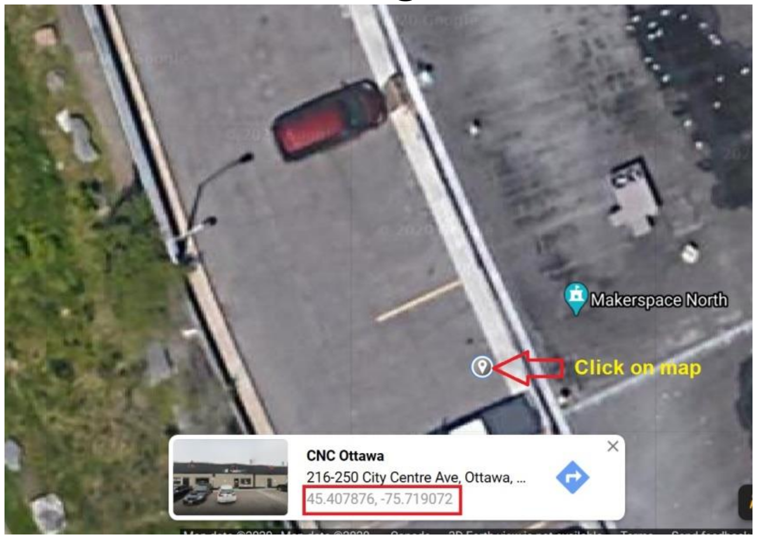
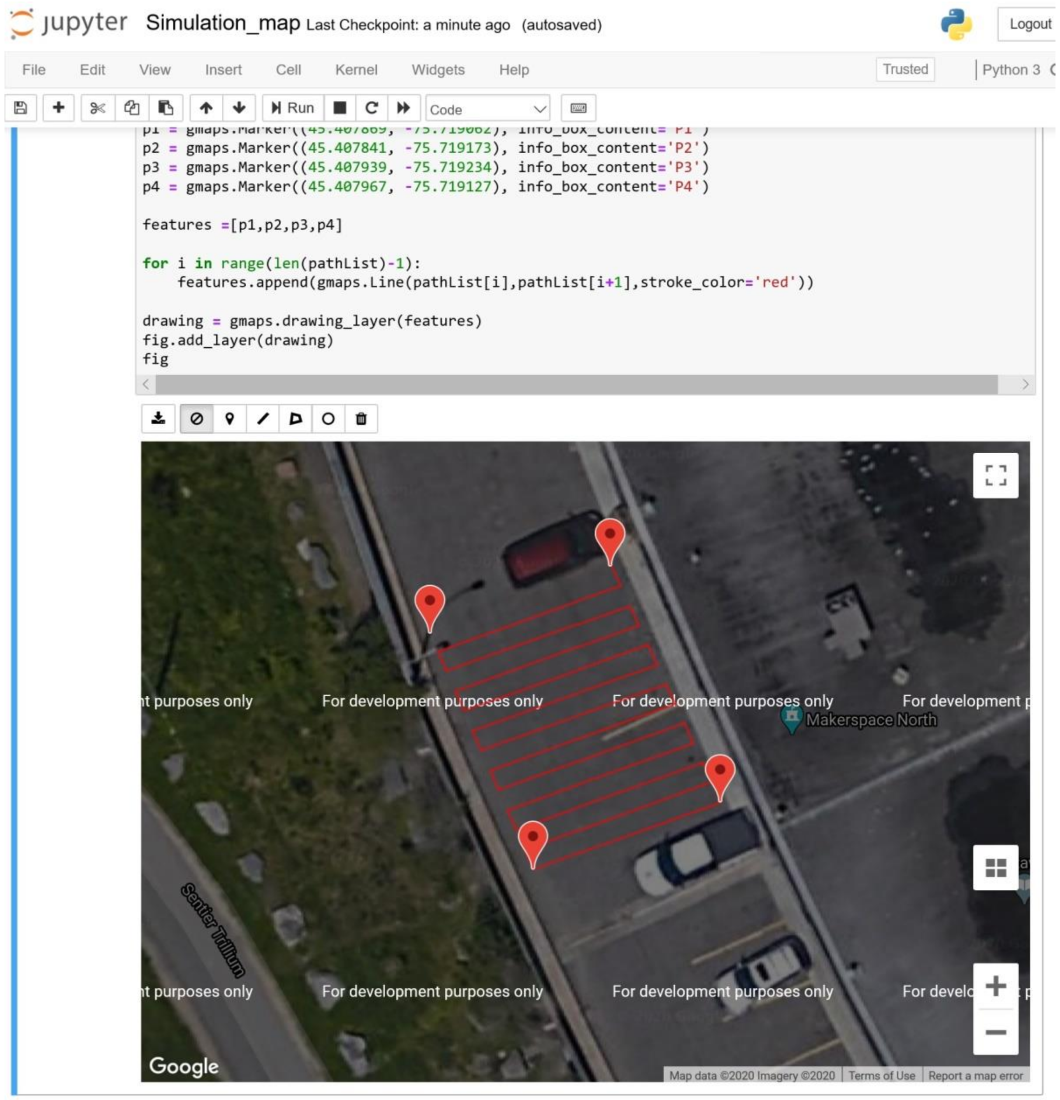
Calculate Path

