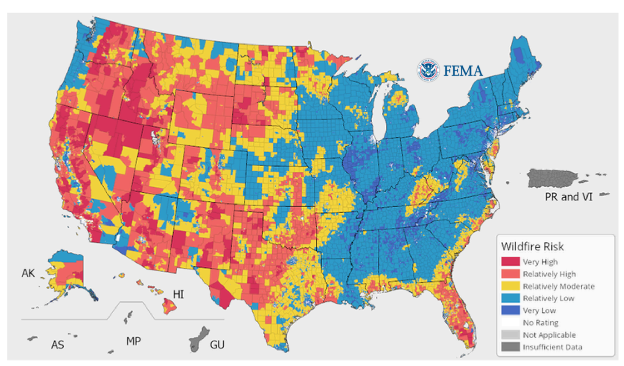
A2D2: An AI-driven Autonomous Drone-based Detection Network for Wildfires

Calvin Zhou, Sophia Yang

Introduction and Problem

Background

- 66k fires burned **7.5M acres** nationwide in 2022 - NIFC
- Annual total cost of wildfires in the U.S. ranges from $394 to **$893 billion** - Congress JEC



Problem Definition

- Current wildfire detection systems are ineffective
  - Stationary camera and sensor networks, satellite
  - Limitations: adverse weather, limited view angles, uncertainty due to distance, speed of detection
- **Drones/UAVs require** significant human efforts due to **manual operations**, impacting their efficiency and restricting usage
  - Manual path planning, human piloting, no autonomous detection

