

Experiments on Vision-Based Simulation and Real World Lane Following and Obstacle Avoidance with Proximal Policy Optimization (PPO)

Thesis Final Defense Presentation

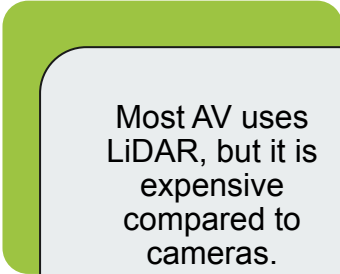
Name: Min Khant Soe

ID: st122277

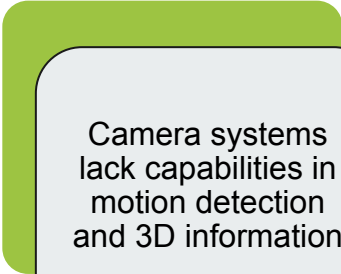
Date: 21/11/2023

Introduction

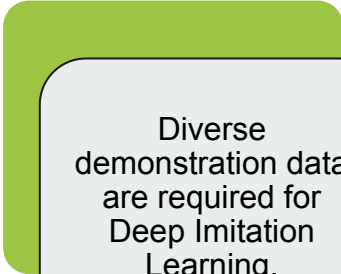
Statements of Purpose



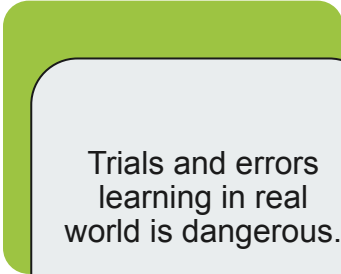
Most AV uses LiDAR, but it is expensive compared to cameras.



Camera systems lack capabilities in motion detection and 3D information




Diverse demonstration data are required for Deep Imitation Learning.



Trials and errors learning in real world is dangerous.

Objective

To conduct experiments to develop a vision-based lane following and obstacle avoidance system with Proximal Policy Optimization (PPO)



To apply PPO agent in the real world

Main Tasks Taken to Achieve the Objectives

- Trained UNet (ResNet-101 Backbone) semantic segmentation
- Established a simulation environment with Asian Institute of Technology (AIT) Map and a 3D golf cart model in the Carla simulator
- Developed and fine-tuned the reward function for PPO
- Trained and tested the PPO agent in Carla
- Evaluated the PPO agent's performance and effectiveness in real-world road conditions at AIT.

Scopes of Thesis

- Only monocular RGB camera is used for the system.
- Semantic Segmentation model is trained for better road scene understanding.
- Training an agent in a simulated environment using the Carla framework to enhance learning in a controlled setting.
- Evaluating the trained model's performance in both the Carla simulation environment and real world only on a straight road, focusing on safe and effective navigation.
- The PPO agent should demonstrate safe and accurate lane-keeping abilities in scenarios without obstacles.
- The PPO agent should exhibit proficiency in not only detecting and avoiding obstacles but also in re-aligning with the original driving lane post-avoidance

Limitation of Thesis

- Performance evaluation of the PPO agent is limited to **clear weather conditions**.
- The focus is solely on **straight roads**, not including the complexities of curved or varied road geometries.
- Study involves only **static obstacles**, not addressing the agent's response to dynamic obstacles.
- Excludes **scenarios** with **obstacles** in lanes **other than the ego lane**, limiting multi-lane applicability.
- Presence of a **driver** during evaluations to intervene in emergencies or unexpected situations.

Hardware Used - Electric Golf Cart



- Property of AICenter, AIT
- Controllable: Throttle, Steering Wheel, Brake
- Controller with 12 V battery
- Emergency Brake System

Hardware Used - USB Camera



- Borrow from Ms. Aphinya Chairat, a Phd Student at AIT
- Web camera with usb2.0
- 2 MP
- 1920x1080 Resolutions
- Manual Focus
- Adjustable Brightness, Contrast

Hardware Used - Computing Devices

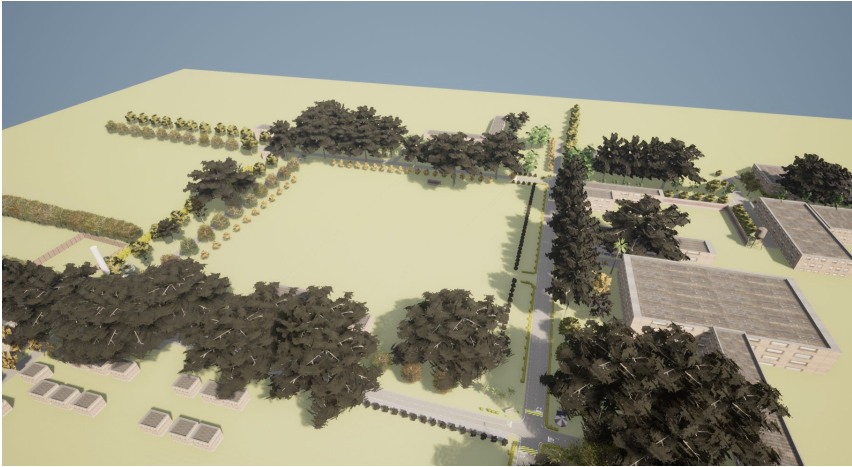
Device	Specification	Purpose of Use	Property of
GPU cloud server	64 GB RAM 4 NVIDIA GeForce RTX 2080 Ti - 11 GB	Semantic Segmentation Model Training	ICT Department, AIT
PC	32 GB RAM NVIDIA GeForce RTX 3060 Ti - 8 GB	PPO Training	AI Center, AIT
Laptop	16 GB RAM NVIDIA GeForce RTX 3060 - 6 GB	Real World Testing	Min Khant Soe

Software Used - CARLA



- An open-source simulator for Autonomous Driving system.
- Core Features
 - Easy to start and use
 - Actors: such as vehicles, pedestrian and traffic
 - Sensors: cameras, LIDAR and RADAR
 - Available ROS Bridge Version

Software Used - AIT Map



- Created by Mr. Siraphop Prasertprasasna, a Phd student at AIT

Software Used - 3D Golf Cart Model



- Provided by Mr. Witoon Wiphusitphunpol, a software engineer at AICenter, AIT

Golf Cart Dimensions

Golf Cart	Height (cm)	Width (cm)	Length (cm)
Carla	179.00	155.00	357.00
Real World	185.00	104.50	302.00

Methodology

Semantic Segmentation - Dataset

- Number of classes: **5**
- Classes Names: **"Road", "Lane Marking", "Vehicle", "People", and "Others"**

Datasets	Training Images	Validation Images	Number of Classes
Mapillary Vistas Dataset	18,000	2000	124
AIT Dataset from AICenter	1760	440	5
AIT-Carla	5440	2410	23
Complete Dataset	25200	4850	5