1- Get Latitude and Longitude coordinates



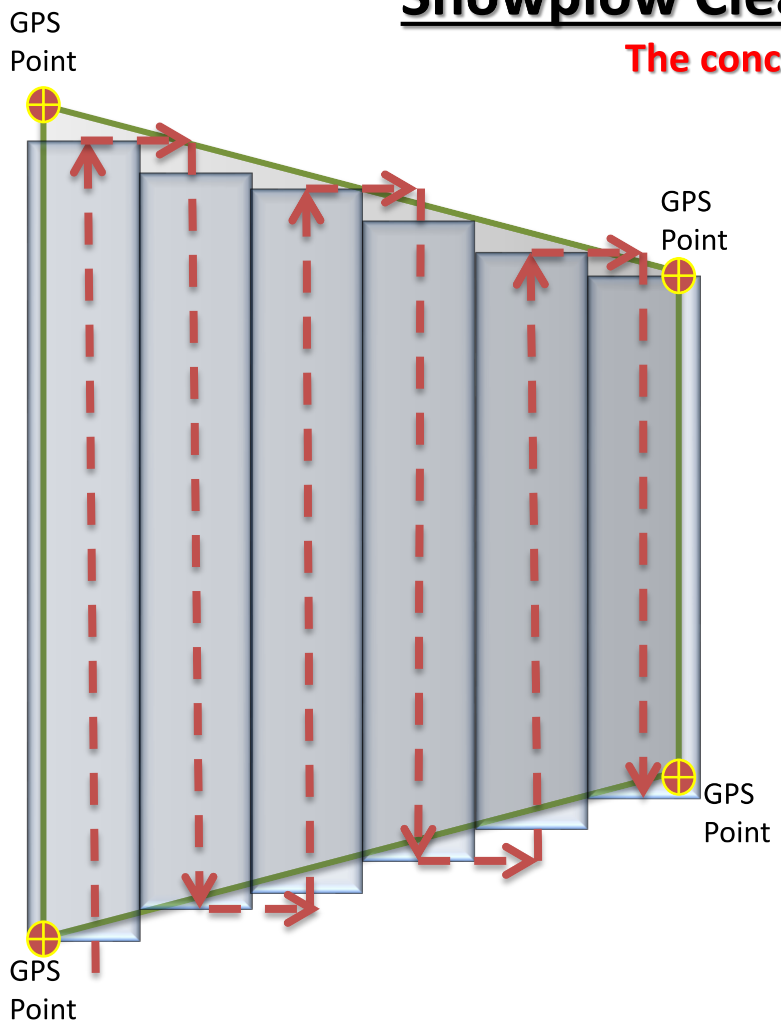
2- Run **Simulation_map.ipynb**
To calculate the waypoint

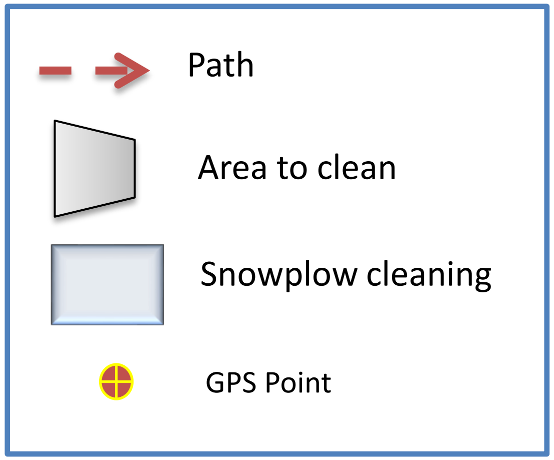

jupyter Simulation_map Last Checkpoint: a minute ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
For development purposes only
Makerspace North
Senter Trillium
Google
Map data ©2020 Imagery ©2020 | Terms of Use | Report a map error

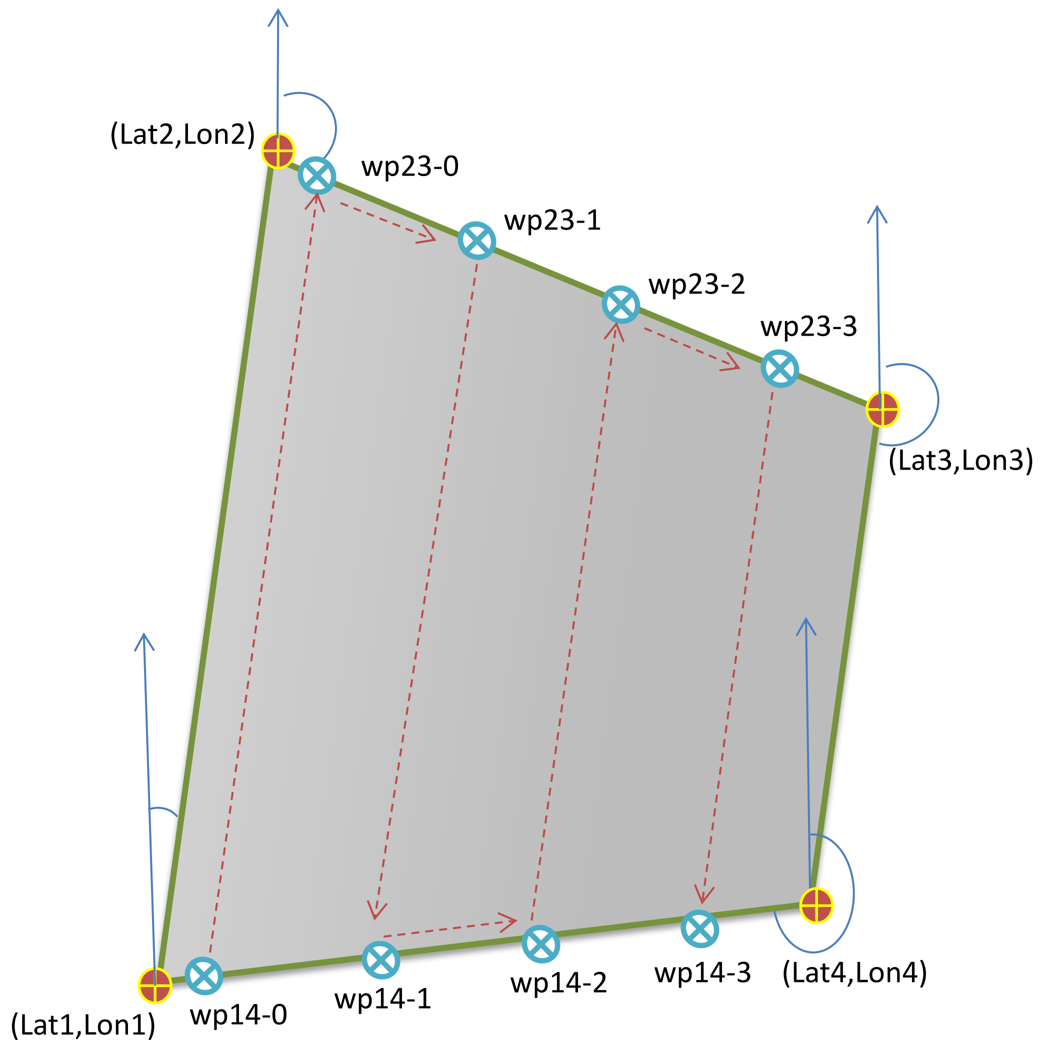
```python
p1 = gmaps.Marker((45.407869, -75.719062), info_box_content='P1')
p2 = gmaps.Marker((45.407841, -75.719173), info_box_content='P2')
p3 = gmaps.Marker((45.407939, -75.719234), info_box_content='P3')
p4 = gmaps.Marker((45.407967, -75.719127), info_box_content='P4')

features =[p1,p2,p3,p4]

for i in range(len(pathList)-1):
    features.append(gmaps.Line(pathList[i],pathList[i+1],stroke_color='red'))

drawing = gmaps.drawing_layer(features)
fig.add_layer(drawing)
fig
```

