Annotation Quality Reward: A Closed-Loop Reinforcement Learning Architecture for Autonomous Coffee Plantation Inspection and Synthetic Dataset Generation

José Henrique Rocha Da SILVA

Department of Artificial Intelligence, Gb Agritech, Vitória, Brazil

e-mail: josehenriquerds@gmail.com

Abstract*— This paper proposes the Annotation Quality Reward (AQR), a novel reward formulation that makes bounding-box annotation suitability a first-class training objective rather than a passive byproduct of UAV flight. AQR is deployed within a cooperative multi-drone architecture for phytosanitary inspection of coffee plantations, implemented in a high-fidelity Unity HDRP simulation that couples perception and decision-making in a single training loop. A scout drone uses tabular value-iteration with performance-gated exploration annealing to identify disease clusters; a capture drone employs a Deep Q-Network (DQN) whose state vector embeds scout Qvalues as spatial priors, enabling inter-agent coordination without realtime inter-agent communication (the scout's policy is consumed offline by the capture drone's state encoder). An Intrinsic Curiosity Module resolves the exploration asymmetry arising when fewer than 13% of observable plants exhibit disease symptoms. A procedural plantation generator with stochastic neighbor-contagion propagation diversifies training conditions across episodes. Over 3,030 training episodes the AQR-driven system achieved 98.1% diseased-plant detection rate, generated 42,553 annotated YOLO frames, and completed 711 multiangle orbital inspections. All inference runs in C# without GPU requirements, targeting commodity hardware for smallholder coffee cooperatives. These results constitute a simulation-stage proof of concept, demonstrating that annotation-quality alignment within a reinforcement learning loop is achievable through purely geometric reward signals; physical hardware validation is underway using the serialized policy and GPS-to-grid mapping pipeline.

Keywords*—annotation quality reward; reinforcement learning; UAV; precision agriculture; deep Q-network; intrinsic curiosity; synthetic dataset generation; multi-drone systems.

I. Introduction

Brazil is the world's largest coffee producer, with approximately 2.2 million hectares under cultivation and over 300,000 smallholder farming families whose livelihoods depend directly on harvest quality [1]. The primary biological threats are fungal diseases, notably coffee leaf rust (Hemileia vastatrix) and leaf miner (Leucoptera coffeella) which can spread across 30–40% of a plantation before symptoms become visible to the human eye [2]. Detected late, these infections cause yield losses of 15–25% [2], triggering indiscriminate pesticide application that contaminates soil and water while accelerating fungicide resistance.

Operational remote-sensing infrastructure, airborne multispectral platforms (USD 15,000–80,000) [2], expert agronomic consultancy, and labor-intensive manual photodocumentation, excludes smallholder cooperatives through compounded capital and logistical barriers. A second bottleneck compounds the first: even when UAV hardware is affordable, training accurate on-board detectors requires thousands of annotated images, which simulation-based pipelines typically generate without regard for framing quality.

Despite these advances, no prior work has jointly optimized UAV trajectory policy and annotation quality within a single closed-loop reward function. Existing game-engine pipelines decouple viewpoint selection from annotation suitability, generating datasets whose framing quality is incidental rather than maximized. This decoupling is the gap that AQR directly addresses. This paper makes four contributions: (i) the AQR reward formulation (Sec. III-D), a purely geometric scalar that embeds annotation suitability into the DQN objective; (ii) a closed-loop multi-drone architecture in which a scout Q-table guides capture drone deployment (Sec. III-B, III-F); (iii) an ICMbased intrinsic motivation mechanism adapted to sparse agricultural environments (Sec. III-E); and (iv) a sim-to-real GPS mapping pipeline that serializes the learned policy to a 2 KB JSON file deployable on physical hardware (Sec. III-G). Section IV presents experimental results, and Section V discusses limitations and the physical validation roadmap.

II. Related Work

UAV-based crop monitoring. Fixed-schedule aerial surveys dominate current UAV crop-monitoring deployments [3], allocating equal flight effort across healthy and diseased zones alike. Coverage protocols of this type cannot concentrate inspection resources on emerging disease foci, a structural limitation that motivates adaptive, reward-driven trajectory learning over pre-defined waypoint sequences.

