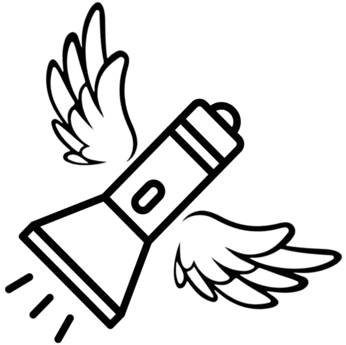


Search and Shine

Project Design Review | SS24 ECE Capstone

Team C7: **Ronit Shah**, Nina Guo, David Wu

Electrical & Computer ENGINEERING

Use Case

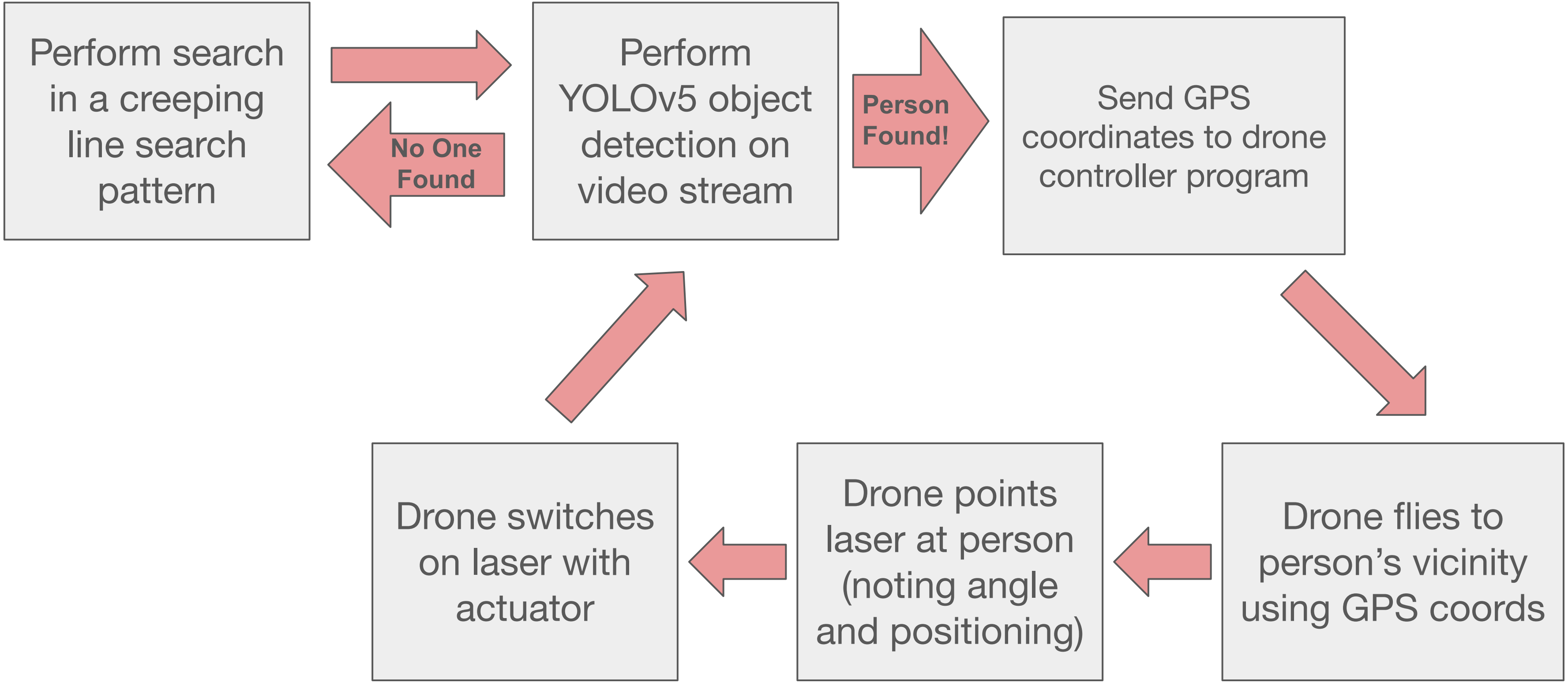
- A search and aid **drone** application that can **autonomously** locate people needing help and shine a **spotlight** on them
  - Our project will use a laser over a spotlight to demonstrate accurate target tracking
- Manually searching for people in disastrous remote areas is **inefficient**, **expensive**, and **error-prone**
- ECE Areas: Software Systems, Signals and Systems

