AI3603 人工智能理论及应用 课程设计选题说明

通过本学期的课程学习，我们学习了人工智能不同方向的理论知识和应用场景。在课程设计中，各位同学将以小组形式，从本文档中的**自动驾驶、强化学习、棋类搜索、生成式模型和目标检测**五个课题中选择一个题目，结合课程所学知识和相关文献资料，提出解决方案并完成课题要求。当前为组队及选题阶段，请各位同学阅读下方选题说明和课题要求，结合自身兴趣完成组队和选题。

1. 组队及选题规则

- **组队要求：**3 人一组，自行组队，每组完成一个课题。
- **课题数量：**共 5 个课题，根据班级人数，每个课题最多允许 4 个小组选择，具体见后续项目中的说明。
- **选题方式：**各小组根据兴趣提交 5 个课题的选题排序，课程团队以“先到先得”，“顺序志愿”为原则，优先满足先提交选题意愿的小组，优先满足小组的第一志愿，依据小组提交顺序和课题容量确定最终选题。

2. 选题流程

- **小组组队：**10 月 15 日 18:00 前，结合兴趣方向自行完成 3 人组队，并将组员信息填写至 https://kdocs.cn/l/crD7ac00CGb2，逾期未完成组队的同学将由助教随机组队。
- **选题意向：**10 月 19 日 18:00 前，阅读本文档题目要求，确定小组选题排序。
- **选题提交：**10 月 19 日 18:00，准时开放问卷链接，统计各小组的选题意愿：https://f.kdocs.cn/g/iSNqJeIF，注意**由[组长]填写，只允许[提交一次]**，若重复提交则以晚提交时间和排序为准。
- **选题确认：**助教根据提交结果确定并公布最终选题结果。

3. 完成过程

- **助教答疑：**每个课题均配备一位责任助教，将与所有选择该课题的小组共同建立微信群，在课题完成过程中共同进行答疑和交流。
- **中期报告：**提交一份 2 页以内的中期报告，总结研究背景、主要问题、拟采用方法和未来计划。
- **成果展示：**最后一节课上各组汇报并展示课题成果，同时完成同学互评。
- **最终提交：**课程结束后，提交纸质版本的总结报告，以及代码、图片、视频等相关文件。

4. 评分标准

- **性能评价（10 分）：**由相同课题 5 个小组的最终运行效果排名决定，各课题具体评价指标见下方题目要求。

- **组间互评（10 分）：**最后一节课上成果展示时，进行组间评分并取平均分，教师和助教均视为独立的小组。
- **组内贡献（10 分）：**项目结束后会发布问卷链接，统计小组成员对任务的贡献程度，该分数由组内其他成员根据任务贡献评分决定。
- **教师评分（10 分）：**教师根据完成度、创新性以及最终提交文件综合评分。

1. Autonomous Driving on Highways (4 groups)

Background: The decision-making in autonomous driving is a critical and difficult task for intelligent vehicles in dynamic transportation environments. Autonomous driving systems constitute multiple tasks where classical supervised learning methods are no longer applicable, such as the prediction of other vehicles' movements, the optimal driving speed in a dynamic changing environment, and diverse configurations of the highways, etc.