Semantic Segmentation - Trained Models

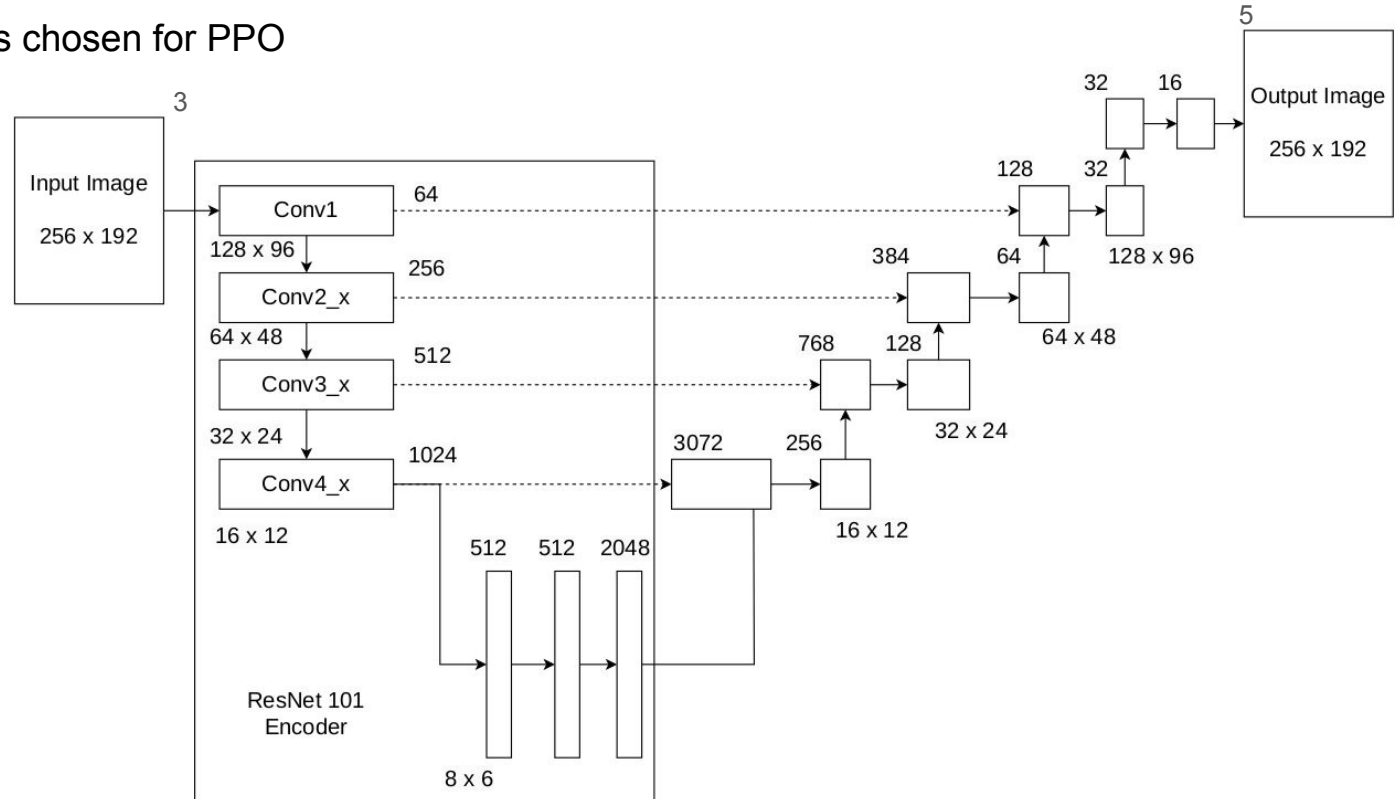
- DeepLab V3+ (Resnet-101 Backbone)
- UNet (ResNet-101 Backbone)
- UNet (EfficientNet-B0 Backbone)
- PIDNet-S

Semantic Segmentation - Training Configuration

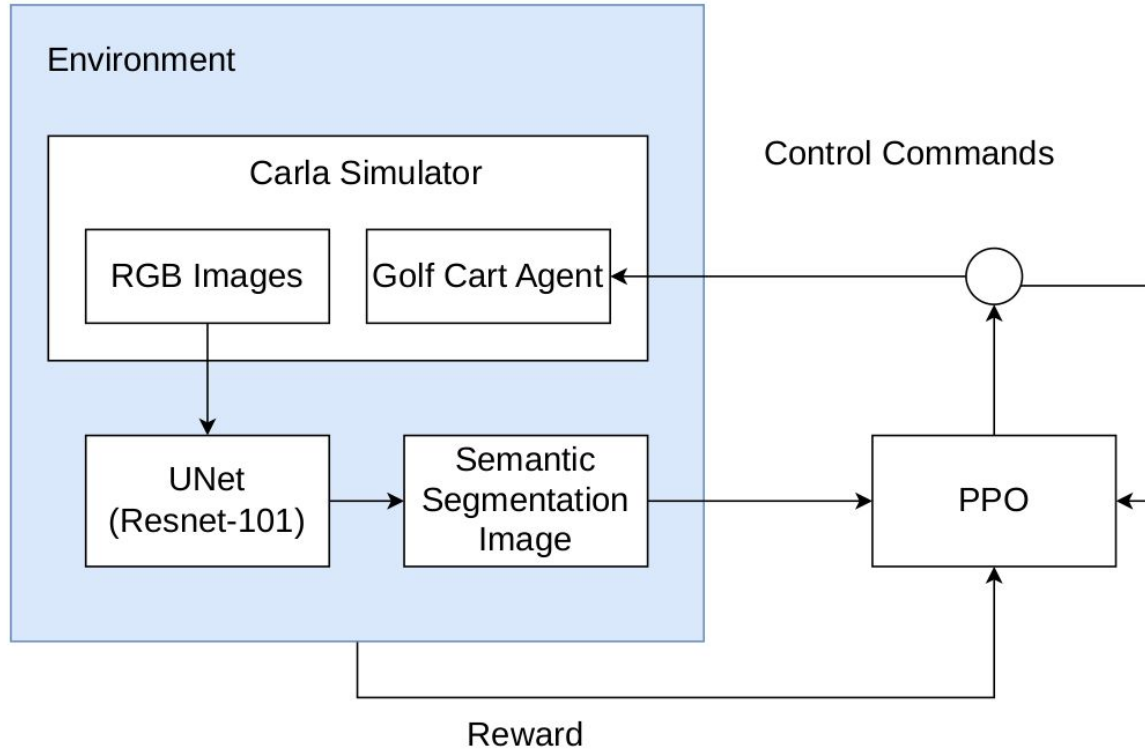
- Image Resolution: 256x192
- Batch size = 32
- Loss: Cross Entropy
- Optimizer: Adam
- Weight Decay: 0.0001
- Maximum Learning rate: 0.01
- Learning Rate Scheduler: One-Cycle Scheduler
- Image Augmentation: Random Cropping, Random Brightness and Contrast, Flipping, Gaussian Noise

Semantic Segmentation - UNet (ResNet-101)

