DEEP RRT*

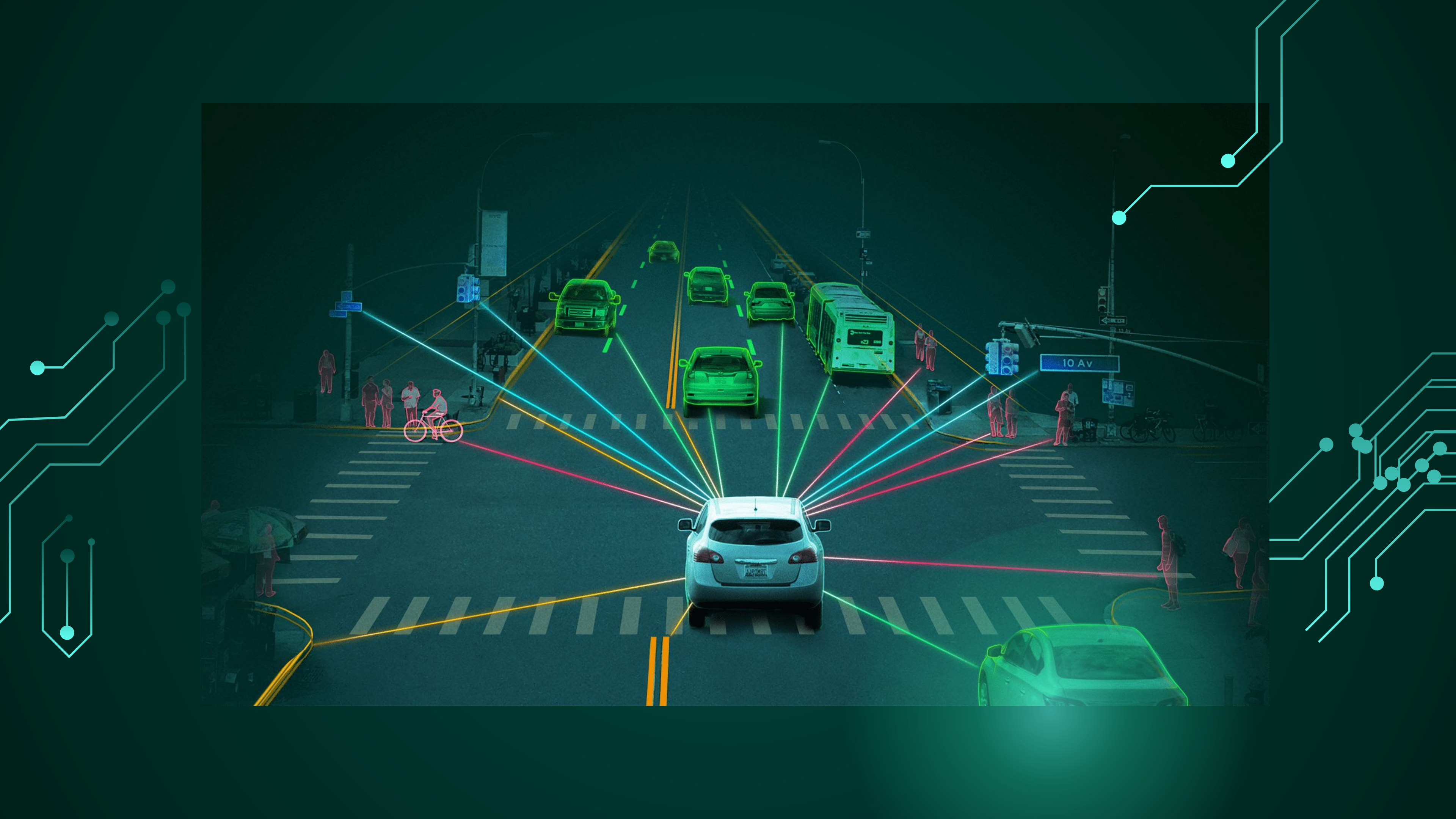
ENPM 661: Planning for autonomous robots