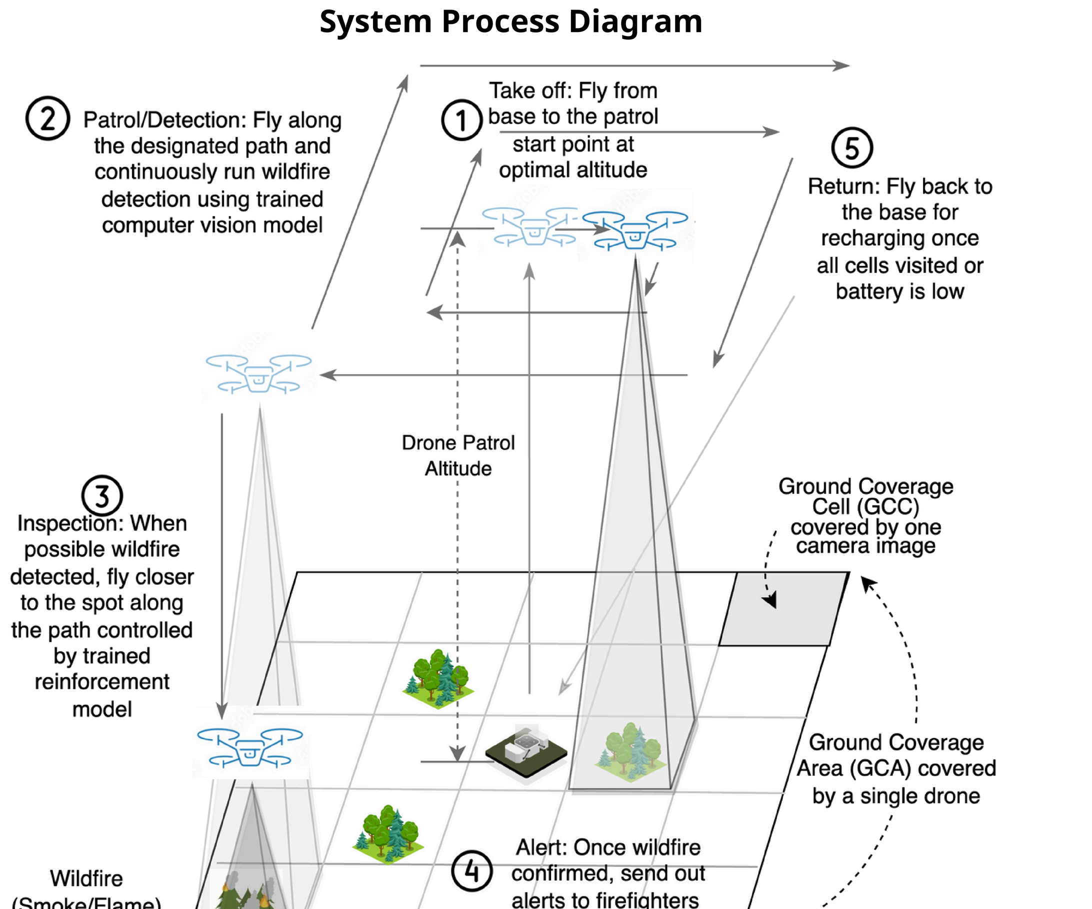
Previous Work

- Deep Computer Vision (DCV) for wildfire detection
  - YOLOv3, v5, v7, v8, Reduce-VGGnet with optimized CNN, FireDETN, Efficient-B5 and DenseNet-201
- Deep Reinforcement Learning (DRL) for object tracking
  - Actor-Critic methods, Deep Deterministic Policy Gradient (DDPG)
- Coverage Path Planning (CPP) for UAV flight paths
  - Motion planning algorithms, path planning for polygon areas, Waypoint planning, cellular decomposition, coverage trajectories for irregular fields, Energy-Aware CPP
- **No previous work has integrated all 3** into a single **drone system** that automates the whole operational process to deliver effective early wildfire detection with minimal human input

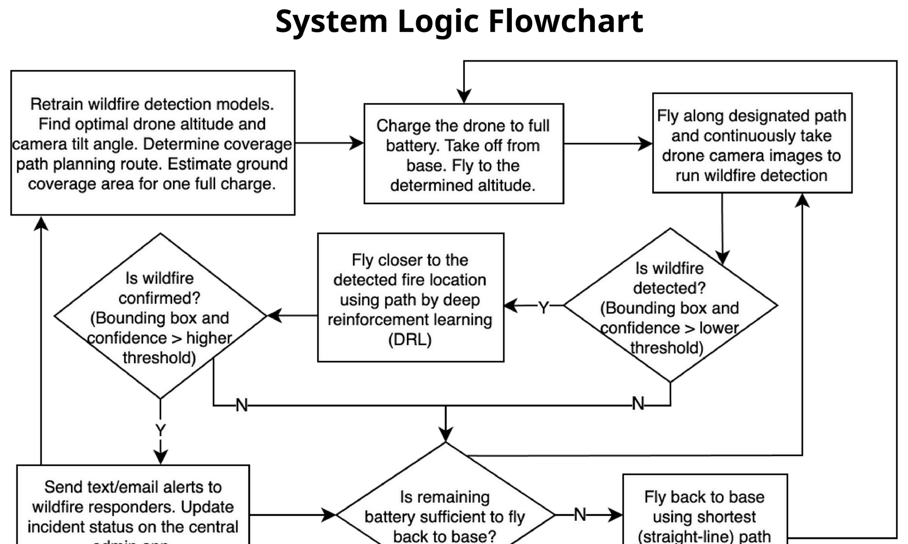
Engineering Goal

- Design and implement **a system fully automating the 5-stage** operational **process** for **drone-based wildfire detection**
- Develop **algorithms** and **models** required in the system flowchart steps
- Conduct **end-to-end simulation tests** to validate the system design