Source: https://drones.wfp.org/index.php/activities

Design Requirements

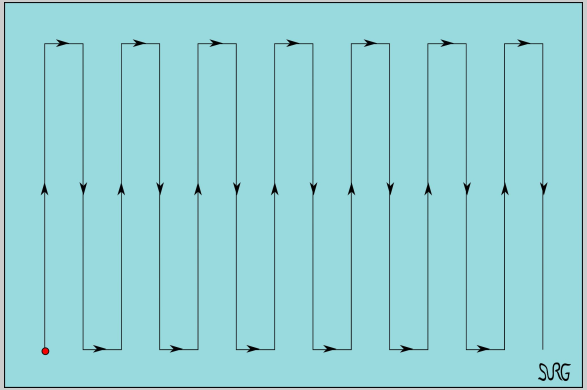
#01 Accurately identify humans in a landscape, and track them	#02 Autonomously comb a landscape inexpensively	#03 Report findings through web-app efficiently
Top-1 Accuracy: > 80% Top-5 Accuracy: > 90%	Visual inspection of images and drone to see if entire arena is covered	Latency of detection, data routing, and result processing: < 5s
Ability to detect humans in different positions (standing, sitting, covered): 90% acc. in GPS coords by server Ability to track humans with the laser	Receive and go to specific coordinates	API built in web application to display where the drone and people are
Offset between laser and actual human: ± 1 feet	Cost effectiveness: <$300	Speedup over sequential YOLOv5 algorithm: 5x