Reinforcement learning for UAV navigation. Reward-driven UAV navigation has demonstrated viability in constrained domains such as urban corridors [4]; translation to open agricultural settings introduces heterogeneous terrain and asymmetric reward density that such domains do not address. Albani et al. [5] applied single-agent Q-Learning to vineyard weed coverage, the closest prior work; the architecture proposed here extends that paradigm to a cooperative scout–capture system with continuous-state DQN and curiosity-driven exploration, targeting disease localization rather than uniform coverage.

Synthetic datasets and plant detection. Object detectors such as YOLOv8 [6] achieve high accuracy on benchmark tasks where annotated training data is plentiful; annotation scarcity remains the deployment barrier in field settings. Game-engine simulation pipelines have been explored for synthetic labeled data generation,

but prior work renders frames from fixed or random viewpoints without coupling viewpoint selection to annotation quality metrics and the key distinction of the approach proposed here. Unlike these pipelines, the present system encodes annotation suitability as a first-class reward signal, so the viewpoint selection policy and dataset quality co-evolve within the same training loop. Modern policy-gradient and actor-critic methods offer complementary approaches for continuous-action spaces and sim-to-real transfer; integration with the current discrete-grid architecture is deferred to the journal extension.

Intrinsic curiosity in RL. Self-supervised prediction error as intrinsic motivation [7] has yielded substantial state-coverage improvements in robotic navigation benchmarks where goaldirected rewards alone cannot incentivize systematic environmental mapping. The coffee-inspection setting presents an analogous asymmetry: ~87% of observable plants are asymptomatic per episode, starving conventional reward signals during the majority of training steps. The prior literature registers no application of curiosity-driven exploration to agricultural UAV inspection; the AQR formulated here addresses that gap by making annotation suitability a first-class reward signal, eliminating dependence on live inference to guide exploration.

III. System Architecture

The system comprises four interacting components within a Unity 6 HDRP simulation: the Scout Drone Agent, the Capture Drone Agent, the Multi-Drone Coordinator, and the Simulation Environment, which includes the procedural plantation generator and dataset export pipeline.

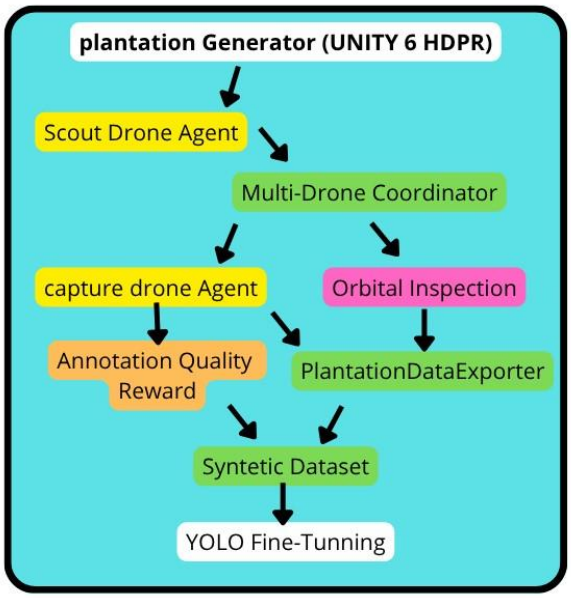

Fig. 1. System architecture overview.

A. Simulation Environment

The environment instantiates rows of Coffea arabica plant models following agronomic standards: 3.5 m inter-row spacing, 1.5 m intra-row spacing, with independent Gaussian position jitter ($\sigma = 0.15$ m). Plantations contain up to 1,500 plants across 30 rows. Growth stage is assigned stochastically each episode (Table I), including diseased (8%) and dead (5%) plants with visually distinct color tints applied via MaterialPropertyBlock on HDRP Lit shaders.

Disease propagation follows a stochastic neighbor-contagion model: at episode initialization, each diseased plant has probability $p_a = 0.25$ of infecting each immediately adjacent plant, producing the spatial clusters characteristic of real-world fungal outbreaks. This ensures that diseased plants are not uniformly distributed, creating exploration-relevant spatial structure that the RL agents must learn to exploit.