By: Sachin Jadhav, Navdeep Singh, Kashif Ansari

TABLE OF CONTENTS

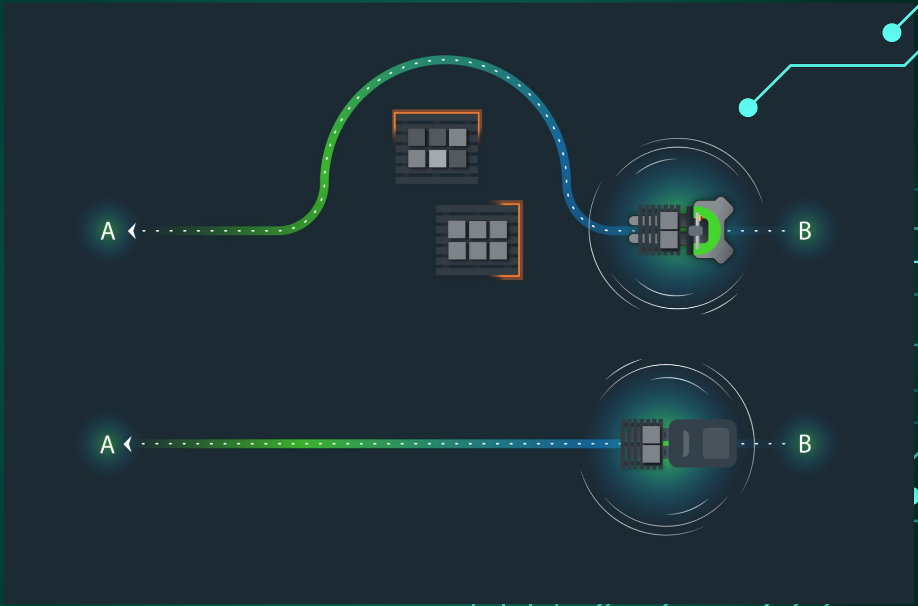
01

INTRODUCTION


10 Av

Need

- Optimal Paths: Choose the best routes considering distance, obstacles, and terrain for quicker and energy-efficient travel
- Collision-Free Navigation: Employ strategies to avoid collisions with static and moving obstacles, ensuring safe robot movement.
- Dynamic Adaptation: Adapt paths in real time to changing conditions like new obstacles or terrain variations for uninterrupted navigation.
- Scalable Planning: Design systems capable of managing complexity and scaling up for efficient navigation in diverse environments.



Path Planning Techniques in Robotics

Grid-Based Algorithms

Dijkstra's Algorithm
A* Algorithm

Sampling-Based Algorithms

Probabilistic Roadmap (PRM)
Rapidly Exploring Random Tree (RRT)

Potential Field Methods

Artificial Potential Field
Vector Field Histogram (VFH)

Optimization-Based Techniques

Linear Programming
Quadratic Programming

Machine Learning-Based Approaches

Deep Reinforcement Learning (DRL)
Neural Network-Based Models

Hybrid Methods

Potential Field + A*
Sampling-Based + Optimization

Our Attempt

- To enhance robotics motion planning by combining traditional sampling-based algorithms with deep learning techniques.

- Using PyTorch for deep learning.
- Combining of RRT* and Motion Planning Networks (MPNet).

- Implementation: PyTorch for algorithms.

- Training neural networks with refinement through RRT* Algorithm provided ground truth values, using NN architecture for improved time complexity and decision-making in path planning.

- Faster collision-free path finding, generalization to new environments, suitability for real-time robotics applications, overcoming computational challenges.

02

Formulation

Workflow

- Split map data into nodes: Start & Goal.
- Known Maps for training, Unknown Maps for testing.
- Feed Maps into RRT algorithm for path generation.
- Encode Known Maps for training.
- Train neural network model to learn map-path mapping.
- During testing, use trained model to predict path steps iteratively.
- Simulate the outcomes into a virtual environment.
- Implement into real world scenario (Static / Dynamic) using appropriate hardware.

