AIAA SciTech Forum
6-10 January 2025, Orlando, FL
AIAA SCITECH 2025 Forum

10.2514/6.2025-1344

Remotely-Processed Vision-Based Control of Autonomous Lighter-Than-Air UAVs with Real-Time Constraints

Caeden Taylor,* Matthew Widjaja,† and and Or D. Dantsker‡
Indiana University, Bloomington, IN 47408

Abstract

This paper presents a remote computer vision processing system for autonomous lighter-than-air (LTA) vehicles that overcomes the computational limitations of onboard processing. The system utilizes a Raspberry Pi Zero 2W as the remote client and a high-performance computer as the remote processing server. The system achieves real-time object detection and control by transmitting images wirelessly to the server, which processes the images using a YOLOv8 model and returns the results to the client. The system is designed to balance detection accuracy, processing delay, and control responsiveness, and is evaluated in the context of the Defend The Republic competition. The results show that the system can detect objects of interest at over 10 frames per second, primarily limited by the communication network's bandwidth. The system's performance is analyzed in terms of image resolution, model size, network bandwidth, and image compression, and the trade-offs between these factors are discussed. The paper concludes that remote computer vision processing is a viable solution for autonomous LTAs, enabling the use of more advanced computer vision models and improving system performance.

Nomenclature

DTR	=	Defend the Republic competition
$FLOP$	=	floating point operations per second
$GPIO$	=	general purpose input output
$GFLOP$	=	billion floating point operations per second
IMU	=	inertial measurement unit
I/O	=	input output
LTA	=	lighter-than-air
PID	=	proportional integral derivative
SBC	=	single board computer
TCP	=	transmission control protocol
UAV	=	unmanned aerial vehicle
$YOLO$	=	You Only Look Once

I. Introduction

As the prevalence of autonomous unmanned aerial vehicles (UAVs) increase, there is a growing demand for systems with sufficient computational capabilities to complete dynamic tasks with minimal human interaction. While most sensors outputs are easily interpreted and utilized, computer vision requires complex algorithms to filter, identify, and track objects in real time in real-world applications.[1]

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

At the Defend The Republic competition, collegiate teams strive to create competitive lighter-than-air (LTA) vehicles with significant physical constraints in a head-to-head competition as an exploration into the research necessary for autonomous aerial vehicles. The targets are neutrally buoyant spherical objects (floating balloons) that must be identified, captured, and delivered to goals of different shapes, with the winner scoring the most goals with varying levels of autonomy.

To maneuver and sense the surrounding environment, LTA systems typically include inertial motion units (IMUs), barometers, ultrasonic range finders, cameras, and other rudimentary sensors for navigation. To detect, capture, and score the metallic mylar balloons, teams must utilize computer vision techniques to track the balloons to be captured over time.[2–11] Simpler computer vision techniques, such as color-based blob detection, struggle to accurately detect and differentiate all types of goals and balloons, particularly in dynamic lighting. More advanced computer vision methods, such as machine learning, yielded more reliable and accurate results.

This paper describes the process used to remotely process images for real-time system constraints across a wireless network with a Raspberry Pi Zero 2W single-board computer as the remote client onboard the LTA and a high-performance computer as the remote processing server. The resultant control scheme allows the system to detect objects of interest at over 10 frames per second, primarily being restricted by the communication network's bandwidth. The solution maintains the performance of the simpler computer vision methods, with the accuracy of machine learning models on a resource-constrained platform.

This paper is organized as follows: Section II describes the rules and major constraints of the DTR collegiate competition that motivates the work. Section III describes the communication protocol developed and used for remote computer vision processing. Section IV provides an analysis system parameters, such as image quality and model size. The paper concludes in Section V with a summary and statement of future work.

II. DTR Competition and Challenge

A. Competition Description

At the Defend The Republic (DTR) competition,[12] collegiate teams strive to use their autonomous lighter-than-air vehicle (LTA) vehicles (i.e., autonomous blimps) to autonomously capture helium balloons and score them into the opponent's goals–effectively Robotic Quidditch. 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, cyber-physical development of the entire system.