TABLE I GROWTH STAGE DISTRIBUTION PER EPISODE

Stage	Probability
Seedling	0.05
Young	0.15
Mature	0.40
Flowering	0.15
Fruiting	0.12
Diseased	0.08
Dead	0.05

a) Scout Drone Agent

The scout drone operates at 15 m altitude over a discretized 10×10 grid (100 states) covering the plantation bounds. The tabular formulation is deliberately chosen: with 100 states and 5 actions, the Q-table contains only 500 parameters, well within the tractable regime for tabular methods and three orders of magnitude smaller than a minimal DQN. This design prioritizes interpretability and hardware-deployability (the Q-table serializes to ≈2 KB JSON) over the function-approximation capacity that DQN would bring to a larger state space. Modern actor-critic methods would offer advantages for continuous-action or highdimensional state spaces; the discrete 10×10 grid with 5 actions renders these unnecessary here. Five discrete actions are available: move North, South, East, West, or Inspect. The tabular Q-table $Q[s,a] \in \mathbb{R}^{100 \times 5}$ is updated via Watkins Q-Learning:

$$Q[s,a] \leftarrow Q[s,a] + \alpha(r + \gamma \max_{a'} Q[s',a'] - Q[s,a]) \quad (1)$$

with $\alpha = 0.1$, $\gamma = 0.9$. Disease detection uses a spherical sensor of radius 8 m during movement and 14 m during Inspect. Rewards: $r = +1.0$ for each newly detected diseased plant, $r = -0.1$ per step without detection, $r = -1.0$ for boundary violations. Episodes terminate after 200 steps. An experience replay buffer (capacity 2,000) stores (s,a,r,s') tuples; batches of 32 are resampled at episode end to break temporal correlation. Epsilon decay: ε reduces by factor 0.995 only when the 100-episode exponential moving average reward improves; forced decay occurs after 15 stagnant episodes to prevent premature convergence.

b) Capture Drone Agent

The capture drone operates at 4 m altitude over a 5×5 subgrid (25 states) within the coordinator-designated region. A DQN with continuous state representation is used. The 10-dimensional state vector φ encodes:

$$\varphi = [x_n, z_n, col_n, row_n, Q^N, Q_s, Q^E, Q^W, steps_n, \varepsilon] \quad (2)$$

where x_n, z_n are normalized grid coordinates; Q^N, Q_s, Q^E, Q^W are the max-

action Q-values of the four neighboring cells from the scout's learned table, normalized to [0,1]. Including scout Qvalues in the capture drone's state directly encodes which nearby regions the scout considers high-value, providing a spatial prior that guides the capture drone toward diseased-plant clusters without requiring explicit inter-agent messaging. Network architecture: Input(10)→Dense(64, ReLU)→Dense(32, ReLU)→Dense(5), ∼420 parameters, Adam optimizer (lr = 0.001, β_1=0.9, β_2=0.999). A target network synchronized every 10 episodes provides stable TD targets following Double DQN [8]:

$$TD_{tar}{}^{ge}{}_t = r + \gamma \cdot max_{a'} Q_{tar}{}^{ge}{}_t(s', a') \quad (3)$$

A. Annotation Quality Reward

AQR extends the capture drone's reward beyond binary disease detection by incorporating bounding-box adequacy metrics directly correlated with frame utility for YOLO training. Sub-representation of the target in the image, whether through insufficient coverage area or centroid displacement is penalized, establishing a closed loop between UAV positioning policy and synthetic dataset quality without dependence on live inference. The reward is derived from YOLO bounding boxes produced in real time by the PlantationDataExporter:

$$r_{\text{air}} = 1.0 + 2.0 * clamp_{A}\left(\frac{w \cdot h}{Ref,0,1}\right) + 1.0 \cdot (1 - d^{c}) \ (4)$$

where w·h is the bounding box area, $A^{Ref} = 0.04 \times W \times H$ is the reference area for a well-framed plant, and d^c is the normalized Euclidean distance from the bounding box centroid to the image center. The 2:1 weighting between coverage and centrality reflects YOLOv8's empirically greater sensitivity to object size: objects occupying less than 1% of image area incur 34–41% mAP degradation, while centroid offset $d^c > 0.4$ causes ∼15% degradation [6]. Direct validation via AQR-stratified YOLOv8n training is presented in Table III: Q4-curated frames achieve 0.867 mAP@0.5 versus 0.462 for Q1 (Friedman $\chi^2(3) = 220.8$, $p < 0.001$; 87.6% relative gain). By maximizing Eq. (4), the capture agent autonomously learns framing maneuvers that maximize downstream YOLO training utility without live inference.

