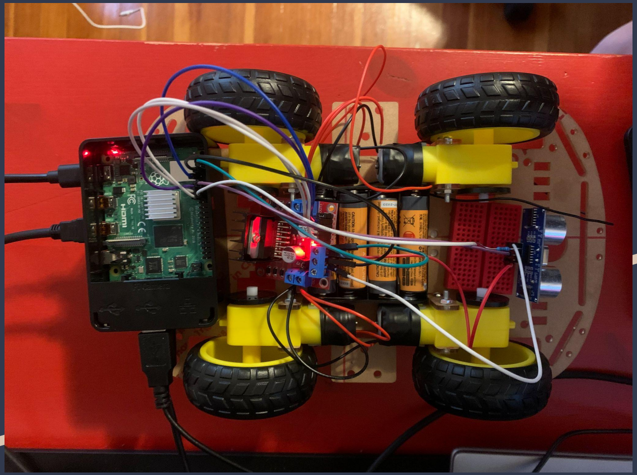
Smart Plant Caretaker Phase 1

Nathan Lee, David Luo, Archie Deng, Ryan Nguyen, Henry Do

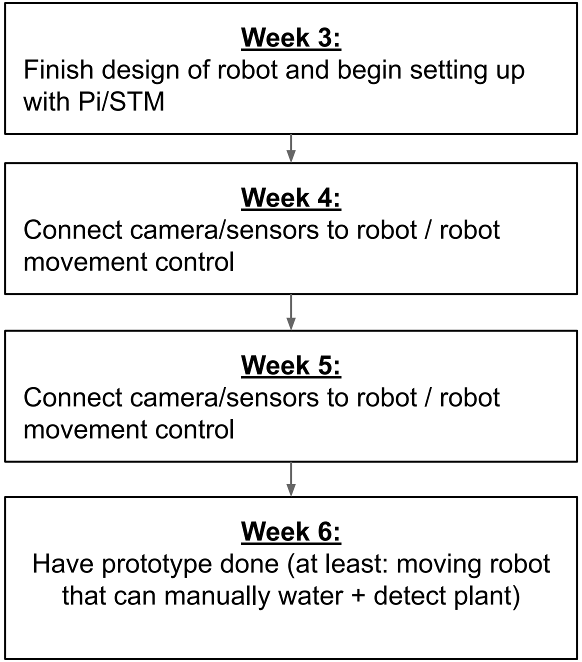
Autonomous Movement



- HC-SR04 Sensor used to sense distance from objects
- L298N H Bridge used to connect motors to raspberry pi
- Turned on different motors by activating different pins
- Used random number generator to pick random movement commands
- Implemented automatic backup when distance from object was less than 25 cm

Robot Movement Deliverables vs. Finished Implementation

Deliverables



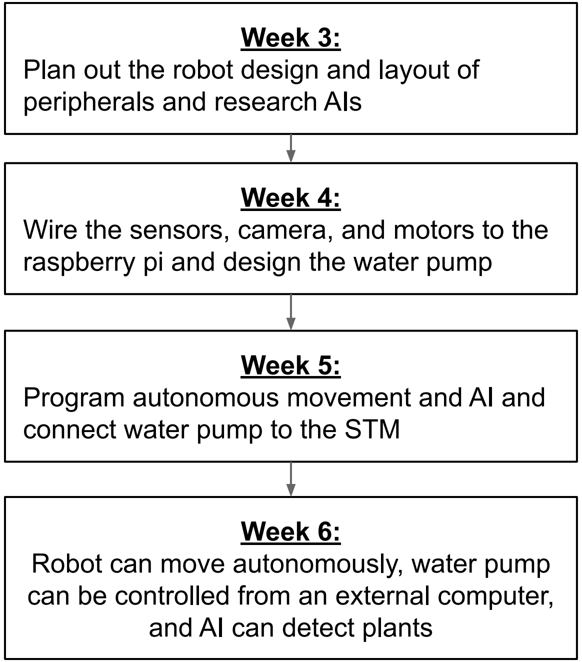
Week 3:
Finish design of robot and begin setting up with Pi/STM

Week 4:
Connect camera/sensors to robot / robot movement control

Week 5:
Connect camera/sensors to robot / robot movement control

Week 6:
Have prototype done (at least: moving robot that can manually water + detect plant)

