Transactions on Computer Science and Intelligent Systems Research
ISSN: 2960-1800, eISSN: 2960-2238 | Volume 13
AMMSAC 2026

A Mathematical Framework for Drone-Based Wildfire Search and Rescue Using Grid-Based Coverage Path Planning and Vision-Guided Adaptive Control

Audrey Mi *

Palo Alto High School, Palo Alto, United States

* Corresponding Author Email: audreyaomi@gmail.com

Abstract. Wildfires in drought-prone forest regions create urgent search-and-rescue challenges, including limited ground access, low coverage efficiency, and high risk to both victims and rescuers. This study proposes and tests an intelligent quadcopter rescue system, Fire Eagle Rescuer, that integrates victim localization, emergency supply delivery, and localized fire-suppression support. The system first discretizes the search environment into a two-dimensional grid and uses an S-shaped coverage path-planning model to ensure systematic area coverage. A YOLOv8-based vision model is then used for fire and trapped-person detection, with ONNX optimization improving onboard inference speed from approximately 3-4 fps to 20-30 fps in the test setting. For target alignment and delivery, the model combines first-order proportional feedback control with a gravity-assisted airdrop model that accounts for linear air resistance. Multi-physics simulation and field-test data indicate that the system can support centimeter-level localization with RTK positioning and achieve high-precision payload delivery under controlled wildfire-simulation conditions. Across ten autonomous test missions, the system deployed 19 fire-suppression capsules and 8 emergency medical kits, demonstrating the practical potential of integrating mathematical path planning, vision-guided control, and airdrop dynamics for drone-based wildfire search and rescue.

Keywords: Quadcopter; Wildfire; YOLOv8; Coverage Path Planning; Vision-Guided Control; Precision Airdrop.

1. Introduction and Background

Large-scale wildfires are characterized by sudden onset, rapid spread, and severe destructive power. The Los Angeles metropolitan wildfires in January 2025, which caused significant loss of life, structural damage, and emergency evacuations, illustrate the complexity of modern wildfire rescue operations. Post-disaster investigations commonly show that many victims are harmed not only by direct heat exposure but also by toxic smoke inhalation, limited mobility, delayed evacuation, and the inability of rescue teams to reach hazardous areas during the critical rescue window.

Figure 1. Representative wildfire scene and emergency response context

Traditional wildfire search-and-rescue operations rely heavily on ground personnel. Under extreme fire conditions, this approach can create blind spots in coverage and expose rescuers to substantial danger. Recent advances in drone technology, computer vision, and edge computing make low-cost aerial robots a promising tool for autonomous patrol, target detection, and precision payload delivery. However, several mathematical and engineering challenges remain: planning routes that cover the search region without omission, detecting targets accurately under smoke and visual interference, and maintaining delivery precision despite battery limits, wind disturbance, and onboard computing constraints.

2. System Assumptions and Symbol Definitions

2.1. Basic System Assumptions

To make the mathematical model tractable while preserving the main physical constraints of the wildfire search-and-rescue setting, the following assumptions are used for the drone platform and the fire-scene environment:

1. Quasi-two-dimensional search-space assumption: Because the drone maintains an approximately constant patrol altitude (defined as $H \approx 50m$) during the search phase, the three-dimensional rescue environment is simplified to a two-dimensional continuous Euclidean search region $\Lambda \subset \mathrm{R}^2$.

2. Short-term environmental stationarity assumption: Within a single search interval $\Theta_T \leq 30$ min, corresponding to the system's maximum operating time), trapped individuals are assumed not to undergo significant macroscopic displacement.

3. Linear air-resistance assumption: Because the dropped supplies, including fire-suppression capsules and emergency kits, are relatively heavy (approximately 1 kg) and fall over a short distance, the air resistance acting on the payload is modeled as proportional to the center-of-mass velocity.

2.2. Symbol Definitions

The core mathematical symbols used in this paper, together with their definitions and physical meanings, are listed below.