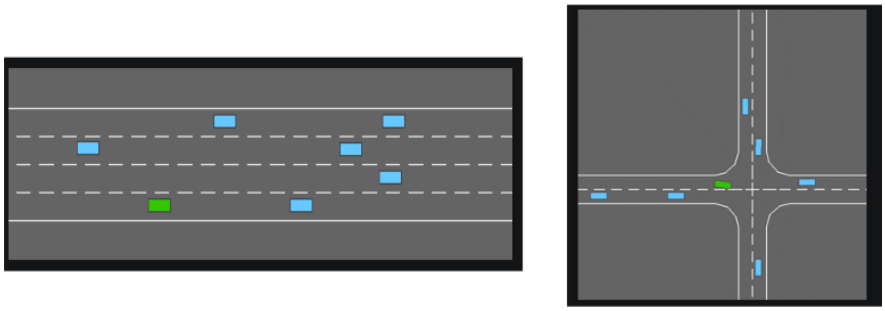

Figure 1: **Left**: Highway Scenario **Right**: Intersection Scenario

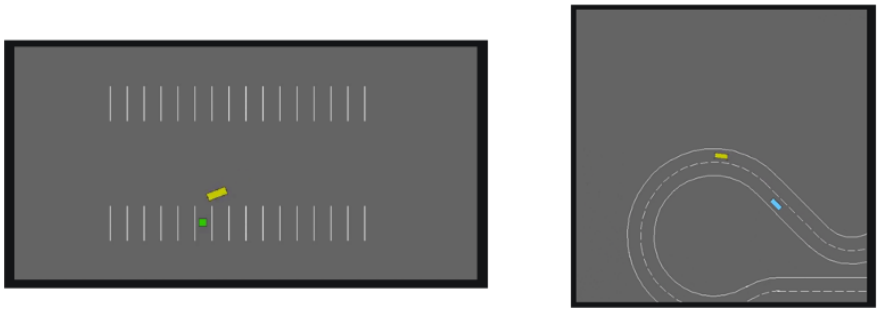

Figure 2: **Left**: Parking Scenario **Right**: Racetracking Scenario

Task Description: In this project, you are required to solve *Four* different driving scenarios as shown in Figure 1 and Figure 2. You are provided with the simulation environment to control the car and complete various tasks with respect to different requirements. The goal is to drive as fast as possible while avoiding potential collisions and fulfilling tasks. The detailed presentation of the simulation can be found in Highway-Env's Documentation (**GitHub link**).

Method Description: Lots of control algorithms could be applied to autonomous driving. For example, traditional control methods, such as model predictive control, rule-based optimal control, and motion planning, are promising and well-developed for autonomous driving. Meanwhile, the learning-based methods have recently achieved significant success in vehicle control, such as the model-free reinforcement learning methods PPO and SAC, and model-based methods like Dyna-Q. Some examples and tutorials about reinforcement learning can be found in the github repository. Reinforcement learning algorithms are recommended and other methods are also acceptable.

Performance Evaluation: We aim to complete tasks as fast as possible while keeping safe. For the highway, intersection and race tracking scenario, the task is to move as quickly as possible. The task of the parking scenario is to park the car in a specific position. The final performance score of each scenario will be computed as a weighted average of the factors:

- **Task Completion**: Progress quickly on the road or satisfy the desired goal destination. Quantified by the ego-vehicle velocity and difference between self-position and target destination.

- **Safety**: Avoid collisions. Quantified by the contact with other vehicles.

The detailed reward function is determined by the reward function and group ranking. For the fairness of evaluation, we use the default configuration for *Parking* task, and we modify the observation type into OccupancyGrid for the other three tasks. Note that although the configuration for each task is fixed during the evaluation, we could modify task configurations during the algorithm's training.

References

[1] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347.

[2] Haarnoja, T., Zhou, A., Abbeel, P., & Levine, S. (2018, July). Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In International conference on machine learning (pp. 1861-1870). PMLR.

[3] Peng, B., Li, X., Gao, J., Liu, J., Wong, K. F., & Su, S. Y. (2018). Deep dyna-q: Integrating planning for task-completion dialogue policy learning. arXiv preprint arXiv:1801.06176.

[4] Leurent, Edouard. An Environment for Autonomous Driving Decision-Making (**Github link**).

[5] Code Base 1: Stable Baselines3 (**Github link**)

[6] Code Base 2: Clean-RL (**Github link**)

[7] Achiam, Joshua. (2018). Spinning Up in Deep Reinforcement Learning. (**Online link**)

2. Robot Control with Reinforcement Learning (4 groups)

Background: Model-free Reinforcement Learning (RL) is gaining prominence in the domain of robot control, holding significant promise. Employing RL for end-to-end locomotion control presents a substantial challenge, particularly when managing high-dimensional, nonlinear systems like quadruped robots.

