Retriever Rover

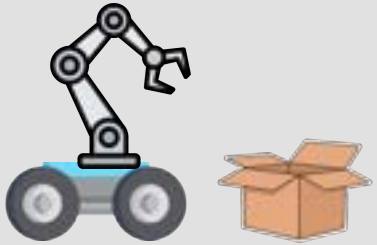
Final Presentation

Team C0: Jasmine Chao, Matthew Mao, **Joonyoung Huh**

Use Case/Application

Nearly a quarter of adults have trouble bending down or kneeling, according to PMC

1. User will send a request via WebApp to retrieve an object (boxes/grocery bags)



2. Robot will navigate and locate object

3. Robot will pick up object

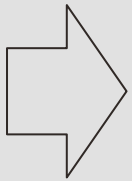
4. Robot will find user and return object at an acceptable height

User Requirements

Delivery Height: 83 cm	95 percentile for knuckle height is 83 cm
Communication between app + robot < 5s	Email takes around 5 seconds to send/receive
Weight of 150 grams	For now, robot's use case is limited to smaller objects, such as medicine or light packages
Successful pickup rate: 70%	Users can rerun. 70% ensures that the first trial has more than a 50% chance of success while having a second rerun has 1 - 0.3 * 0.3 = 91%

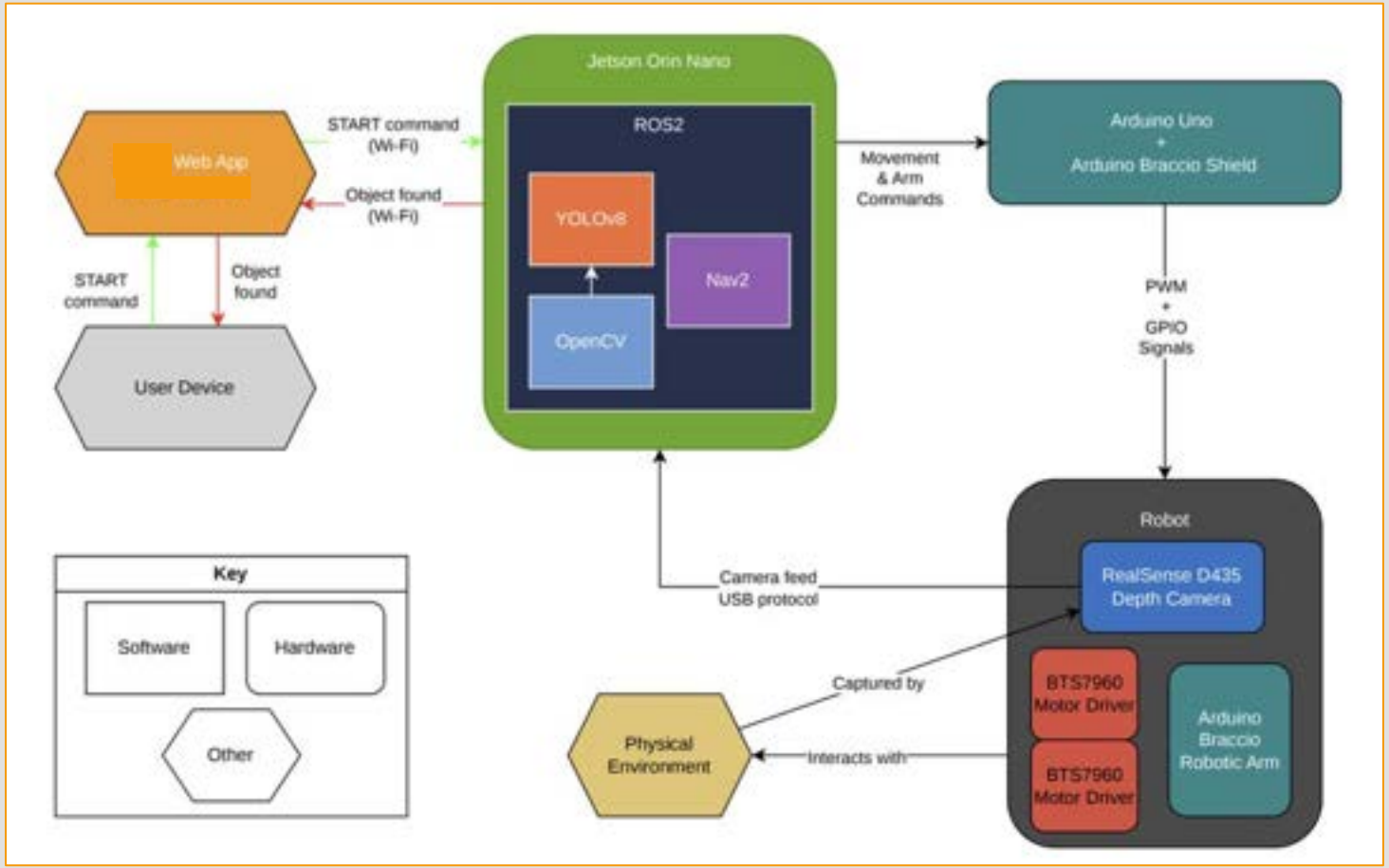
Design Requirements

Detection: 90%
Pickup: 90%
Navigation: 90%

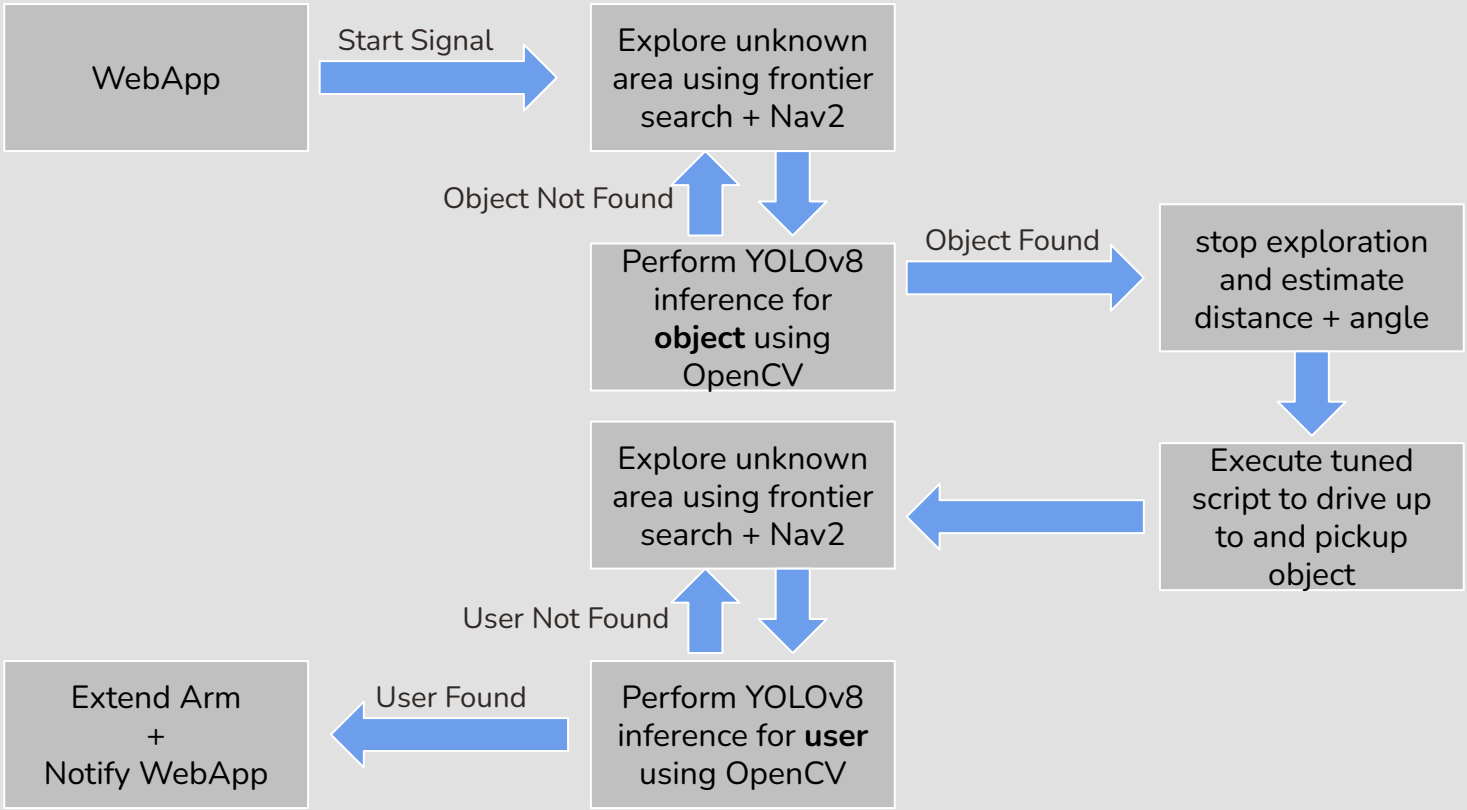


chaining all of their probabilities together allows for:
90% * 90% * 90% = 72.9% > 70%
(USER REQUIREMENT)