System Process Diagram



System Logic Flowchart



Methods and Procedures

Estimation of Ground Coverage Area (GCA) Size

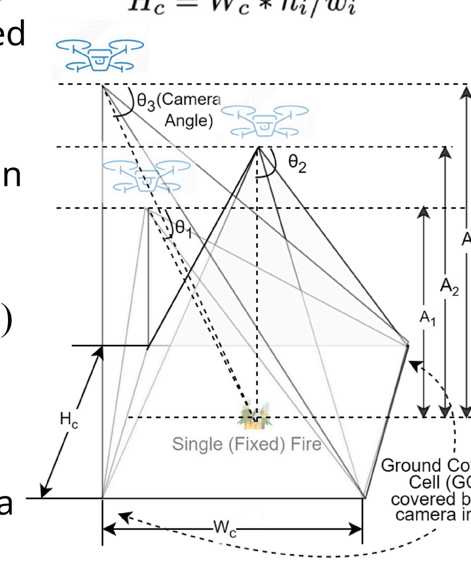
- GCA is **the ground area** that a single drone can patrol on a full charge
  - Calculated from the size and number of **Ground Coverage Cells (GCC)**, which is the **projection** of the ground cell to a camera image
- When the camera faces directly downward at an angle θ of -90° and altitude of A, the GCC width W_c & height H_c are given by:
  - Where w_i and h_i are the image width & height

$$W_c = 2 * A \qquad H_c = W_c * h_i / w_i \tag{1}$$

- When θ is not −90°, the drone needs to be placed at an **offset** for the camera to cover the same ground area as a down-facing camera
- Horizontal and vertical offset ϕ_x and ϕ_y are given by:

$$\phi_x = \frac{W_c}{2} + A * tan(90 - abs(\theta) - \frac{\alpha}{2}) \qquad \phi_y = \frac{H_c}{2} + A * tan(90 - abs(\theta) - \frac{\beta}{2}) \tag{2}$$

  - Where θ is the camera tilt angle (-90° to 0°) from the horizontal, α and β are the camera horizontal and vertical **field of view** (FOV)



Grid Search for Altitude and Camera Angle

- Use a grid search-based algorithm to find the optimal drone altitude and camera tilt angle delivering best wildfire detection

Algorithm 1 Grid search for altitude and camera angle

Inputs: Drone camera lens FOV in horizontal α and in vertical β, Wildfire detection rate threshold d_{min}
Output: Altitude A and camera angle θ producing wildfire detection rate $\geq d_{min}$ given greatest A
Set wildfire location $F \leftarrow (0,0,0)$
while average detection rate $d \geq d_{min}$ **do**
  $A \leftarrow A_{start}$
  Move drone to (ϕ_x, ϕ_y, A) calculated from equation (2)
  for $\theta = -90, -85, \ldots, -5$ **do**
    $A_{opt}, \theta_{opt}, d_m \leftarrow A, \theta, 0$
    Set detection count $d_v \leftarrow 0$
    for $i = 1, 2, \ldots, N$ **do**
      Move drone to another location (x_i, y_i, A) in the GCC range
      Take a camera image and run wildfire detection
      if wildfire detected and the actual wildfire is inside detection bounding box **then**
        $d_v += 1$
      end if
    end for
    Calculate average detection rate $d_a = d_v / N$
    if $d_a > d_m$ **then**
      $A_{opt}, \theta_{opt}, d_m \leftarrow A, \theta, d_a$
    end if
  end for
  $A \leftarrow A + A_{step}$
end while
return $A_{opt}, \theta_{opt}, d_m$