Task Description: This project mandates the utilization of reinforcement learning techniques to train a locomotion control policy for the **Unitree Go1** quadruped robot model, with the goal of achieving a basic velocity tracking task.

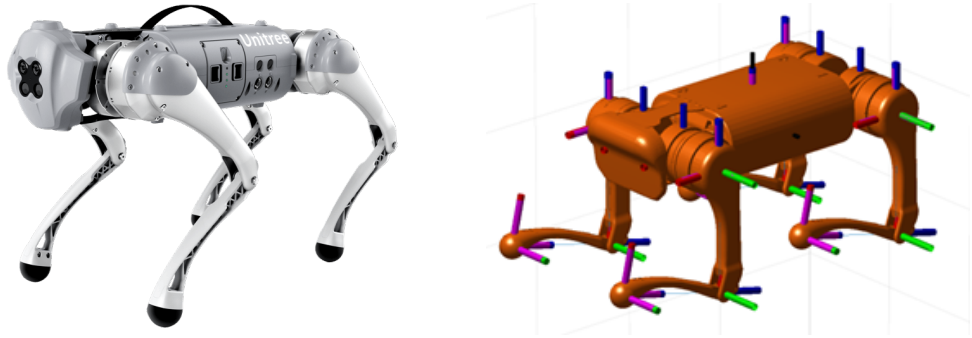

Figure 3: **Left**: Go1 Robot **Right**: Go1 URDF model

This task will be executed within the **Isaac Gym** simulation environment, an NVIDIA-developed prototype for physics simulation tailored to GPU-accelerated reinforcement learning research.

