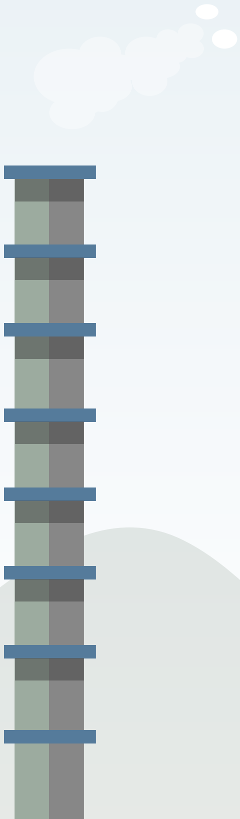
## EMBELLISHER:

# TRASH COLLECTING ROBOT

By:  Ella Lee, Ritu Pathak & Hirani Sattenapalli

# OUR DRIVE

## USE-CASE DEFINITIONS

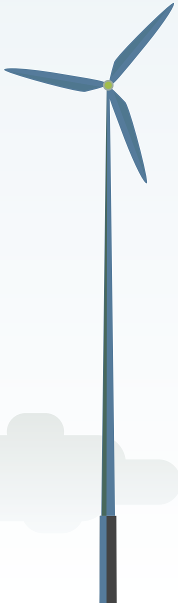


### PROBLEMS

- Increased pollution
- Sanitization hazards
- Intensive human labor
- High cleaning costs
- Public safety

### NEEDS

- Identify garbage on sidewalks
- Pick up and collect garbage
- Navigate autonomously
- Restrictive budget
- Avoiding obstacles



# USE-CASE REQUIREMENTS

| | |
|---|---|
| **Object Classification of ML Model** | At least 95% mean average precision for Yolo V7<br>80% mean average precision for tiny Yolo V7 |
| **Obstacle Avoidance** | At least 95% success rate |
| **Efficiency** | At least 90% pick up rate |
| **Object Pickup Speed** | Less than 45 seconds |
| **Robot Operation Duration** | Within a span of 2 to 4 hours |
| **Voltage Duration** | About 2 hours |

# QUANTITATIVE DESIGN REQUIREMENTS

| | |
|---|---|
| **ML Subsystem Performance** | At least 15 frames per second rate for inferencing pipeline |
| | Returning bounding boxes at<br>- A confidence interval of 0.68<br>- A precision of 0.95<br>- Recall of 1.0 |
| **Movement and Pick-Up Mechanism** | Speed of the robot to be roughly 0.92ft/s<br>- max distance of 18.4ft<br>- 20s to reach and pick up the trash |
| **Coordination/ Path planning** | Robot's ability to come back to its starting location<br>- Avg distance from initial to final coordinates < 2-3 ft (~60cm) |
| **Power Supply** | At least 28000 mAH<br>Last approximately 2.9 hours when supplying 5v @ 3a |

# SYSTEM APPROACH

## 01 OBJECT CLASSIFICATION

**Train:** ~~YOLOv7 and~~ YOLOv7-tiny
**Datasets**: Soda Cans, Water Bottles, and Crumpled Paper
**Parts**: Jetson Nano Orin, ~~e-CAM50_CUNX~~ (+ Intel Realsense Camera)
**Software**: openCV, PyTorch, TensorRT

## 02 MOTOR CONTROL

- Robot rotates and navigates to detect trash
- Roller pushes garbage in, which is carried by conveyor belt to storage

**Parts**: ~~RPi 5~~ (+RPi 4), mecanum wheels, Rollers, linear shafts, ~~stepper motors~~ (+DC motors), rubber belt, ultrasonic sensors
**Software**: ~~ROS2 Humble~~, Python

## 03 INTEGRATION

Ethernet connection between devices
Unit tests for each component
Integration tests defined for testing + metrics