Finished Implementation



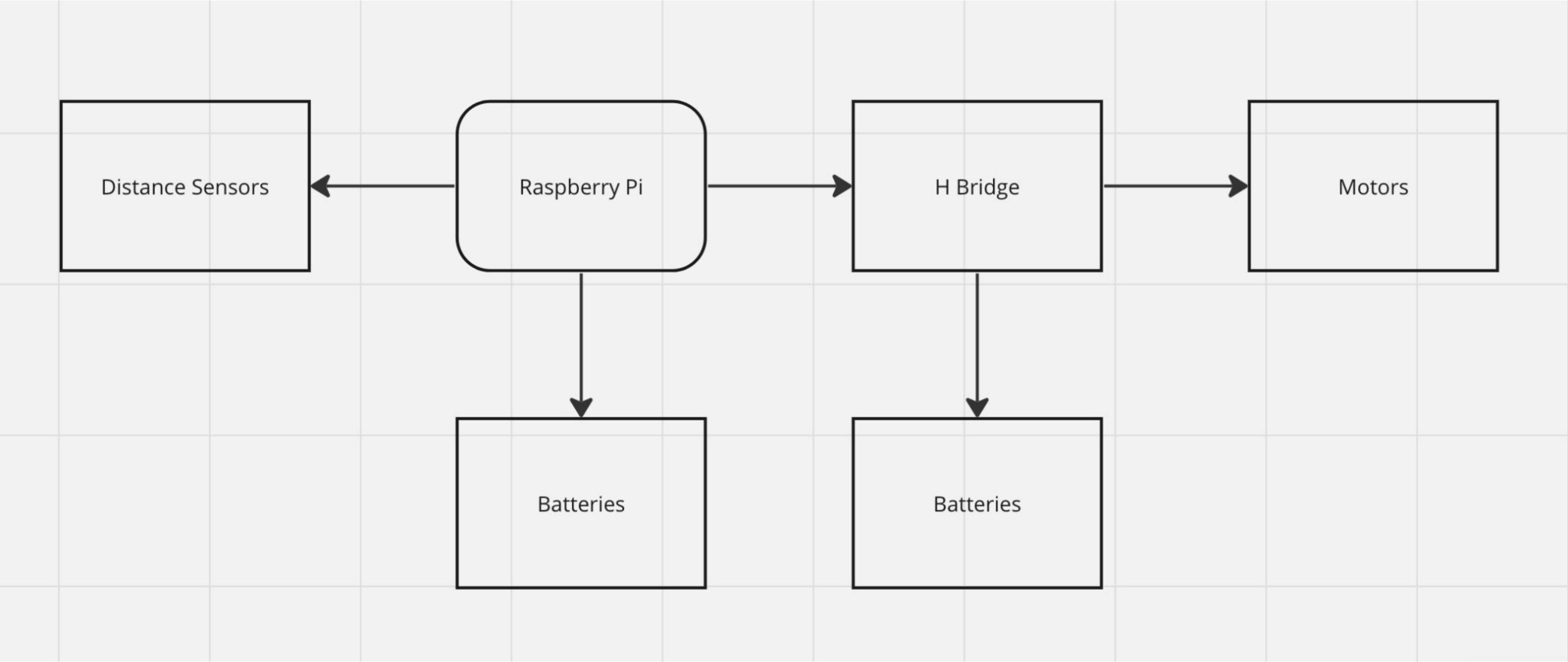
Week 3:
Plan out the robot design and layout of peripherals and research AIs

Week 4:
Wire the sensors, camera, and motors to the raspberry pi and design the water pump

Week 5:
Program autonomous movement and AI and connect water pump to the STM

Week 6:
Robot can move autonomously, water pump can be controlled from an external computer, and AI can detect plants