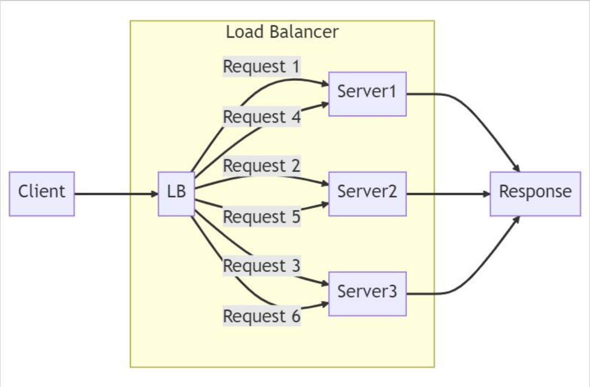
Solution Approach



Justifications for Solution

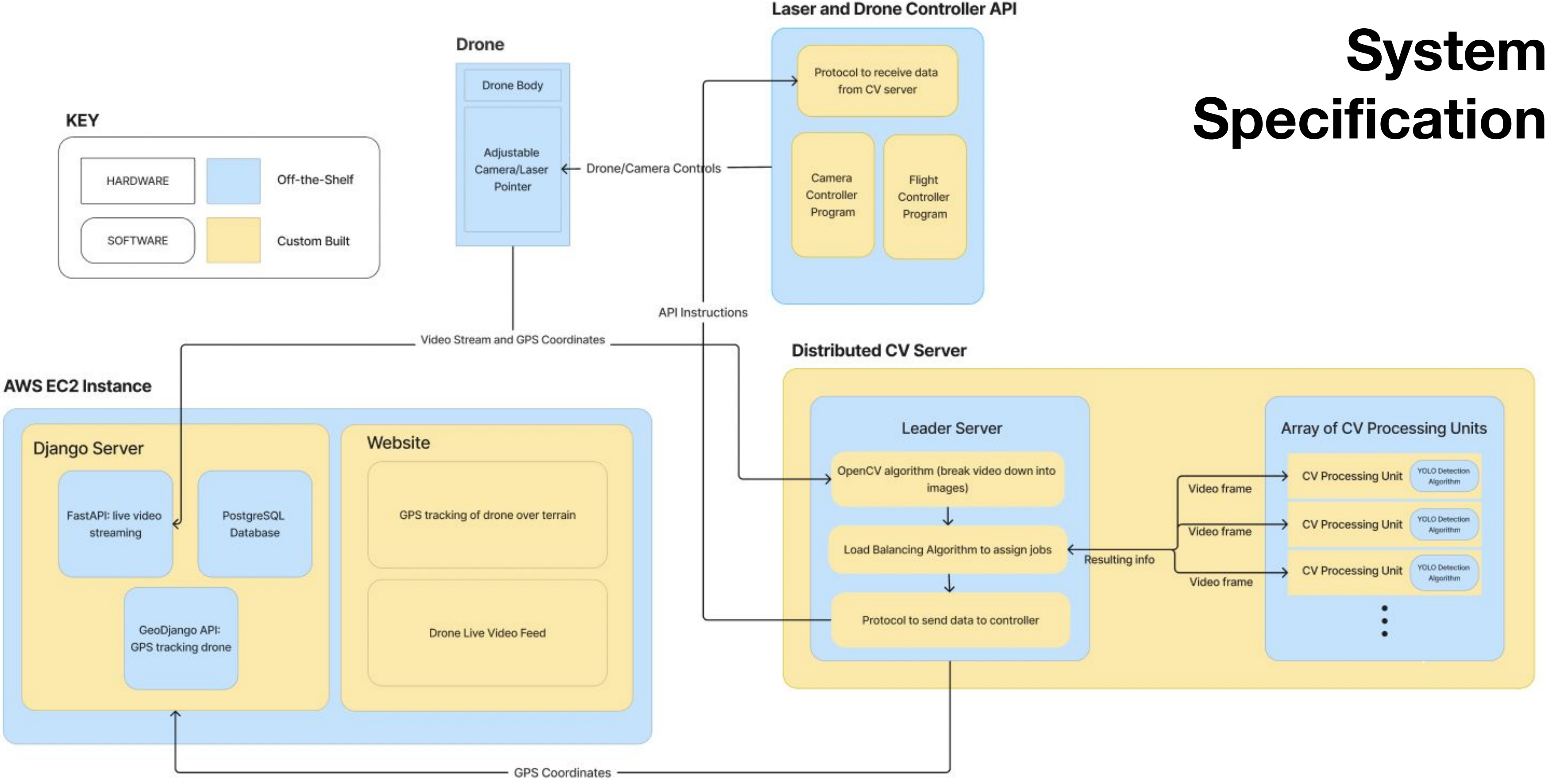
- **Creeping line search** can comb an area efficiently and thoroughly
- **Load Balancing** Algorithms to speed up YOLOv5 algorithm and distribute tasks
- **Public Safety Considerations:** Greater health protection for individuals in need
- **Social Factors:** Providing aid without discrimination and protection for workers
- **Economic Considerations:** Cost effectiveness and less need for manual intervention





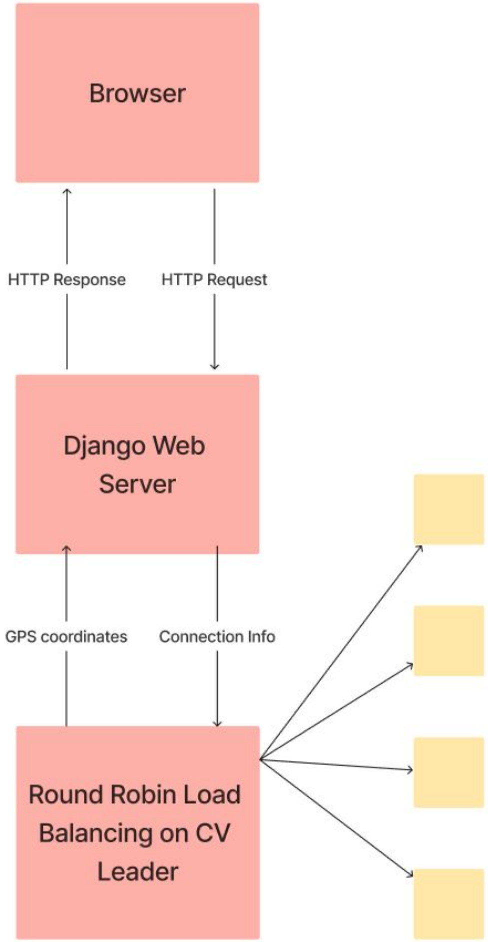
Sources:
https://en.m.wikipedia.org/wiki/File:Creeping_line_search_pattern.png
https://medium.com/javarevisited/load-balancing-algorithms-that-can-be-used-in-java-applications-6f605d1bf19