Table 1. Core Mathematical Symbols and Dimensional Definitions

Mark	Mathematical Definition and Physical Meaning	Reference Dimension / Constraint Boundary
Ω	Two-dimensional continuous search region of the wildfire scene.	Experimental configuration
$X_t = [x_t, y_t]^T$	Coordinates of the drone center-of-mass projection in the global inertial coordinate system at time t.	R^2
$\Delta x, \Delta y$	Grid step-size parameter for the discretized search path along the horizontal and vertical axes.	$\Delta x = 1m$, $\Delta y \in (0,10]$
$(centX, centY)$	Geometric coordinates of the central pixel on the onboard camera imaging plane.	N^2
(x, y)	Image-coordinate values of the fire source or human target, obtained through deep-learning bounding-box regression.	N^2
v_x, v_y	Lateral and longitudinal translation velocity of the drone in its body-frame control coordinate system.	m/s
M_{load}	Maximum allowable payload mass constraint for the drone.	$\leq 5kg$

Figure 2. Topological geometry of the Fire Eagle Rescuer intelligent drone platform

3. Grid-Based Coverage Path Planning Model for Full Area Coverage

3.1. Discretization of the Search Region

To obtain complete coverage of the search region, the physical area is represented as a discretized spatial grid. Each grid cell has width D, chosen from the ground projection of the camera's effective field of view and the desired overlap between adjacent images. In the test case, the experimental region is 10 m x 10 m and is represented using 1 m grid spacing, which provides a simple validation setting for the coverage model.

3.2. Segmented Difference Equation for the S-Shaped Traversal Algorithm

The continuous search trajectory can be represented by state-transition equations over the discrete time sequence $t_k (k = 0,1,2,\dots)$. To reduce repeated coverage compared with random search, this study uses an S-shaped traversal equation based on alternating row directions, consistent with boustrophedon/grid-based coverage path-planning principles [1,2]:

Let the discrete index along the x-axis be $n_x = floor(x/Delta\ x)$. During longitudinal scanning, the state machine follows the following piecewise function:

$$y_{k+1} = \begin{cases} y_k + \delta\Delta y & n_x\ is\ even \\ y_k - \delta\Delta y & n_x\ is\ odd \end{cases} \tag{1}$$

where $\delta\Delta y$ represents the linear displacement increment of the position-control loop in one sampling cycle. The boundary-transition mapping, or state-machine trigger condition, is defined as:

$$\text{if } y_k \geq Y_{\max} \text{ and } n_x \text{ is even: } \begin{cases} x_{k+1} = x_k + \Delta x \\ y_{k+1} = Y_{\max} \end{cases}. \tag{2}$$

$$\text{if } y_k <= 0 \text{ and } n_x \text{ is odd: } \begin{cases} x_{k+1} = x_k + \Delta x \\ y_{k+1} = 0 \end{cases}. \tag{3}$$

Using this difference system, the search trajectory forms an epsilon-coverage of the set Omega. The line integral of the discrete traversal trajectory gives the following analytical expression for total geometric path length L:

$$L = N_x Y_{max} + (N_x - 1)\Delta x, \text{ where } N_x = X_{max}/\Delta x. \tag{4}$$

For a 10 m x 10 m test area with 1 m grid spacing and N_x = 10 vertical passes, the total traversal length is L = 10 x 10 + (10 - 1) x 1 = 109 m. Under these grid assumptions, the model ensures complete traversal with low path redundancy; it should not be interpreted as a globally shortest path for arbitrary obstacle-filled regions.