03

Methods

```
τa ← {xstart}
τb ← {xgoal}
τ ← Ø
Reached ← False
for i ← 0 to N do
    xnew ← Pnet(Z, τa(end), τb(end))
    τa ← τa ∪ {xnew}
    Connect ← steerTo(τa(end), τb(end))
    if Connect then
        τ ← concatenate(τa, τb)
        return τ
return Ø
```

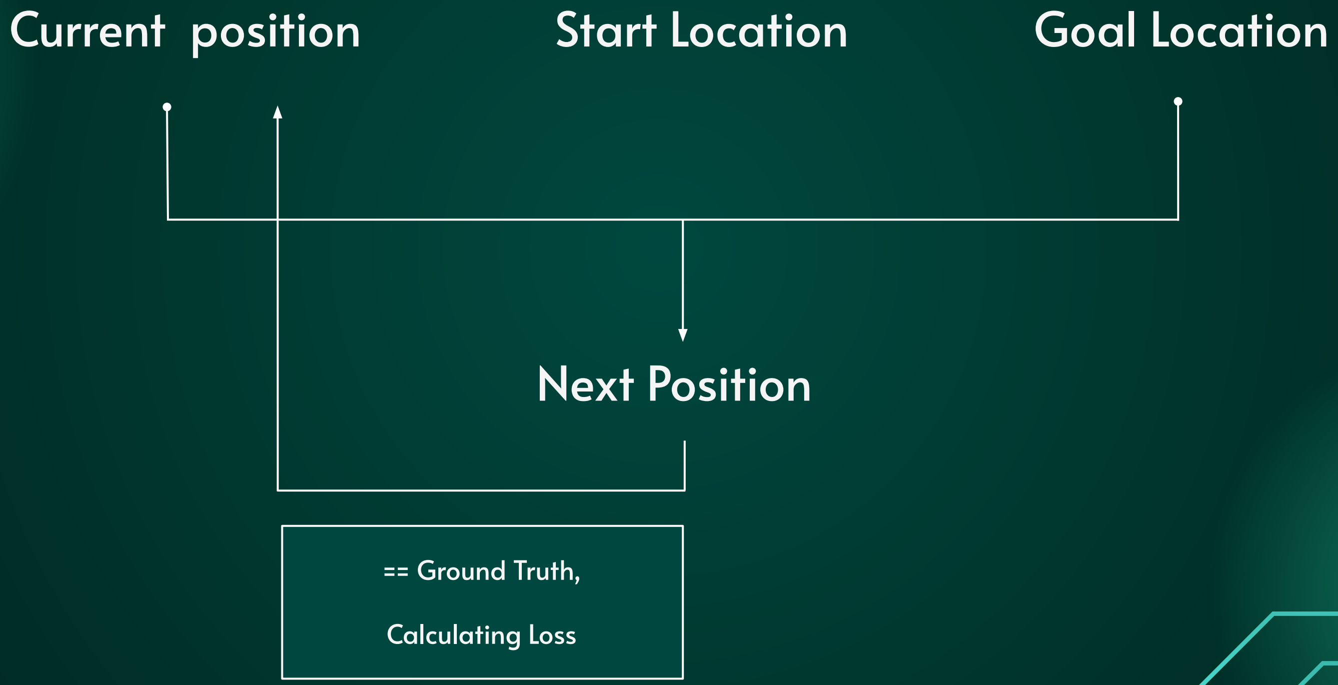
Planner Pseudocode

Contractive Auto-Encoder (C A E)