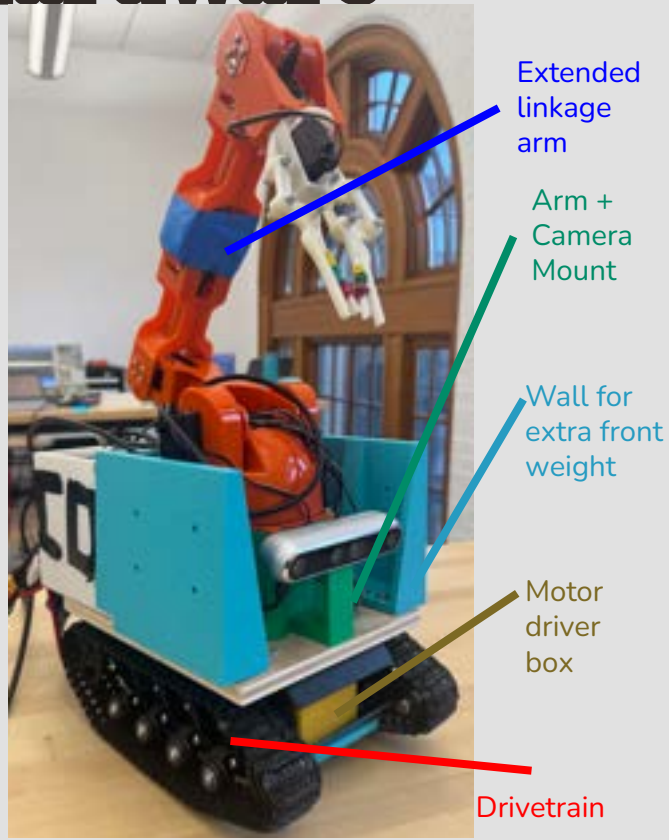
Block Diagram

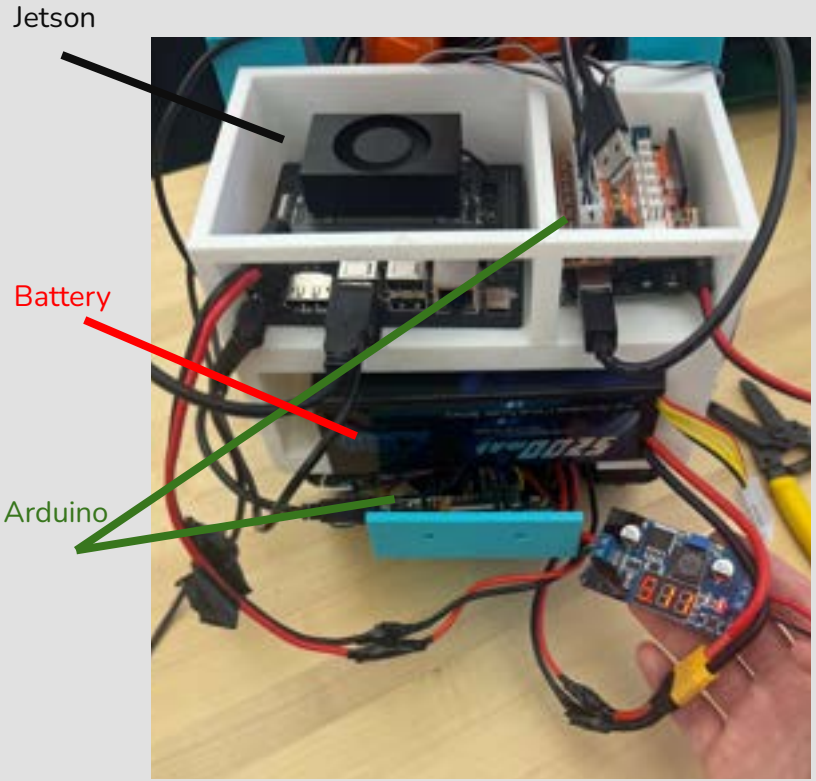


Complete Solution - software

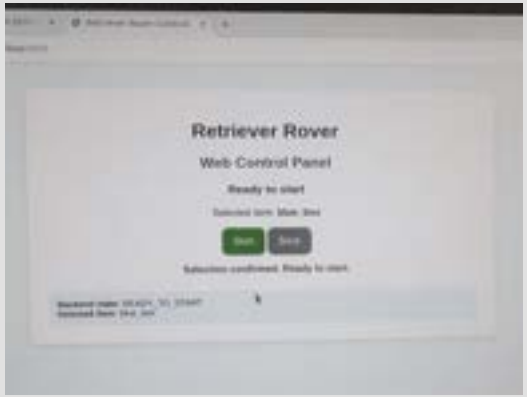


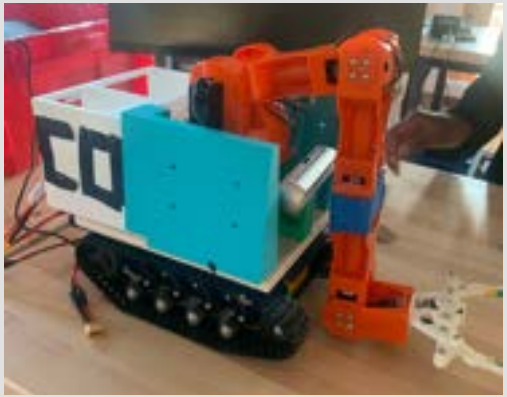
Complete Solution - Rover Hardware





Final Demo Plans





- Robot will be in a predetermined area will four walls and obstacles
- Audience member can place object anywhere in room (not against a wall or corner)
- Audience will then leave the room and press START on the webapp
- We will then watch as the robot navigates and picks up object
- Robot will start searching for a human.
- We will enter the room and show how the robot stops and raises the object to an accessible height

CV Testing

	0.2m	0.5m	1m
Target Object	12/12	12/12	12/12
Foot	4/4	4/4	4/4
Shoe that gets predicted as foot (want this to be low)	2/4	0/4	0/4

Blue box: > 0.90 for prediction, Foot: > 0.6 for prediction

Units of results: correctly identified orientations / total number of orientations tested

Navigation Testing

	Robot can simply drive forward	One Obstacles	Two Obstacles
Results	5/5	2/5	0/5

Conducted 5 trials each

Units of results: successful trials / total trials

https://drive.google.com/file/d/1NFleo90KV4k2dqpT-z9HnwvZLBGEgBv6/view?usp=sharing

Pick up Testing

	0 degrees	45 degrees right	45 degrees left
Robot starts very tilted (45 degrees)	3/3	1/3	2/3
Robot starts slightly tilted (< 15 degrees)	3/3	2/3	2/3

Conducted 3 trials each, overall success: **72%** (might have to retune motor drivers b/c they broke)