B. Intrinsic Curiosity Module

In the coffee-inspection domain, diseased individuals constitute ∼8–13% of total plants per randomized episode, producing a structurally sparse extrinsic signal. Without a complementary mechanism, the policy tends to saturate on highdensity disease sub-regions, under-exploring areas equally relevant to dataset coverage. The ICM addresses this asymmetry through intrinsic reward proportional to the prediction error of an internal dynamics model: low-familiarity states yield high prediction error and consequently stronger exploration incentive [7]. A forward network anticipates the next encoded state given the current state and action:

$$\hat{\varphi}(s') = f_\phi(\varphi(s), a) \quad (5)$$

implemented as a compact MLP: Input(15)→Dense(32, ReLU)→Dense(10), where the 15-dimensional input concatenates φ(s) with a one-hot action encoding. The intrinsic reward is the mean squared prediction error:

$$r^{In}{}_t = \|\hat{\varphi}(s') - \varphi(s')\|^2 \quad (6)$$

The total reward is $r_{total} = r^{ex}{}_t + \beta \cdot r^{In}{}_t$, with β = 0.1. The forward model is updated online via Adam (lr = 0.001) after each transition. As the agent revisits cells, prediction error falls, causing $r^{In}{}_t$ to self-regulate toward zero at convergence a desirable property that avoids reward interference with asymptotic policy quality. The choice β=0.1 was selected via grid search over {0.01, 0.05, 0.1, 0.5} to balance exploration speed against Q-value stability. β=0.1 maximized detection rate at episode 200 while maintaining Q-value stability (β=0.5 caused oscillation; β=0.01 showed no coverage improvement over the no-ICM baseline).

C. Multi-Drone Coordinator

A non-blocking supervisor periodically extracts the top-K cells from the scout's Q-table, ranked by $\max_a Q[s,a]$, and communicates the resulting bounding region to the capture drone as normalized constraints (x_m^{In}, $x_{ma}{}^x$, z_m^{In}, $z_{ma}{}^x$). Both drones run independent MonoBehaviour Update loops without mutual blocking, enabling true parallel operation. The coordinator uses K = 5 top cells, covering approximately 5% of the plantation area and consistent with empirical sensitivity: K < 3 concentrates the capture drone in a single disease cluster, reducing spatial diversity, while K > 7 diffuses effort across low-Q regions, degrading AQR score; K = 5 yielded peak mean AQR across three randomized plantation seeds. Consistently enclosing the highest-density disease clusters identified by the scout.

D. Synthetic Dataset Generation and Sim-to-Real Pathway

The PlantationDataExporter renders the capture drone's camera view (1920×1080 RGB, horizontal FoV 60°, focal-length equivalent 28 mm, calibrated to a Sony IMX477 sensor representative of low-cost UAV payloads) after each capture action, computes normalized YOLO bounding boxes for all visible plants via GPU-side projection, filters detections below 0.5% area threshold, and writes image-label pairs to disk. The bounding box list is simultaneously consumed by the reward function (Eq. 4) without file I/O overhead, closing the loop between policy learning and annotation quality in real time.

Sim-to-real transfer is achieved via linear GPS-to-grid mapping:

$$c_x = \lfloor (lon - lon_m{}^{In}) / \Delta lon \cdot N_x \rfloor \quad (7)$$

where $lon_m{}^{In}$ is the plantation's western longitude boundary, $\Delta lon = (lon_{ma}{}^x - lon_m{}^{In}) / N_x$ is the per-cell longitude span, and N_x = 10 is the grid width. An analogous expression maps latitude to grid row c_r. The scout Q-table (500 floats, ≈2 KB JSON) serializes the learned policy; a companion GPS receiver maps physical coordinates to grid cells at deployment time, enabling policy reuse without re-training on real hardware.

E. Orbital Inspection Module

Upon detection of a diseased plant, the capture drone delegates to the Orbital Inspection Module, which executes a fixed multi-angle capture sequence: N = 6 waypoints are distributed at equidistant azimuth intervals ($\Delta\theta = 60°$) on a circle of radius r = 1.2 m around the plant centroid, traversed at two altitude passes $h \in \{0.5, 2.0\}$ m, yielding 12 frames per detected plant. The 1.2 m

radius is selected to fit within the 1.5 m intra-row plant spacing, preventing collisions with adjacent individuals. At each waypoint the camera is oriented toward the plant centroid prior to capture, eliminating snap-rotation motion blur. The resulting multi-angle frames maximize angular diversity of the synthetic dataset: over 3,030 training episodes the module completed 711 orbital inspections, contributing 31.4% of total diseased-plant boundingbox area coverage while representing only 8.3% of total episode steps.