Robot Movement



Distance Sensors

Raspberry Pi

H Bridge

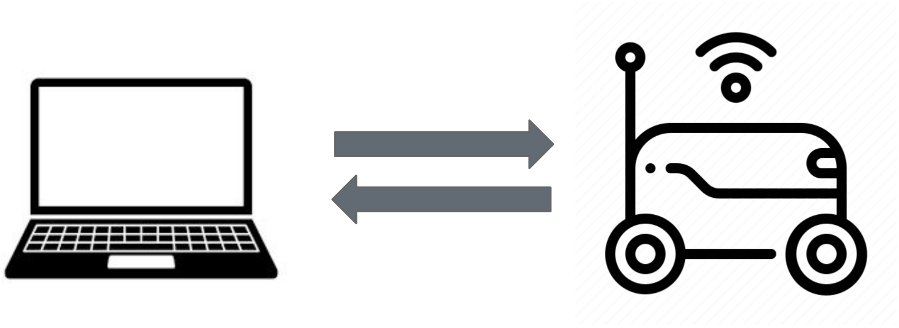
Motors

Batteries

Batteries

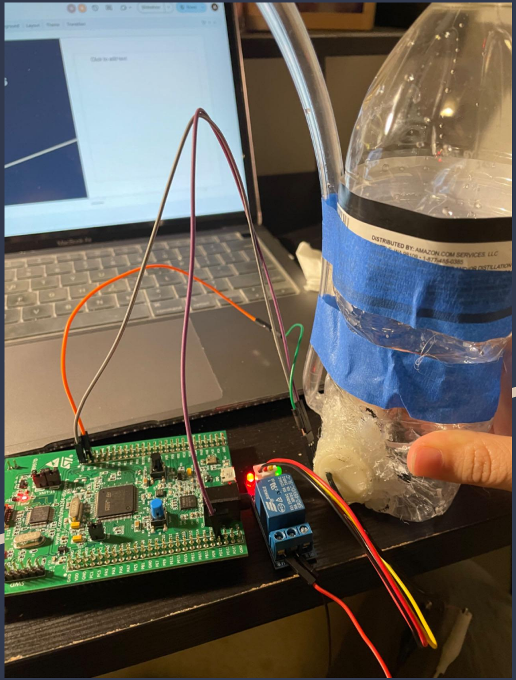
Wifi Communication

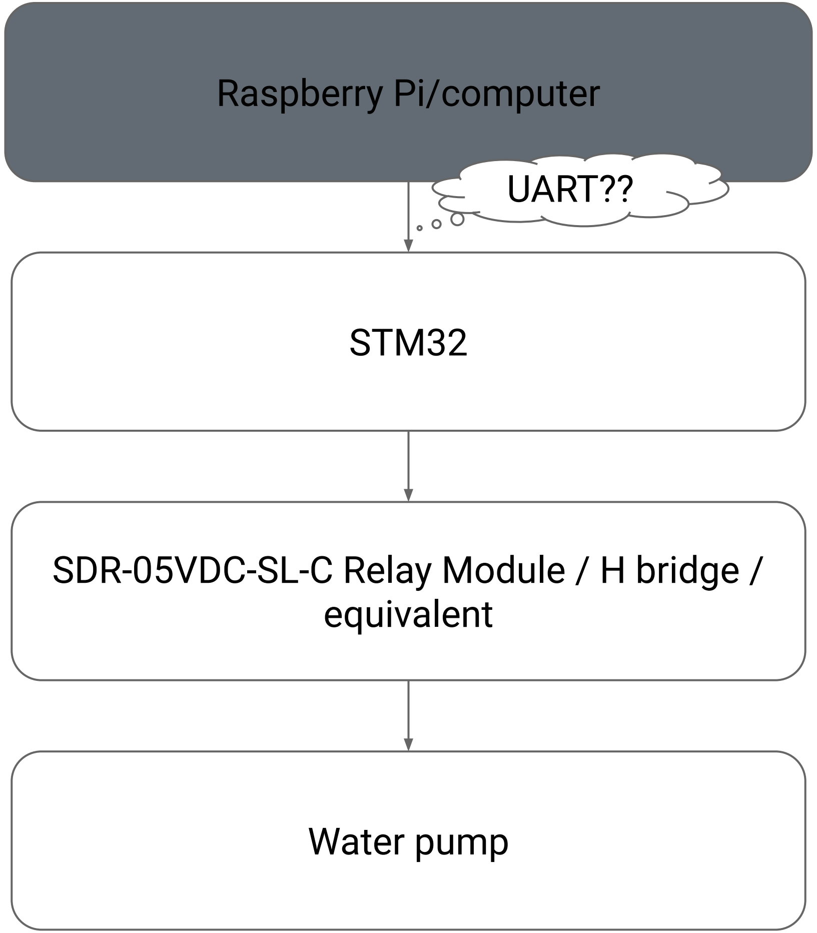
- Used an edimax adapter to allow the robot to connect to wifi
- Allowed the robot to be controlled by an external computer and data to be sent back by the camera



Watering Capabilities

We can control the water pump with the on-board switch





Raspberry Pi/computer

UART??