Snowplow Cleaning Path

The concept

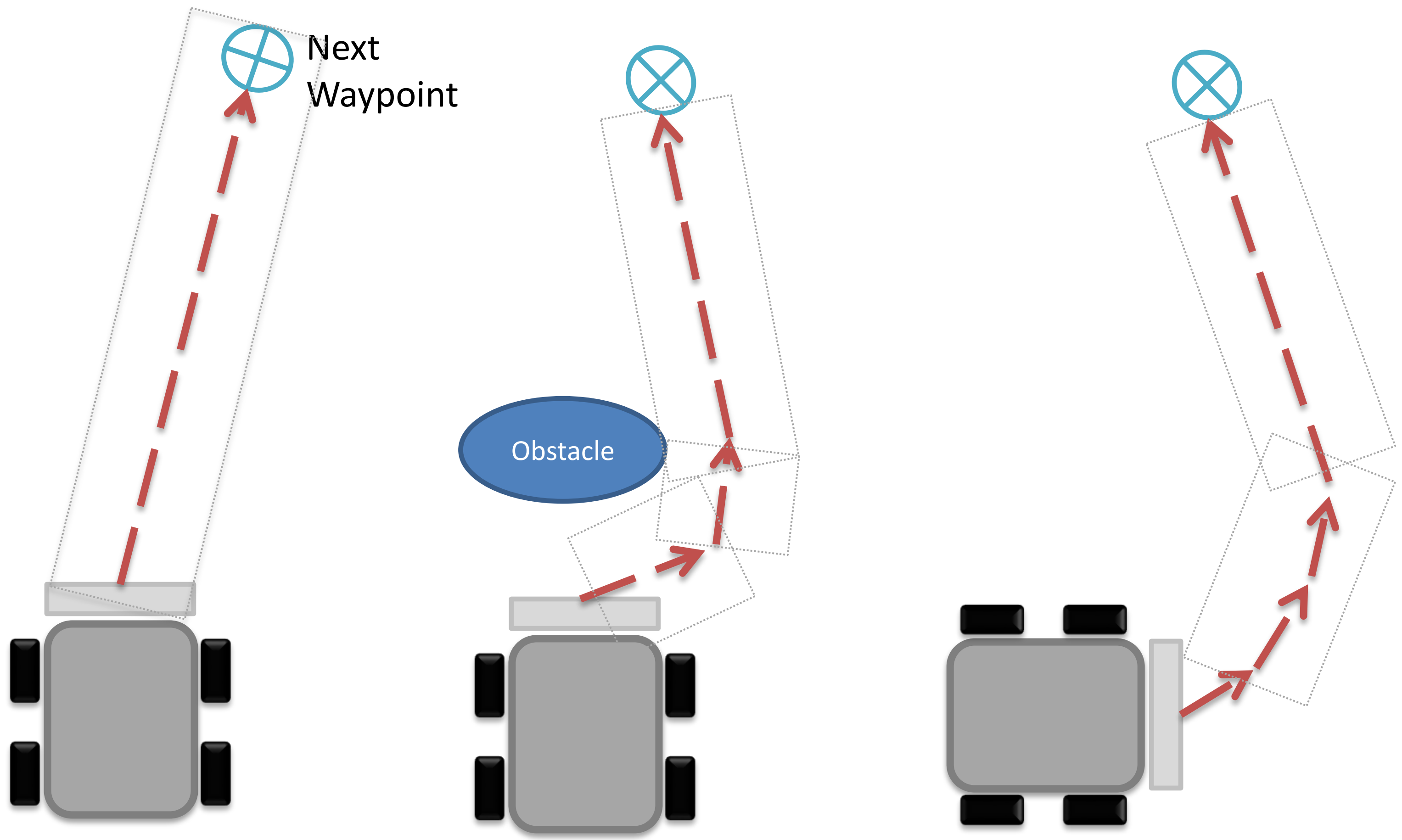




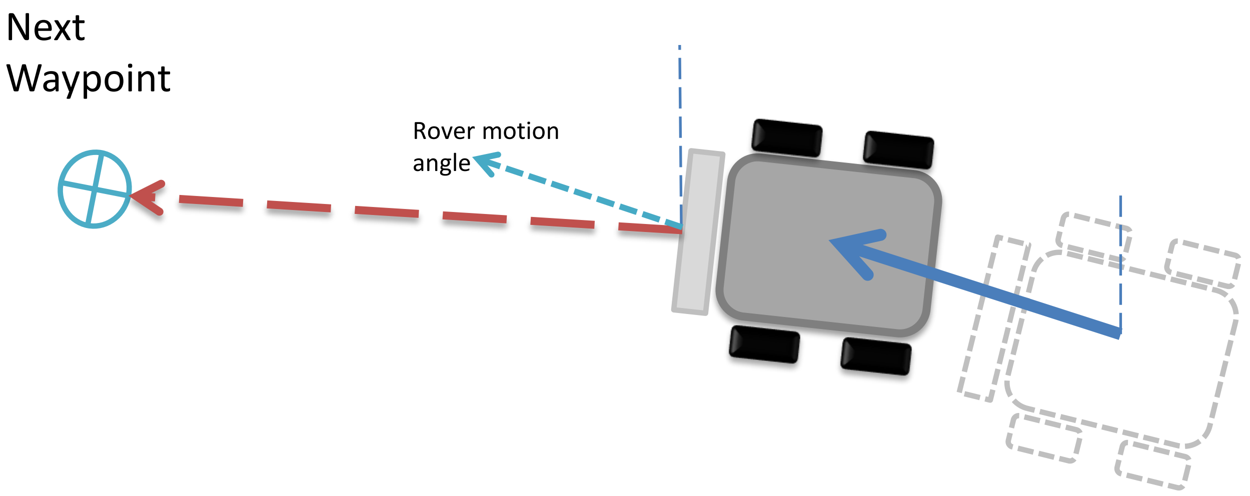


Bearing angle is measured clockwise from the north

Robot to move following waypoints in sequence



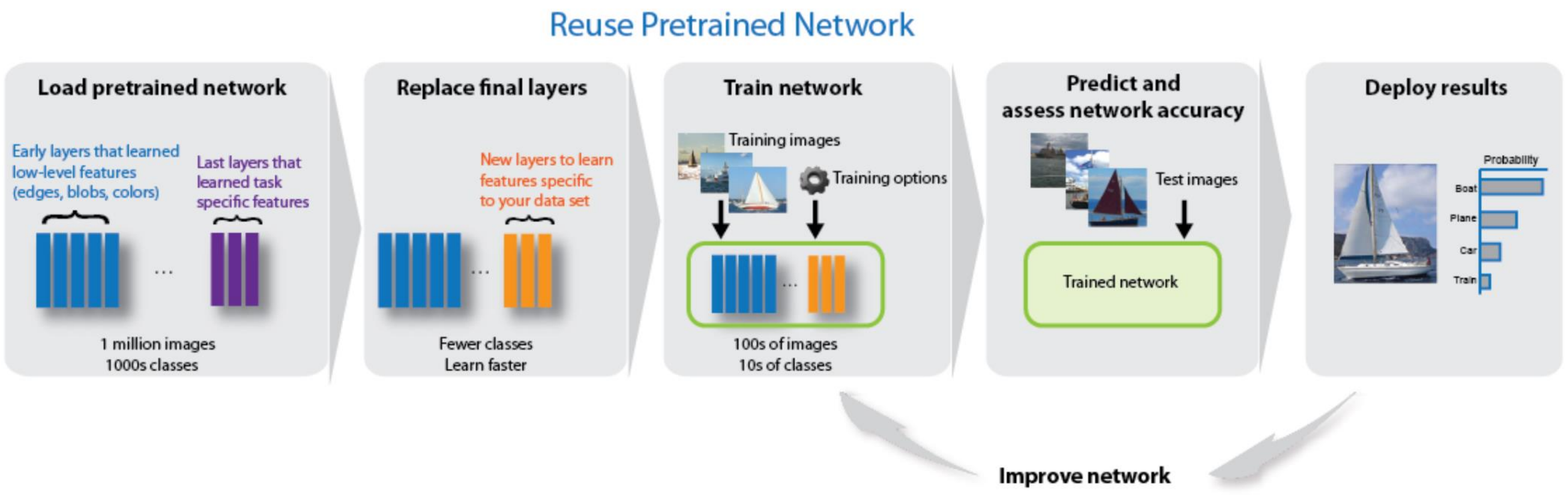
Rover Navigation Blueprint


Next
Waypoint
Rover motion