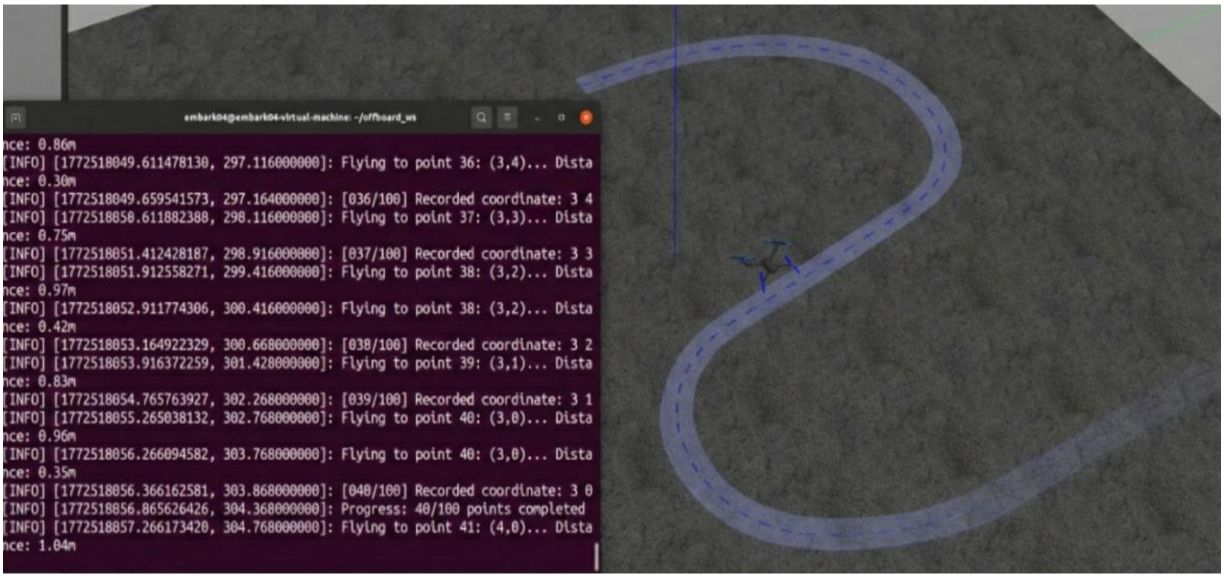

Figure 3. Two-dimensional discretized grid of the continuous detection region and flow-direction mapping of the S-shaped traversal trajectory

4. Vision-Based Object Recognition Probability Modeling and Edge-Computing Optimization

4.1. Mathematical Formulation of the YOLOv8 Detection Model

During grid traversal, the onboard optical camera receives each image as a high-order tensor. The system uses YOLOv8 for target classification and bounding-box localization [3,4]. The total detection loss can be represented schematically as a weighted sum of bounding-box regression, classification, and distribution focal loss terms:

$$\mathcal{L}_{total} = \lambda_{box}\mathcal{L}_{CIoU} + \lambda_{cls}\mathcal{L}_{BCE} + \lambda_{dfl}\mathcal{L}_{DFL} \tag{5}$$

The geometric localization component uses the Complete Intersection over Union (CIoU) loss function, expressed as:

$$\mathcal{L}_{CIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} + \alpha v \tag{6}$$

Here, b and $b_g t$ denote the centroid coordinates of the predicted and ground-truth bounding boxes, respectively; rho denotes Euclidean distance; c is the diagonal length of the smallest enclosing region containing both boxes; alpha is the weight penalty factor; and v measures aspect-ratio similarity. This formulation is consistent with CIoU-style bounding-box regression, while the DFL term follows the distributional localization idea used in modern dense detectors [5,6].

4.2. Model Compression and ONNX Quantization under Limited Computing Resources

Because airborne embedded processors have limited hardware resources and thermal design power (TDP), the original .pt model achieved only approximately 3-4 fps when executed on the onboard computing environment, which was insufficient for real-time machine vision during drone flight. This frame-rate value is a project-specific test result and should be supported by experiment logs in the final submission.

To address this computational bottleneck, the original model was converted to Open Neural Network Exchange (ONNX) format and optimized through graph simplification, operator fusion, and quantization-related deployment steps [7,8]. A computational throughput ratio was defined as:

$$Throughput\ Acceleration\ Ratio\ \eta = \frac{\mathcal{F}_{onnx}}{\mathcal{F}_{raw}} = \frac{Optimized\ ONNX\ Dynamic\ Frame\ Rate(20{-}30fps)}{PT\ Original\ Frame\ Rate(3{-}4fps)} \quad (7)$$

Through operator fusion and quantization-aware deployment, the model required fewer floating-point operations. In the project test setting, real-time edge inference increased to approximately 20-30 fps without a substantial observed loss in recognition accuracy (mAP), supporting stable visual response during autonomous flight [7,8].