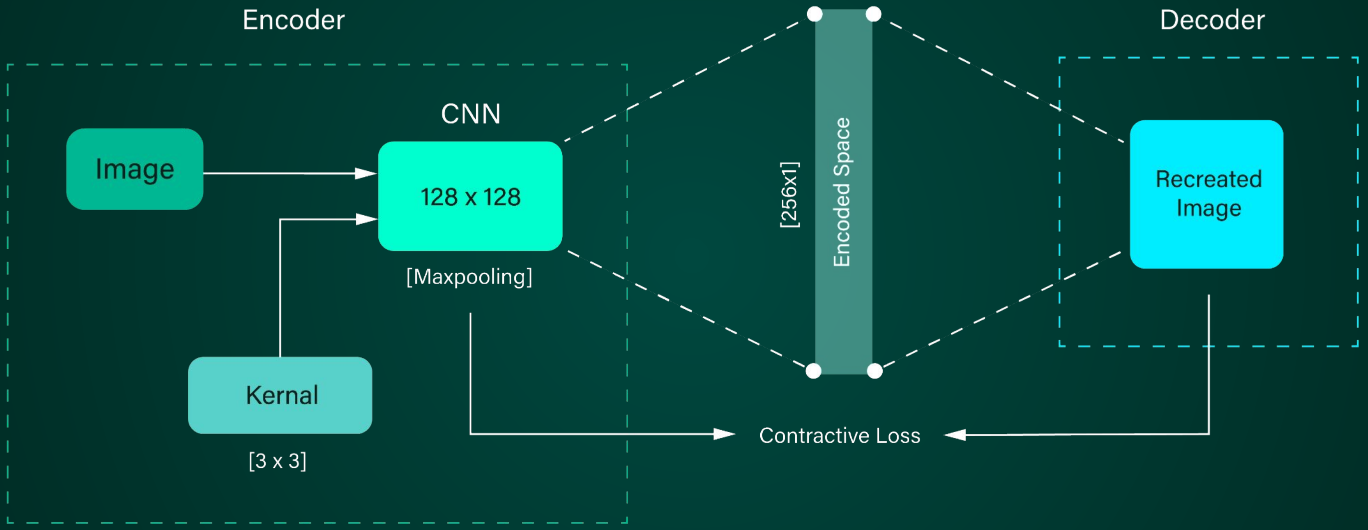
Encoder **Decoder**

MSE + Contractive Loss

Neural Planner



CAE


Encoder
Decoder
CNN
Image
128 x 128
[Maxpooling]
Kernal
[3 x 3]
[256x1]
Encoded Space
Recreated Image
Contractive Loss