System Specification



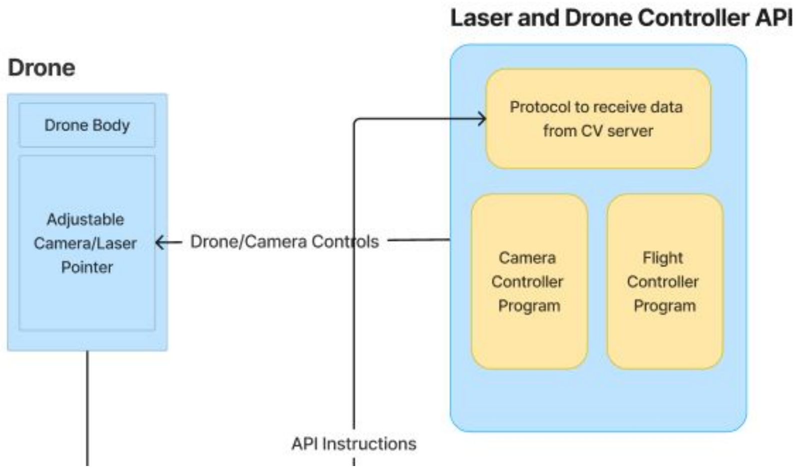
Implementation Plan: Server and Web Application

- Browser communicates with **Django** web server to fetch info like GPS coordinates
- **OAuth** and **SMS verification** for secure login to website
- Custom-made **Round Robin** load balancing runs on CV leader node and distributes tasks to CV nodes
- **OpenCV** on CV leader to break video down into frames
- **YOLOv5** object detection algorithm runs on CV nodes (https://github.com/danhilltech/goyolov5)

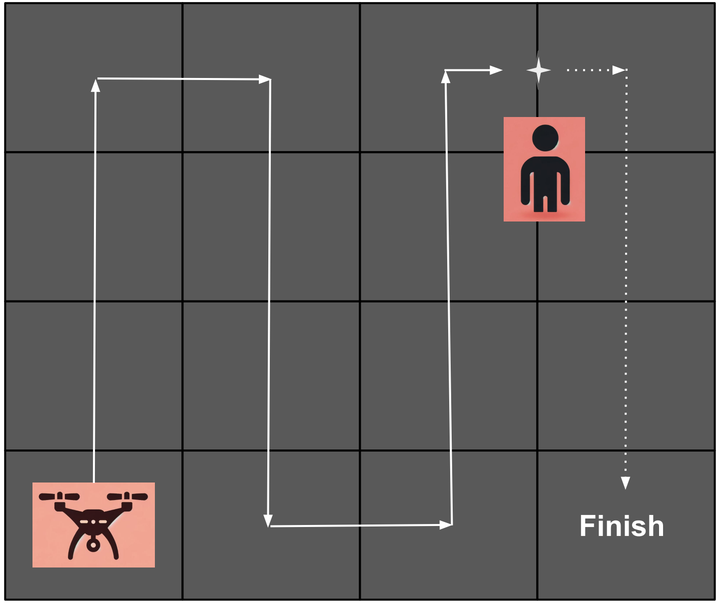


Implementation Plan - Drone and Laser (and API)

- Drone has software API (Mavlink protocol), an adjustable camera, and a GPIO.
  - In discussion with Prof. Basti Scherer
- Flight control prog uses software API to control **movement.**
- Camera control prog uses software API to control the camera **AND** the laser
  - Laser is strapped onto the camera
  - On/off is controlled through the GPIO
    - Actuator-controlled switch
  - Must be *light* and *energy efficient*



Scenario Testing Setup

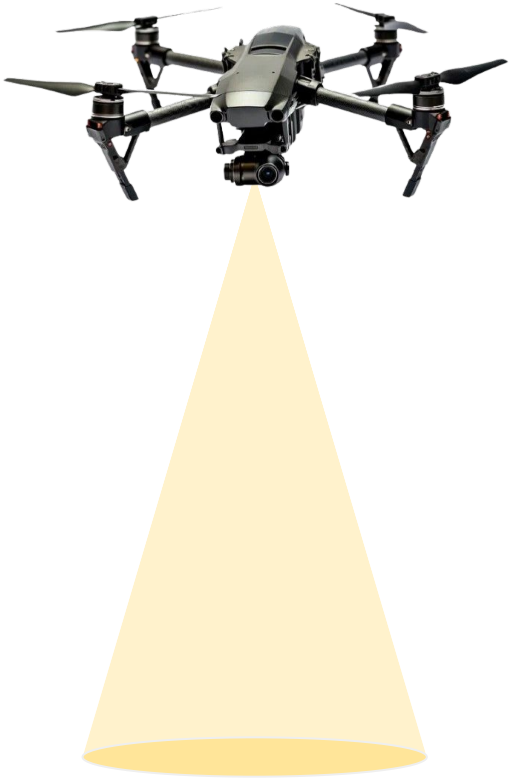


- Drone Arena Scaife Hall (20’ x 20’)
- Black Foam Tiling
- Arena divided into grid
- Additional Obstructions (balls, plants, jackets -> obstruct view of drone)
- Detect person on flight path and spotlight them
- Future tests: person moves after being pointed at; laser follows