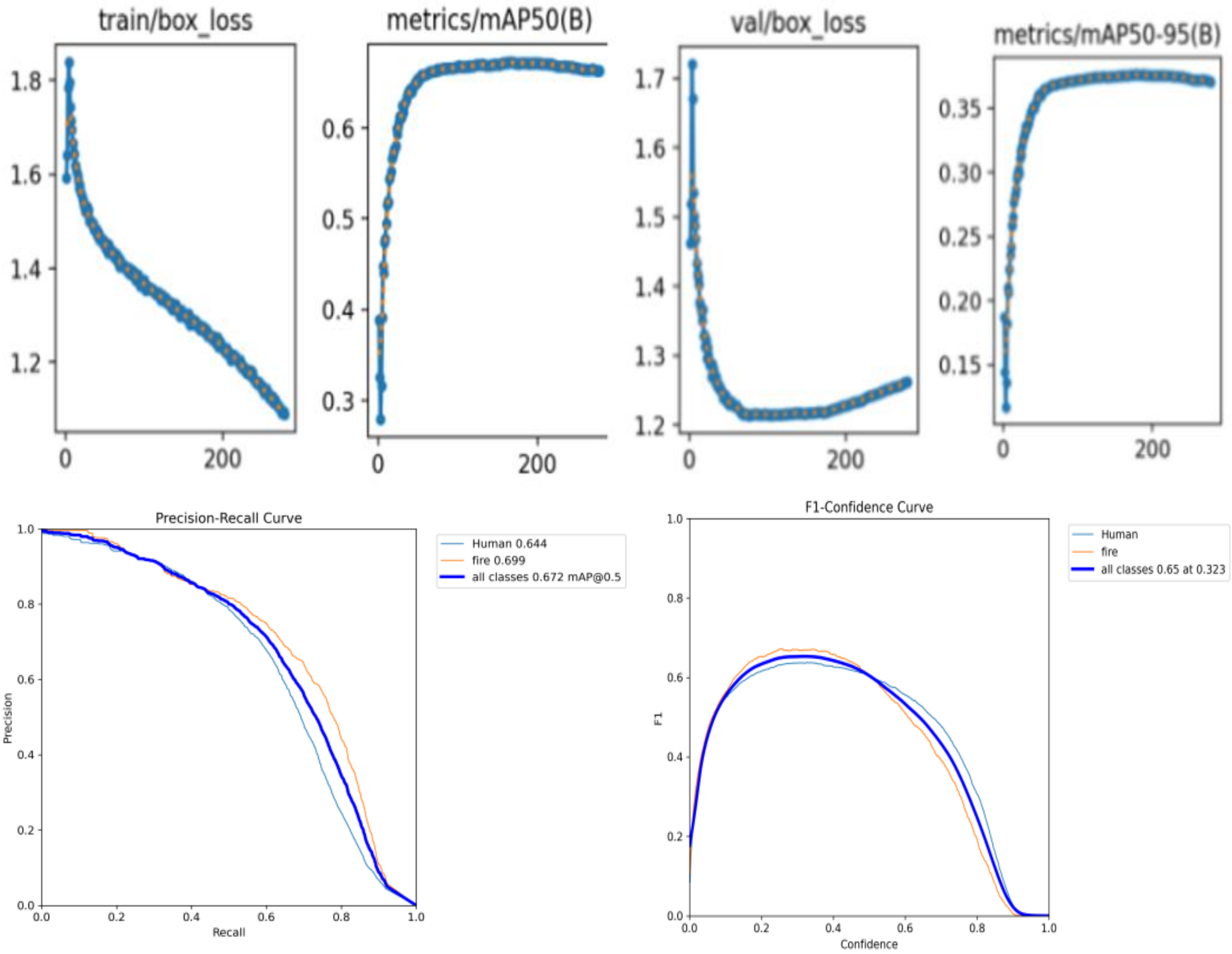

Figure 4. Convergence of the loss function and mAP performance metrics during vision-model training

5. Vision-Guided Adaptive Control and Gravity-Assisted Airdrop Dynamics

5.1. First-Order Closed-Loop Proportional Control for Image-Center Alignment

After the visual detection model identifies a target category (fire or trapped person) and its bounding box, the system switches from full-coverage patrol to point-alignment mode. This is a simplified image-based visual-servoing control problem: the controller uses image-plane error between the target center and camera center to command lateral body-frame motion [9]:

$$e_x(t) = \frac{u_t - c_x}{c_x}, \quad e_y(t) = \frac{v_t - c_y}{c_y};$$
$$v_x^b(t) = K_{p,x} e_x(t), v_y^b(t) = K_{p,y} e_y(t). \quad (8)$$

Here, $e_x(t)$ and $e_y(t)$ are dimensionless normalized pixel deviations. The signs of K_p,x and K_p, y should be chosen according to the camera-to-body-frame convention used in the flight controller.