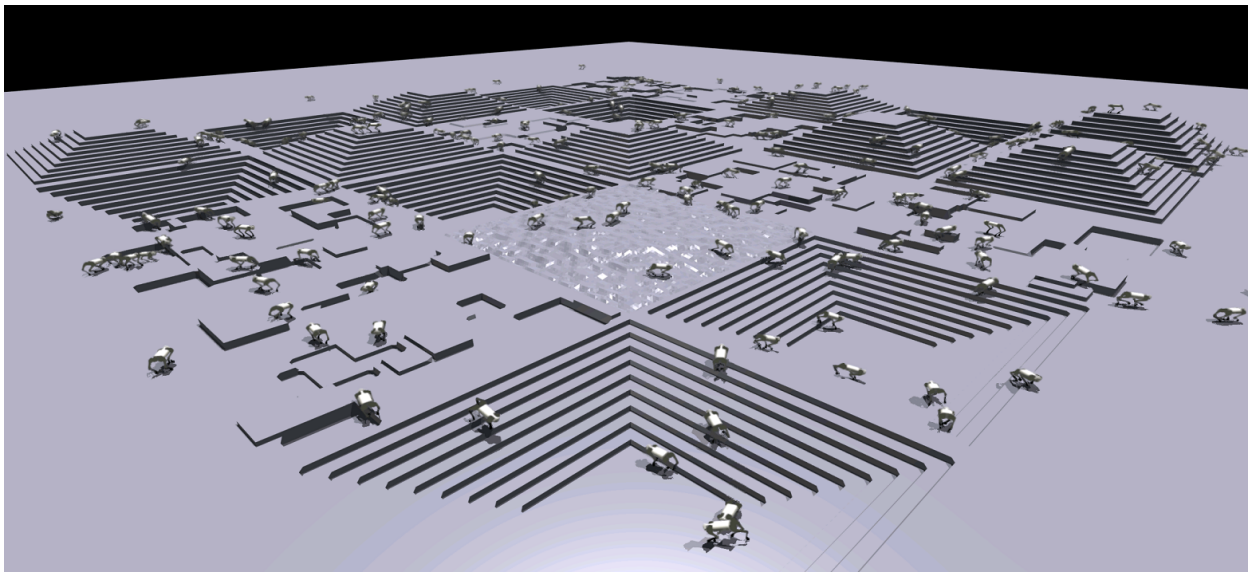

Figure 4: Isaac Gym simulation environment

Method Description: Isaac Gym provides the capability to monitor multiple facets of the robot's state, encompassing parameters like trunk velocity and pose, along with joint position and velocity, and even the local height map. Additionally, we have the capacity to control these 12 joints using position-based commands. Our principal objective is to design a policy network that takes the observed state and velocity command as input and produces target positions for the 12 joints accordingly. Leveraging the established state and action definitions, the utilization of multiple rewards to incentivize or penalize the robot's motion is recommended. Employing a model-free RL algorithm, such as PPO, is

advisable for the training process. This will enable the robot to execute movements in alignment with the designated trunk velocity command.

Performance Evaluation: Our ultimate evaluation criteria will comprehensively assess the accuracy, agility, and stability of the robot's locomotion within the default terrain [4]. The final performance score will be computed as a weighted average of the three factors.

- **accuracy**: quantified by the difference between practical velocity and command velocity.
- **agility**: quantified by the maximum velocity attainable when applying the forward velocity command.
- **stability**: quantified by the transverse accelerate velocity and yaw's angular acceleration when exclusively applying the forward velocity command.

Notice: A Linux system, typically Ubuntu with AMD64 architecture, and an NVIDIA GPU with VRAM $\geq$ 4GB are required.

References

[1] Rudin, N., Hoeller, D., Reist, P., & Hutter, M. (2022, January). Learning to walk in minutes using massively parallel deep reinforcement learning. In Conference on Robot Learning (pp. 91-100). PMLR.

[2] Isaac Gym Simulator Installation Package (**JBox link**).

[3] Go1 URDF model (**JBox link**).

[4] Code Base: Isaac Gym Environments for Legged Robots (**GitHub link**).

[5] Docker Image Base: Installed isaacgym & legged_gym & rsl_rl (**DockerHub link**).

3. Board Game AI Agent (4 groups)

Background: The application of AI in chess games has a long history and has achieved remarkable accomplishments. Chess games are a type of strategy game with well-defined rules and a discrete state space, including games such as Chess, Go, and Checkers. These games have been a fertile ground for AI research due to their complexity and the strategic decision-making required.

Task Description: This project aims to develop an AI agent capable of playing the game of Gomoku by implementing a search algorithm. Gomoku is a board game that is played on a square grid, where two players take turns placing their stones on the intersections of the grid. The objective of the game is to create a straight line of five stones horizontally, vertically, or diagonally. Board games are competitive games with clear objectives and rules, where players strive to achieve victory through strategic planning and decision-making. Search algorithms are designed to assist computer agents in exploring the game tree and making informed moves.

