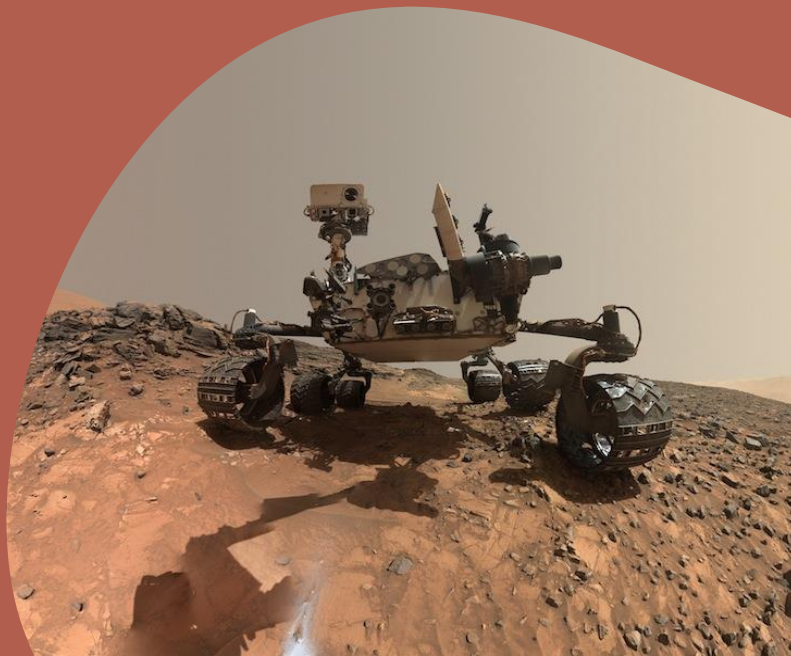
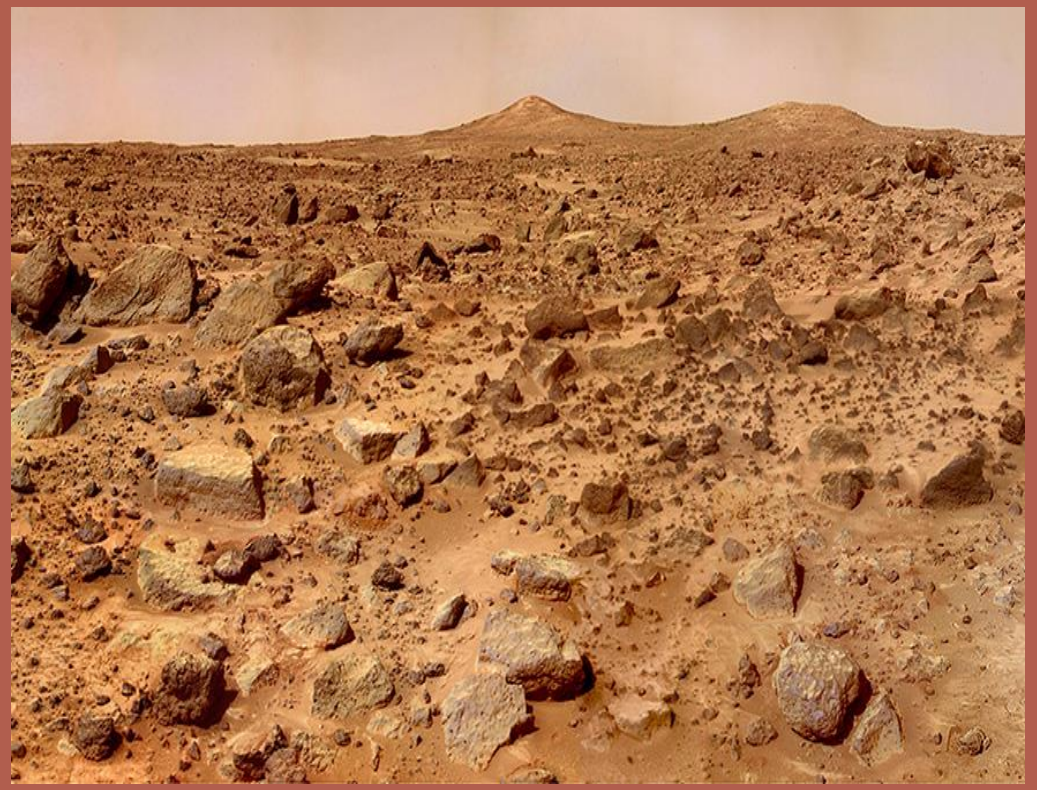
Rover Maneuvering with AI