Coverage Path Planning (CPP) for Patrol Stage

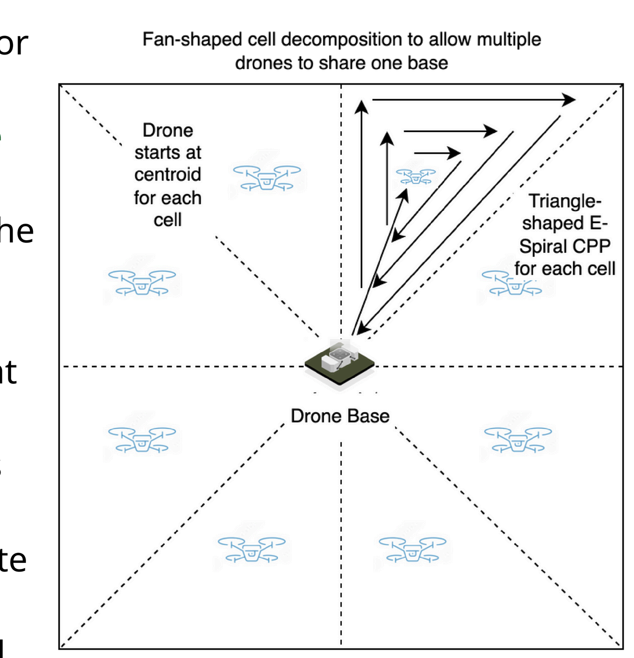
- To traverse the GCA, an Energy-Aware Spiral (**E-Spiral**) CPP model is employed
  - Advantages: guarantees **complete coverage**, allows for continuous flight as the **rotation** of the drone accounts for all areas
- Starts at center GCC and returns to the center for charging (base placed at the center for a single drone) after completing the path
- The Ground Coverage Area (GCA) for one drone given its **max** battery **flight distance** is constructed from the number of GCCs
  - Determines patrol stage range

Algorithm 2 E-Spiral Coverage Path Planning

Inputs: Drone max flight distance D_m, Ground coverage cell width W_c and height H_c
Output: A list of waypoints for E-Spiral CPP path
Set waypoints list $wps \leftarrow [\,]$
Append starting waypoint $(\phi_x, \phi_y, 0)$ to wps
Set flight distance $d \leftarrow 0$
Set direction pointer $i \leftarrow 0$
while $d < d_m$ **do**
  $(x, y, z) \leftarrow$ last element in wps
  if $i = 0$ **then**
    $x_1 \leftarrow x, y_1 \leftarrow y - H_c, d \leftarrow d + H_c$
  else if $i = 1$ **then**
    $x_1 \leftarrow x - W_c, y_1 \leftarrow y, d \leftarrow d + W_c$
  else if $i = 2$ **then**
    $x_1 \leftarrow x, y_1 \leftarrow y + H_c, d \leftarrow d + H_c$
  else if $i = 3$ **then**
    $x_1 \leftarrow x + W_c, y_1 \leftarrow y, d \leftarrow d + W_c$
  end if
  if (x_1, y_1, A) not in wps **then**
    Append (x_1, y_1, A) to wps
    $i \leftarrow 0$ if $i = 3$ else $i + 1$
  else
    $i \leftarrow 3$ if $i = 0$ else $i - 1$
  end if
end while
Return: wps

Multi-Drone Network

- E-Spiral is the optimal CPP for a multi-drone network with **non-rectangular GCAs due to minimal overlap**
- Drones fly to the center of the triangular GCAs and spiral outward for patrolling
- A shared large drone base at a fire outpost reduces base installation costs and allows for shared charging
- All drone bases communicate to a central management module hosted on the cloud infrastructure, forming a **scalable and collaborative** drone network