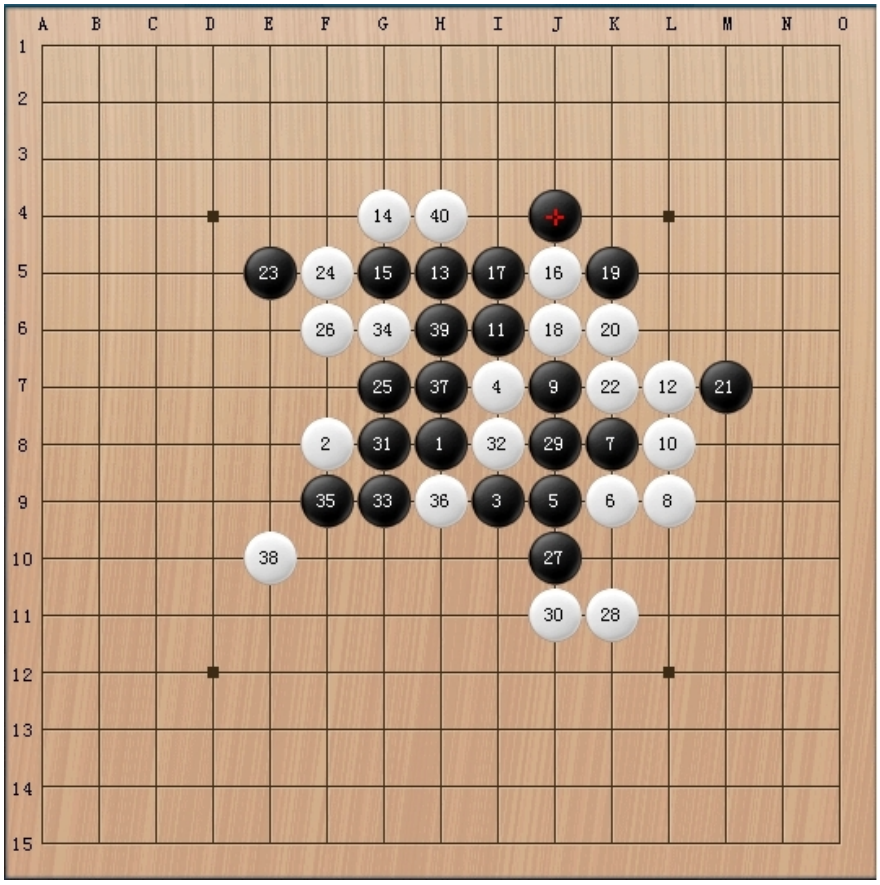

Figure 5: Gomoku game

Method Description: In board games, the game state can be represented as a tree structure, where each node represents a possible game state and the edges represent the possible moves that lead to subsequent states. Search algorithms, such as Monte Carlo Tree Search (MCTS), traverse this tree to find the optimal path or move sequence that maximizes the agent's chances of winning. The MCTS algorithm consists of four main steps: selection, expansion, simulation, and backpropagation, as shown in Fig.3. This project aims to create an intelligent system capable of making optimal moves and defeating pure-MCTS opponents. To enhance the capabilities of the AI agent, integration of deep neural networks into the MCTS algorithm is encouraged. (It is also permissible to explore search algorithms

other than MCTS-based algorithms in this project.)

Performance Evaluation: The performance evaluation will be conducted by comparing the AI agent against pure-MCTS opponents. This approach ensures a fair and standardized assessment, allowing for a direct comparison of the winning rate with the same number of simulations. The reference code of Gomoku Environment and pure-MCTS opponents can be found in the (**JBox**).