Dey, Arnab & Aravind, Anku



Trivia

v/s



Considerations

- Importance of terrain in navigation
  - Relate to self-driving car or robotic vacuum cleaner
- Spirit, Opportunity, Curiosity getting stuck in sand
  - Spirit ended mission thus
  - Purgatory Dune, Hidden Valley, etc
- Limited mission duration in terms of Sols – Mars days
- Earth/Mars data bandwidth limitations – 60 MB/Sol
- Perseverance’s earth weight is 2260 lb-wt
- SPOC-Lite & VeeGer

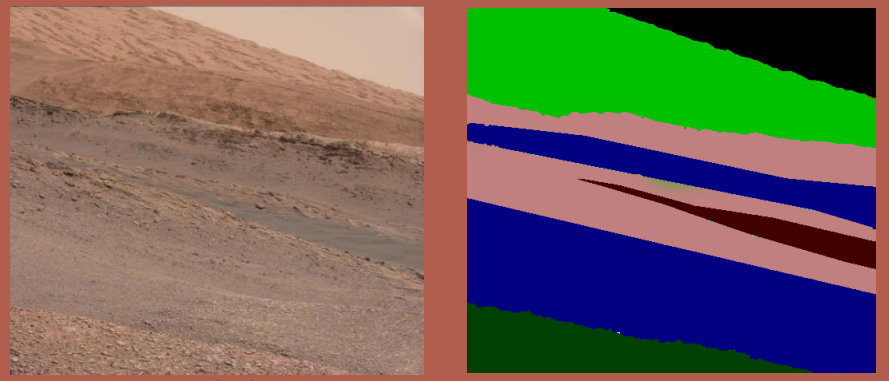
Global & Local Planners

- Global planner
  - For initial heuristics
  - Needs a segmented satellite image of the mission site
  - Semantic Segmentation not implemented
    - Preprocessing constraints
    - Lack of labelled data
    - Difficulty to identify and distinguish terrains
  - Assumption: segmented image is available
- Local planner
  - To avoid unfavorable terrain
  - Needs segmented local image from onboard cameras

Source: MSL Curiosity

Semantic Segmentation

- DeepLabV3+[1]
  - NASA SPOC[2] is based on DeepLab
  - TensorFlow model
- Transfer Learning
  - Xception65_coco_voc_trainaug – initial checkpoint [1]
    - Pretrained on PASCAL VOC, COCO & ImageNet dataset
    - 21 classes
- Training
  - 110 images – our own dataset(acquired images and labelled it)
  - 6 Classes
    - 5 Terrains
    - Background (sky)
  - 10000 epochs (stepped approach 1800, 4000, 10000)
  - Optimizer: Momentum
  - Loss function: Cross-entropy loss
  - Metrics: mIOU
  - Time: 11.25 to 16.75 hours on Chameleon Cloud



Source: MSL Curiosity

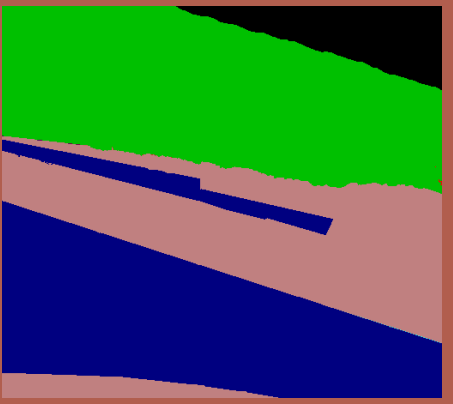
1. GitHub.“Tensorflow/Models.https://github.com/tensorflow/models
2. https://nasa-jpl.github.io/SPOC/

Results

- After 10000 iterations
  - Cross-entropy loss : 0.2593
  - Accuracy – 62.41%
- Inference
  - Small dataset
  - Biased
    - the classes with better accuracy had more images to train
  - Labelling errors
    - At times, it's difficult to define boundary for terrains
    - It's a tedious task
  - Require fine tuning
  - Require more epochs
    - Around 30000

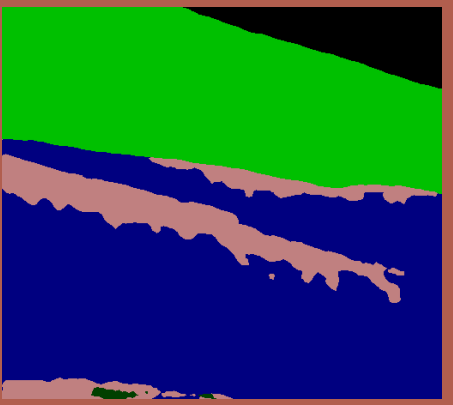
Class	mIOU
Terrain1	0.7144
Terrain2	0.2127
Terrain3	0.5446
Terrain4	0.3955
Terrain5	0.941
Background	0.9361
Average	0.62405