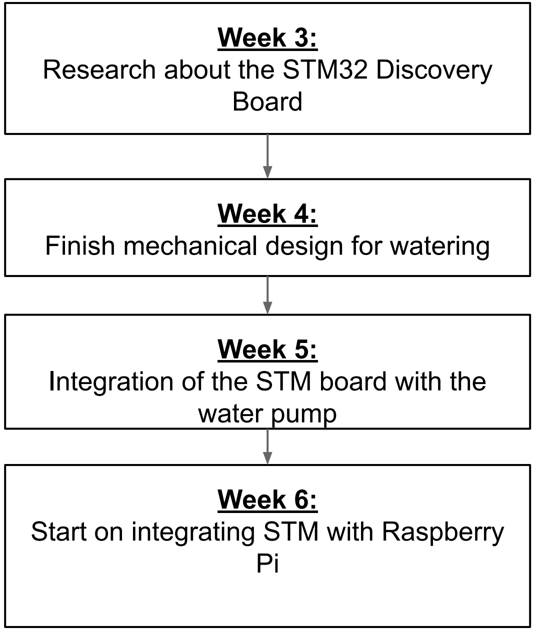
STM32

SDR-05VDC-SL-C Relay Module / H bridge / equivalent

Water pump

Watering Capability Deliverables vs. Finished Implementation

Deliverables



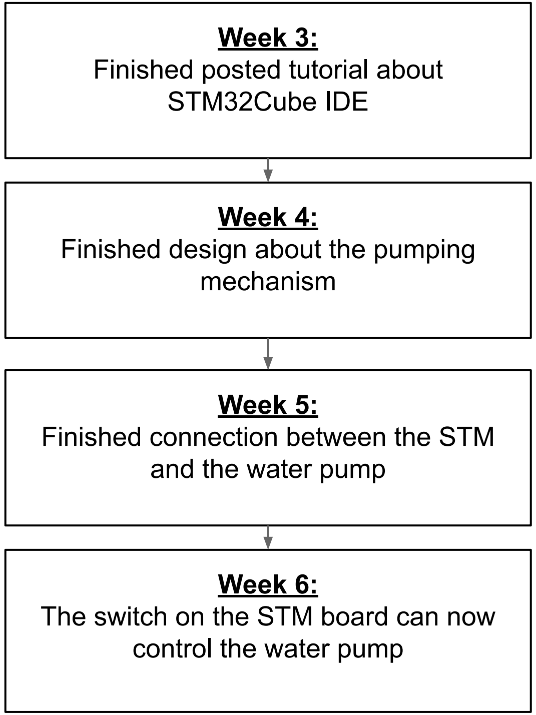
Week 3:
Research about the STM32 Discovery Board