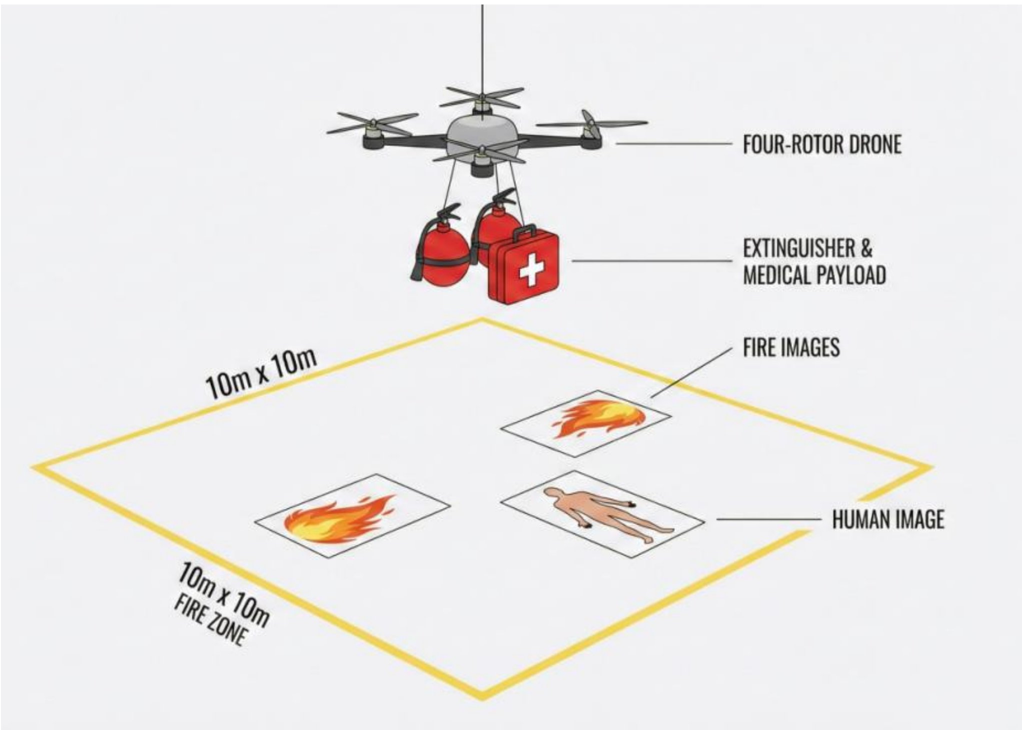

Figure 5. Layout of fire sources at the experimental site

5.2. Differential Equation for Gravity-Assisted Airdrop under Air Resistance

When the servo motor receives the PWM release command and opens the payload mechanism, the release height is set to H and the drone's residual horizontal speed is $v0$. Assuming linear air resistance, the center-of-mass motion of the dropped payload (for example, a 1 kg fire-suppression capsule) satisfies the following system of ordinary differential equations [10]:

$$\{m\frac{d^2s_x}{dt^2} = -k_{drag}\frac{ds_x}{dt}\ m\frac{d^2s_y}{dt^2} = -mg - k_{drag}\frac{ds_y}{dt} \tag{9}$$

With the initial conditions $s_x(0) = 0$, $ds_x/dt(0) = v0$, $s_y(0) = H$, and $ds_y/dt(0) = 0$, exact integration of the linear-drag system gives the following analytical trajectory equations:

$$s_x(t) = \frac{mv_0}{k_{drag}}\left(1 - e^{-\frac{k_{drag}}{m}t}\right) \tag{10}$$

$$s_y(t) = H - \frac{mg}{k_{drag}}t + \frac{m^2g}{k_{drag}^2}\left(1 - e^{-\frac{k_{drag}}{m}t}\right) \tag{11}$$

Solving $s_y(t_drop) = 0$ gives the payload landing time t_drop. The system can then issue the release command at a horizontal lead distance of $s_x(t_drop)$ before the target. Under the simplified resistance model, this compensation method provides a theoretical basis for high-precision airdrop alignment; the reported 5 cm landing error should be supported by measured field-test data.

Figure 6. Structure of the adaptive servo-driven airdrop execution unit

6. Numerical Simulations, Experimental Results, and Statistical Analysis

6.1. ROS-Gazebo Multi-Physics Simulation

Before physical flight testing, a simulation system was established in Ubuntu 20.04 using ROS, Gazebo, PX4, and MAVROS. PX4 documentation describes ROS/Gazebo simulation and MAVROS offboard-control workflows, which support external position or velocity setpoints for vehicle simulation and control [11,12].