angle

Obstacle detection

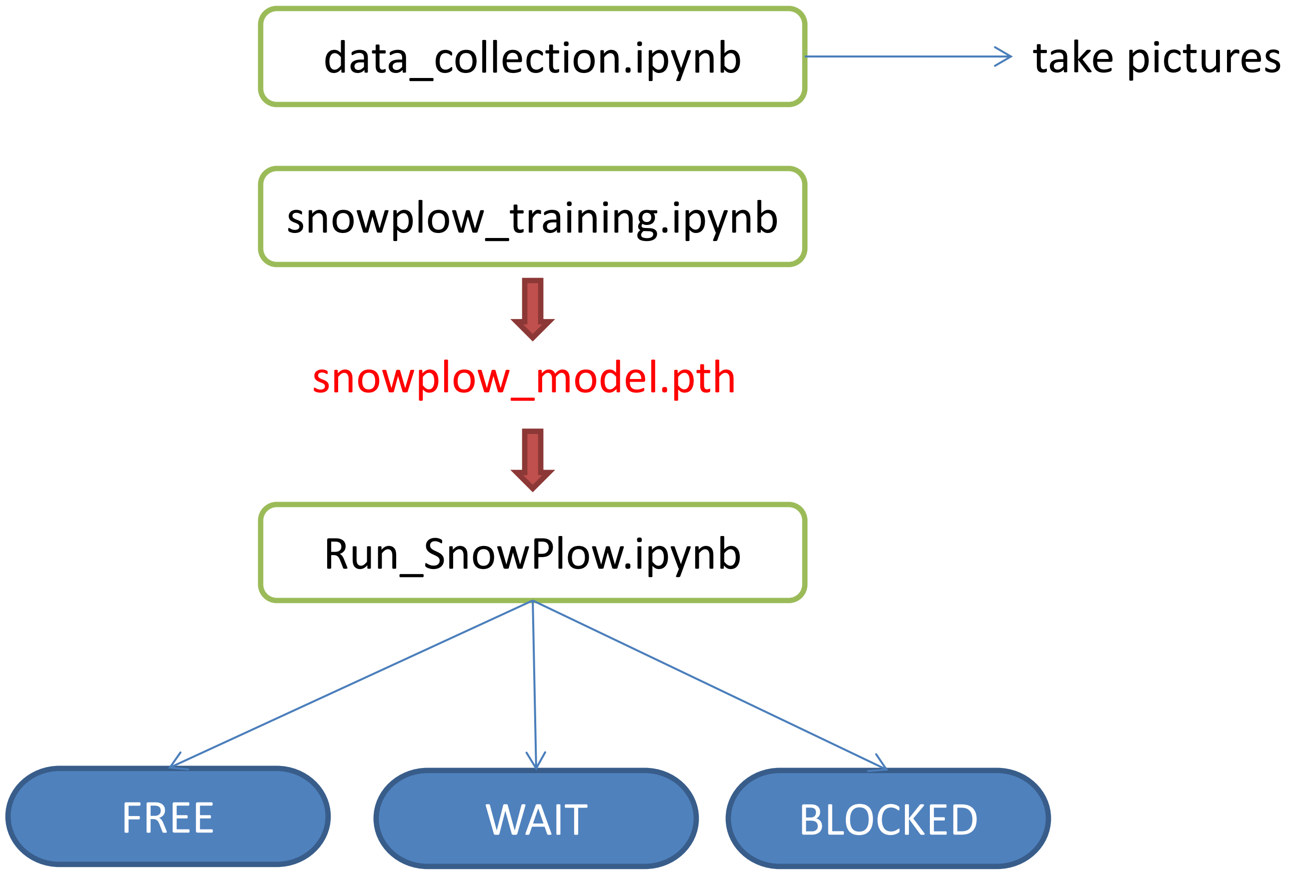
CNN System

AlexNet is a convolutional neural network that is 8 layers deep. We load a pre-trained version of the network trained on more than a million images from the ImageNet database. The pre-trained network can classify images into 1000 objects



