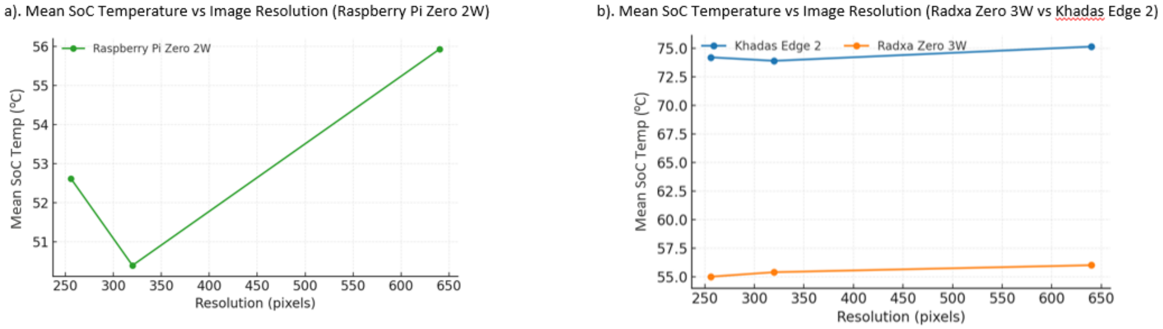

Figure 8: Mean SoC temperature after 60 s of inference at each input resolution.

C. Implications for LTA flight

Only the NPU-equipped boards satisfy both real-time criteria at 256–320 px inputs. The Raspberry Pi Zero 2 W, while extremely light (13 g), exceeds the p_{95} latency budget by roughly an order of magnitude, and although off-board Wi-Fi inference–with the raspberry pi as the edge device–can achieve sub-100 ms under ideal conditions ($\leq$ 100 ms[12]), the added variability from packet loss and airtime contention undermines its stability.

Latency wise, Khadas Edge 2 achieves the lowest median and often meets the 100 ms p_{95} bound at 256x256, but it runs hot without a heatsink: at 27 g it more than doubles the payload mass of the 13 g Radxa Zero 3W and operates close to its throttling threshold, which amplifies instability (CV 5–46 %). By contrast, Radxa Zero 3W (13 g) remains well below its thermal limit (CV 6–7 %), sustains $p_{95} \leq 100$ ms in all test resolutions, and imposes a minimal mass penalty (shown in Figure 8). Its combination of low weight, consistent latency, and thermal headroom makes it the safest and most balanced choice for closed-loop real-time control on buoyancy-limited LTA UAVs.

In summary: a CPU-only solution is untenable despite its light mass, Khadas maximizes throughput at the cost of both weight and thermal stability, and Radxa delivers the best trade-off of latency, jitter, and payload mass.

Table 3: Inference-latency statistics and end-of-run temperatures across 15 trials. Latencies are in milliseconds.

Board	Input	Median	p50	p95	p99	CV (%)	Temp (°C)
Raspberry Pi Zero 2 W	256x256	808	815	1340	1510	38	53
	320x320	1050	1070	1680	1830	41	54
	640x640	4300	4400	6120	6420	47	56
Radxa Zero 3W	256x256	61	63	86	95	6	54
	320x320	77	79	97	105	7	55
	640x640	227	230	260	280	7	55
Khadas Edge 2	256x256	17	18	32	38	54	74
	320x320	20	21	38	45	61	75
	640x640	45	46	150	170	76	75

V. System Design

The results in Section IV show that a lightweight NPU-equipped SBC with a small form factor can deliver sub-100 ms, low-variance inference without exceeding the modest thermal, weight, and power limits of LTA UAVs. This finding motivates a shift from our legacy architecture,[12] in which camera frames were streamed via Wi-Fi to a ground station for processing, and toward a fully self-contained design centered on the Radxa Zero 3W. By moving all perception and control inference on board, we eliminate key points of failure, such as dependence on the communication link, inference instability, and guarantee that the control loop always closes, even during brief telemetry dropouts.

A. On-board perception-control model

The perception control loop can be broken into three discrete processes. First, a camera is utilized over the standard Camera Serial Interface (CSI) connection. A CSI camera is used due to it's low Size, Weight, Power and Cost (Swap-C). A Python program then interfaces with the appropriate driver, and receives a low resolution image. This image is then pipelined into Radxa's package for neural processing unit (NPU) acceleration for object detection. Due to this process taking place onboard, there is no communication latency across the network, preventing any concerns regarding data dropout or time delay. The NPU sub-module then returns an array of classifications, confidences and bounding boxes for each object detected within the image frame.