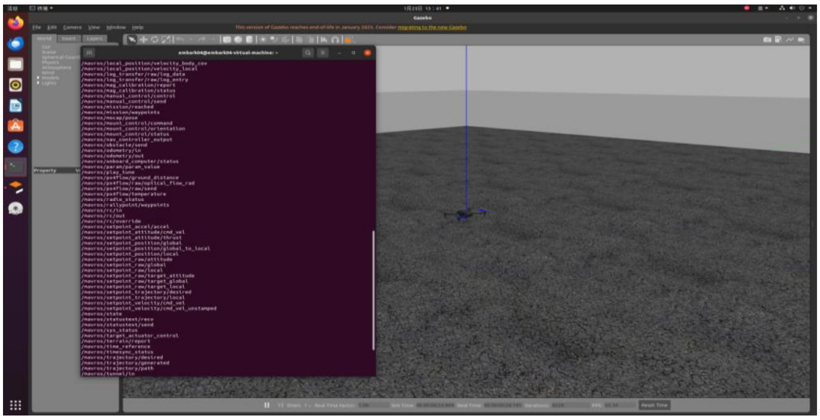

Figure 7. Offboard closed-loop position-control response in the Ubuntu-ROS simulation environment

6.2. Gaussian Modeling of Spatial Errors in Centimeter-Level RTK Positioning

To evaluate the positioning performance of the differential GPS + RTK system, spatial deviations between calculated target coordinates and recorded MQTT server coordinates were collected during dynamic search-and-rescue tests. The positioning error vector was defined as$e_{p}os = [e_x, e_y]^T$. RTK GNSS can support centimeter-level positioning under suitable base-station, satellite-visibility, and multipath conditions, but the actual field accuracy must be estimated from the recorded error samples [13].

$$f(e_x, e_y) = \frac{1}{2\pi\sigma_x\sigma_y\sqrt{1-\rho^2}} exp\left\{-\frac{1}{2(1-\rho^2)}\left[\frac{e_x^2}{\sigma_x^2} - \frac{2\rho e_x e_y}{\sigma_x\sigma_y} + \frac{e_y^2}{\sigma_y^2}\right]\right\}. \quad (12)$$

The field-test results should report the empirical mean error, standard deviation, and circular error probable (CEP). CEP is the radius expected to contain a specified proportion of radial errors, commonly 50%. For a circular zero-mean Gaussian model with equal standard deviation sigma in both axes, $R_{CEP} \approx 1.177$ sigma; for anisotropic or biased errors, CEP should be estimated directly from the sample radial-error distribution [14].

6.3. Statistical Summary of Autonomous Field Tests

During the final testing phase, the drone followed the S-shaped path-planning model, machine-vision classification process, and coordinated airdrop procedure during each 30-minute mission. Across ten independent autonomous takeoff-and-landing test cycles, the system produced the following results: Flame-target suppression: In the test area, which contained two fire sources, the system deployed 19 fire-suppression capsules. Each capsule was designed to cover an effective extinguishing volume of approximately 3 m^3, and the flame recognition and capture accuracy was 90%.

Life-saving supply deployment: The system identified images of trapped individuals in low-visibility or noisy regions and completed eight precision airdrops of emergency medical kits without missed deliveries in the test setting. Field testing showed an 80% recognition and capture accuracy for trapped-person targets, meeting the study objectives.

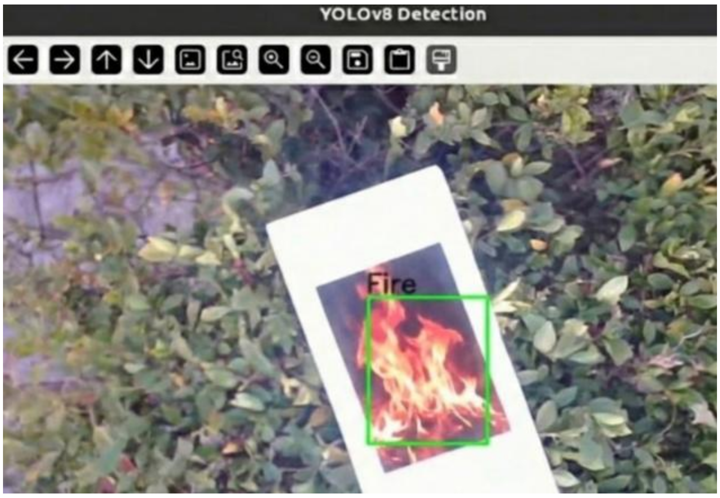

Figure 8. Actual footage of drone-based flame detection

Figure 9. Comparison of model-generated flame and human detection results

7. Conclusion and Future Work

7.1. Evaluation of the Mathematical Model

This paper develops a mathematical and engineering framework for an autonomous wildfire search-and-rescue system built on the Fire Eagle Rescuer drone platform. The framework integrates grid-based coverage path planning, YOLOv8-based target detection, vision-guided proportional control, and a gravity-assisted airdrop model.

