Self-Driving by Deep Reinforcement Learning on AirSim
深層強化学習によるAirSim上での自動運転について

s1250074 SUGA Yuta, Supervisor: Prof. LI Xiang

1. Summary of The Research

In recent years, autonomous driving by AI has attracted remarkable attention. In this thesis, the goal is to train a self-driving car by deep reinforcement learning using camera images as input. There exist several environments where we can train our cars on a computer. Among them, I chose AirSim on Unity, which was developed in 2018. Because it can be developed together with Unity ML-Agents, an environment for deep reinforcement learning.

Specifically, the learning algorithms Proximal Policy Optimization (PPO) [1] and Soft Actor-Critic (SAC) [2] are used to train the car. By using PPO, I examined whether the type and number of cameras were related to the increase in reward. Then I compared the best results among them with the results learned by SAC. As a result, I concluded that SAC, which accumulates experience, is inefficient in the image-based training, and PPO, which can update policies many times in a short period, is more suitable.

2. Approach/Methodology

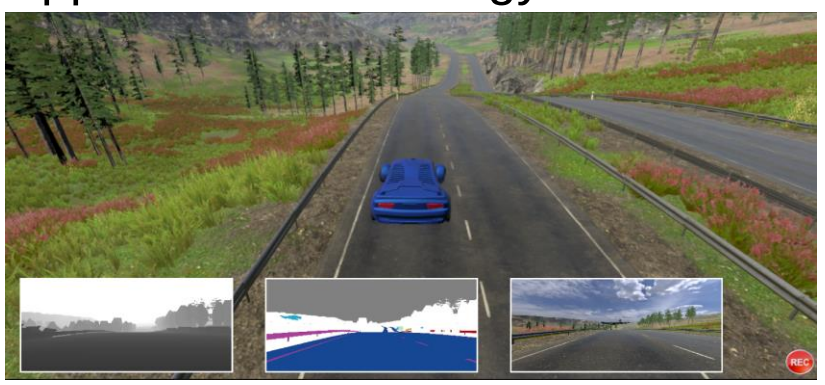

Figure 1: Screenshot of training on the highway in Windridge City.

The car is trained on the highway in the “Windridge city” with AirSim on Unity (Figure 1). This car uses three cameras (Normal, Segmentation, Depth) to train. In particular, the Segmentation Camera plays a major role in detecting the lines of the road (Figure 2). By color-coding the road, lines and other objects can be detected, and noise is eliminated to some extent.

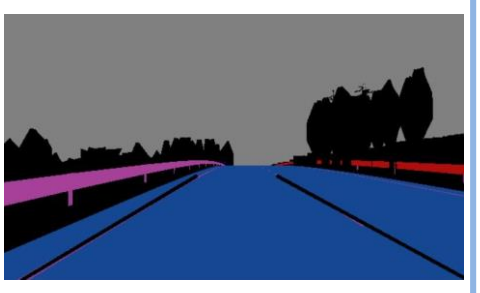

Figure 2: Screenshot of the view from the camera with segmentation

Not crashing, not straying from the lines of the road, and driving at a constant speed are all factors that determine good driving for AI.

3. Experiment Result

In my research, I compared the environment with cameras trained with PPO and with SAC. In the SAC environment, since we need to accumulate a large amount of experience, the agent used one camera and the number of pixels was set to 100*100 horizontally and vertically.

The following results were obtained after 500,000 steps of training in both environments (Figure 3). The blue graph shows the results for PPO and the orange graph shows the results for SAC. Figure 3 shows the reward for every 1000 steps. From the results, we can see that there is a big difference between PPO and SAC: PPO sometimes can exceed 1.0, while the maximum value of SAC is almost -6.0. This may because PPO updates its policies frequently while SAC updates its policies after accumulating a large amount of experience.

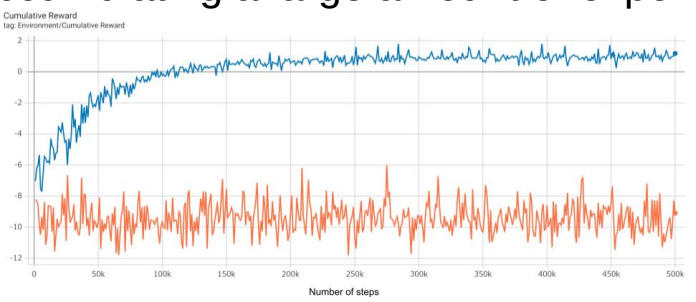

Figure 3: The reward of PPO (blue) and SAC (orange).

4. Conclusion and Future Tasks

The advantage of the SAC is that the number of training sessions can be shortened by using past experience. However, the results in the previous section show that its learning speed is inferior to PPO. This may be partly because this environment is difficult to obtain reward in an insufficiently trained state. In conclusion, PPO, which updates the policy frequently from a small number of failures, is more suitable for this environment than SAC, which finds an appropriate operation method from a large number of failures. Also to be able to drive carefully to know pedestrians, traffic lights, and various signs, sensors that can acquire more information are needed.

References

[1] Schulman, John, et al. "Proximal policy optimization algorithms." (2017).

[2] Haarnoja, Tuomas, et al. "Soft Actor-Critic: Off-Policy maximum entropy deep reinforcement learning with a stochastic actor." (2018)