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

Each game has two 30 minute halves and 30 minute halftime. During each 30 minute half, there are six 5 minute intervals, with 30 seconds of manual blimp control, and 4.5 minutes for autonomous flight. In order to score points, teams must capture and score balls through goals placed on opposite ends of the field. Teams advance by scoring the most points during a game, with different point values based on the level of autonomy used to complete a task. Scoring a goal manually is worth 1 points. Autonomous scoring of a game ball, with manual assistance (capturing) is worth 3 points. And capturing and scoring the ball with uninterrupted autonomy is worth 10 points. This point scoring scheme heavily incentivizes autonomy.

The playing field consists of two sets of three uniquely shaped goals hung from the roof of the competition space at either end. Each set of goals have one triangle, square, and circle in either yellow or orange. The circle has an interior diameter of 36.5 inches, and an outer diameter of 44.5 inches. The square has an inside leg length of 38 inches, with there outside length being 46 inches. The triangular goal is an equilateral triangle hung upside down, with the total height from base to tip being 55 inches. The goals are made from plywood and retro-reflective tape in order to enhance visibility. Fig. 2 diagram presents a 2D birds-eye view of an example DTR match.

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,[13] and computation.

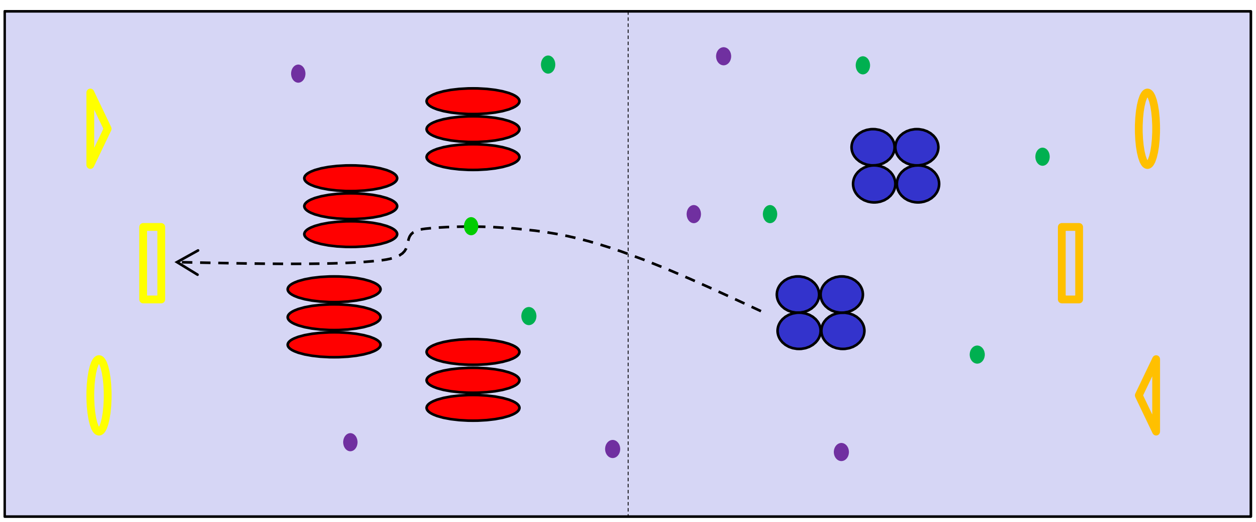
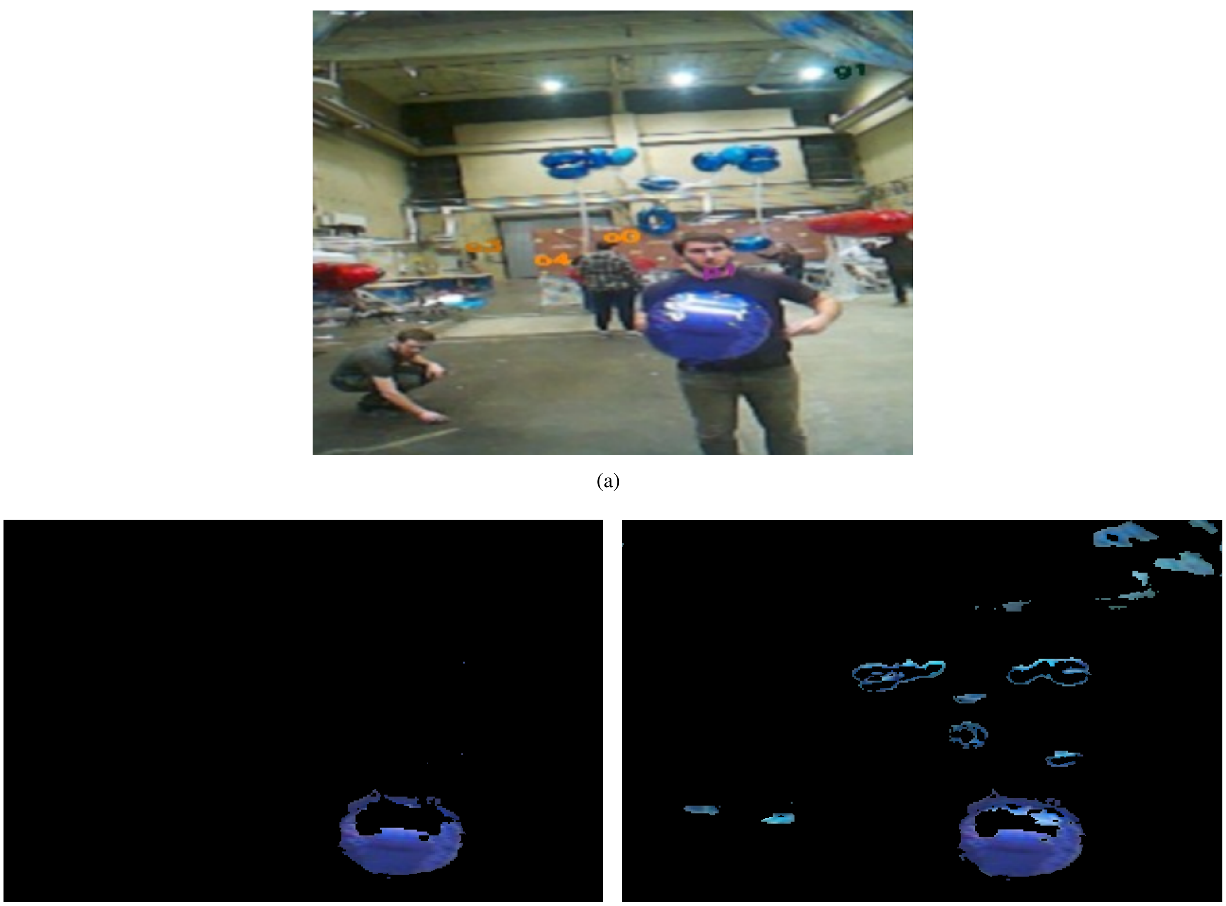


Figure 2. Birds-eye view representation of DTR match layout.

B. Computer Vision Challenge

In order to detect, capture, and score metallic mylar balloons in retro-reflectively taped goals, teams must utilize computer vision techniques to properly identify and track the position of game balloons over time. Simple computer vision techniques, such as color-based blob detection, struggle to adequately detect all types of goals and balloons, particularly in dynamic lighting. For example, Fig. 3 visualizes one flaw with color-based image filtering. Due to the overlap in the color spectrum between purple "goal" balloons, "blue" enemy blimps, and the blue sky, false positives were difficult to filter out. In sub-figure (a), the original image is shown. The sub-figures (b) and (c) show the effects of under-tuning and over-tuning the color filters, respectively. In order to improve game ball idenification and tracking, more advanced computer vision methods, such as machine learning, must be used and yield significantly better accuracy results. Model architectures such as single-shot detection Mobilenet, YOLO, and EfficientDet all work well to consistently identify all goals and game balls within sight.