UNet (ResNet-101) is chosen for PPO



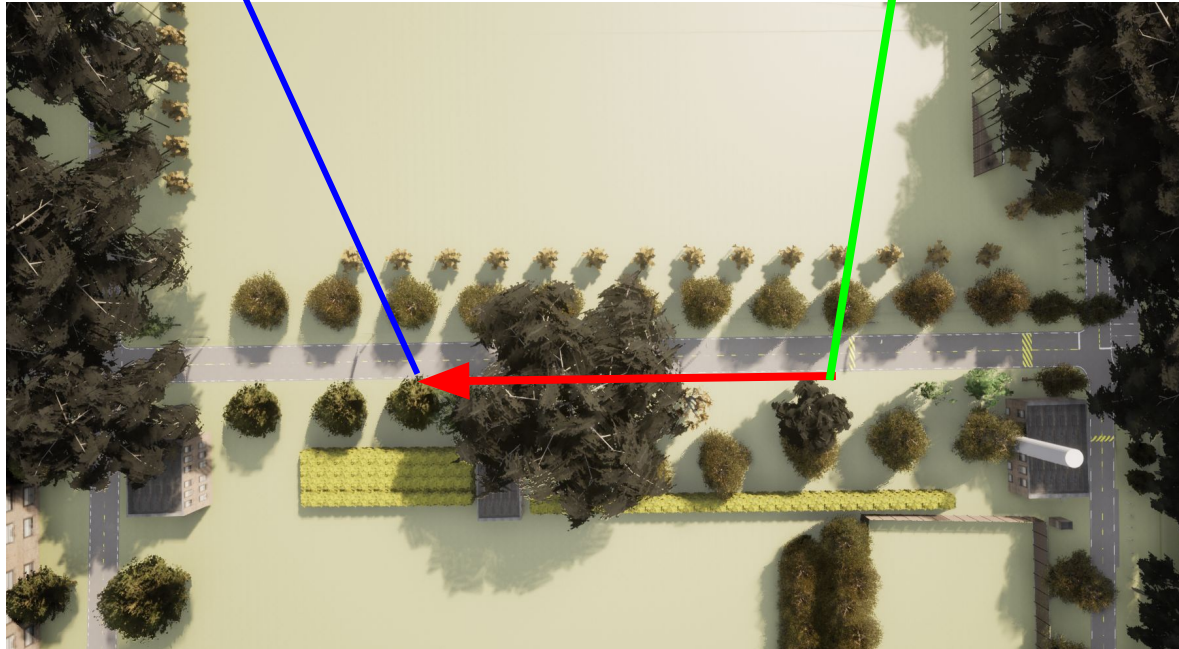
PPO - System Overview for Training



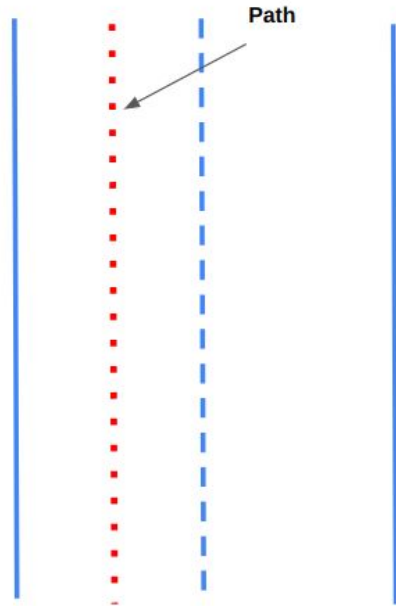
PPO - Training Environment - Location

End Point: 90 meters

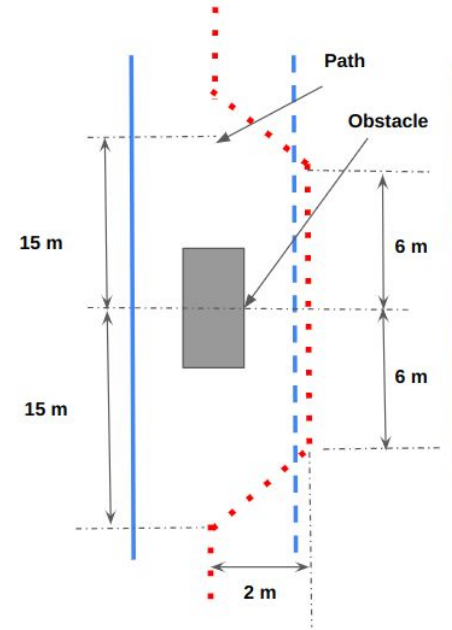
Start Point: 0 meter



PPO - Training Environment - Driving Path



Path without obstacle



Path with obstacle

PPO - Action Space - Continuous Action Type

- Steering Command (a^{st}): $[-1, 1]$
 - Negative value represents to turn left
 - Positive value represents to turn right
 - Max steer value (st_{max}) is manually set.
 - If $a^{st} \geq st_{max}$,
 - $a^{st} = st_{max}$
 - If $a^{st} \leq -st_{max}$,
 - $a^{st} = -st_{max}$
 - Otherwise,
 - $a^{st} = a^{st}$

PPO - Action Space - Continuous Action Type

- Steering Command (a^{st}):
 - **Action Smoothing** technique is applied

$$a_t^{st} = (1 - \alpha) \cdot a_{t-1}^{st} + \alpha \cdot a_t^{st,current}$$

where α is smoothing factor, and set to 0.1.

PPO - Action Space - Continuous Action Type

- Throttle + Brake Command ($a^{\text{th_brake}}$): $[-1, 1]$
 - Throttle Command (a^{th})
 - if $a^{\text{th_brake}}$ is zero or positive value,
 - $a^{\text{th}} = a^{\text{th_brake}}$
 - otherwise: $a^{\text{th}} = 0$
 - Throttle Command (a^{br})
 - if $a^{\text{th_brake}}$ is negative value,
 - $a^{\text{br}} = 1$ (apply brake)
 - otherwise: $a^{\text{br}} = 0$

PPO - Action Space - Continuous Action Type

- Throttle Command (a^{th}):
 - Max throttle value (th_{max}) is manually set.
 - If $a^{th} \geq th_{max}$,
 - $a^{th} = th_{max}$
 - otherwise,
 - $a^{th} = a^{th}$
- Brake Command (a^{br}):
 - Max brake value (br_{max}) is manually set.
 - If $a^{br} = 0$,
 - $a^{br} = 0$
 - otherwise,
 - $a^{br} = br_{max}$

Action Value Difference Between Simulation and Reality

- The control values are difference between simulation and real world.
- Throttle:
 - Simulation
 - 38 % of max throttle value -> around 5 km/h
 - 60 % of max throttle value -> around 15 km/h
 - Percent values are manually recorded in simulation.
 - Real
 - 13 % of max throttle value -> around 5 km/h
 - 30 % of max throttle value -> around 15 km/h
 - Percent values are manually recorded by observing the speedometer app on phone.
 - Therefore, **throttle max value for PPO training** was set to 38 % for 5 km/h and 60 % of for 15 km/h.
- Brake:
 - Brake response of real electric golf cart is very slow.
 - Therefore, **brake max value for PPO training** was set to 50 %.

Action Value Difference Between Simulation and Reality

- The control values are difference between simulation and real world.
- Steer:
 - Simulation
 - 70 % is max steer value -> $[-0.7, 0.7]$
 - if steer value is 0, steer angle is zero.
 - Manually observed in simulation
 - Real
 - 100 % is max steer value -> $[-1, 1]$
 - When steer value is 0, steer angle is not zero, due to friction of the wheel and analog steering value.
 - Manually observed by giving some percent value to the electric golf cart.
 - Therefore, **Steer max value for PPO training** was set to 70 %.