Testing, Verification, and Metrics

Requirements	Testing	Metrics
Accurately identify humans in a flat landscape	Unit test images of people in different flat environments, body parts hidden, multiple people	Top-1 Accuracy: > 80% Top-5 Accuracy: > 90%
Autonomous flight control	Checkpoint tests of creeping search and targeted coord movement	Can fly in pre-specified pattern, and correctly navigate to laser-pointing position if person found (± 3 feet)
Low latency	Time taken to send, process, and receive flight and video data	Latency of detection, data routing, and result processing: < 5s
Point light to person's GPS location	Comparison tests between person's actual coords vs laser-pointed coords vs calculated coords using altitude and scraping video frame data	Offset in person location and GPS calculation: ± 1 feet Offset in person location and laser-pointed location: ± 1 feet
Power consumption	Flight time of drone with and without laser + actuator attachment	Maintain <5 minute loss of flight time when searching

Alternative Approach: Fixed Spotlight



- Pointing a laser at a target requires very high precision (distance amplifies error)
- Goal of search and shine can still be maintained on a simpler level with a "UFO" approach
  - Drone now flies directly above target
    - Metric: ± 1 feet
  - Laser is strapped onto the drone
  - On/off is still controlled through the GPIO
    - Actuator-controlled switch
  - Must be light and energy efficient

Schedule

task/milestone	description	person/people with primary responsibility	start date	finish date	Wed 2/7	Wed 2/14	Sun 2/18	Wed 2/21	Wed 2/28	Fri 3/1	Wed 3/6	Wed 3/13	Wed 3/20	Wed 3/27	Wed 4/3	Wed 4/10	Wed 4/17	Sun 4/21
Obtain materials	Perform research on correct materials to buy (motors, Arduino components, etc.)~~, and purchase them~~ - changed to obtaining from Prof. Basti	David	2/7	2/14	x	x												
Load Balancing Algorithm (1/2)	Perform research on optimal load balancing algorithms and implement in Go	Ronit	2/7	2/14	x	x												
General Website Setup	Create base template for Django website, do front/backend setup of site, and research hosting live video streaming	Nina	2/7	2/14	x	x												
Design Presentation Slides	Design Presentation Slides	All	2/14	2/18		x	x											
Begin coding of flight program logic	Learning to program with the software API for movement and camera control	David	2/14	2/21		x	x	x										
Load Balancing Algorithm (2/2)	Implement the load balancing algorithm in Go, and finish unit testing. Research fetching the video stream from the DCC	Ronit	2/14	2/21		x	x	x										
Implement Live Video Stream and GPS Tracking	Create working displays of live video feed and gps tracking of drone, extra features to improve UX	Nina	2/14	2/21		x	x	x										
GPS Coord Movement	Use software API to be able to integrate with CV server data (target coord movement)	David	2/21	2/28				x	x									
Extract Real Time Data + Calculation Algo	Implement routing to send flight data to website and cv server, GPS calculation of human location	Nina	2/21	2/28				x	x									
Computer Vision Algorithm on the Distributed Server	Implement the Computer Vision Algorithm on the CV processing units. Integrate with video frames and unit tests	Ronit	2/21	2/28				x	x									
Perform laser assembly - drone GPIO and actuator implementing, spring break	Work on attaching the laser, and using drone GPIO to control actuator to power the laser	David	2/28	3/13					x	x	x	x						
Deploy website, UI/UX, + Fully integrate	Deploy code on EC2, and implement a better UI/UX, Fully integrate and test all components are connected efficiently	Nina & Ronit	2/28	3/13					x	x	x	x						
Laser pointer accuracy	Coordinate math to be able to point laser correctly, accounting for angle and positioning; Work with CV server to handle location and pointing logic	Ronit & David	2/28	3/13					x	x	x	x						
Test initial creeping search	Test autonomy and creeping search, inital pointing accuracy involving GPS coords	All	3/13	3/20								x	x					
Test entirety of DCC	Making sure everything is integrated and working with server and website	All	3/20	3/27									x	x				
Slack	Slack time	All	3/27	4/21										x	x	x	x	x
Final Presentation Slides	Final Presentation Slides	All	4/17	4/21													x	x