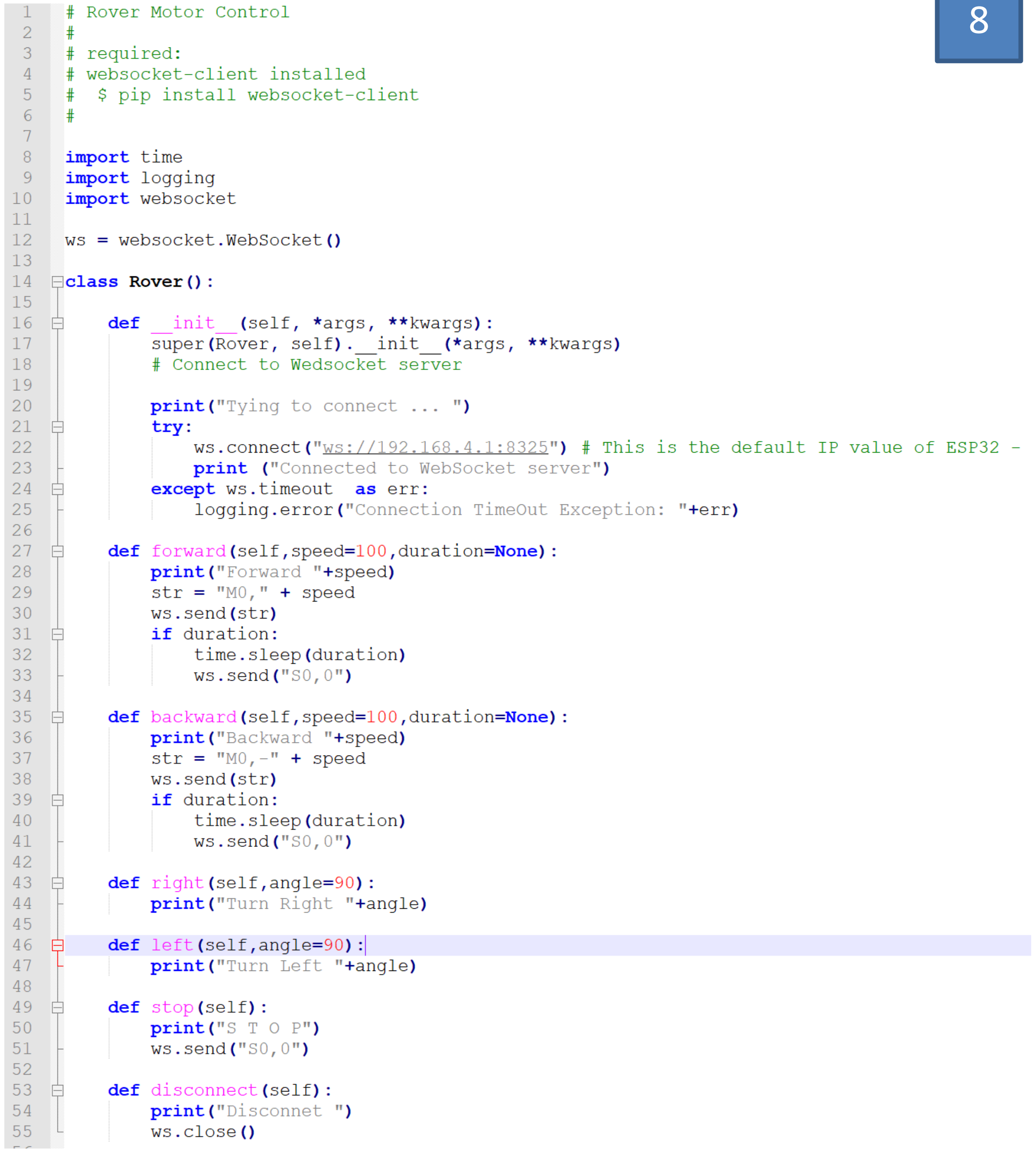
Check https://jetbot.org

Check https://github.com/NVIDIA-AI-IOT/jetbot/wiki


data_collection.ipynb
take pictures
snowplow_training.ipynb
snowplow_model.pth
Run_SnowPlow.ipynb
FREE
WAIT
BLOCKED

Python Classes

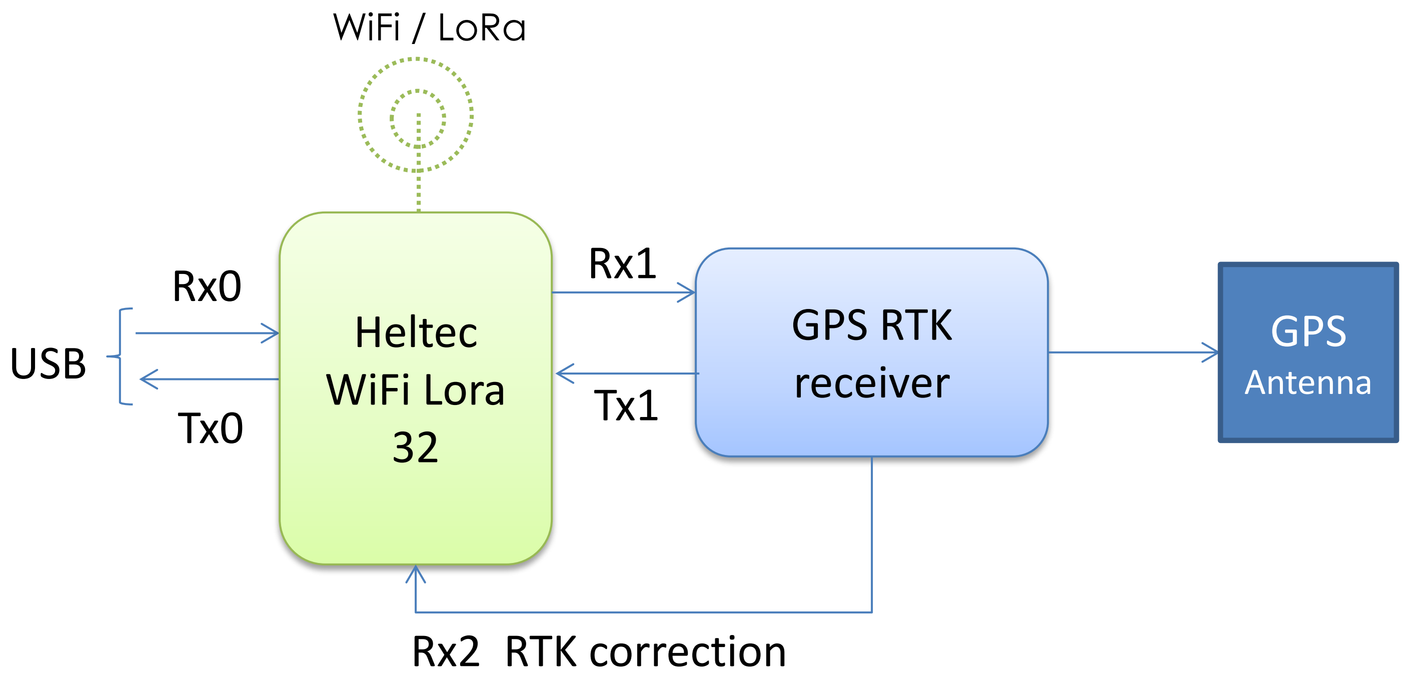
Required to connect Notebook and Camera, ESP32, ...



```
# Rover Motor Control
#
# required:
# websocket-client installed
#   $ pip install websocket-client
#

import time
import logging
import websocket

ws = websocket.WebSocket()

class Rover():

    def __init__(self, *args, **kwargs):
        super(Rover, self).__init__(*args, **kwargs)
        # Connect to Wedsocket server

        print("Tying to connect ... ")
        try:
            ws.connect("ws://192.168.4.1:8325") # This is the default IP value of ESP32 -
            print ("Connected to WebSocket server")
        except ws.timeout  as err:
            logging.error("Connection TimeOut Exception:  "+err)

    def forward(self,speed=100,duration=None):
        print("Forward  "+speed)
        str = "M0," + speed
        ws.send(str)
        if duration:
            time.sleep(duration)
            ws.send("S0,0")

    def backward(self,speed=100,duration=None):
        print("Backward  "+speed)
        str = "M0,-" + speed
        ws.send(str)
        if duration:
            time.sleep(duration)
            ws.send("S0,0")

    def right(self,angle=90):
        print("Turn Right  "+angle)

    def left(self,angle=90):
        print("Turn Left  "+angle)

    def stop(self):
        print("S T O P")
        ws.send("S0,0")

    def disconnect(self):
        print("Disconnet  ")
        ws.close()
```