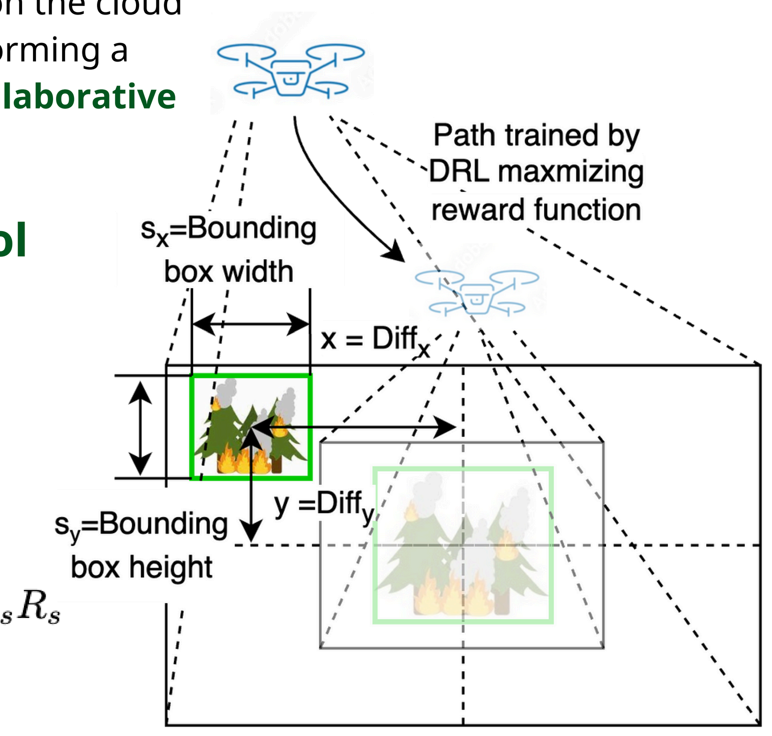


Deep Reinforcement Learning (DRL) for Inspection Stage Flight Control

- Using DRL for close-distance inspection stage maneuvers for confirmation and more details
- **Double Deep Q-Learning (DDQN) model with a memory tree**
  - Advantages: decoupled action selection and value estimation, more dynamic due to less overestimation, and learns better from past episodes
- Model considers the current detection as an environmental state value driven by the reward function for close-up inspection and better detection, given by:
- **Reward Components** - DRL reward function

$$Reward = k_x R_x + k_y R_y + k_s R_s$$

  a. Bounding box size ratio R_s calculated by: $R_s = \frac{s_x * s_y}{w_i * h_i}$ controls flight in z direction, coefficient of k_s
  b. Bounding box distance ratios R_x and R_y from center calculated by: $R_x = \frac{Diff_x}{w_i/2}$ $R_y = \frac{Diff_y}{h_i/2}$ controls drone flight in x and y directions, coefficients of k_x and k_y