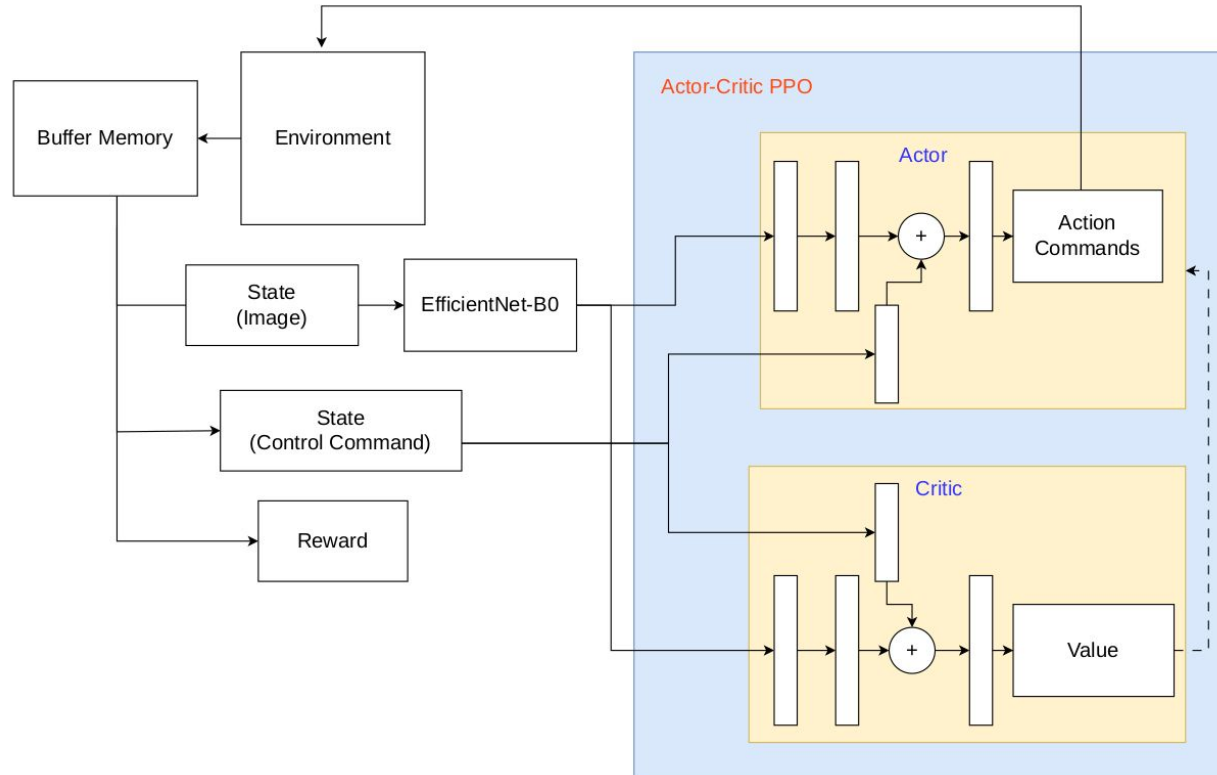
PPO - State Space

- Image State Input:
 - Semantic Segmentation Image: 80x60 Resolution, 3 Channels
 - 3 stacked consecutive frames
 - Shape of Image State: [batch, 9, 80, 60]
- Control State Input:
 - $[a_{t-1}^{st}, a_{t-1}^{th}, a_{t-1}^{br}]$
 - Shape of Control State: [batch, 3]

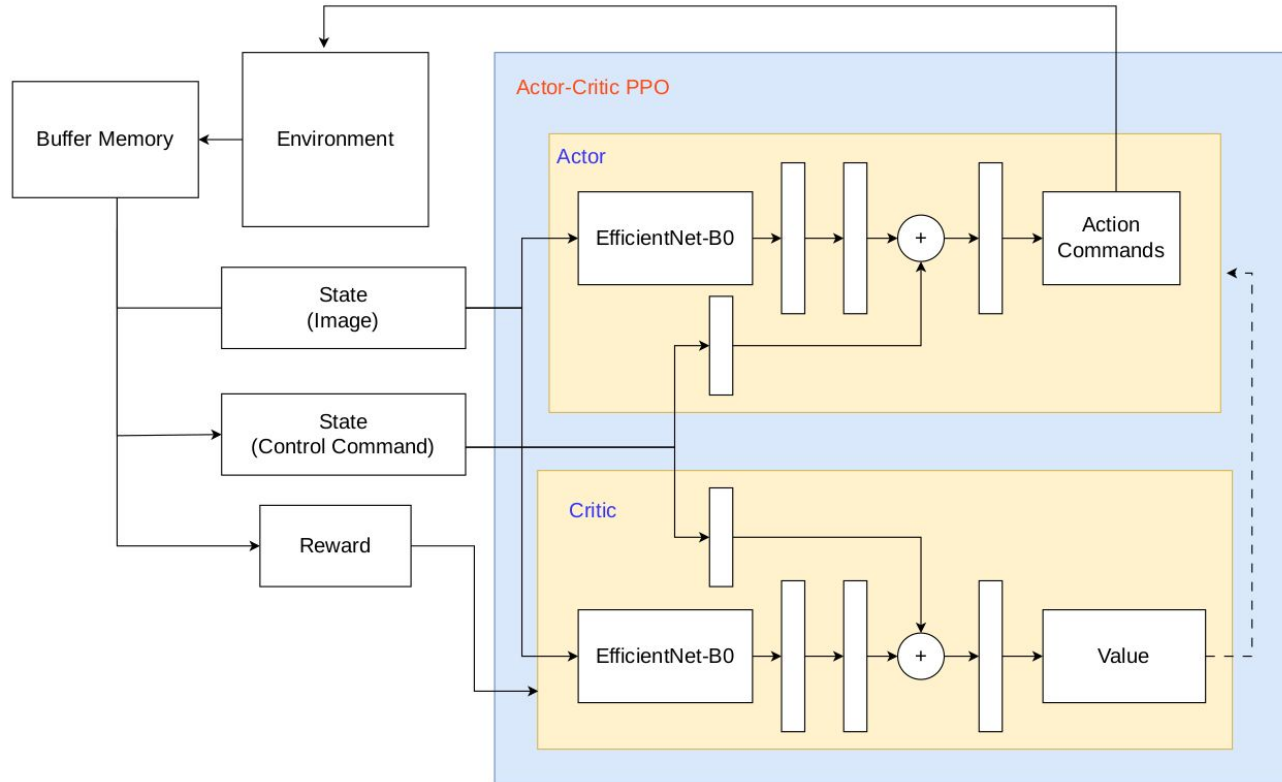
PPO - Policy Network - State Encoder: EfficientNet-B0

Stage i	Operator \hat{F}_i	#Channels \hat{C}_i	#Layers \hat{L}_i
1	Conv3x3	32	1
2	MBConv1, k3x3	16	1
3	MBConv6, k3x3	24	2
4	MBConv6, k5x5	40	2
5	MBConv6, k3x3	80	3
6	MBConv6, k5x5	112	3
7	MBConv6, k5x5	192	4
8	MBConv6, k3x3	320	1
9	Conv1x1 & Pooling & FC	1280	1