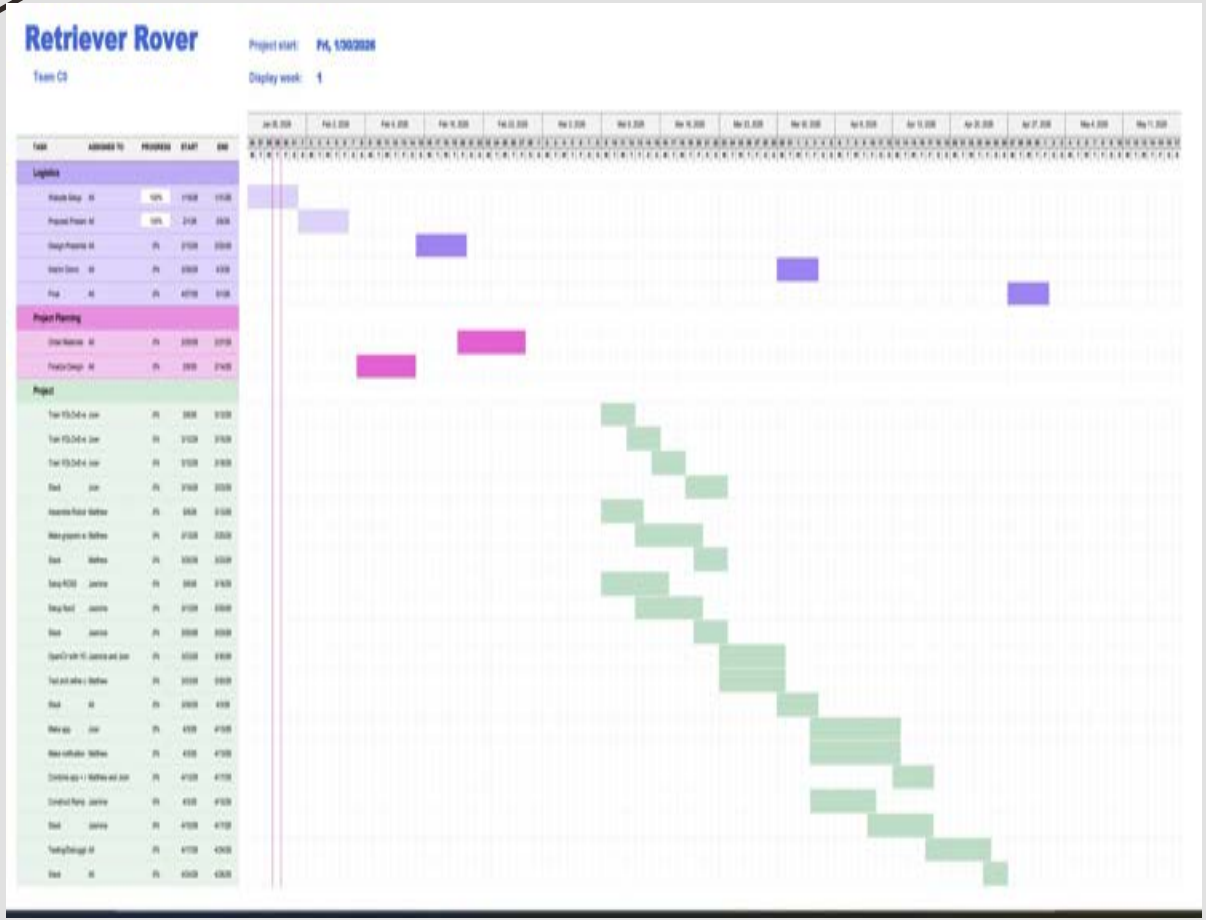
Units of results: successful trials / total trials

https://drive.google.com/file/d/1WMnJW8Sr1qFEygUyHeFJEJMbV1grieAr/view?usp=sharing

Design Tradeoffs

What we considered:	What we went with
Variable camera angle (set by servo) ● Able to see farther ● Easier to detect humans (can use online model) ● COMPLICATED	**Fixed camera** ● Less complexity (one less motor to cause failure) ● Angle of our camera pointing down: had to train a custom model for feet/shoes
Raise the height of our standoff by 6” ● Increases chance of tipping over ● Makes it harder to grab object because we are so high up	**Extend the length of arm by 6”** ● Increases torque on servos ● Led to a lot of servo horn stripping
Creeping line search ● Very simple algorithm ● Easier ● Cannot avoid obstacles well	**Nav 2 + Frontier search** ● A lot more complex ● Obstacle detection and avoidance available

Project Management



Remaining Tasks:

- Tune pickup
- Tune Navigation
- Integrate firmware for raising an object to user-accesible height
- Build the full pipeline
- Replace current placeholders on webapp with actual rover-side actions

Lessons

- Buy extras and backups!!!
  - Hardware often breaks, and at the worst times possible!
- If you can't find a specific part, 3D printing is always a viable option
- Dividing up a very large project into many components is helpful
  - Each arduino has its own task, Jetson acts as the head telling subcomponents when to do their task
- We gained a lot of experience with YOLO models
  - Always be sure to vary your data
- Make sure you install the correct software packages when using Jetson