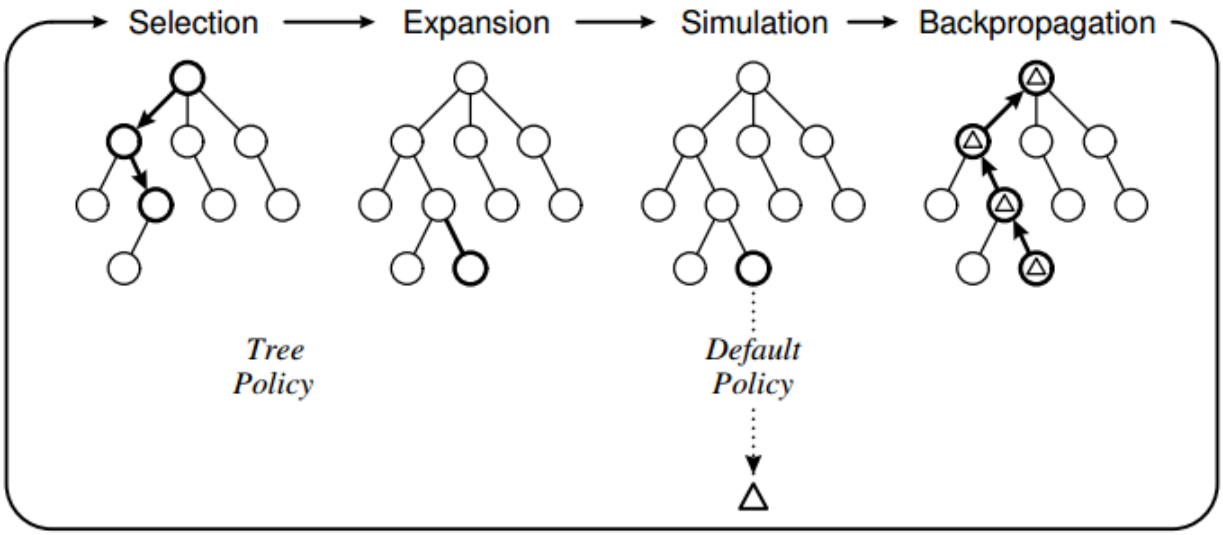

Figure 6: Monte Carlo Tree Search algorithm

References

[1] Schrittwieser, Julian, et al. "Mastering atari, go, chess and shogi by planning with a learned model." Nature 588.7839 (2020): 604-609.

[2] Wang, Yuan. "Mastering the game of Gomoku without human knowledge." (2018).

[3] Gomoku with Monte Carlo Tree Search (MCTS) lectures (**YouTube link**).

[4] Code Base: Gomoku Environmen & pure-MCTS opponents (**JBox link**).

4. Image-to-Image translation with generative model (4 groups)

Background: The generative model has advanced tremendously in recent years and is even capable of mimicking museum-worthy works of artists through their unique style. But creating masterpieces is thought of to be, well, more art than science. So can data science, in the form of GANs, trick classifiers into believing you've created a true art?

Task Description: In this project, your task is to translate realistic photo images into mural-style paints with provided datasets (**JBox link**). In the datasets, photo images are the source domain and mural images are the target domain as shown in Fig. 7, where the target domain images correspond to translated images generated with the CycleGAN. The raw data for mural images are also provided, and extra preprocessing and augmentation can be applied to the mural images in the training dataset.