The current LTA vehicles (as of Spring 2024 DTR) use a Raspberry Pi Zero 2W due to the required tradeoff of mass and computational power, as set by the game rules.[9] However, the limited memory and computational power of the Raspberry Pi Zero 2W leads to slow processing times and insufficient memory to handle the large datasets and computations required by machine learning models, often resulting in poor performance or even crashes. The lack of a graphics processing unit on the Raspberry Pi Zero 2W means that the model cannot leverage parallel processing capabilities, which are the main contributor for increasing the inference speed of deep learning models like YOLOv8.[14] Thus, to compensate for the weak computational performance, a remote image processing server and protocol were developed in order to transfer and run inference on the vehicle's image. The images are sent over WiFi to a high-performance server, returning the results in a threaded process using a single instance of the Ultralytics YOLO model.



(a) (b) (c)

Figure 3. A comparison of color-based image filtering: (a) Original unfiltered image, (b) over-tuned purple ball filter, and (c) under-tuned purple ball filter with enemy balloons.

III. System Design

Historically, variants of Single Board Computers (SBCs) are used as the main board for processing and interfacing with all other system components. SBCs typically provide a Linux-based operating system, with a select number of USB ports and a standard set of 40 general purpose input-output (GPIO) pins for interfacing with discrete sensors all in a form factor similar to a credit card. While these systems excel in convenience and weight, their computational performance is the primary bottleneck for computer vision tasking.

Despite the rapid advancements in mobile silicon performance and efficiency, these systems are still not optimized for the complex tensor operations required for complex computer vision models. Computational performance, typically measured in terms of floating point operations per second (FLOP's) can be used to estimate the performance of a given system with a machine learning model. For example, the You Only Look Once (YOLO) models can be quantified in terms of parameter size and number of operations per image inference. YOLOv8's smallest model size, the "nano" model, requires 8.7 billion floating point operations (GFLOP) per inference pass.[15]

Given the total FLOPs per second a system can process and the number of FLOPs a machine learning model uses, an onboard inference time can be estimated. The Raspberry Pi Zero 2W, the most common SBC used at the Defend the Republic Competition, is capable of processing roughly 5.1 GFLOPS per second. Given the YOLOv8 tiny model takes 8.7 GFLOPS, the system is only capable of processing less than 1 frame per second.

By shifting the required computation to a ground station server with no performance constraints, frames can be processed upwards of 60 frames per second. The primary bottleneck for the protocol shifts from on-board processing power to network bandwidth for image transfer. Fig. 4 visualizes the complete hardware system from the camera to the server with a WiFi connection for data transfer and estimated compute power of both the ground system and onboard the vehicle for comparison.