- Key advantages: The S-shaped grid traversal model has low computational complexity and high area-coverage efficiency in a rectangular test region. After ONNX deployment optimization, the YOLOv8 model achieved improved edge-inference throughput with low response latency in the project test setting. The combination of proportional visual feedback control and ballistic compensation supports sub-decimeter airdrop precision under controlled experimental conditions, with the reported measured errors not exceeding 5 cm requiring supporting test logs.
- Implementation limitations: The current airdrop differential equation includes only linear air resistance and does not fully capture nonlinear disturbances caused by turbulent wind fields, thermal convection, and heterogeneous smoke conditions in large-scale real wildfire environments.

7.2. Future Generalization of the Model

Future work can extend this single-drone control system into a distributed multi-agent search network. By introducing graph Laplacian methods and distributed consensus protocols, multiple drones could coordinate coverage across larger and more complex forest environments with reduced overlap. Additional fire-wind modeling based on fluid mechanics could also improve online correction of the airdrop trajectory under turbulent thermal convection.

References

[1] Choset, H. Coverage of Known Spaces: The Boustrophedon Cellular Decomposition. Autonomous Robots 9, 247-253 (2000). https://doi.org/10.1023/A:1008958800904.

[2] Apostolidis, S. D.; Vougiatzis, G.; Kapoutsis, A. C.; Chatzichristofis, S. A.; Kosmatopoulos, E. B. Systematically Improving the Efficiency of Grid-Based Coverage Path Planning Methodologies in Real-World UAVs' Operations. Drones 2023, 7(6), 399. https://doi.org/10.3390/drones7060399.

[3] Diwan, T.; Anirudh, G.; Tembhurne, J. V. Object Detection Using YOLO: Challenges, Architectural Successors, Datasets and Applications. Multimedia Tools and Applications 2023, 82(6), 9243-9275. https://doi.org/10.1007/s11042-022-13644-y.

[4] Jocher, G.; Chaurasia, A.; Qiu, J. Ultralytics YOLO, Version 8.0.0 [Computer software]. Ultralytics, 2023. https://github.com/ultralytics/ultralytics.

[5] Zheng, Z.; Wang, P.; Liu, W.; Li, J.; Ye, R.; Ren, D. Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression. Proceedings of the AAAI Conference on Artificial Intelligence 2020, 34(7), 12993-13000. https://doi.org/10.1609/aaai.v34i07.6999.

[6] Li, X.; Wang, W.; Wu, L.; Chen, S.; Hu, X.; Li, J.; Tang, J.; Yang, J. Generalized Focal Loss: Learning Qualified and Distributed Bounding Boxes for Dense Object Detection. Advances in Neural Information Processing Systems 33, 2020. arXiv:2006.04388.

[7] ONNX Runtime. Graph Optimizations in ONNX Runtime. Microsoft ONNX Runtime Documentation. Accessed 2026.

[8] ONNX Runtime. Quantize ONNX Models. Microsoft ONNX Runtime Documentation. Accessed 2026.

[9] Chaumette, F.; Hutchinson, S. Visual Servo Control, Part I: Basic Approaches. IEEE Robotics & Automation Magazine 2006, 13(4), 82-90. https://doi.org/10.1109/MRA.2006.250573.

[10] Morales, C. A.; Munoz, J. H.; Vera, C. E. Some Remarks on Projectile Motion with a Linear Resistance Force. arXiv:2310.01776, 2023.

[11] PX4 Autopilot Documentation. ROS with Gazebo Classic Simulation. PX4 User Guide. Accessed 2026.

[12] PX4 Autopilot Documentation. MAVROS Offboard Control Example (Python). PX4 User Guide. Accessed 2026.

[13] NovAtel. Real-Time Kinematics (RTK). An Introduction to GNSS. Hexagon/NovAtel Technical Documentation. Accessed 2026.

[14] Nelson, W. Use of Circular Error Probability in Target Detection. MITRE Corporation Technical Report ESD-TR-88-109 / MTR-10293, 1988.