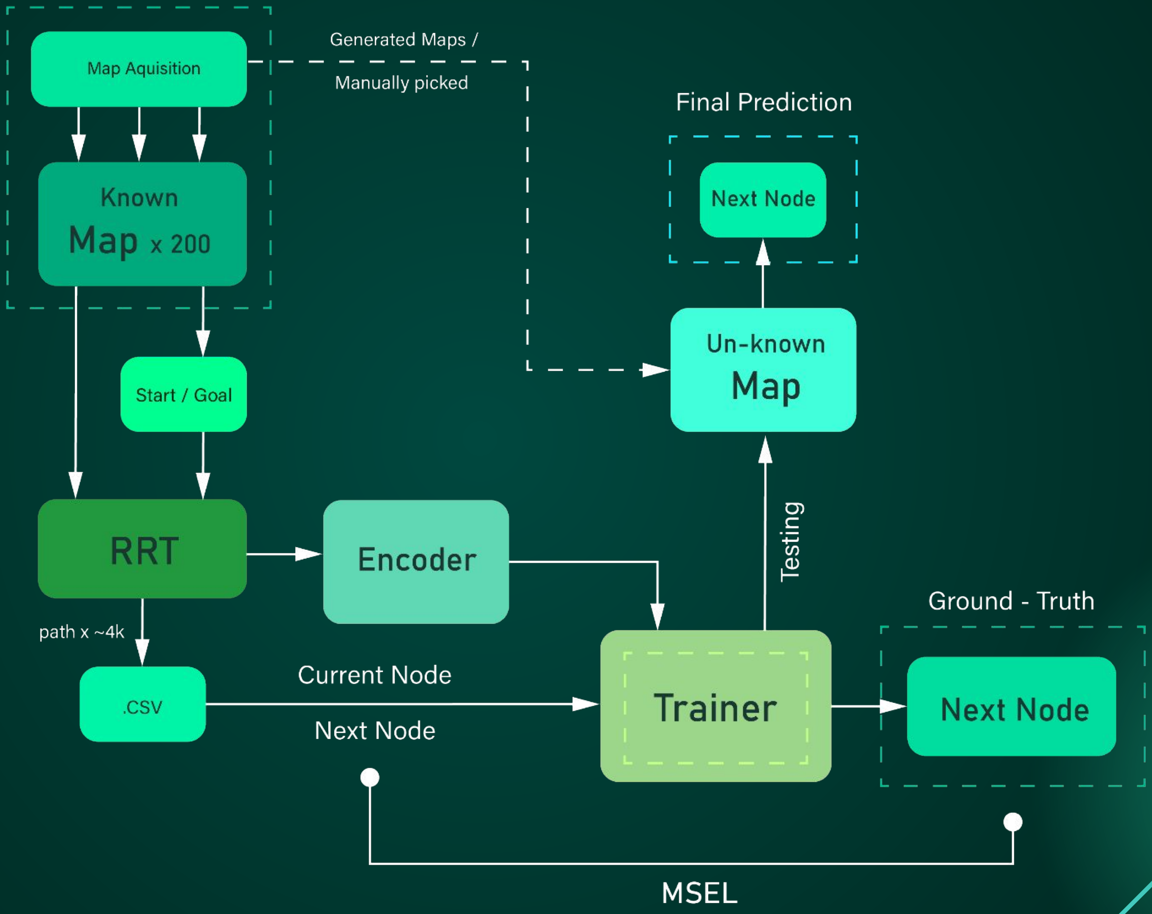
Generating Datasets

```
8,27
5,103,133940,247324,99
6.760649125548583,93.78668133543
9.325648428521857,89.49473788569227
9.922695775612356,84.53051242089596
10.33290345073481,79.54736786516906
14.932405211368259,77.58660690219327
19.89464279600004,78.19995684608156
24.201950782035123,75.66084323825729
28.125599070256943,72.56165235831233
31.837996827652834,69.21230870755298
36.62655005158725,70.6509740687636
38.38246550844562,65.96944003509837
41.4977366119822,62.05854679295151
36.731221913498395,60.548468109856536
33.66383057615008,56.599910096662285
```

200 Randomized maps

40000 Paths

Overview

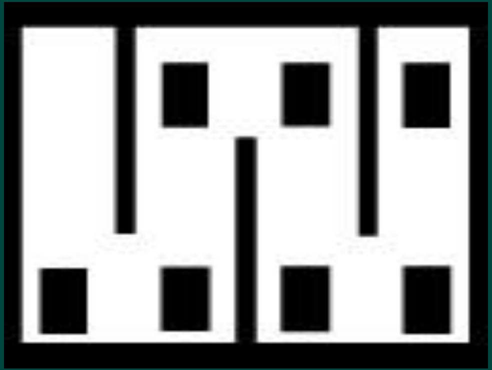

Map Aquisition
Generated Maps /
Manually picked
Final Prediction
Next Node
Known
Map x 200
Un-known
Map
Start / Goal
RRT
Encoder
Testing
Ground - Truth
path x ~4k
Current Node
.CSV
Trainer
Next Node
Next Node
MSEL

04

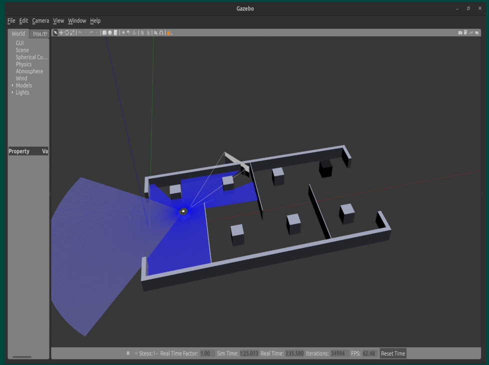
Simulation

ROS

```python
# Define the action set
action_set = [(0, RPM1), (RPM1, 0), (RPM1, RPM1), (0, RPM2), (RPM2, 0), (RPM2, RPM2), (RPM1, RPM2), (RPM2, RPM1)]
```