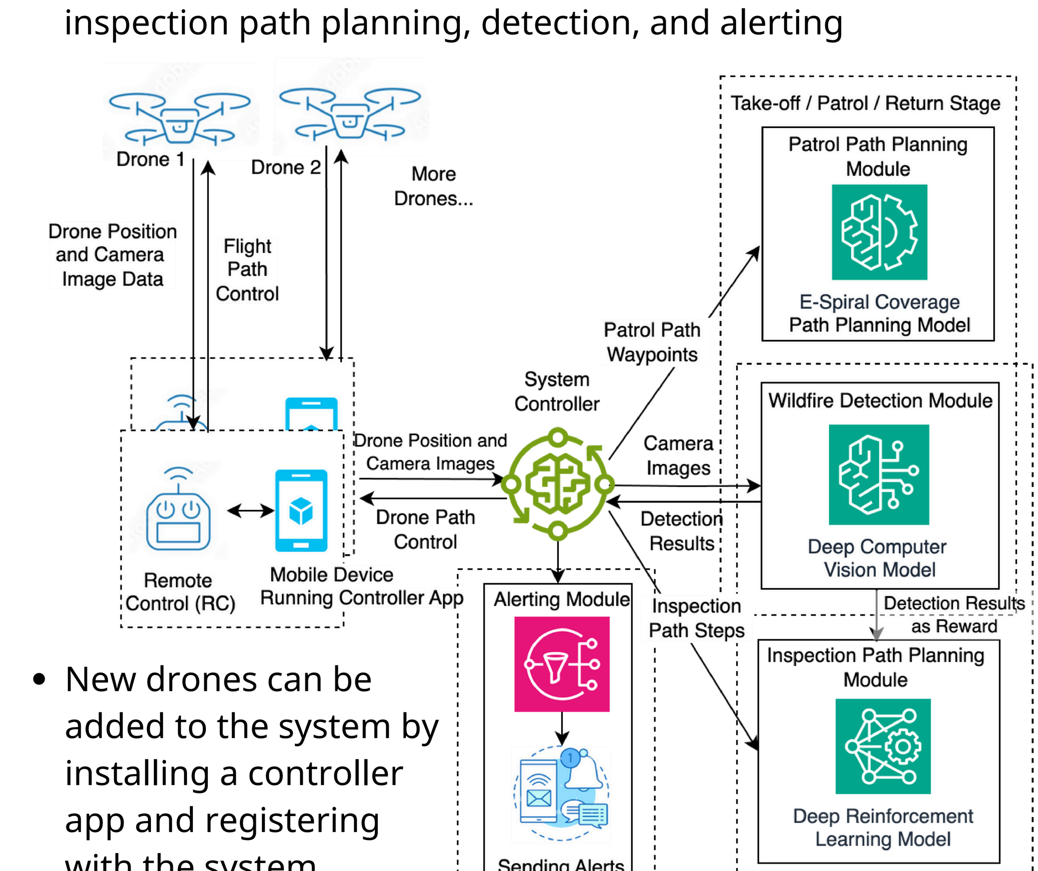


Deep Computer Vision (DCV) for Wildfire Detection

- Implemented 5 DCV models by conducting training, validation, and testing on the **FLAME** open wildfire dataset
  - YOLOv8, Faster R-CNN, DETR, EfficientDet, and RetinaNet, evaluated using mean average precision calculated at an Intersection over Union (IoU) threshold of 0.50 (**mAP50**)
    - mAP50 used to select model that is more stable and **consistent** to be trained further with transfer learning
- **YOLOv8** selected as best model over Faster R-CNN due to speed (15 ms vs 80 ms per inference) despite lower mAP50
  - Faster inference speed reduces training time for the DRL model

System Architecture

- Consists of **5 modules**: controller, patrol path planning, inspection path planning, detection, and alerting
- New drones can be added to the system by installing a controller app and registering with the system



Results

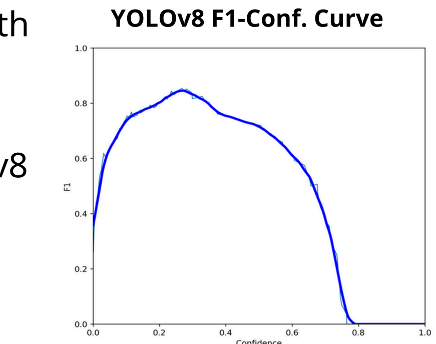
Wildfire Detection Tests

- Created simulation environment for testing using Microsoft AirSim (open-source simulator built on Unreal Engine) with virtual assets
- Below is a table of all 5 DCV models tested on FLAME with mAP50 evaluation and their unique features

Model	YOLOv8	Faster R-CNN	DETR	EfficientDet	RetinaNet
mAP50	0.913	0.916	0.748	0.663	0.763
Description and Uniqueness	Single-shot with modified CSPDarknet 53 with self-attention mechanism and feature pyramid network (FPN)	Two-shot with a region of interest (ROI) pooling layer and region proposal network (RPN)	Transformer on CNN backbone and feed-forward network (FFN)	Single-shot with weighted bi-directional feature pyramid network (BiFPN) scaling method	Single-shot with focal loss function to address class imbalance during training