PPO - Policy Network - Shared-State Encoder

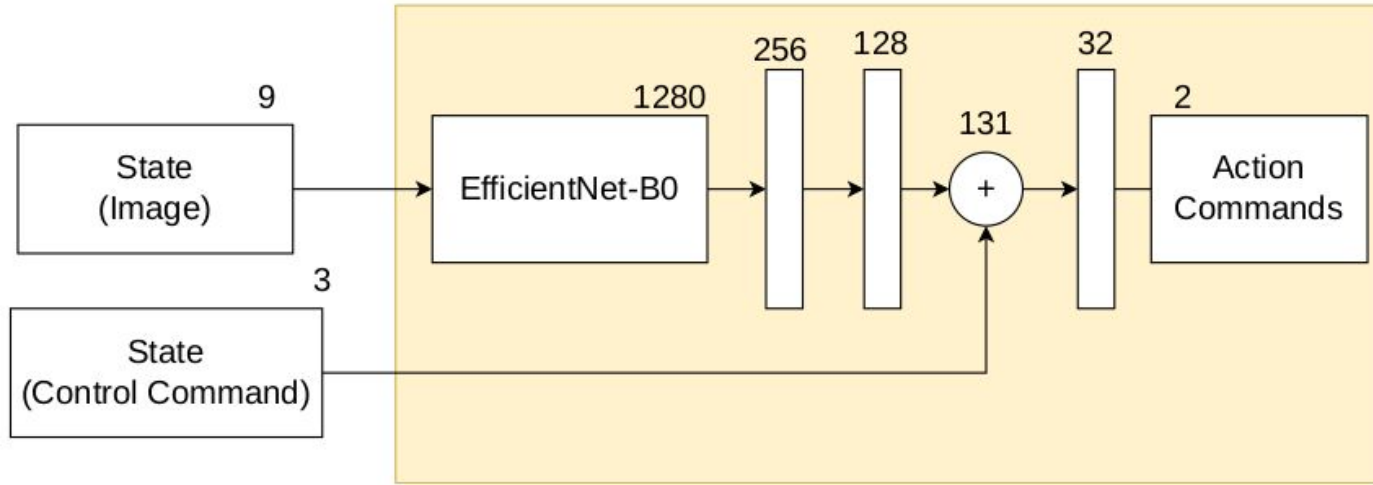


PPO - Policy Network - Separated-State Encoder



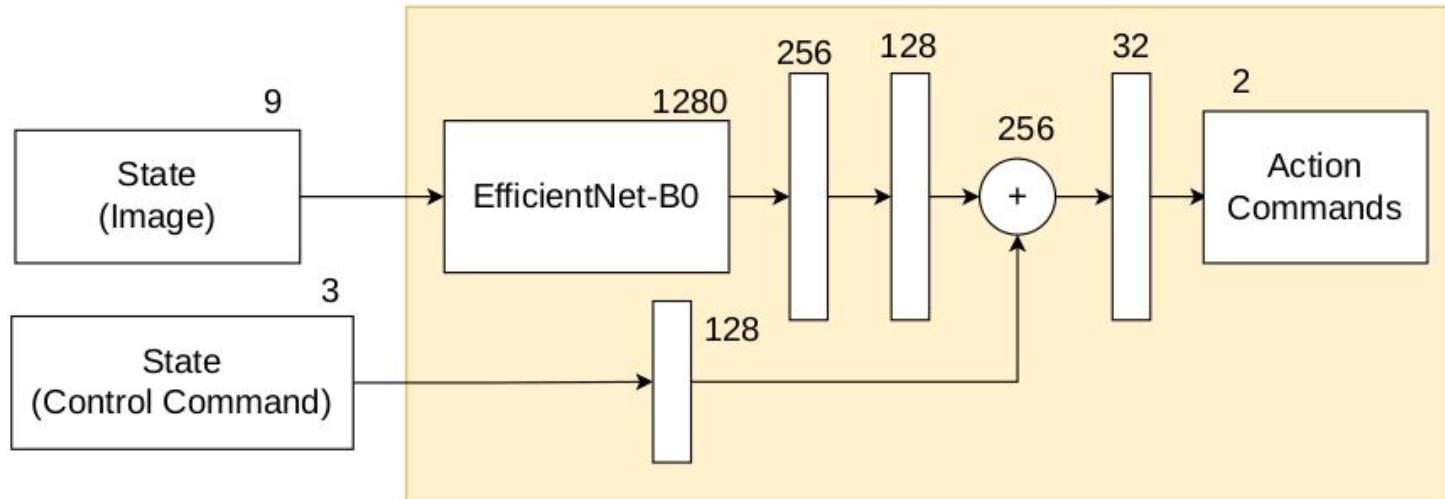
PPO - Policy Network - Direct-concatenation Method

Directly concatenate to the output of 2nd linear after state encoder

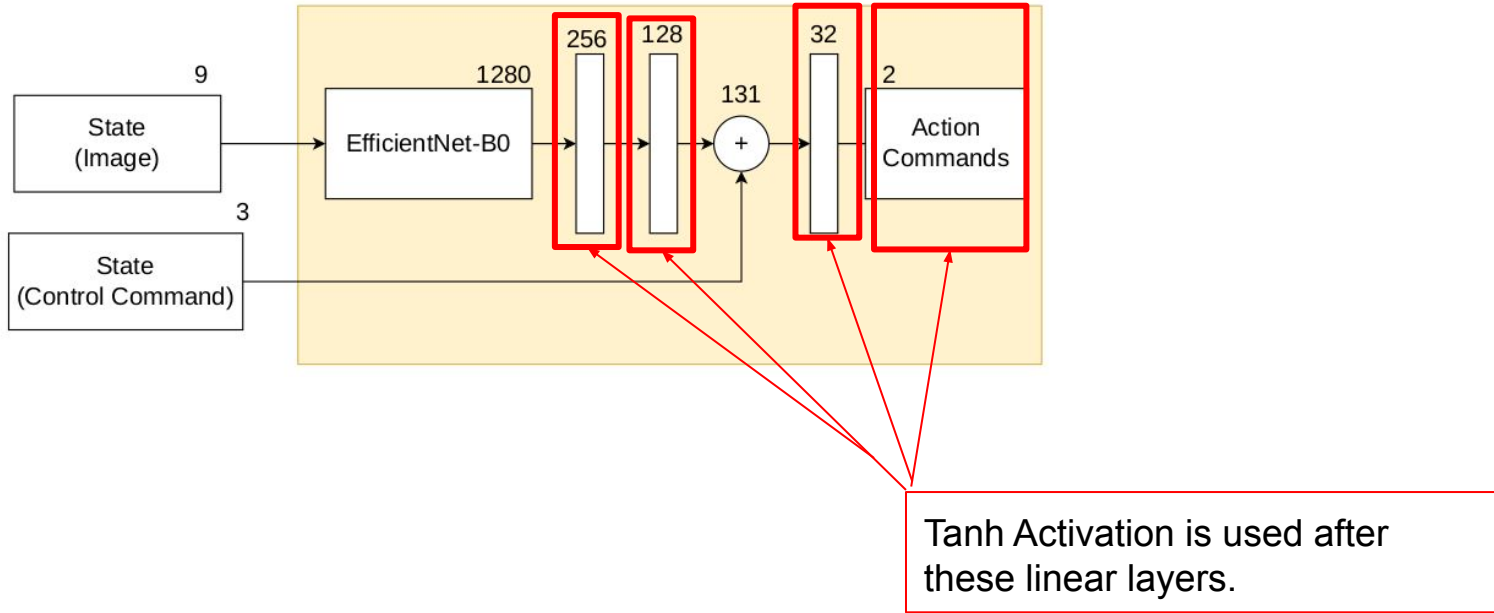


PPO - Policy Network - Expand-concatenation Method

Expand control state channel with a linear layer, then concatenate to the output of 2nd linear after state encoder



PPO - Policy Network - Activation Function



PPO - Curriculum Training Approach

- Lane Following
 - a. Spawn the agent at the middle point of ego lane
 - b. Initial goal distance is set as 10 meters
 - c. Everytime agent achieve to the goal distance, it will be spawn at random location (fixed starting distance) of the ego lane.
 - d. Every 5 achievement, goal distance is increased by 10 until 90 meters.

PPO - Curriculum Training Approach

- Obstacle avoidance
 - a. Trained model from Lane following is used.
 - b. Initial obstacle is spawned at near the left edge of ego lane (left lane) and the forward distance of 20 meters from the agent.
 - c. Initial goal distance = 1 meter + the distance of the obstacle
 - d. Everytime agent achieve to the goal distance, it will be spawn at random location (fixed starting distance) of the ego lane.
 - e. After 5 achievement, goal distance is increased by 10 until 50 meters.
 - f. When agent complete 5 times at 50 meters, obstacle location is moved by 0.5 meters to the right side until it reach to the near of right edge of ego lane.

PPO - Training Configuration

- Policy Clipping Parameter: 0.22
- Batch Size:
 - Initial 300
 - Increment by 50 up to max 800 after every 5 successful completion
- Epochs:
 - Max: 100
 - Used Early Stopping to prevent overfitting
- Learning Rate (Actor): 0.0005
- Learning Rate (Critic): 0.0006
- Discounted Factor: 0.985
- Entropy bonus coefficient:
 - 0.02 for lane following task
 - 0.03 for obstacle avoidance
- Value Loss coefficient: 0.5

PPO - Reward Function

Reward function for each time step taken by PPO agent

$$R = r_{\text{follow_path}} + r_{\text{complete}} + r_{\text{bonus}} - (p_{\text{speed}} + p_{\text{crash}} + p_{\text{time_step}})$$

PPO - Reward Function

Crash Penalty:

$$p_{\text{crash}} = \begin{cases} 1700, & \text{if stopped too long} \\ 400, & \text{if out of road or obstacle collision} \\ 700, & \text{if time out} \\ 0, & \text{Otherwise} \end{cases}$$

PPO - Reward Function

Time Step Penalty:

$$p_{\text{time_step}} = 1$$

Speed Penalty:

$$\text{speed_diff} = |\text{player_speed} - \text{target_speed}|$$

$$p_{\text{speed}} = c_s \cdot (\text{speed_diff})^2$$

where c_s is the speed penalty factor and set to 2

PPO - Reward Function

Distance to the path from the front axle (d_{front}) and distance to the path from back axle (d_{back}) are calculated as follow.

$$d_{\text{front}} = \sqrt{(x_{\text{front}} - x_{\text{path_front}})^2 + (y_{\text{front}} - y_{\text{path_front}})^2}$$

$$d_{\text{back}} = \sqrt{(x_{\text{back}} - x_{\text{path_back}})^2 + (y_{\text{back}} - y_{\text{path_back}})^2}$$

- The coordinates (x front , y front) represent the midpoint of the front axle rod of the vehicle.
- The coordinates (x path front , y path front) correspond to the closest waypoint on the path ahead of the vehicle.