Original image



Our label

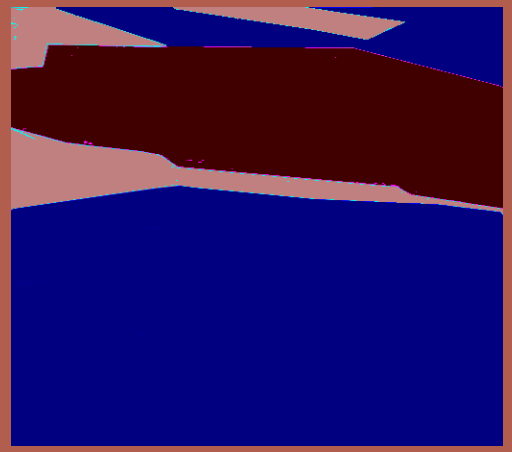
"To be honest, this prediction was better than our label"



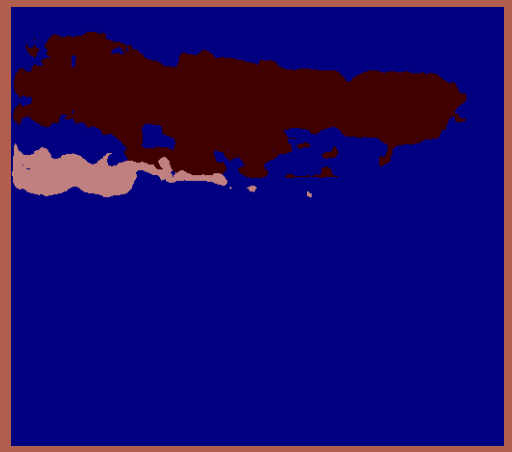
Predicted label

More Visualization

Original image

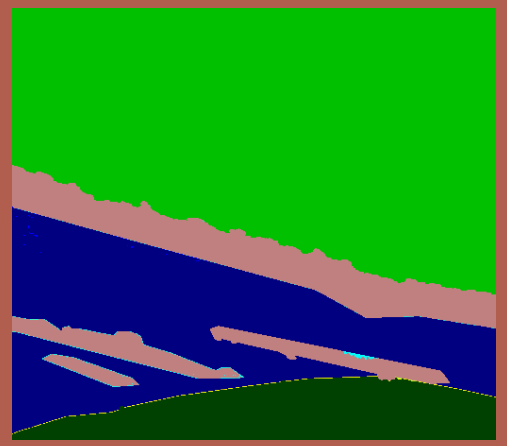


Our label

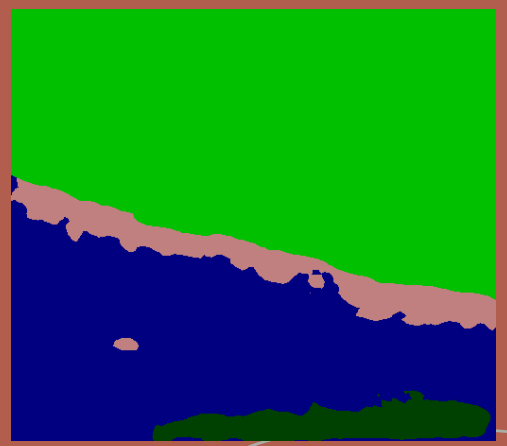


Predicted label

Original image



Our label

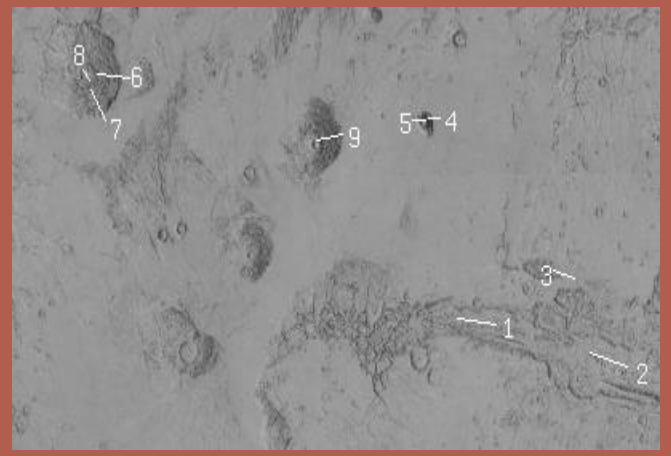