Week 4:
Finish mechanical design for watering

Week 5:
Integration of the STM board with the water pump

Week 6:
Start on integrating STM with Raspberry Pi

Finished Implementation



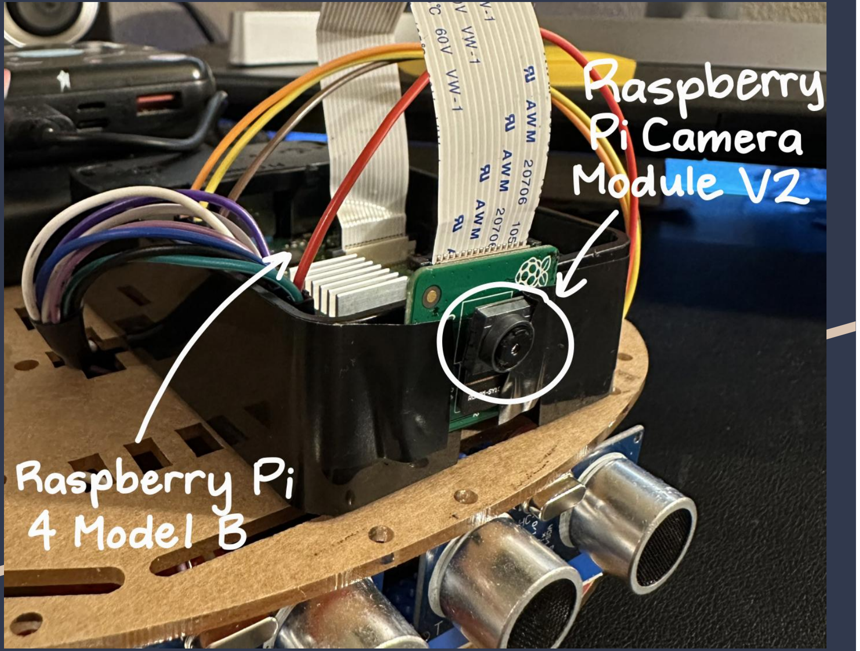
Week 3:
Finished posted tutorial about STM32Cube IDE

Week 4:
Finished design about the pumping mechanism

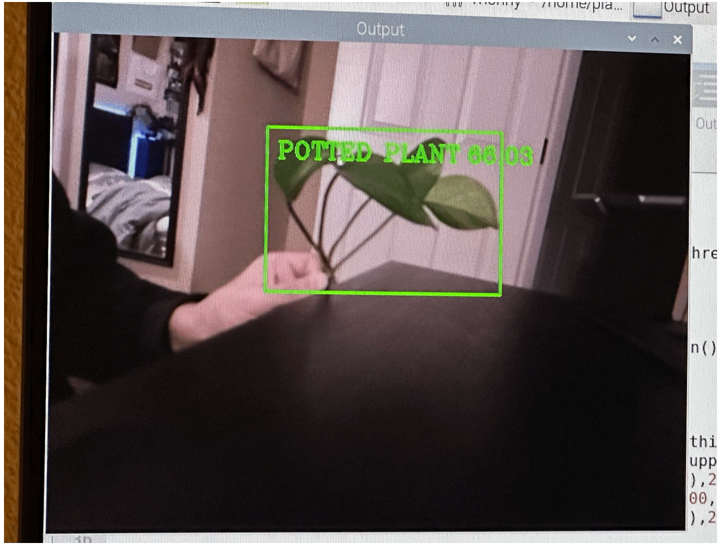
Week 5:
Finished connection between the STM and the water pump

Week 6:
The switch on the STM board can now control the water pump

AI Image Recognition

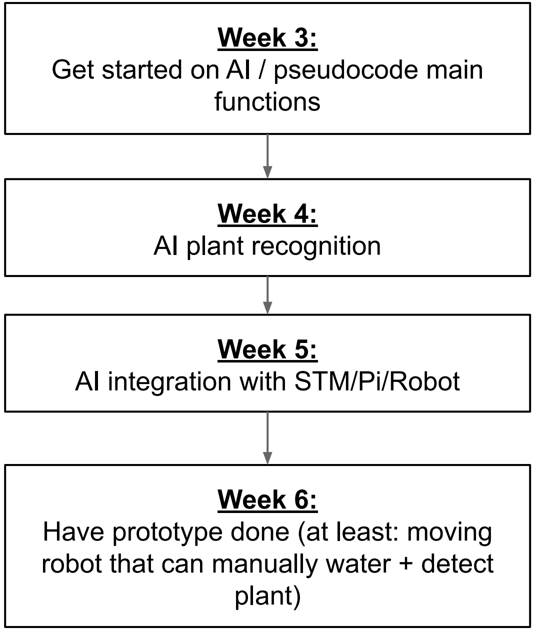


- Used Raspberry Pi and Camera Module
- Used OpenCV for computer vision software
- Distinguish what is and is not a plant
- Found “COCO” common object default dataset that contained “potted plant”
- Used Python script to detect object in camera frame and focus just on the “potted plant”