PPO - Reward Function

Forward Movement Reward:

$$r_{\text{move_forward}} = \begin{cases} c_f, & \text{if moving forward} \\ 0, & \text{otherwise} \end{cases}$$

where c_f is the moving forward factor and used a constant value of 6.

Following the Path Reward:

$$r_{\text{follow_path}} = \left(\frac{r_{\text{move_forward}}}{d_{\text{front}} + 1} \right)^2 + \left(\frac{r_{\text{move_forward}}}{d_{\text{back}} + 1} \right)^2 - (d_{\text{front}})^2 - (d_{\text{back}})^2$$

PPO - Reward Function

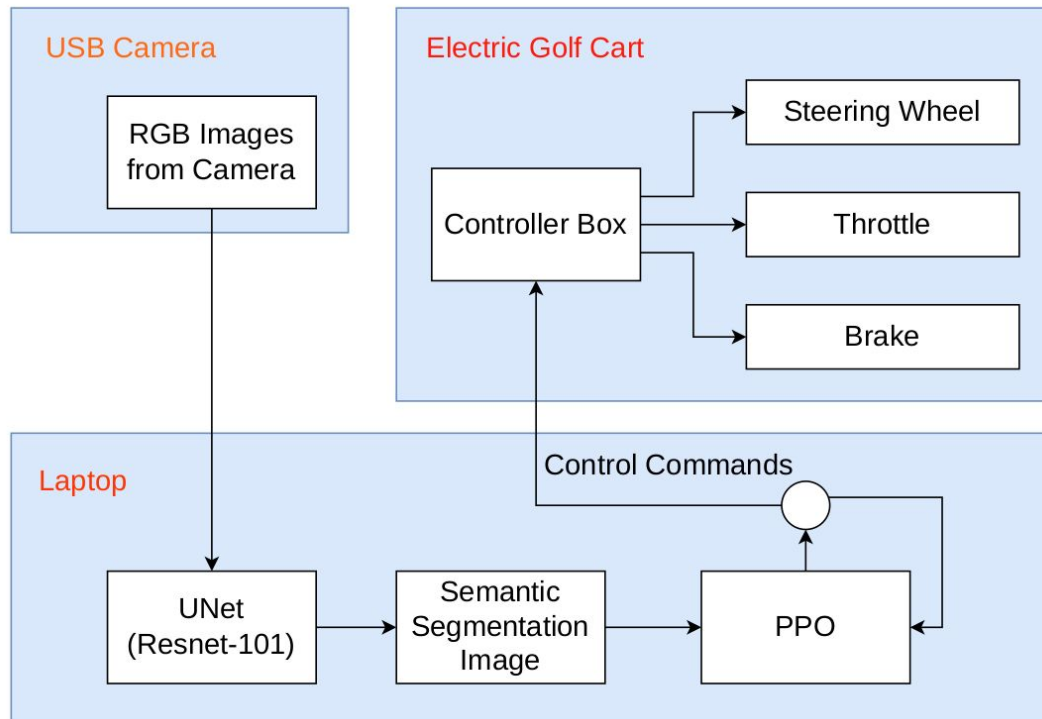
Bonus Reward:

$$r_{\text{bonus}} = 5 \cdot \left\lfloor \frac{\text{distance_traveled}}{10} \right\rfloor$$

Completion Reward:

$$r_{\text{complete}} = \begin{cases} r_{\text{bonus}} - d_{\text{front}}, & \text{if PPO agent complete the task} \\ 0, & \text{Otherwise} \end{cases}$$

System Overview for Real World Testing



Experiments

Semantic Segmentation

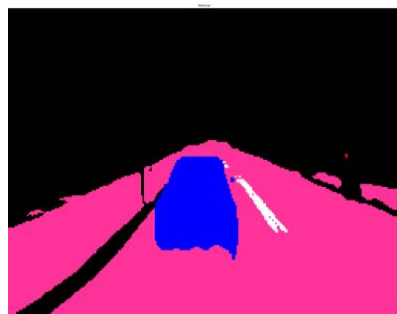
Model	Mean IoU	FPS
DeepLab V3+ (ResNet-101)	0.739	16
UNet (ResNet-101)	0.776	14
UNet (EfficientNet-B0)	0.742	18
PIDNet-S	0.760	11

FPS values are tested on the laptop.