Position System

RTK GPS Base Station



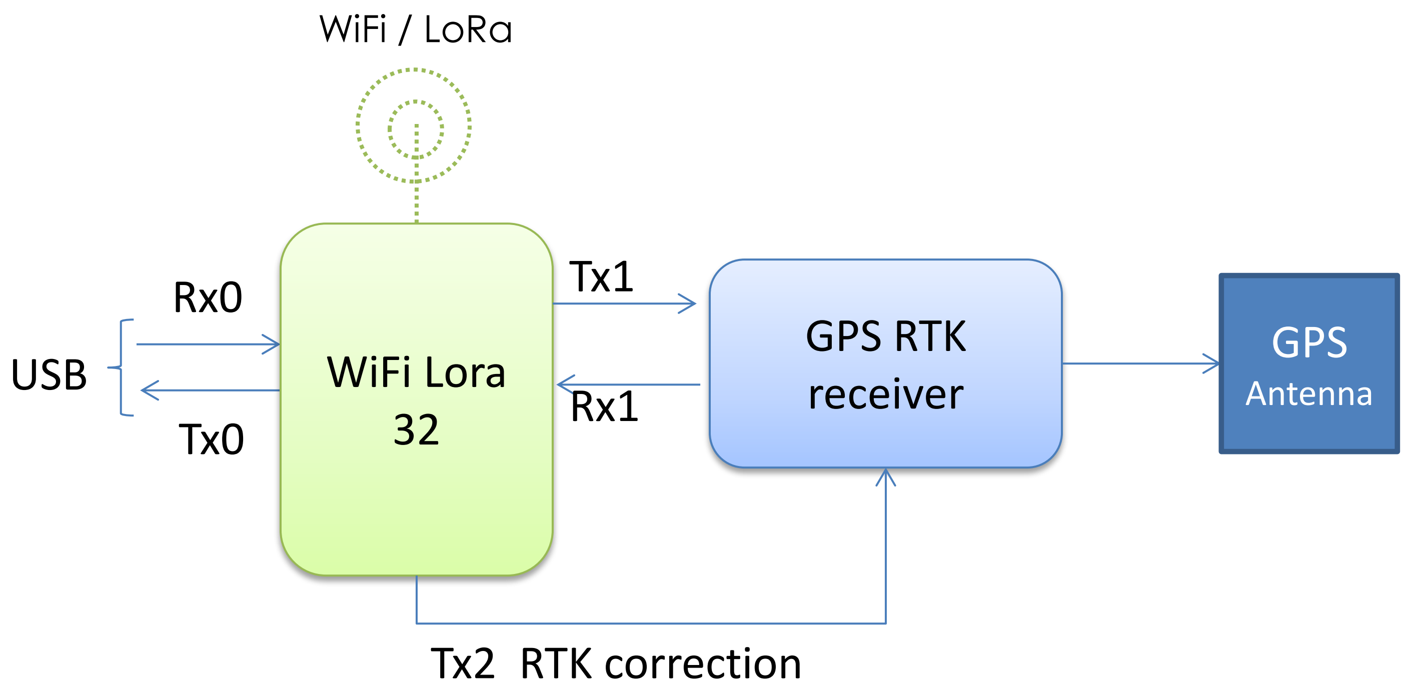
ESP32

Rx2 Receive RTK correction

Send correction data WiFi or LoRa

Note: TX and Rx are referred to ESP32

RTK GPS Rover