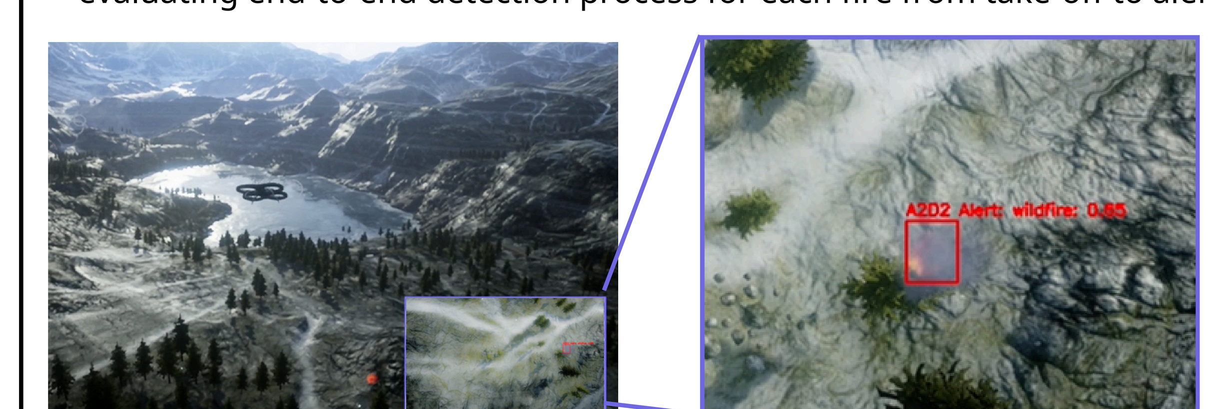
- Selected YOLOv8 had a mAP50 of **0.913** with the fastest detection speed of **15ms**, a fraction of the next fastest, Faster R-CNN
- After transfer learning, the selected YOLOv8 was able to detect with a mAP50 of **0.90**, comparable to the performance on FLAME

YOLOv8 F1-Conf. Curve



End-to-End Experimental Tests

- **Test setup:** Placing 100 wildfires at random locations in the GCA and evaluating end-to-end detection process for each fire from take-off to alert.



- Above is a screenshot of the full system in action with the functional take-off, spiral patrol, detection, and close inspection process
- At the right is a table of the system's results from the tests
  - Inspection path planning used DDQN model which trained faster than A2C with similar results

Altitude, A	80 ft (from Grid Search)
Camera angle, θ	45° (from Grid Search)
Detection Rate	0.99
DDQN Loss	10.634

Conclusion

Summary

- A2D2 enables a drone to emulate the heuristic operational procedure of a human pilot, involving searching, closely flying over for confirmation, and sending alerts
- The system is a breakthrough innovation that significantly elevates the efficiency and effectiveness of early wildfire detection

Future Work

- Transition from simulation to field tests with support from CAL Fire and local fire departments
- Continue developing this system to make it a modularized platform extensible to other tasks and use cases like post-fire monitoring, remote sensing, geo-mapping, search and rescue, etc.

Selected References

- Saydirasulovich, S. N., Mukhiddinov, M., Djuraev, O., Abdusalomov, A., & Cho, Y.-I. (2023). An Improved Wildfire Smoke Detection Based on YOLOv8 and UAV Images. Sensors, 23(20), 8374.
- Cabreira, T. M., Franco, C. D., Ferreira, P. R., & Buttazzo, G. C. (2018). Energy-Aware Spiral Coverage Path Planning for UAV Photogrammetric Applications. IEEE Robotics and Automation Letters, 3(4), 3662–3668. https://doi.org/10.1109/LRA.2018.2854967
- Kaufmann, E., Bauersfeld, L., Loquercio, A., Müller, M., Koltun, V., & Scaramuzza, D. (2023). Champion-level drone racing using deep reinforcement learning. Nature, 620(7976), 982–987. https://doi.org/10.1038/s41586-023-06419-4
- Shah, S., Dey, D., Lovett, C., & Kapoor, A. (2017, May 15). AirSim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles. arXiv.Org. https://arxiv.org/abs/1705.05065v2
- Shamsoshoara, A., Afghah, F., Razi, A., Zheng, L., Fulé, P. Z., & Blasch, E. (2021). Aerial imagery pile burn detection using deep learning: The FLAME dataset. Computer Networks, 193, 108001. https://doi.org/10.1016/j.comnet.2021.108001
- van Hasselt, H., Guez, A., & Silver, D. (2016, February 12). Deep Reinforcement Learning with Double Q-learning. Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence. AAAI '16: The 13th AAAI Conference on Artificial Intelligence.

Acknowledgments: Bradley Fulk for mentorship, CAL Fire, US Forest Service, San Diego FD, Menlo Park FD, and Orange County FD for valuable insight and advice