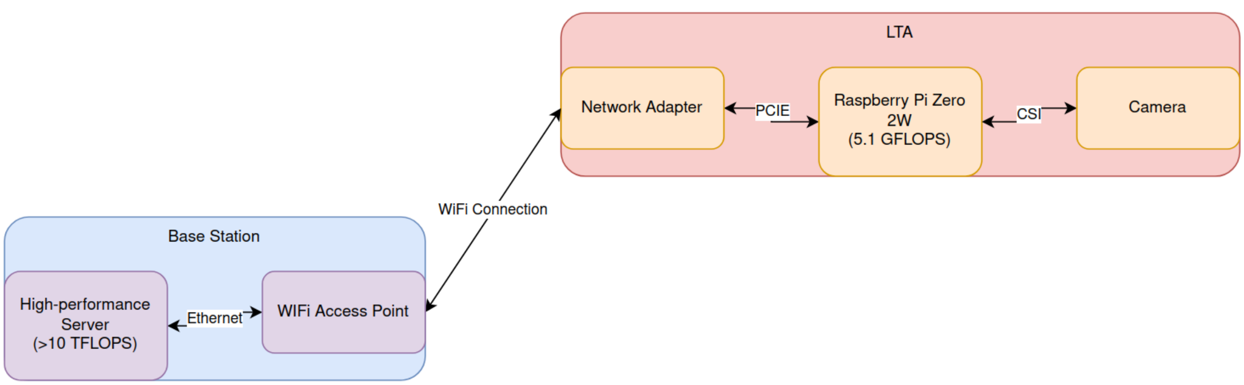


Figure 4. Visualization of system with communication link and estimated processing power

IV. Architecture Development

A remote image processing protocol was developed that takes 3 discrete steps per image for the real-time system. These steps are the client request, server process, and server response. Fig. 5 depicts a typical transaction between a client and a server to process and return the resultant data. The larger green blocks represent the data being passed over the WiFi link, with each colored inner block representing the type of data being passed and its format. The protocol's speed is primarily dependent on the current bandwidth between the client and server, with image transmission taking the bulk of the time for the request.

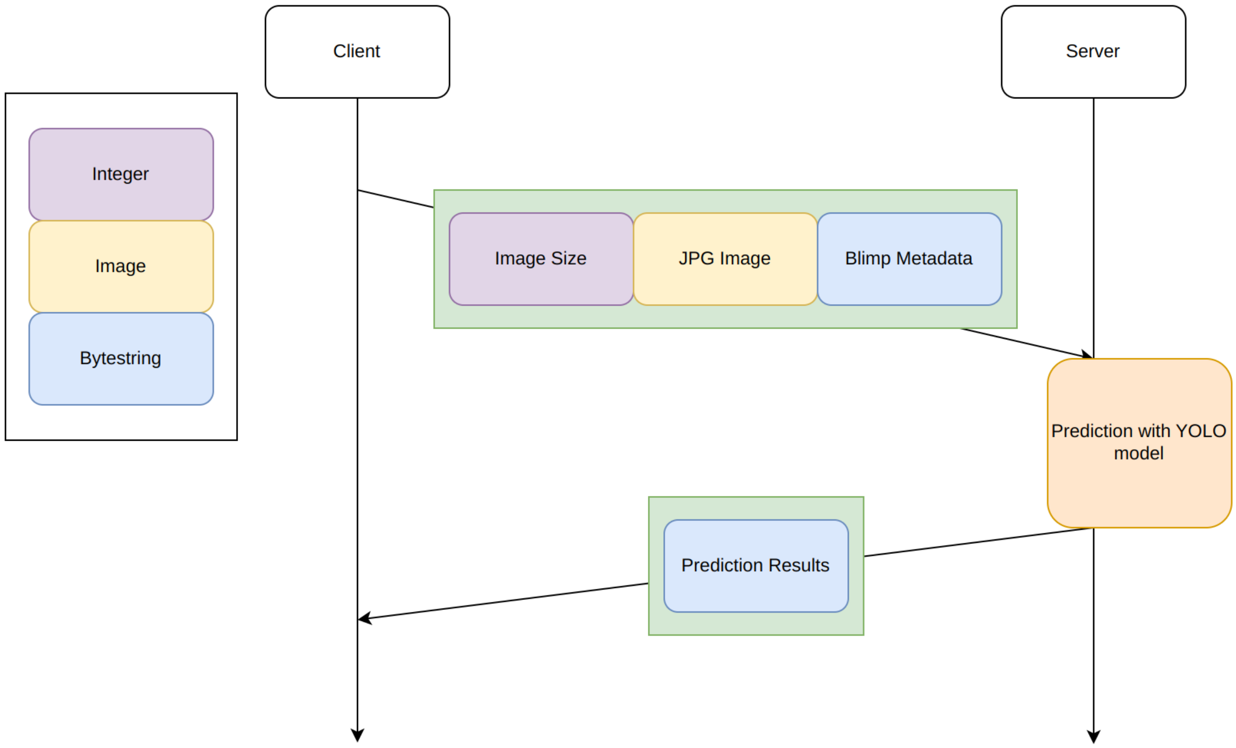


Figure 5. Communication Protocol Diagram representing what data is sent during a single frame of remote vision processing