Next, a state machine determines which object type should be targeted next, and filters detections appropriately. If in frame, the center of the object is computed using the four corners of the bounding box, and using the optical properties of the camera's lens, a relative angular error and relative altitude error are computed. Both errors are fed into PID controllers to control the vehicle's motors.

Finally, the bounding box, classification, and confidence are drawn on the image itself for human-interpretable real-time game monitoring. The image is converted into a compressed JPG image, and broken into packets to be streamed across a UDP socket on the network. In order to avoid any form of latency, the UDP protocol is used to bypass any retransmission. Without retransmission, data is continuously streamed at a quick and consistent rate, as real-time monitoring can be lossy and not impact the system's performance.

This process is repeated in a loop throughout the duration of a game, with the state machine dynamically selecting what object type should be targeted. This approach yields a consistent processing time, which is required for real-time autonomous system control.

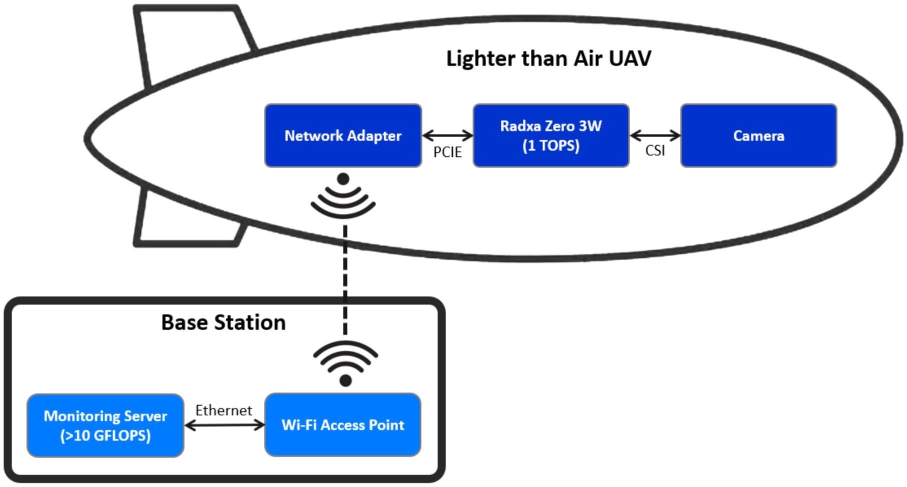

Figure 9: Visualization of system with communication link and estimated processing power.

VI. Conclusions and Future Work

This study confirms that a fully self-contained perception–control loop is now practical for lighter-than-air UAVs with Defend the Republic-competition constraints. Benchmarks in Section IV show that an RK3566-based Radxa Zero 3W sustains median inference latencies of 70–80 ms with a coefficient of variation below 7%, while stabilizing at 55 °C—well beneath its thermal-throttle threshold.[20] Because real-time control budgets cap round-trip latency at 100 ms[12] as stated in Section II, the board offers ample headroom and cleanly replaces the earlier off-board pipeline. By migrating vision, sensor fusion, and control entirely on-board, we eliminate the variability inherent to wireless round-trips and maintain stable flight even during multi-second telemetry drop-outs. These results establish on-board autonomy as a robust baseline for future Defend the Republic seasons and similarly resource-tight missions. Furthermore, this approach's independence from external computation or communication opens the door for its integration into fixed-wing, high-speed, maneuverable, or hybrid UAVs facing similar constraints, thus broadening its applicability beyond the DTR competition context.

Future work will involve thoroughly investigating quantization techniques and tailoring model optimizations for NPUs and SBCs. The promising performance of the NPU-based boards to handle higher loads also gives the green light to begin testing localization techniques. This will allow us to create advanced trajectories based on knowledge of objectives and obstacles in the environment. We will also employ advanced dataset augmentation and training strategies to improve object detection.[15]

References

[1] Messinger, S., *Modeling, Adaptive Control, and Flight Testing of a Lighter-than-Air Vehicle Validated Using System Identification*, Master's thesis, The Pennsylvania State University, 2022, Master's Thesis.

[2] Mathew, J. P., Karri, D., Yang, J., Zhu, K., Gautam, Y., Nojima-Schmunk, K., Shishika, D., Yao, N., and Nowzari, C., "Lighter-Than-Air Autonomous Ball Capture and Scoring Robot – Design, Development, and Deployment," *arXiv preprint arXiv:2309.06352*, 2023.