128*128 Map



Gazebo Classic

Turtlebot3 @RAL

05

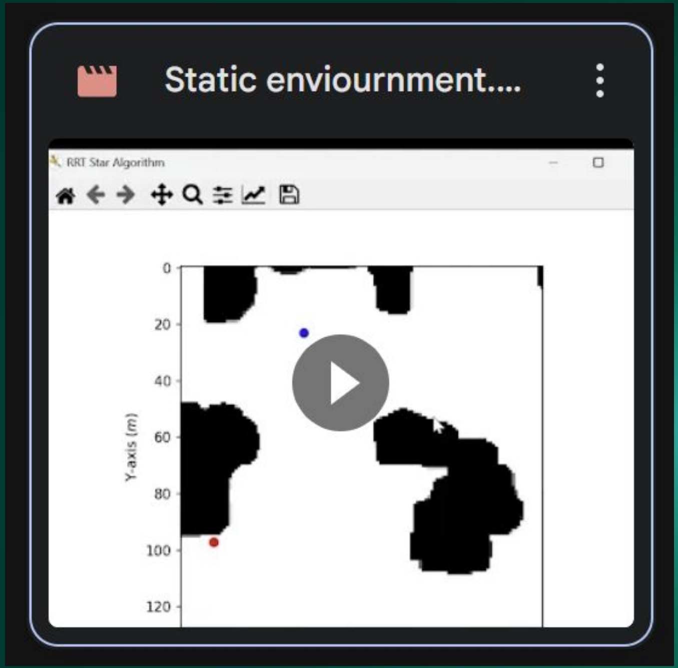
Implementation & Challenges

Challenges

- RRT data generator being stuck at different intervals.
- Varying dimensions of encoder output disrupting matrix multiplication.
- Oversized dimensions (128*128) of dataset.
- Data collected not meeting the required format for training the network.

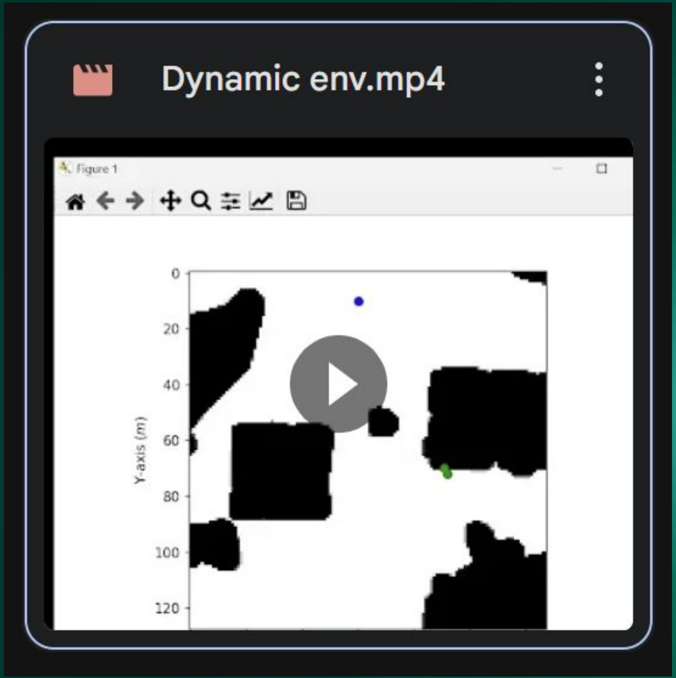
06

Outcomes



Immobilized obstacles in the same environment

Click on text



Mobilized obstacles in varying environments

Click on text

RESOURCES

- **Research papers referred:**
- 1) *Motion Planning Networks:* / https://arxiv.org/abs/1806.05767
- 2) *Deep RRT*:* / https://ojs.aaai.org/index.php/SOCS/article/view/21803

Group 37

DOES ANYONE HAVE ANY QUESTIONS?