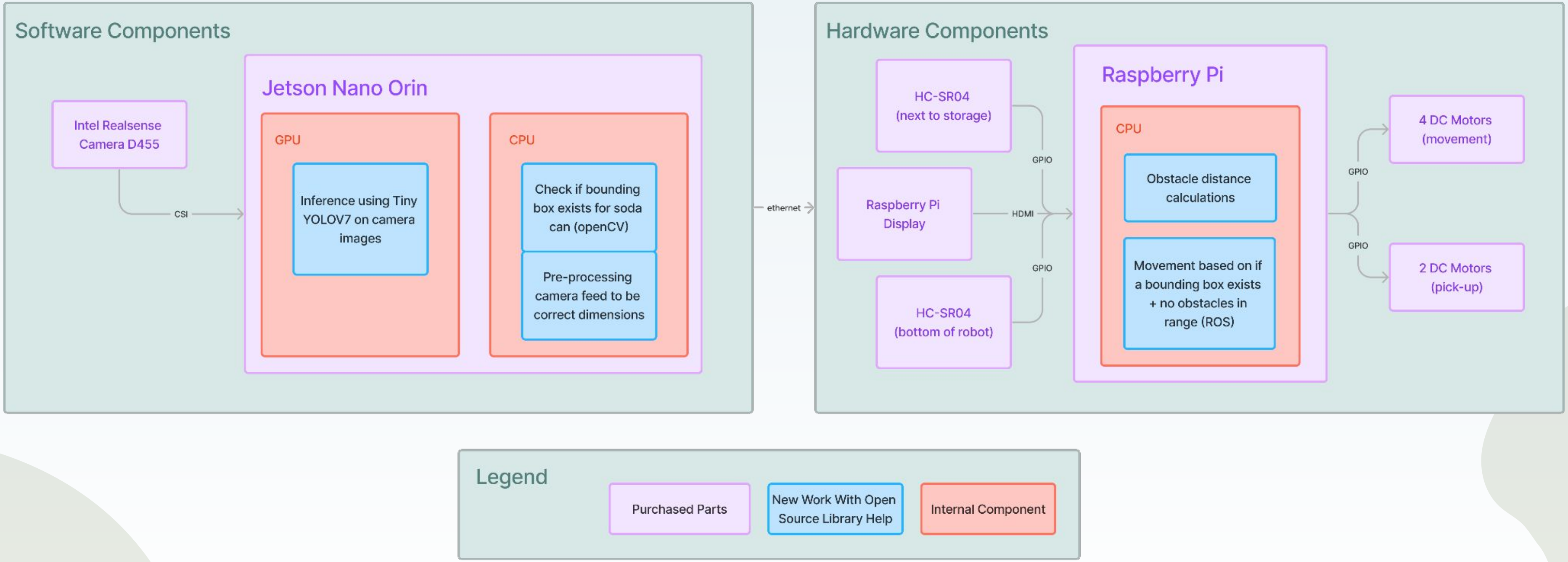
## 04 COORDINATION/ PATH-FINDING

Define path for robot to return to original location given coordinates

**Parts:** Gps module

**Software:** ~~VSLAM~~, Python package

# SYSTEM DIAGRAM



## Software Components

Intel Realsense Camera D455

Jetson Nano Orin

GPU

Inference using Tiny YOLOV7 on camera images

CPU

Check if bounding box exists for soda can (openCV)

Pre-processing camera feed to be correct dimensions

## Hardware Components

HC-SR04 (next to storage)

Raspberry Pi Display

HC-SR04 (bottom of robot)

Raspberry Pi

CPU

Obstacle distance calculations

Movement based on if a bounding box exists + no obstacles in range (ROS)