AI Image Recognition Deliverables vs. Finished Implementation

Deliverables



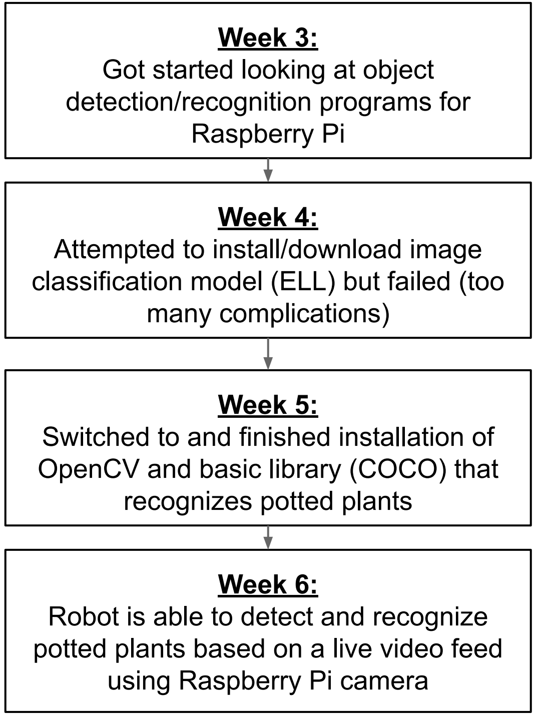
Week 3:
Get started on AI / pseudocode main functions

Week 4:
AI plant recognition

Week 5:
AI integration with STM/Pi/Robot

Week 6:
Have prototype done (at least: moving robot that can manually water + detect plant)

Finished Implementation



Week 3:
Got started looking at object detection/recognition programs for Raspberry Pi

Week 4:
Attempted to install/download image classification model (ELL) but failed (too many complications)

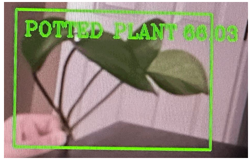
Week 5:
Switched to and finished installation of OpenCV and basic library (COCO) that recognizes potted plants

Week 6:
Robot is able to detect and recognize potted plants based on a live video feed using Raspberry Pi camera

microsoft/**ELL**
Embedded Learning Library

OpenCV

COCO
Common Objects in Context



What’s Next? (AI Image Recognition)

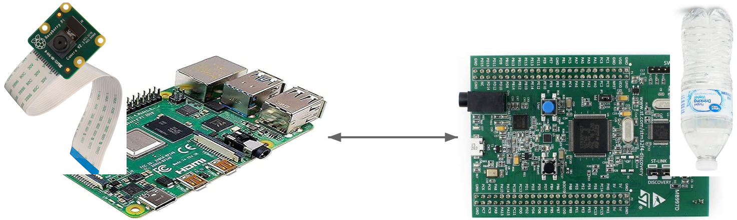
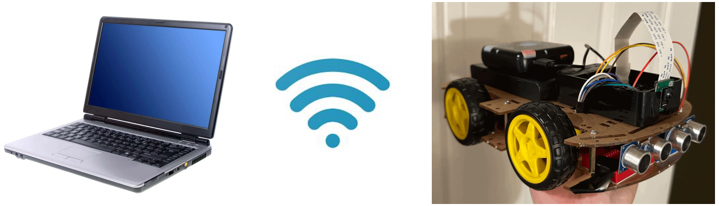
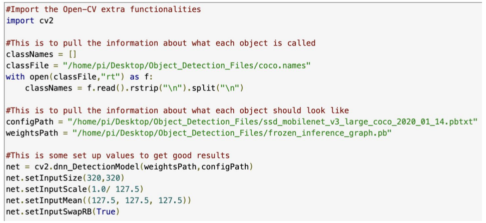


Figure out how to connect the Raspberry Pi to the STM board to control water release



Work on finding a way to control image detection without being plugged in (wirelessly)

Work on finding/creating another library that best represents small plants in garden settings (Lobe AI also in consideration)