Semantic Segmentation



Input Image



UNet (ResNet-101)
Output Image



UNet (EfficientNet-B0)
Output Image



PIDNet-S
Output Image

Semantic Segmentation



Input Image



UNet (ResNet-101)
Output Image

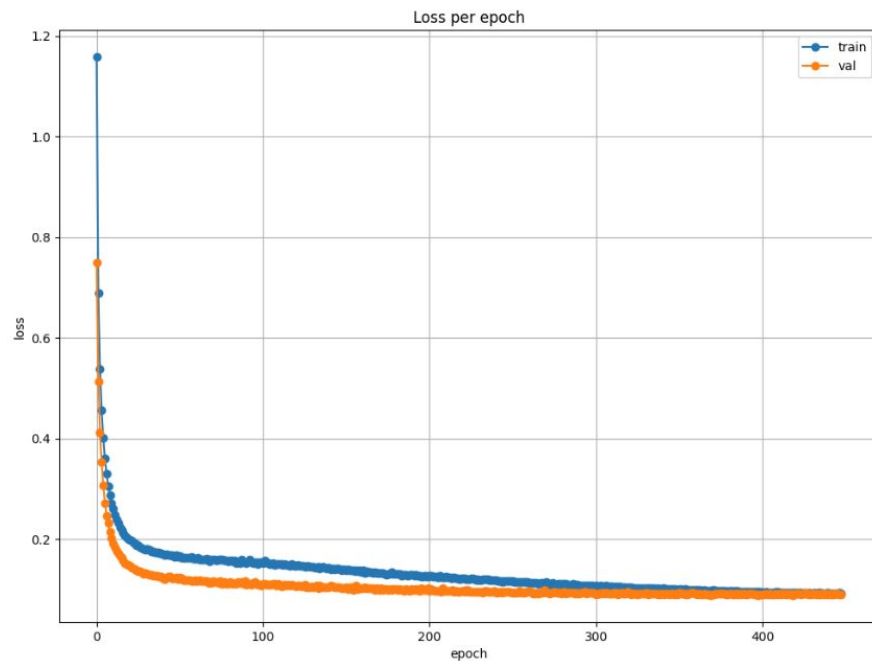


UNet (EfficientNet-B0)
Output Image

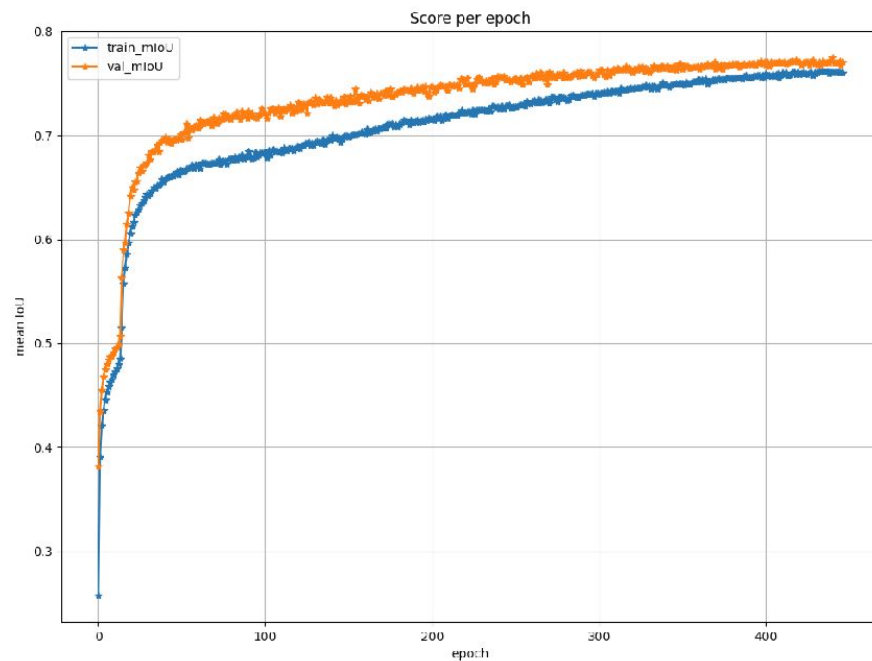


PIDNet-S
Output Image

UNet (ResNet-101) Plots



Loss Plot



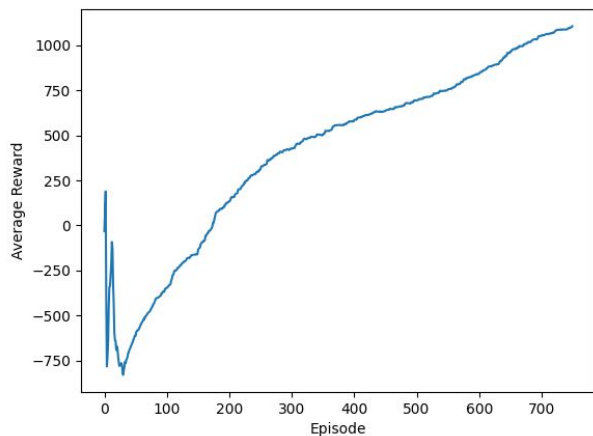
Mean IoU Plot

Semantic Segmentation Testing at AIT

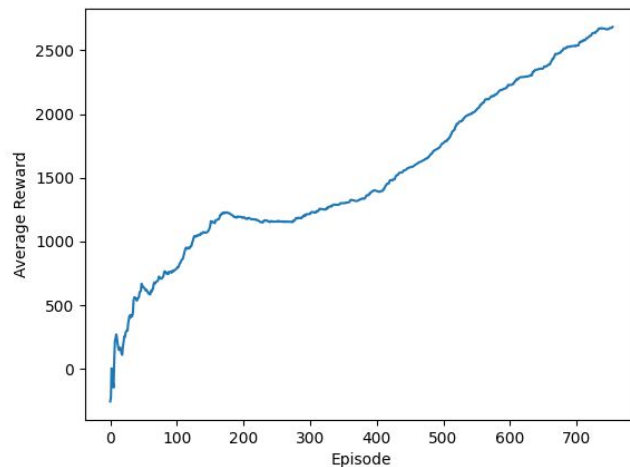


PPO - Shared-state encoder vs Separated-state encoder

- Used **Expand-Concatenation** Method for Control State Integration.
- PPO couldn't learn and perform well with shared-state encoder.
- With separated-state encoder, it could complete the final goal distance of the lane following task at both 5 km/h and 15 km/h.



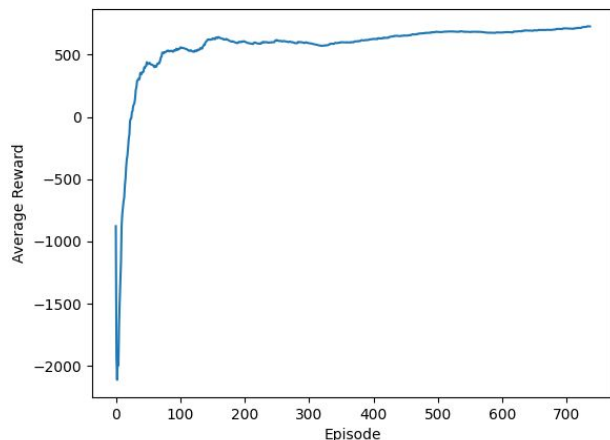
Average Reward Plot for
Shared-state encoder method



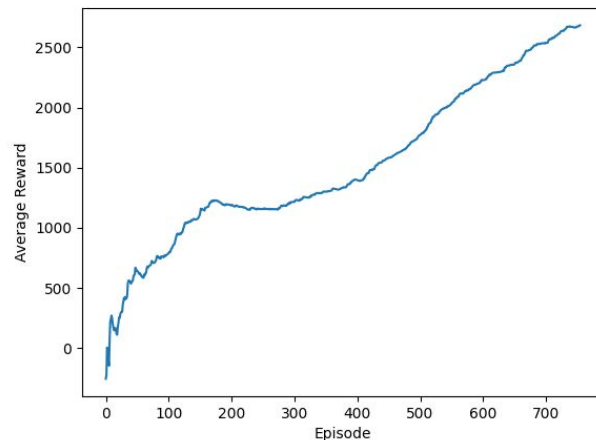
Average Reward Plot for
Separated-state encoder method

PPO - Direct-Concatenation vs Expand-Concatenation

- Trained using **separated-state encoder** architecture.



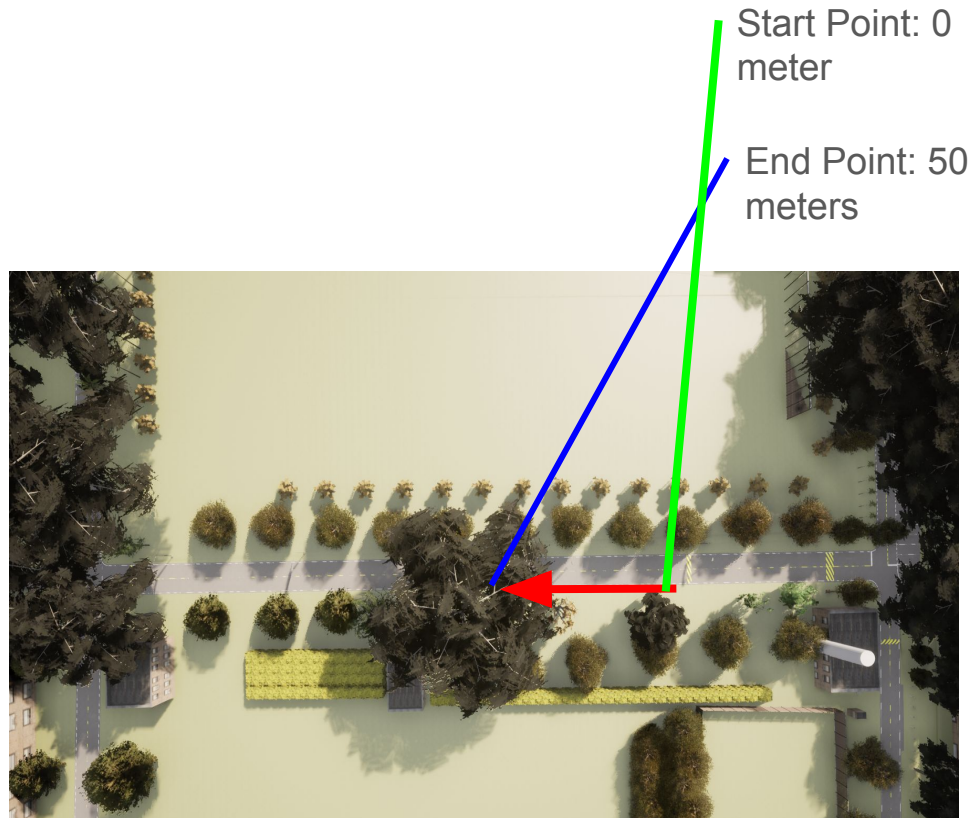
Average Reward Plot for
Direct-Concatenation method