The following subsections describe the specific protocols, formats, and details used for each step of the process applied for the real-time system. The following subsection will establish the system's minimum requirements and evaluate various common communication protocols. Next, the three main parts of the data exchange protocols are explained in detail for transmitting the image and returning the prediction results.

A. Selection and Justification of Wireless Communication Links

Based on the size constraints of the DTR competition venue, the communication protocol used must have a range of at least 200 feet in open air with a bandwidth to support our chosen image resolution at a rate of at least 10 frames per second. To estimate the required bandwidth of the communication link, four images at different resolutions were measured with JPEG and PNG compression, with the uncompressed file size for comparison. Table 1 presents the image formats that a wireless protocol should support transmitting at a rate of 10 times per second.

WiFi provides a balance between reliability, range, and throughput that is suitable for the requirements of the system. Real world throughput due to distance and interference was 2 Mbps and real-time controls require a system response rate of at least 10 frames per second to be processed. Thus, a resolution of 320 by 320 with 90% compression JPEG

Resolution:	96x96 pixels	160x160 pixels	320x320 pixels	640x640 pixels
Uncompressed (24bit/pixel):	27.65 KB	27.65 KB	307.2 KB	1.23 MB
PNG (Compressed 24bit/pixel):	5.36 KB	14.88 KB	59.51 KB	238.04 KB
JPG (90% compression):	940.9 B	2.61 KB	10.45 KB	41.82 KB

Table 1. Image Resolution and Compression Comparison

encoding was used to effectively utilize the communication bandwidth, accounting for any overhead from additional lower-level processes in the protocol, such as packet retransmission, or image reassembly.

B. Client Request

Once a network is established between the server and the vehicle(s), the vehicle connects to a Transmission Control Protocol (TCP) network socket created by the server. The first two bytes sent by the client create a 16 bit integer that represents the length of the following encoded image in bytes. Next, the client sends the image in bytes with JPG compression. This step is the most time-consuming part of the protocol due to the limited bandwidth of the WiFi link. Finally, the vehicle transmits its own metadata to the server, which includes its current state and estimated position.

C. Server Process Architecture

Once connected to the network, the server opens a TCP socket on a specified port, waiting for any client(s) to connect. A trained YOLOv8 model based on the "tiny" variant is used for object detection on six goal shapes/colors and both mylar balloon colors. The model is loaded in the global memory space once, and is used for all inference across all connected systems. Upon a new connection, a managing thread is created and waits for the client to send an image.

As the threads are still apart of the single running process, all threads can independently access the single loaded model in shared memory space, reducing the overhead associated for multiple models existing in memory. Due to the high compute power available on the ground, inference time was negligible to the overall system. Utilizing a single threaded process for managing multiple vehicles yields additional benefits, such as better scalability and centralized fleet management.

D. Server response

Once the thread receives the processed results of the image from the global model, results are returned as a byte string over the network to the client. The processed image is also shown on the servers display to effectively manage all clients.

V. Trade-off Analysis

In order to develop a real-time object detection and control framework, several key variables must be balanced to achieve optimal performance. The primary objectives are to maximize detection accuracy while minimizing processing and communication latencies to ensure responsive control. The main factors influencing these objectives include image resolution, YOLO model size, network bandwidth, and image compression. Table 2 summarizes the impact of these variables on system performance. The number of arrows indicates the relative magnitude of each effect, providing a visual representation of the trade-offs involved. Below, the affects of the factors are discussed, followed by the implementation performed on the LTA vehicle.