4 DC Motors (movement)

2 DC Motors (pick-up)

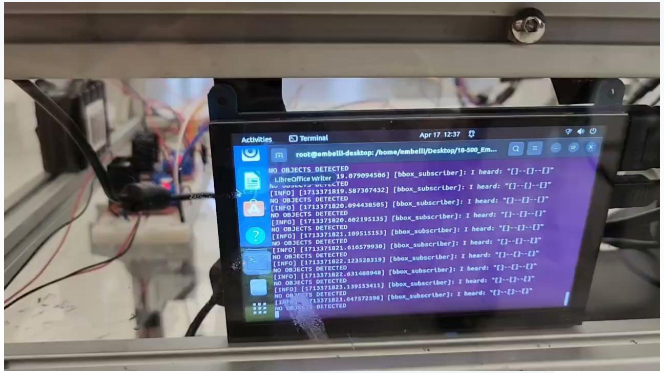
## Legend

Purchased Parts

New Work With Open Source Library Help

Internal Component

# COMPLETE SOLUTION



```
NO OBJECTS DETECTED
[illegible]19.079094506] [bbox_subscriber]: I heard: "[]--[]--[]"
NO OBJECTS DETECTED
[INFO] [1713371819.587307432] [bbox_subscriber]: I heard: "[]--[]--[]"
NO OBJECTS DETECTED
[INFO] [1713371820.094438505] [bbox_subscriber]: I heard: "[]--[]--[]"
NO OBJECTS DETECTED
[INFO] [1713371820.602195135] [bbox_subscriber]: I heard: "[]--[]--[]"
NO OBJECTS DETECTED
[INFO] [1713371821.109515153] [bbox_subscriber]: I heard: "[]--[]--[]"
NO OBJECTS DETECTED
[INFO] [1713371821.616579930] [bbox_subscriber]: I heard: "[]--[]--[]"
NO OBJECTS DETECTED
[INFO] [1713371822.123528319] [bbox_subscriber]: I heard: "[]--[]--[]"
NO OBJECTS DETECTED
[INFO] [1713371822.631488948] [bbox_subscriber]: I heard: "[]--[]--[]"
NO OBJECTS DETECTED
[INFO] [1713371823.139553411] [bbox_subscriber]: I heard: "[]--[]--[]"
NO OBJECTS DETECTED
[INFO] [1713371823.647572596] [bbox_subscriber]: I heard: "[]--[]--[]"
NO OBJECTS DETECTED
```

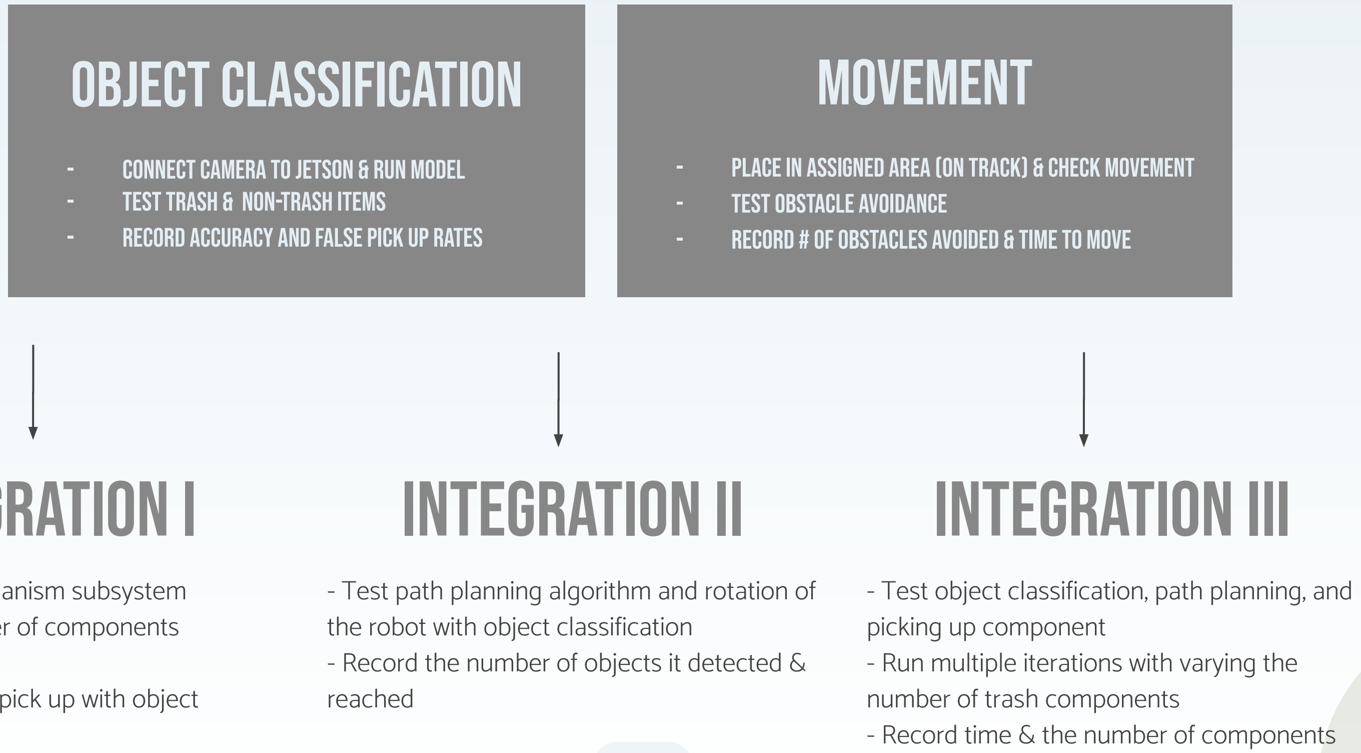
## ETHICAL CONSIDERATIONS

- **Public Safety & Welfare:**
  - Avoid collision with humans & street animals
  - Ensure operating location is the sidewalk
- **Social & Environmental Factors:**
  - Plan to have the robot during times of the day when the streets are less crowded
  - 10am-12am or 1pm-3pm