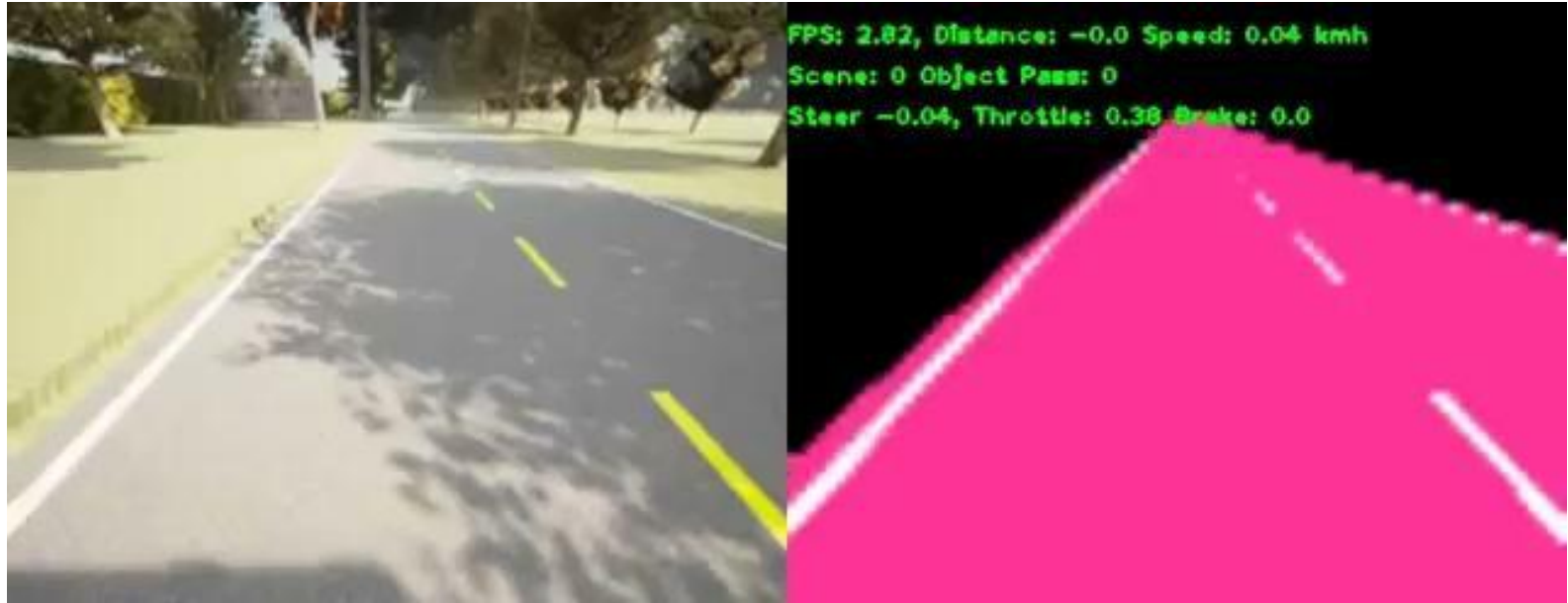
Average Reward Plot for
Expand-Concatenation method

Evaluation of PPO agent in simulation - Lane Following

- **Location:** The road between football field and 711
- **Direction:** same as in training
- **Distance:** Around 50 meters
- **Speed:** 5 km/h
- **Success Rate:**
 - 75 % for 5 km/h



Evaluation of PPO agent in simulation - Lane Following



Sample record for 50 meters at Speed 5 km/h

Evaluation of PPO agent in simulation - Lane Following

- **Location:** The road between football field and 711
- **Direction:** same as in training
- **Distance:** Around 90 meters
- **Speed:** 5 km/h and 15 km/h
- **Success Rate:**
 - 15 % for 5 km/h
 - 20 % for 15 km/h



Evaluation of PPO agent in simulation - Lane Following



Sample record for 90 meters at Speed 5 km/h

Evaluation of PPO agent in simulation - Lane Following



Sample record for 90 meters at Speed 15 km/h

Evaluation of PPO agent in simulation - Obstacle Avoidance

- **Location:** The road between football field and 711
- **Direction:** same as in training
- **Distance:** Around 90 meters
- **Speed:** 5 km/h
- **Success Rate:**
 - 99 % success rate of avoidance
 - 0 % success rate of turning back to ego lane



Evaluation of PPO agent in real world - Lane Following

- During the agent driving electric golf cart,
 - Steering Wheel does not follow the command value sometime.
 - It lags behind the command value.
- To minimize this effect,
 - Action smoothing factor is reduced to half of the value used in the training.

Evaluation of PPO agent in real world - Lane Following

- **Location:** The road between football field and 711
- **Direction:** same as in training
- **Distance:** Around 85 meters
- **Speed:** 5 km/h and 15 km/h
- **Success Rate:**
 - 10% for 5 km/h
 - 15 % for 15 km/h



Evaluation of PPO agent in real world - Lane Following



Sample record at Speed 5 km/h

Evaluation of PPO agent in real world - Lane Following



Sample record at Speed 15 km/h

Evaluation of PPO agent in real world - Lane Following

- **Location:** The road between football field and 711
- **Direction:** opposite direction as in training
- **Distance:** Around 65 meters
- **Speed:** 5 km/h
- **Success Rate:**
 - 10% for 5 km/h



Evaluation of PPO agent in real world - Lane Following



Sample record at Speed 5 km/h

Recommendation and Conclusion

Recommendation

- Consider **worst-case** scenarios for **semantic segmentation training** and include that in training data to minimize noise.
- Understand how the human driver will react if similar scenarios are given, and **formulate** the **reward** carefully.
- After reward function is built, **test it manually first** with diverse scenarios to see if it works.
- Ensure that the **final reward minimum** for each step is **0 or a positive value** to encourage the agent to move forward.
- Apply **action smoothing** to reduce the shaking behavior.
- Before starting any DRL training experiments, **inspect the hardware limitations** of the actual car.

Recommendation

- Start **training** the model in the **simplest possible environment** and gradually adjust parameters and the reward function for more complex environments. This approach helps the agent adapt more quickly.
- **Expose** the agent to as many **different scenarios** as possible to ensure comprehensive learning and adaptability.
- **Experiment** with both types of state encoder architectures, and also both command state concentration methods.
- **Visual monitoring** during training is crucial to quickly identify and rectify any issues, saving time and improving the training process.

Recommendation

- Use a camera with fixed settings for visual input on the car. In this research, an adjustable camera was used, but it required frequent adjustments for optimal results in semantic segmentation.
- Utilize a gps sensor module to set the path for DRL agent; this could help agent aware of the positions.
- Utilize a speed sensor module to feed the driving speed of the agent, so that agent can understand more. Motion captures from stacking frames works but not seems to be a effective methods.
- One more area to explore is that experimenting with 4 cameras, locating at top front, at two side mirrors, and at the back; this will enhance the understanding of the road scene for DRL agent. This could cost more computation power but could improve the performance of the DRL agent.

Conclusion

- Showcased the feasibility of employing **Proximal Policy Optimization (PPO)** with just a single **monocular camera** in autonomous vehicle systems.
- Established that images from **semantic segmentation** are sufficient for obstacle avoidance, offering an alternative to conventional lidar and depth estimation methods.
- Employed a **curriculum learning** approach for efficient and effective training of the PPO algorithm.
- Conducted a comparative analysis between direct-concatenation and expand-concatenation methods for control state integration into the PPO policy network, finding that the **expand-concatenation** method yielded **superior** results.
- Executed experiments comparing shared and separated-state encoder architectures within the PPO policy network, finding that **separated-state encoders** lead to **superior** learning outcomes.
- **Tested** the PPO agent's performance in both simulated environments and real-world settings.

Thank You