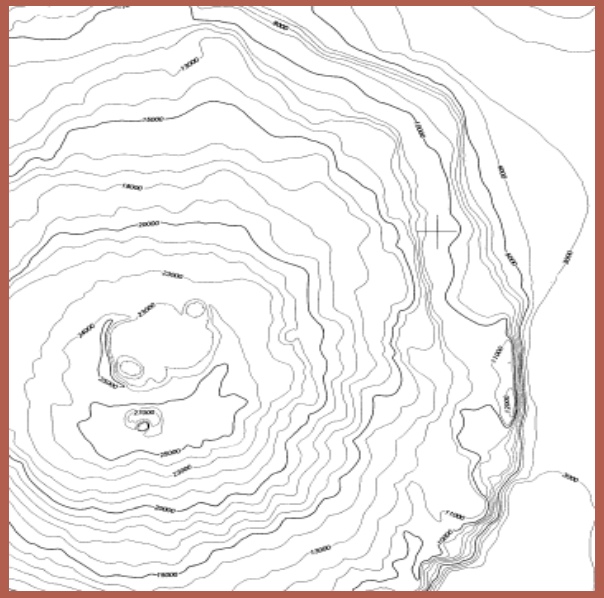
Predicted label

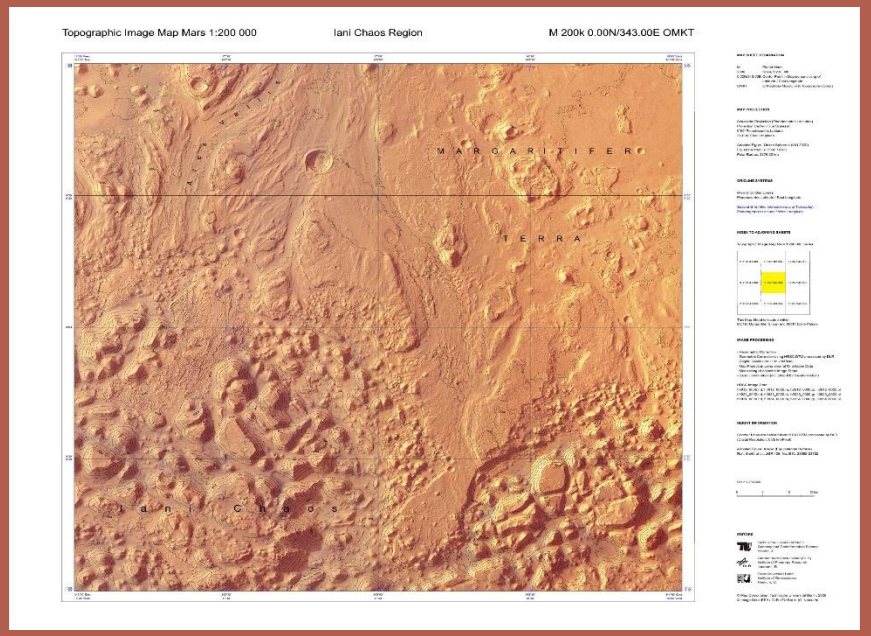
Challenges

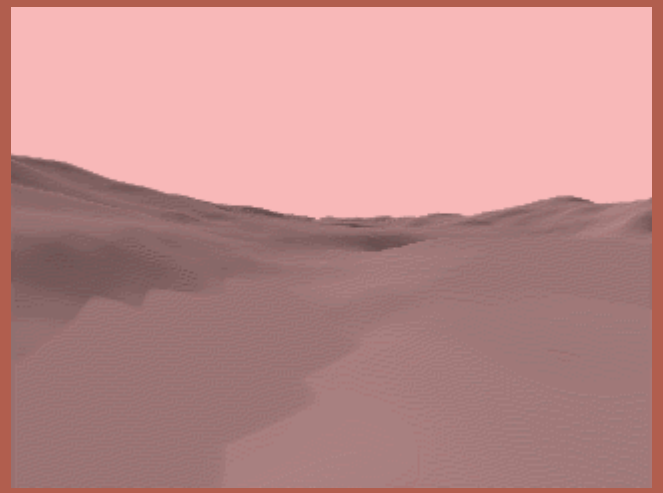
- Lack of labelled data
- ML and DL is fairly new
  - Bad judgements
- DeepLabV3+
  - Lack of documentation
    - Difficult to tailor it precisely to our needs
  - Setup was a challenge
    - Installation instructions are outdated
  - Xception65 backbone involves huge computation

