Autonomous New Planet Rover

Jiajia Chen
Robotics, Autonomous Vehicles, Drones, and AI
Summer 2021

Content

- Problem Statement
- Project Plan
- Challenges
- Data Collection and Tagging
- Model Training
- Combine Multiple Use Cases
- Demo 1
- Demo 2
- Lessons Learned and Future Work

Problem Statement

As humans explore outer space, we need lots of autonomous new planet rovers which can -

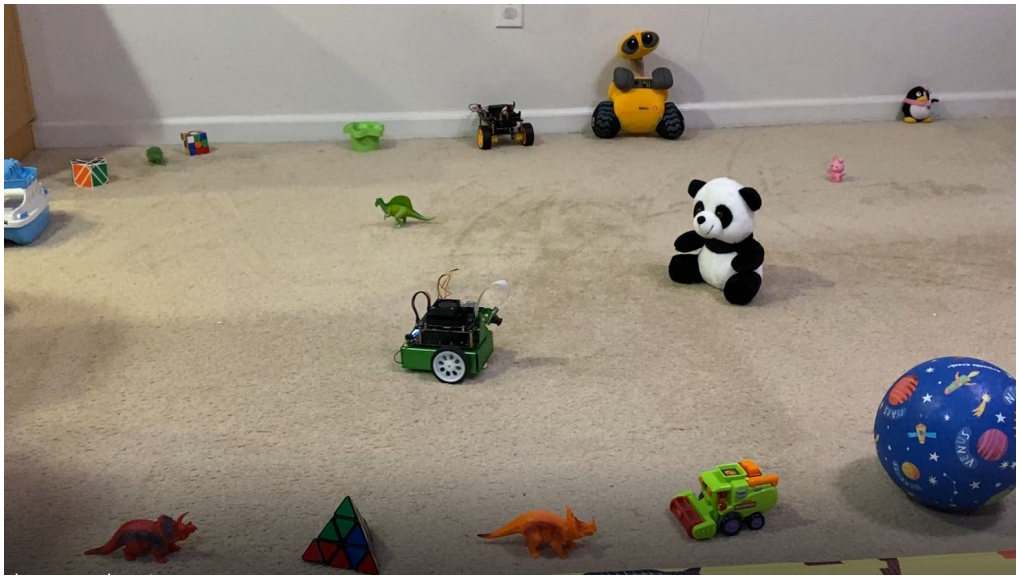
- Continuously navigate and explore details at the surface and transmit data of interest
- Protect itself from damage by avoiding crashing into obstacles or falling over edges
- Pay special attention to certain objects or events and steer safely there to observe

Due to vast distances, human intervention is infeasible. The rovers must be fully autonomous.

Project Plan

A plan that works for a solo project with a financial budget below $500 and time allocation of 10 hr/week for 5 weeks

- Hardware:
  - Waveshare Jetbot AI Kit
  - Jetson Nano Development Kit 4GB
- Software and development tools: Nvidia image of Jetbot, Python, Jupyter Lab server, Keras, PyTorch, NumPy, Nodejs, Github, Ubuntu, Balena Etcher
- Data collection: 2 classes, ~250 images
- Model training:
  - AlexNet + transfer learning
  - COCO (Common Objects in Context)
- Use cases:
  - Collision avoidance
  - Object recognition
  - Goal Setting & Object following



Challenges

- Hardware
  - One motor of the Waveshare Jetbot doesn’t work properly resulting in the Jetbot’s slight leaning towards left when it’s programmed to go straight forward
  - WiFi module of the Sparkfun Jetbot doesn’t work at all
  - Received a wrong chassis board for the Sparkfun unit
  - HDMI port had issues
  - OLED didn’t work
  - Image lagging due to the bandwidth of WiFi

Spinning around

Imaging lagging

With a tail

Data Collection and Tagging

No resource to collect millions of images for training a neural network from scratch

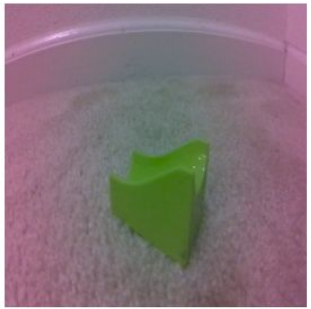
- 2 classes:
  - Free: 120 images
  - Blocked: 130 images

```
[5]: display(image)
     display(widgets.HBox([free_count, free_button]))
     display(widgets.HBox([blocked_count, blocked_button]))
```