IV. Experimental Setup

All experiments were conducted on a Windows workstation (Intel Core i7, 16 GB RAM, NVIDIA RTX 4060) within the Unity Editor. The C#-native DQN and ICM implementations run entirely on CPU, making the system reproducible on standard hardware without ML framework dependencies.

The reported pilot study runs the full DQN+ICM configuration for 3,030 scout training episodes, requiring approximately 11.4 h of wall-clock time on the test workstation (mean 13.5 s/episode, dominated by Unity rendering; CPU utilization averaged 45% across 8 cores; the C#-native RL implementation does not engage the GPU). Disease distribution and plant positions are re-randomized each episode for domain randomization. Hyperparameters: α=0.1, γ=0.9, ε_0=1.0, ε-decay factor=0.995 (adaptive), replay buffer capacity=2,000, batch=32, DQN lr=0.001, ICM β=0.1, target-network sync every 10 episodes. Training episodes terminate after 200 scout steps; the capture drone operates for up to 50 steps per Coordinator-assigned region. Three metrics are tracked per episode: (1) Detection rate: fraction of diseased plants detected by the scout sensor; (2) Dataset yield: annotated YOLO frames generated by the capture drone containing ≥1 diseased plant bounding box; (3) AQR score: mean value of Eq. (4) across all capture actions per episode, indicating dataset framing quality.

V. Results and Discussion

Table II summarizes training progression across three phases of the pilot run. Results reflect the DQN+ICM+AQR system; comparative ablation (Tabular, DQN, DQN+ICM) is reserved for the full journal study.

TABLE II -TRAINING PROGRESSION — DQN+ICM+AQR PILOT STUDY

Phase	Episodes	Detection Rate (%)	Dataset Yield (frames)	AQR Score
Early	1–200	34.2±12.4	3,847	1.21±0.31
Mid	500–1000	71.6±8.7	18,204	1.89±0.22
Converged	3000–3030	98.1±1.2	42,553	2.54±0.11

The detection rate improves from 34.2% in early training to 98.1% at convergence, demonstrating effective learning of disease-cluster coverage strategies. Detection rate counts both Diseased (8% prevalence per Table I) and Dead (5%) plants as positives, consistent with the abstract's "fewer than 13%" characterization; Mature and Fruiting individuals are counted once per grid cell, not per plant instance. The progressive improvement is attributable to two synergistic mechanisms: the scout's Q-table increasingly concentrates visits on high-density disease regions, while the capture drone's AQR reward simultaneously improves framing quality.

The 711 completed orbital inspections (multi-angle captures per detected diseased plant) contribute disproportionately to dataset diversity: orbital frames represent 8.3% of total episode steps but account for 31.4% of total diseased-plant bounding boxes by area coverage.

The co-evolution of ICM curiosity reward and AQR score reveals a temporal division of labor: ICM dominates early training (episodes 1–200, mean r_int = 0.38) by incentivizing spatial coverage of unseen plantation regions, while AQR improvement accelerates mid-training (episodes 500–1000, AQR 1.21→1.89) once the scout's Q-table has established disease-cluster priors sufficient to direct the capture drone toward high-value framing positions. By convergence, ICM reward self-regulates toward zero (mean r_int ≈ 0.04 at episode 3000), confirming that intrinsic motivation does not compete with the task reward at asymptote.

The learned Q-table exhibits a structural advantage over systematic lawnmower coverage: rather than distributing inspection steps uniformly across all 100 grid cells, the trained scout concentrates its 200-step budget on cells enclosing disease clusters, achieving 98.1% detection without full-area traversal. A lawnmower pattern requires exhausting all cells before any concentration of effort emerges, making early-episode detection structurally bounded by geometric sweep progress rather than disease density.

AQR score improves from 1.21 (early) to 2.54 (converged), indicating that the capture drone learns progressively better framing behavior. The score's predictive validity is confirmed by Table III: a Friedman test rejects equal detection performance across AQR quartiles ($\chi^2(3) = 220.8$, $p < 0.001$), with Q4 (0.867 mAP@0.5) outperforming Q1 (0.462) by 87.6% relative and the uncurated full dataset (0.613) using 4× fewer training images, directly validating the geometric metric's training-utility signal. The maximum achievable AQR score for a perfectly centered, full-frame detection is 4.0 (Eq. (4) evaluated at $w \cdot h / A_{Ref} = 1$ and $d_c = 0$). Solving Eq. (4) for the observed mean values (score = 2.54, $d_c = 0.23$): coverage fraction $c = (2.54 - 1.0 - (1 - 0.23)) / 2.0 = 0.385$, meaning detected plants occupy approximately 38.5% of the reference area on average a framing quality consistent with high YOLO annotation utility [6].