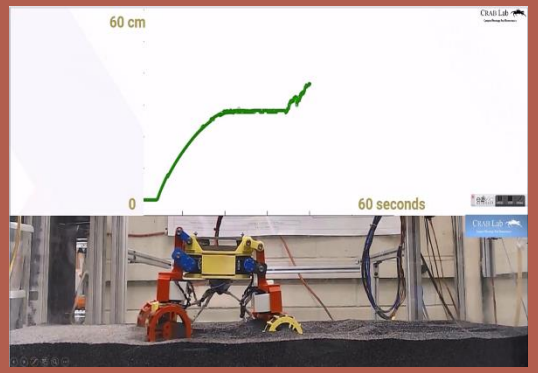
Weights Associated with Paths











References

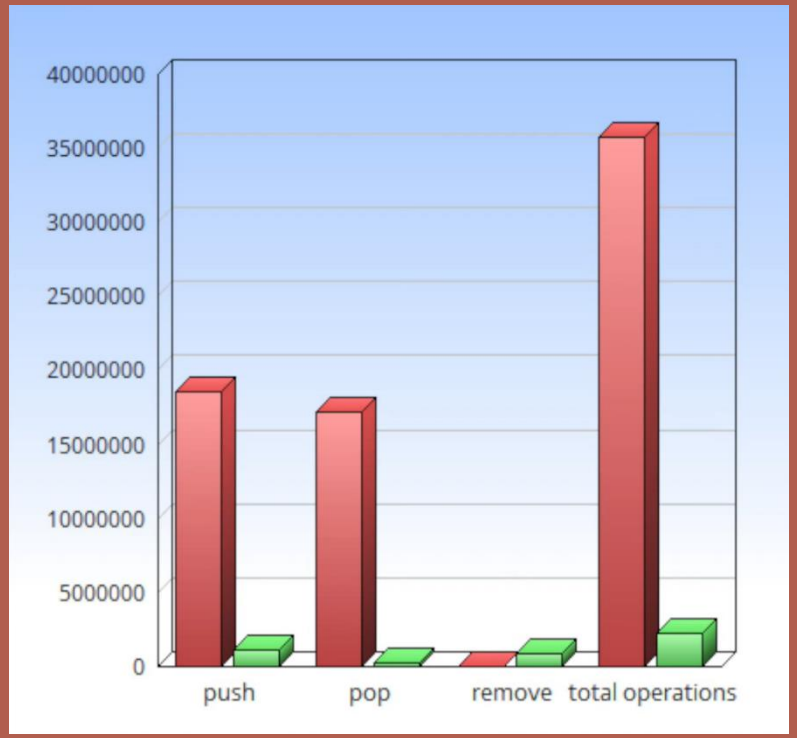
https://www.msss.com/http/ps/dtm/marsterrain.html

https://videos.space.com/m/AW2uzBeR/future-rovers-may-walk-too-see-a-prototype?list=9wzCTV4g

http://www.esa.int/Science_Exploration/Space_Science/Mars_Express/The_first_hiking_maps_of_Mars

Algorithms

- Searches
  - Informed v/s Uninformed search
  - Deliberative v/s Reactive control
- Heuristic Search: A*, LPA*, D*-lite, etc
  - f = g + h
  - h → Manhattan, Eucledian, Diagonal
- Comparison of A* vs D* lite operations
  - D* lite more efficient for more obstacles
  - Visits lesser states, optimal generalization



Reference

https://www.youtube.com/watch?v=skK-3UfcXW0

Algorithms

- Lifelong planning A*
  - Incremental version of A* with changing weights of edges
  - Static start and end nodes
- D*-lite
  - 2002 article by Koenig and Likhachev - AAAI
  - Similar to LPA* except start position = current position
  - Search direction from goal to start – thus impacting g calculation
  - Priority Queue reordering → $0 \leq h(s, s') \leq c^*(s, s')$ and $h(s, s'') \leq h(s, s') + h(s', s'')$
  - Calculated information not thrown away like A*

Reference

https://github.com/samdjstephens/pydstarlite/wiki/Understanding-the-differences-between-LPA*-and-D*-Lite

Conclusion

- Overall, we've achieved our goals
  - Terrain classification was successful from onboard cameras
  - Should add satellite image-based terrain classification
  - A* algorithm logic works successfully of with global planning
  - Working on improving D*-lite for more local planning
- Entirely different AI technologies can be combined to obtain useful results
- Just an exploration of AI techniques we think are implementable

THANK YOU

Entire package is available with instructions in
https://drive.google.com/drive/folders/183Ix3LEi-Kw7S5FyZ23agQ9IGdyegL0y?usp=sharing