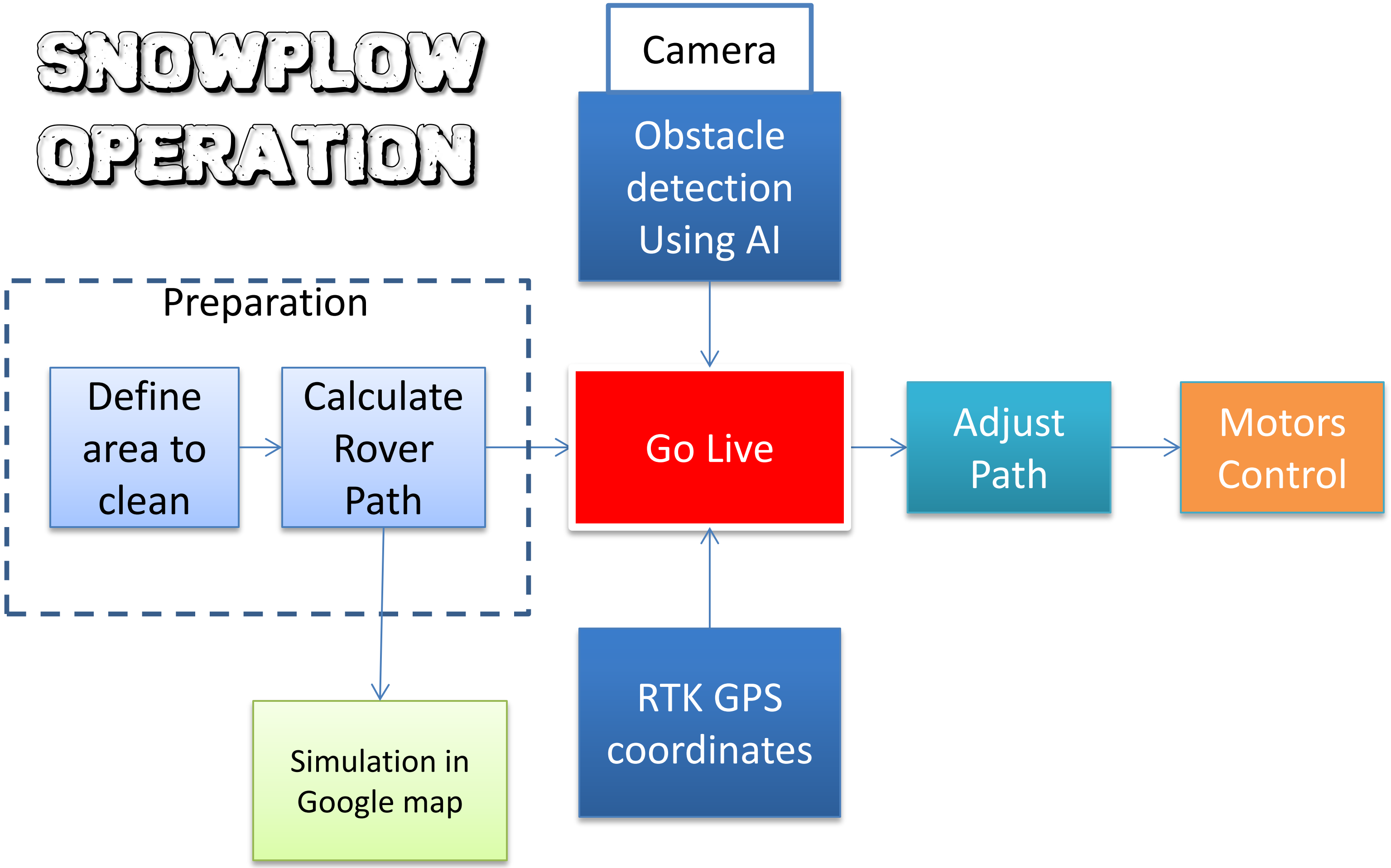
ESP32

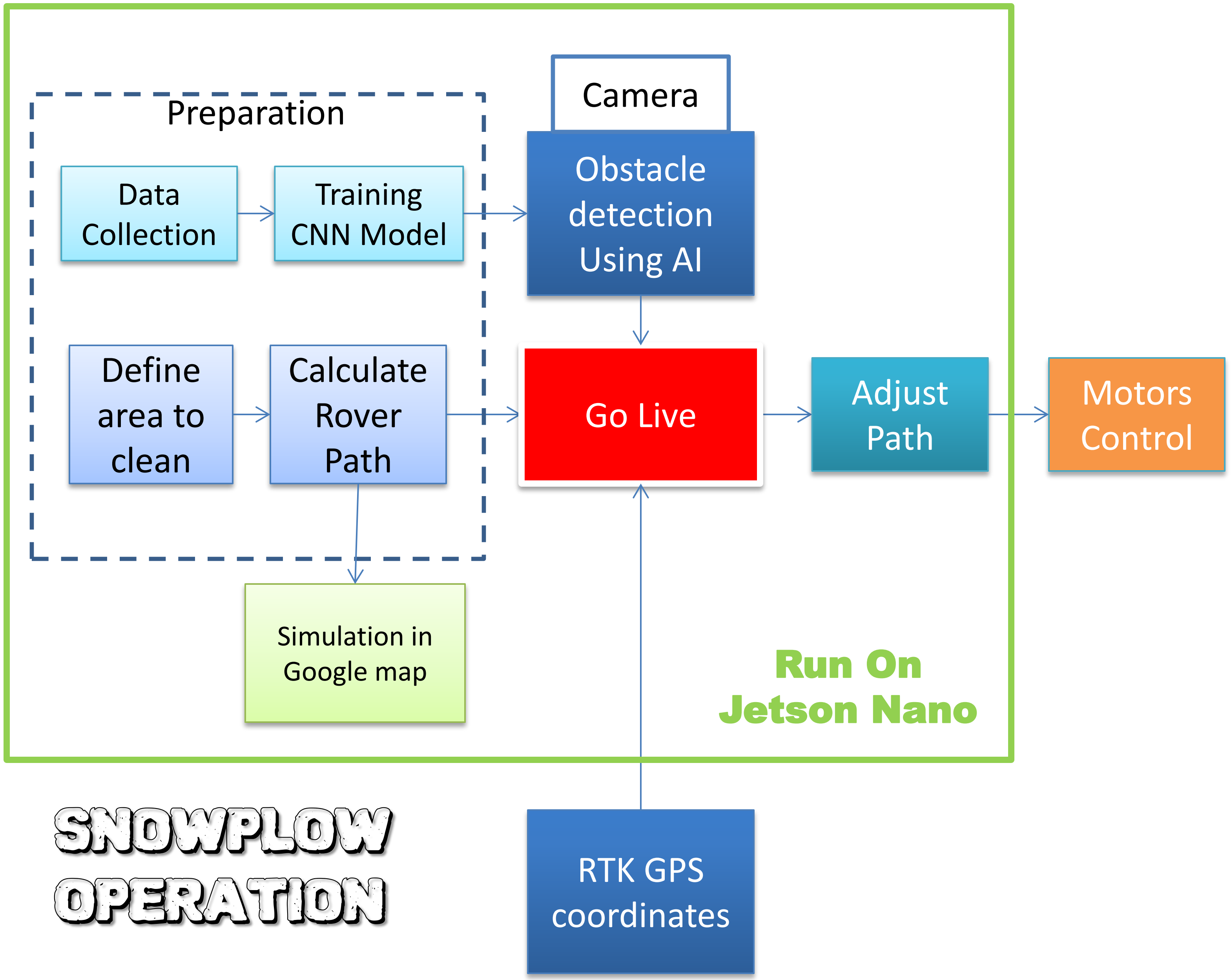
Receive RTK correction Wifi or LoRa

Tx2 Send RTK correction to GPS

Rx1 Get GPS corrected coordinate

SNOWPLOW OPERATION





SNOWPLOW OPERATION