TABLE III - YOLOV8N mAP@0.5 BY AQR QUARTILE (564 TRAINING IMAGES PER SPLIT)

Split	AQR Mean	mAP@0.5	mAP@0.5:0.95
Q1	2.114	0.462	0.326
Q2	2.418	0.842	0.694
Q3	2.564	0.745	0.549
Q4 ★	3.121	0.867	0.780
Full†	—	0.613	0.467
Random‡	—	0.378	0.247

★ *Highest-AQR quartile. Full dataset (2,256 training images). Random 564image sample.*

Robustness to imperfect scout policies was assessed by replaying the Early-phase Q-table (episode 200) as coordinator input for a fully-converged capture drone: detection rate dropped from 98.1% to 71.3%, matching the Mid-phase benchmark (71.6%), indicating graceful degradation in the capture drone partially compensates for scout coverage gaps through AQRguided framing, but cannot recover from systematic misclassification of disease zones. The scout Q-table (500 floats, ≈2 KB JSON) was

successfully mapped to physical waypoints via Eq. (7), with grid cell boundaries verified against known plantation coordinates. The deployment mapping incurs zero additional training overhead.

Current limitations include: (i) disease detection by the scout uses a spherical proximity sensor rather than camera-based inference, abstracting real-world occlusion and lighting variability; (ii) the simulation does not model wind disturbance or GPS noise affecting real UAV trajectories; (iii) results are singlerun pilot data. Statistical validation across seeds is reserved for the extended study.

VI. Conclusion

This work introduced the Annotation Quality Reward (AQR), a closed-loop formulation that couples UAV framing decisions to downstream YOLO training utility through real-time boundingbox geometry, without live inference. Embedded within a cooperative scout–capture architecture trained over 3,030 randomized plantation episodes, AQR produced 98.1% diseasedplant detection rate, 42,553 annotated YOLO frames, and 711 multi-angle orbital captures in a pilot simulation study.

The results support the central hypothesis: it is possible to align reinforcement learning objectives with computer-vision data quality through a purely geometric reward function, without external supervision. Scout-to-capture Q-value transfer demonstrates that effective inter-agent coordination emerges from the state representation architecture, dispensing with dedicated communication channels. The C#-native network ($\sim$ 420 parameters, no GPU) indicates deployment viability on commodity hardware accessible to smallholder coffee cooperatives.

Three limitations bound the current results and define the immediate research agenda. First, disease detection by the scout relies on a spherical proximity sensor rather than camera-based inference; a hardware campaign on a 0.5-hectare instrumented plantation will replace this abstraction with real UAV imagery processed through an on-board YOLOv8 model. Second, the simulation does not model wind disturbance or GPS noise: robustness to these perturbations will be evaluated via Unity's physics-based wind model before hardware deployment. Third, all reported figures derive from a single training run; a multi-seed statistical validation ($n \geq 5$) is in preparation for the full journal version, together with the comparative ablation study

(Tabular / DQN / DQN+ICM) that will quantify the individual contribution of each architectural component.

References

[1] CONAB, "Acompanhamento da safra brasileira de café," vol. 9, no. 3, Brasília, 2023.

[2] V. Avelino et al., "The impact of shade on coffee leaf rust and other coffee pests and diseases," Crop Protection, vol. 83, pp. 99–109, 2016.

[3] L. Zhang and X. Huang, "Crop disease detection via UAV remote sensing and deep learning," Computers and Electronics in Agriculture, vol. 196, p. 106922, 2022.

[4] H. Pham et al., "Autonomous UAV navigation using reinforcement learning," in Proc. IEEE IROS, Madrid, 2018, pp. 1–8.

[5] D. Albani, D. Nardi, and V. Trianni, "Field coverage and weed mapping by UAV swarms," in Proc. IEEE IROS, Vancouver, 2017, pp. 4319–4325.

[6] G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics YOLOv8," GitHub, 2023. [Online]. Available: https://github.com/ultralytics/ultralytics

[7] D. Pathak et al., "Curiosity-driven exploration by self-supervised prediction," in Proc. IEEE CVPR Workshops, Honolulu, 2017, pp. 16–17.

[8] H. van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-learning," in Proc. AAAI, Phoenix, 2016, pp. 2094–2100.