Parameter	Detection Accuracy	Proccessing Delay	Control Responsiveness
Image Resolution (↑)	↑↑↑	↓	↓↓↓
Image Resolution (↓)	↓↓↓	↑	↑↑↑
Model Size (↑)	↑↑	↓	↓
Model Size (↓)	↓↓	↑	↑
Bandwidth Availability (↑)	-	↑↑	↑↑↑
Bandwidth Availability (↓)	-	↓↓	↓↓↓
Image Compression (↑)	↓↓	↑↑	↑↑↑
Image Compression (↓)	↑↑	↓↓	↓↓↓

Table 2. Impact of changing parameters on detection accuracy, processing delay, and real-time control responsiveness. Arrows indicate the direction of impact: ↑ for positive, ↓ for negative, with the number of arrows representing the significance.

A. Performance Factors

1. Image Resolution

Increasing the image size enhances detection accuracy due to the higher resolution, which allows the system to capture finer details critical for object detection. These observations align with findings by Yan et al.,[16] which demonstrate that higher image quality significantly improves detection accuracy, particularly for objects at greater distances. However, larger image sizes result in significantly higher data volume, leading to increased processing and communication latencies.

Alternatively, reducing the image size decreases latencies and improves responsiveness, as smaller images require less time for processing and transmission. While this sacrifices some detection accuracy, this trade-off can be practical in latency-sensitive applications where rapid feedback is prioritized over precision.

2. Model Size

A larger YOLO model size improves detection accuracy by capturing more complex features and patterns in the input data. This capability makes larger models suitable for applications that demand high precision and detailed object recognition. However, the increased complexity of the model demands more computational resources, resulting in higher processing latency. Such latency challenges have been addressed in works such as Uddin et al.,[17] which emphasize the balance required between computational load and responsiveness.

Conversely, a smaller model size reduces processing latency, improving system responsiveness. While this approach limits detection capabilities, the streamlined performance may be advantageous in scenarios that prioritize speed and resource efficiency.

3. Bandwidth Availability

Higher bandwidth availability significantly reduces communication latency, enabling faster data transmission. This improvement in responsiveness is crucial for real-time object detection systems operating in resource-constrained environments. Yuan et al.[18] evaluated the impact of video bit rate and resolution on object detection accuracy and range, highlighting the importance of bandwidth in minimizing end-to-end delays. However, high bandwidth may not always be available in mobile or remote settings, limiting the applicability of bandwidth-intensive methods.

On the other hand, lower bandwidth increases communication latency, directly impacting responsiveness and overall system performance. Systems designed to function effectively under varying bandwidth conditions must incorporate adaptive communication strategies to ensure reliable operation.

4. Image Compression

Image compression reduces the size of transmitted data, decreasing both processing and communication latencies. This makes it an effective solution for improving responsiveness in real-time systems. For example, Galanopoulos et al.[19] analyzed the trade-offs of image compression levels on detection systems, showing its efficacy in latency-critical applications. However, high compression levels can significantly degrade image quality, affecting detection accuracy by obscuring important features.

Minimal compression, in contrast, preserves image quality and ensures accurate detection. Balancing compression to reduce latency while maintaining sufficient image quality remains crucial for systems that require both accuracy and real-time performance.

B. Implementation

For this effort, the detection requirements were tailored to the specific conditions of the DTR competition, which was held in an area approximately twice the size of a standard NCAA basketball court. The primary targets included large yellow and orange goals, distinguishable purple and green balloons, and small LTA vehicles. The relatively simple nature of these objects provided the flexibility to adjust resolution and model complexity without compromising detection reliability.