[3] Mendoza, A., Lovelace, A., Potter, S., and Koziol, S., "Sensor Fusion Image Processing for Autonomous Robot Blimps," *2023 IEEE 66th International Midwest Symposium on Circuits and Systems (MWSCAS)*, 2023, pp. 312–316.

[4] Simmons, J., Lovelace, A., Tucker, D., Mendoza, A., Coates, A., Alonzo, J., Li, D., Yi, X., Potter, S., Mouritzen, I., Smith, M., Banta, C., Hodge, R., Spence, A., and Koziol, S., "Design and Construction of a Lighter than Air Robot Blimp," *2023 ASEE GSW*, No. 10.18260/1-2-1139-46327, ASEE Conferences, Denton, TX, June 2024, https://peer.asee.org/46327.

[5] Xu, J., D'antonio, D. S., Ammirato, D. J., and Saldaña, D., "SBlimp: Design, Model, and Translational Motion Control for a Swing-Blimp," *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2023, pp. 6977–6982.

[6] Li, K., Hou, S., Negash, M., Xu, J., Jeffs, E., D'Antonio, D. S., and Saldaña, D., "A Novel Low-Cost, Recyclable, Easy-to-Build Robot Blimp For Transporting Supplies in Hard-to-Reach Locations," *2023 IEEE Global Humanitarian Technology Conference (GHTC)*, IEEE, 2023, pp. 36–42.

[7] Nojima-Schmunk, K., Turzak, D., Kim, K., Vu, A., Yang, J., Motukuri, S., Yao, N., and Shishika, D., "Manta Ray Inspired Flapping-Wing Blimp," *arXiv preprint arXiv:2310.10853*, 2023.

[8] Taylor, C. and Dantsker, O. D., "Lighter-Than-Air Vehicle Design for Target Scoring in Adversarial Conditions," AIAA Paper 2024-3896, AIAA Aviation Forum 2024, Las Vegas, NV. 2024.

[9] Dantsker, O. D., "Design, Build, and Fly Autonomous Lighter-Than-Air Vehicles as a Project-Based Class," AIAA Paper 2024-4375, AIAA Aviation Forum 2024, Las Vegas, NV. 2024.

[10] Dantsker, O. D., "Integrating Unmanned Aerial Systems into the Intelligent Systems Engineering Curriculum," Paper 2024-1185, Congress of the International Council of the Aeronautical Sciences, Florence, Italy. 2024.

[11] Taylor, C. and Dantsker, O. D., "Lighter-Than-Air-Vehicle Design for Adversarial Defense," AIAA Aviation Forum 2025, Las Vegas, NV. 2025.

[12] Taylor, C., Widjaja, M., and Dantsker, O. D., "Remotely-Processed Vision-Based Control of Autonomous Lighter-Than-Air UAVs With Real-Time Constraints," AIAA Paper 2025-1344, AIAA SciTech Forum 2025, Orlando, FL. 2025.

[13] Yuan, Z., Azzino, T., Hao, Y., Lyu, Y., Pei, H., Boldini, A., Mezzavilla, M., Beheshti, M., Porfiri, M., Hudson, T. E., Seiple, W., Fang, Y., Rangan, S., Wang, Y., and Rizzo, J.-R., "Network-Aware 5G Edge Computing for Object Detection: Augmenting Wearables to "See" More, Farther and Faster," *IEEE Access*, Vol. 10, 2022, pp. 29612–29632.

[14] Jayanth, R., Gupta, N., and Prasanna, V. K., "Benchmarking Edge AI Platforms for High-Performance ML Inference," *arXiv preprint arXiv:2409.14803*, 2024.

[15] Indiana University, Aerospace Systems Lab, "Defend The Republic Fall 2024," `https://www.iu-dtr.com/`.

[16] George Mason University, Patriot Pilots, "Defend The Republic Spring 2024," `https://www.sparx.vse.gmu.edu/`.

[17] Cox, B., Dantsker, O. D., and Deters, R. W., "Propulsion System Testing Instrumentation for Multi-Rotor and Lighter-Than-Air UAVs," AIAA Paper, 2025-0252, AIAA SciTech Forum 2025, Orlando, FL. 2025.

[18] Choe, C., Choe, M. K., and Jung, S., "Run Your 3D Object Detector on NVIDIA Jetson Platforms: A Benchmark Analysis," *Sensors*, Vol. 23, No. 8, 2023, pp. 4005.

[19] Raspberry Pi Foundation, *Raspberry Pi configuration: config.txt documentation*, Raspberry Pi, 2025.

[20] Radxa, *User Manual for Radxa Zero3*, Radxa, 2023.