# DESIGN TRADE-OFFS

| Subsystem | Definition & objectives | Criteria | Considered Alternatives |
|---|---|---|---|
| **Machine Learning** | Classification of trash components | Time<br>Accuracy<br>False Positives | - Pre-Trained vs Custom ML Model<br>- Custom Dataset vs Existing Dataset<br>- Precision vs Recall |
| **Motion Control & Pickup** | Movement of robot + collection of trash | Movement & Pick up Speed | - Different types of motors<br>  - Higher torque, less speed<br>- Scoop vs robotic arm vs roller<br>  - Primary trade off: speed & ease of access<br>  - Durability & consistency |
| **Path Planning Algorithm** | Defines robot ability to get back to starting location | Engineering Complexity | - Issac ROS VSLAM<br>- AprilTags<br>- Distance Matrix Service by Google Maps |

# TEST, VERIFICATION, AND VALIDATION



## OBJECT CLASSIFICATION

- CONNECT CAMERA TO JETSON & RUN MODEL
- TEST TRASH & NON-TRASH ITEMS
- RECORD ACCURACY AND FALSE PICK UP RATES

## MOVEMENT

- PLACE IN ASSIGNED AREA (ON TRACK) & CHECK MOVEMENT
- TEST OBSTACLE AVOIDANCE
- RECORD # OF OBSTACLES AVOIDED & TIME TO MOVE

## INTEGRATION I

- Test pick-up mechanism subsystem
- Record the number of components collected
- Test movement + pick up with object classification

## INTEGRATION II

- Test path planning algorithm and rotation of the robot with object classification
- Record the number of objects it detected & reached

