Teammates:

1. Piyush Jena - pj2400
2. Kashish Goel - kg3044

Introduction:

Reinforcement learning (RL) has emerged as a powerful approach for training autonomous driving systems. Unlike traditional rule-based or model-based approaches, RL algorithms learn to make decisions based on trial and error interactions with the environment, enabling the development of more adaptive and robust systems.

In this project, we focus on the application of reinforcement learning to autonomous driving. Autonomous driving is a challenging task that requires an agent to navigate complex environments while ensuring the safety of passengers and other road users. It has the potential to revolutionize transportation by improving road safety, reducing congestion, and providing accessibility to those who are unable to drive. RL has been successfully applied to a range of autonomous driving tasks, from lane following and obstacle avoidance to complex maneuvers such as merging and intersection negotiation.

Proposed Project:

The problem we aim to tackle is to develop a RL-based algorithm for self-driving car control. The objective is to train the self-driving car to navigate through different driving scenarios in a safe and efficient manner. In this project, we use the CARLA [1] simulator as our simulation environment. Starting with a simple DQN implementation that fetches state using a CNN, we explore various approaches to improve the performance. Furthermore we improve our model and pipeline using some of the top performers in the CARLA Leaderboard.

Methodology:

We plan to start with a simple DQN to predict actions. The available actions include accelerate, brake and steering actions like turn left and right as detailed by an implementation by pythonprogramming group [2]. It uses a pre-trained Xception [3] model with the top layer replaced by a 3 neuron output layer for the actions - accelerate, turn left and turn right. It uses Q-learning algorithm with ϵ-greedy exploration. The model didn't perform well and the model car went around in circles. Changing the reward for collision enabled the model to perform better in following the lanes but a small deviation caused the car to oscillate or crash. Moreover the network was notorious for taking a huge amount of time to train.

To address these issues, we survey the available literature and plan to incorporate changes in two key areas: vision and safe exploration.

1. **Vision**: We change the CNN from a pre-trained Xception [3] to a CNN module trained on images from CARLA for semantic segmentation task. As detailed in [4], this has shown marked improvement in the performance. Also, it is common knowledge now that self-driving cars use very advanced models for semantic segmentation for identifying pedestrians, other cars, traffic signals among others. We will be adding more layers instead of just 1 layer to increase the model capacity to accommodate the huge number of states. We will not be training the encoder (CNN) which would substantially decrease the number of trainable parameters. It has been described in [4] that the encoder performance didn't significantly improve the overall performance.
2. **Safe exploration**: As found in the base implementation, it was very easy for the car to crash. So, we will be exploring the use of a new reward scheme, SARSA and Imitation Learning.
    a. **Reward -** A new reward scheme has been explored in [5]. It penalizes the agent if it deviates from the center of the lane, or if it is not aligned along the lane. It also linearly increases the reward according to the speed. All this will significantly help in alleviating the issues found in the base implementation. We plan to incorporate this reward scheme in our implementation.

b. **SARSA** - As we found in the cliff-walking example, SARSA algorithm tended to choose a safer path by walking away from the cliff due to look-ahead. We want to explore if this is true in the case of autonomous driving.
c. **Imitation Learning** - As described in [5], we see that historically Imitation Learning algorithms have performed better than Reinforcement Learning. As humans, we also learn driving through this method. We want to explore the performance of imitation learning in this case. We also want to explore the possibility of improving the training time of RL algorithms by setting the initial policy $\boldsymbol{\pi}_0$ to the learned values from imitation learning instead of random policy to address safety issues related to exploration. The algorithm proposed in [9] achieves 78% success in the CARLA benchmark and can be used as a starting point to further train using reinforcement learning to converge to optimal Q-values.

We will be comparing the performance of the developed algorithm against top-performing models by running the CARLA benchmark. We also plan to compare the training times and loss. We will also be comparing the performance in different test environments to see how well the algorithm generalizes in different cities and weather conditions.

Conclusion:

This proposed project aims to develop a reinforcement learning-based algorithm for self-driving car control using the CARLA simulator as the simulation environment. The objective is to train the self-driving car to navigate through different driving scenarios in a safe and efficient manner. The proposed changes in vision and safe exploration are expected to enhance the overall performance of the algorithm. This project has the potential to contribute significantly to the development of more adaptive and robust autonomous driving systems, thereby improving road safety, reducing congestion, and providing accessibility to those who are unable to drive.

Citations:

1. CARLA Team, "CARLA: An Open Urban Driving Simulator," https://carla.org/. [Accessed: Mar. 21, 2023].
2. PythonProgramming.net team, "Reinforcement Learning Agent for Self-Driving Autonomous Cars in Carla with Python," PythonProgramming.net, 2021. [Online]. Available: [Link]. [Accessed: Mar. 21, 2023].
3. Francois Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1251-1258.
4. Gharaee, Z., Holmquist, K., He, L., & Felsberg, M. (2021, January). A bayesian approach to reinforcement learning of vision-based vehicular control. In 2020 25th International Conference on Pattern Recognition (ICPR) (pp. 3947-3954). IEEE.
5. Toromanoff, M., Wirbel, E., & Moutarde, F. (2020). End-to-end model-free reinforcement learning for urban driving using implicit affordances. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 7153-7162).
6. R. Zhang, P. Liang, Y. Li, C. Zhang, Y. Liu, "Learning by Cheating," in Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 1246-1254.
7. S. Kim, S. Yun, S. Yoo, "Transfuser: Imitation Learning with Transformer-based Data Augmentation," in Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 5559-5568.
8. M. Zhao, J. Wu, T. Zhou, L. Yang, J. Yuan, "Multi-Modal Fusion Transformer for End-to-End Autonomous Driving," in Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 2677-2686.
9. Y. Shen, T. Zhao, J. Li, Y. Wang, T. Li, Y. Li, X. Mao, "End-to-End Urban Driving by Imitating a Reinforcement Learning Agent," in Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 5315-5324.