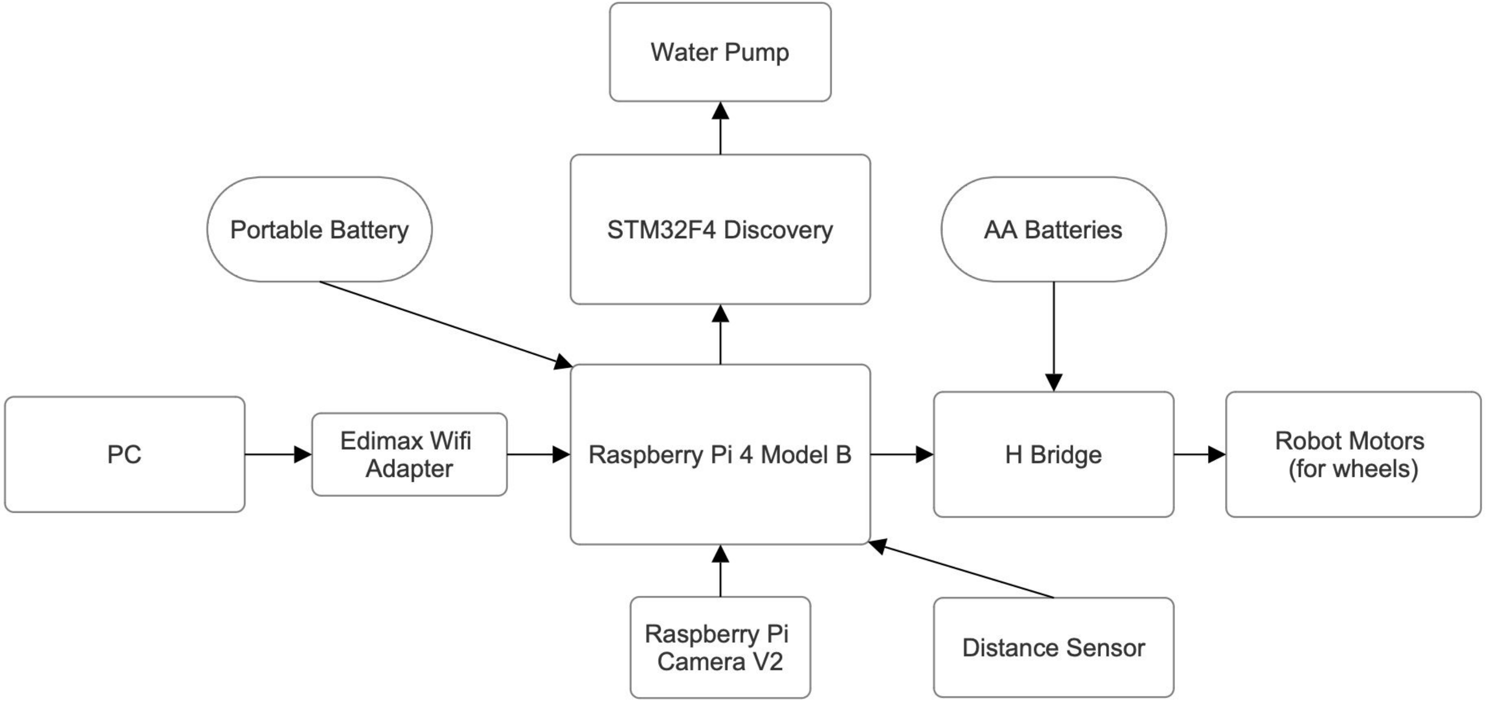


```
#Import the Open-CV extra functionalities
import cv2

#This is to pull the information about what each object is called
classNames = []
classFile = "/home/pi/Desktop/Object_Detection_Files/coco.names"
with open(classFile,"rt") as f:
    classNames = f.read().rstrip("\n").split("\n")

#This is to pull the information about what each object should look like
configPath = "/home/pi/Desktop/Object_Detection_Files/ssd_mobilenet_v3_large_coco_2020_01_14.pbtxt"
weightsPath = "/home/pi/Desktop/Object_Detection_Files/frozen_inference_graph.pb"

#This is some set up values to get good results
net = cv2.dnn_DetectionModel(weightsPath,configPath)
net.setInputSize(320,320)
net.setInputScale(1.0/ 127.5)
net.setInputMean((127.5, 127.5, 127.5))
net.setInputSwapRB(True)
```

Raspberry Pi

Figure out how to make the OpenCV program run automatically (instead of whenever “Run” is pressed)

UML Diagram of Our Smart Plant Caretaker



Thank You