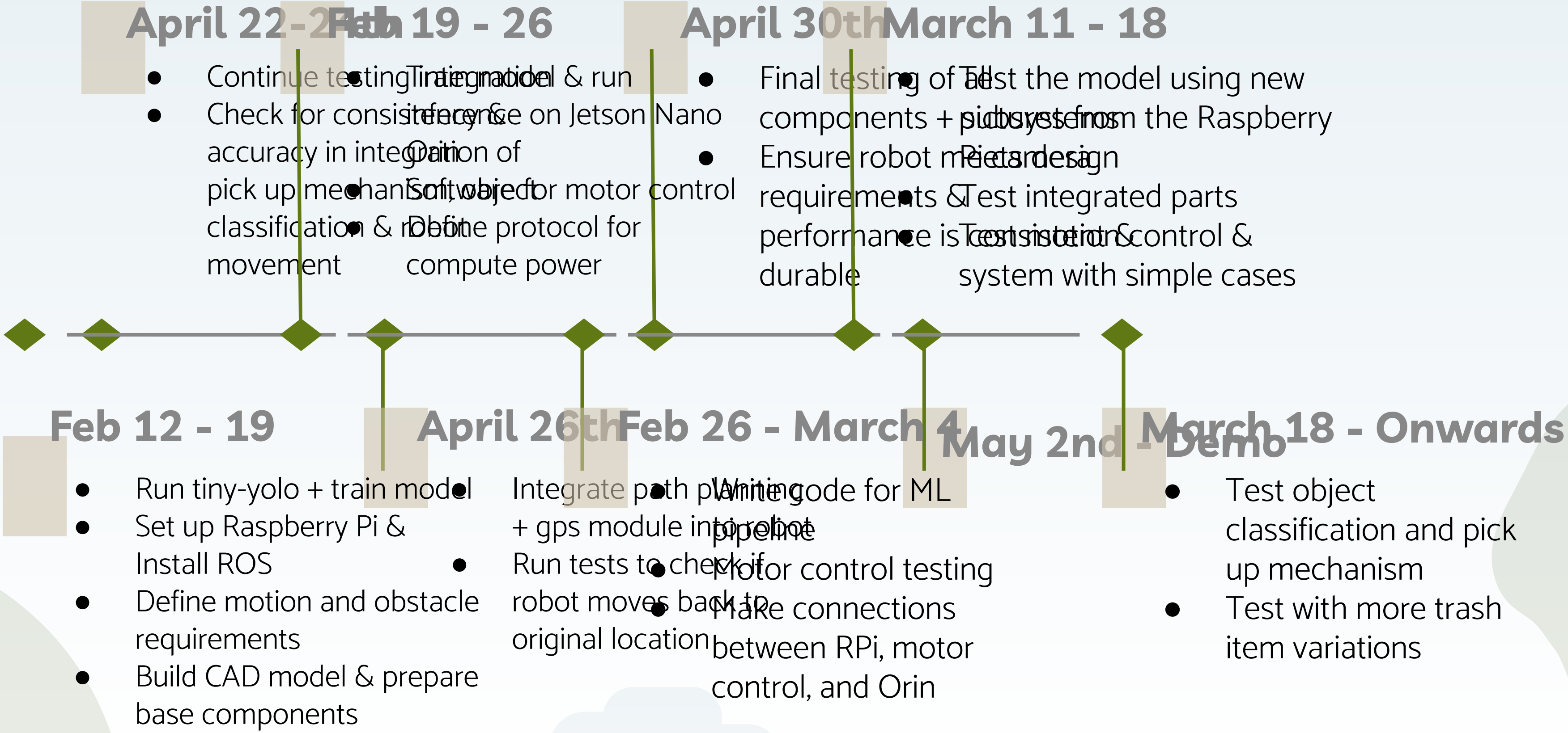
## INTEGRATION III

- Test object classification, path planning, and picking up component
- Run multiple iterations with varying the number of trash components
- Record time & the number of components

# VERIFICATION & VALIDATION

| Subsystems | Measurements | Test Inputs | Passing Test Requirements | Results |
|---|---|---|---|---|
| **Machine Learning** | Inference speed<br>Conf. Threshold<br>Avg. Precision<br>Avg. Recall | 1. Camera feed<br>2. Test dataset<br>3. False positive visual inspection | 1. Run inferencing @ **15 FPS**<br>2. Conf. Threshold >= **0.68**<br>3. Precision >= **0.95**<br>4. Recall = **1.0** | 1: **~115 FPS** for inferencing, **~103 FPS** for full ML pipeline<br>2 & 3: **mAP = 0.978** avg. across classes @ **0.68 conf.** threshold<br>2 & 4: **Recall = 0.975** avg. across classes @ **0.68 conf.** threshold |
| **Path Planning** | Distance to original location | 1. Initial coordinates<br>2. Final coordinates | 1. Avg distance between initial and final coordinates < 60-90cm | **Ongoing testing** |
| **Motion Control + Pick up Mechanism** | Pick up speed<br>Items collected<br>Operation Time | 1. Trash components | 1. Pick up Time < 20 seconds<br>2. Operate for 2 hours<br>3. > 90% of classified trash items are collected | **Pick up time:** 15 seconds<br>**Operation time:** 2-2.5 hours<br>**Success rate:** 9 out of 10 |

# PROJECT MANAGEMENT



## Feb 12 - 19

- Run tiny-yolo + train model
- Set up Raspberry Pi & Install ROS
- Define motion and obstacle requirements
- Build CAD model & prepare base components

## Feb 19 - 26

- Train model & run inference on Jetson Nano Orin
- Software for motor control
- Define protocol for compute power

## Feb 26 - March 4

- Writing code for ML pipeline
- Motor control testing
- Make connections between RPi, motor control, and Orin

## March 11 - 18

- Test the model using new pictures from the Raspberry Pi camera
- Test integrated parts
- Test motion control & system with simple cases

## March 18 - Onwards

- Test object classification and pick up mechanism
- Test with more trash item variations

## April 22-24th

- Continue testing integration
- Check for consistency & accuracy in integration of pick up mechanism, object classification & robot movement

## April 26th

- Integrate path planning + gps module into robot
- Run tests to check if robot moves back to original location

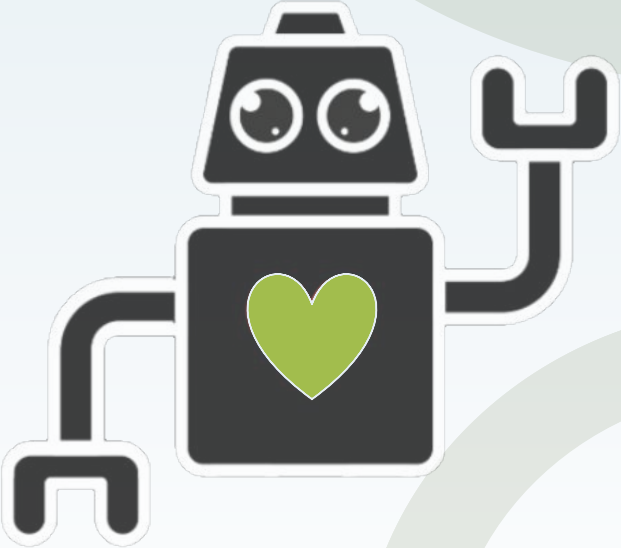
## April 30th

- Final testing of all components + subsystems
- Ensure robot meets design requirements & performance is consistent & durable

## May 2nd - Demo

# THE EMBELLISHER

A bot that seeks to purify and depollute our urban areas



Clean ♲ Improve ♲ Vitalize Our Environment