1 add free

0 add blocked

```
[5]: display(image)
     display(widgets.HBox([free_count, free_button]))
     display(widgets.HBox([blocked_count, blocked_button]))
```



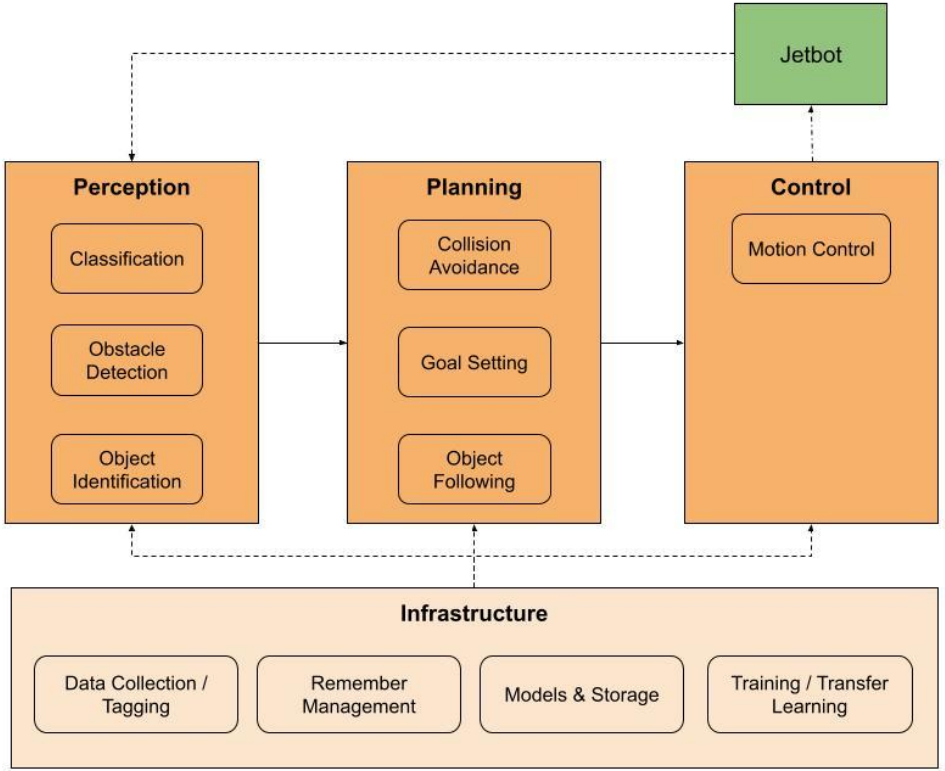
1 add free

1 add blocked

Model Training

Use a pre-trained model and perform transfer learning to train the model with my own 250 images of obstacles in my test environment.

- AlexNet
- COCO Dataset



Combine Multiple Use Cases

Problem Statement		
Pay special attention to certain objects or events and steer safely there to observe	Protect itself from damage by avoiding crashing into obstacles or falling over edges	Continuously navigate and explore details at the surface and transmit data of interest
Goal Setting & Object Following	Collision Avoidance	Object Recognition
		

Demo


display(round_widget)
display(objectofinterest_widget)
Python 3
8 hours ago
8 hours ago
8 hours ago
9 hours ago
seconds ago
8 hours ago
8 hours ago
blocked
0.01
speed
0.20
turn gain
0.20
[]]]

Lessons Learned and Future Work

Lessons Learned

- Working with actual hardware exposes you to all kinds of real-life issues.
- There will absolutely be bottlenecks in hardware, networking, processing, memory, storage, etc, so based on the use cases of priority, you have to make several trade-offs.
- Training for fully autonomous vehicles in a remote planet requires training not only for safe states, but also for how to get back to a safe state from an unsafe one.

Future Work

- Add more cameras and sensors to get better perception of relative altitude of different points in the image
- Apply deep reinforcement learning
- motion backwards away from moving objects
- Take the rover out among the soil, pebbles, rocks, ditches, etc

https://github.com/acjj0/autonomous-rover/