To optimize real-time performance, the “nano” variant of the YOLOv8 model was selected, paired with a 160x160 image resolution. The reduced model size minimized processing latency, aligning with the need for rapid decision-making during flight. Similarly, the low image resolution reduced communication latency, allowing for faster data transfer and improved responsiveness of the PID control loop. These choices reflect the trade-offs discussed earlier: lower image resolution and simplified model complexity were effective given the simplicity of the objects and the emphasis on real-time control.[19]

By balancing these factors, the system achieved a high frame rate, ensuring reliable detection and responsive vehicle control. This configuration successfully combined accuracy and latency trade-offs, delivering consistent performance within the constraints of the competition and the processing capabilities of the LTA platform.

VI. Conclusions and Future Work

Remote computer vision processing was successfully implemented using a Raspberry Pi Zero 2W as the remote client on a LTA vehicle used in the Spring 2024 Defend the Republic Competition. Multiple wireless protocols were compared, WiFi was chosen due to its high throughput, range, and onboard availability. Three main steps are used to remotely processing each frame, including the client request, server architecture, and server response. Moving the process off-board demonstrated substantial decreased inference time, being primarily constrained by the communication bandwidth.

Building on the findings of this effort, future work will explore methods to enhance system efficiency and adaptability, in preparation for future DTR competitions[20] as well as other efforts that rely on real-time computer vision. Training object detection models on compressed images offers a promising approach for reducing communication latency without compromising accuracy, as suggested by recent research.[21] Advanced architectures, such as YOLOv9 and YOLOv10, could also be investigated to improve detection performance, particularly when fine-tuned on application-specific datasets. A transition to fully on-board processing presents an opportunity to eliminate reliance on wireless networks, thereby increasing system resilience in environments with limited or unstable connectivity.

Additionally, optimizing network usage remains a critical consideration. Techniques like bandwidth-adaptive transmission and advanced compression strategies can ensure reliable performance under constrained communication conditions. Benchmarking YOLO variants across different edge devices and incorporating adaptive strategies, such as dynamic resolution adjustments, may further refine the balance between accuracy and latency. Additionally, developing models tailored to specific challenges, such as detecting small or fast-moving objects in complex scenarios, would enhance the system's versatility and make it more effective for diverse applications.

References

[1]Kanellakis, C. and Nikolakopoulos, G., "Survey on computer vision for UAVs: Current developments and trends," *Journal of Intelligent & Robotic Systems*, Vol. 87, 2017, pp. 141–168.

[2]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.

[3]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.

[4]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.

[5]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.

[6]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.

[7]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.

[8]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.

[9]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.

[10]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.

[11]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.

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

[13]Cox, B., Dantsker, O. D., and Deters, R. W., "Lighter-Than-Air Vehicle Design for Target Scoring in Adversarial Conditions," AIAA SciTech Forum 2025, Orlando, FL. 2025.

[14]Ultralytics, "Model Prediction with YOLOv8," `https://docs.ultralytics.com/modes/predict/`.

[15]Sohan, M., Ram, T., and Ch, V., *A Review on YOLOv8 and Its Advancements*, Springer, 01 2024, pp. 529–545.

[16]Yan, L., Qin, Y., and Chen, J., "Scale-Balanced Real-Time Object Detection With Varying Input-Image Resolution," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 33, No. 1, 2023, pp. 242–256.

[17]Uddin, M. N., Sakibul Islam Sakib, M., Nawer, S., and Mohona, R. T., "Improved Fire Detection by YOLOv8 and YOLOv5 to Enhance Fire Safety," *2023 26th International Conference on Computer and Information Technology (ICCIT)*, 2023, pp. 1–6.

[18]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.

[19]Galanopoulos, A., Valls, V., Iosifidis, G., and Leith, D. J., "Measurement-driven Analysis of an Edge-Assisted Object Recognition System," *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, 2020, pp. 1–7.

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

[21]Bhowmik, N., Barker, J. W., Gaus, Y. F. A., and Breckon, T. P., " Lost in Compression: the Impact of Lossy Image Compression on Variable Size Object Detection within Infrared Imagery ," *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, IEEE Computer Society, Los Alamitos, CA, USA, June 2022, pp. 368–377.