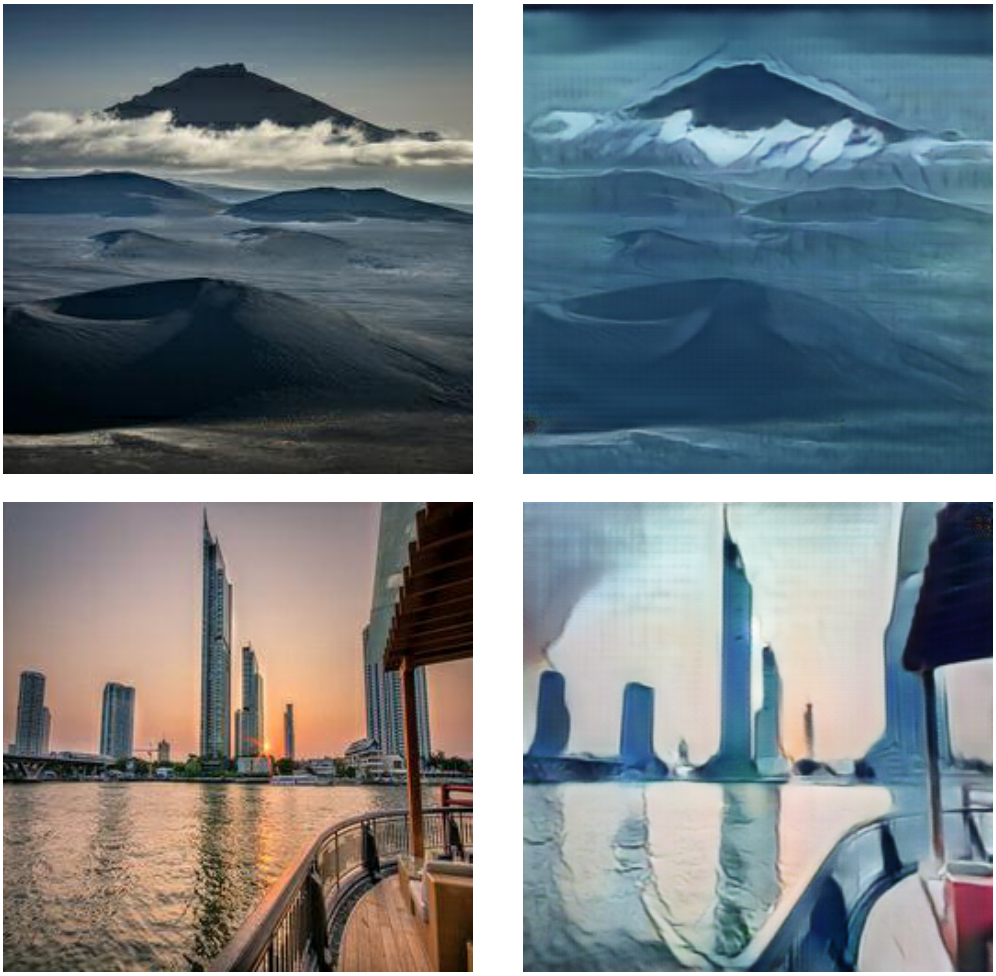

Figure 7: **Left**: Source domain **Right**: Target domain

Method Description: It is a Image-to-Image transtlation problem in this project. Pix2pix is first proposed for paired translation. Further, CycleGAN, DiscoGAN, and DualGAN relax the task to unpaired datasets and StarGAN extends the diversity to multiple domains. In recent years, the diffusion model has beat GAN on image synthesis. With conditional generation, the image condition enables the diffusion model to generate specific domain images with high diversity. A simple Cycle-GAN demo is also released in **Reference** with implementation (**Github**). To enhance the performance of diversity, we recommend the integration of the diffusion model in your deep neural networks.

Performance Evaluation: The performance is evaluated with the test images in the datasets. FID (**Github**) will be calculated between the generated target domain results and the training datasets. At least one translated result for each test image is required in the submission. Optionally, to encourage the diverse generation, 10 extra sets of results for the same test datasets can be submitted. The diversity will be evaluated with LPIPS (**Github**). Pairwise LPIPS among the outputs from the same test image is calculated and averaged for all test images. The complexity of the model is also benchmarked with the number of parameters.

- **Discrepancy**: quantified by the FID between generated results and the trainning datasets.
- **Diversity**: quantified by the LPIPS among the outputs from the same test image.

Notice: NVIDIA GPU with VRAM $\geq$ 4GB is required.

References

[1] J. -Y. Zhu, T. Park, P. Isola and A. A. Efros, "Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks," 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 2017, pp. 2242-2251, doi: 10.1109/ICCV.2017.244.

[2] Y. Choi, M. Choi, M. Kim, J. -W. Ha, S. Kim and J. Choo, "StarGAN: Unified Generative Adversarial Networks for Multi-domain Image-to-Image Translation," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 2018, pp. 8789-8797, doi: 10.1109/CVPR.2018.00916

[3] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. In Proceedings of the 34th International Conference on Neural Information Processing Systems (NIPS'20). Curran Associates Inc., Red Hook, NY, USA, Article 574, 6840–6851.

[4] Dhariwal, Prafulla, and Alexander Nichol. "Diffusion models beat gans on image synthesis." Advances in neural information processing systems 34 (2021): 8780-8794.

[5] J. Ho and T. Salimans, "Classifier-Free Diffusion Guidance." arXiv, Jul. 25, 2022. doi: 10.48550/arXiv.2207.12598.

[6] Cycle-GAN Demo (**JBox link**)

5. Traffic Object Detection (4 groups)

Background: Perception is a fundamental component and the very first step of autonomous driving systems. It plays a critical role in ensuring the safety, reliability, and effectiveness of self-driving vehicles. Traffic object detection is a common task for road perception using computer vision methods. It refers to the process of identifying and locating various objects and obstacles in a scene related to road traffic. These objects can include vehicles, pedestrians, cyclists, road signs, traffic lights, and other elements commonly found on roadways.

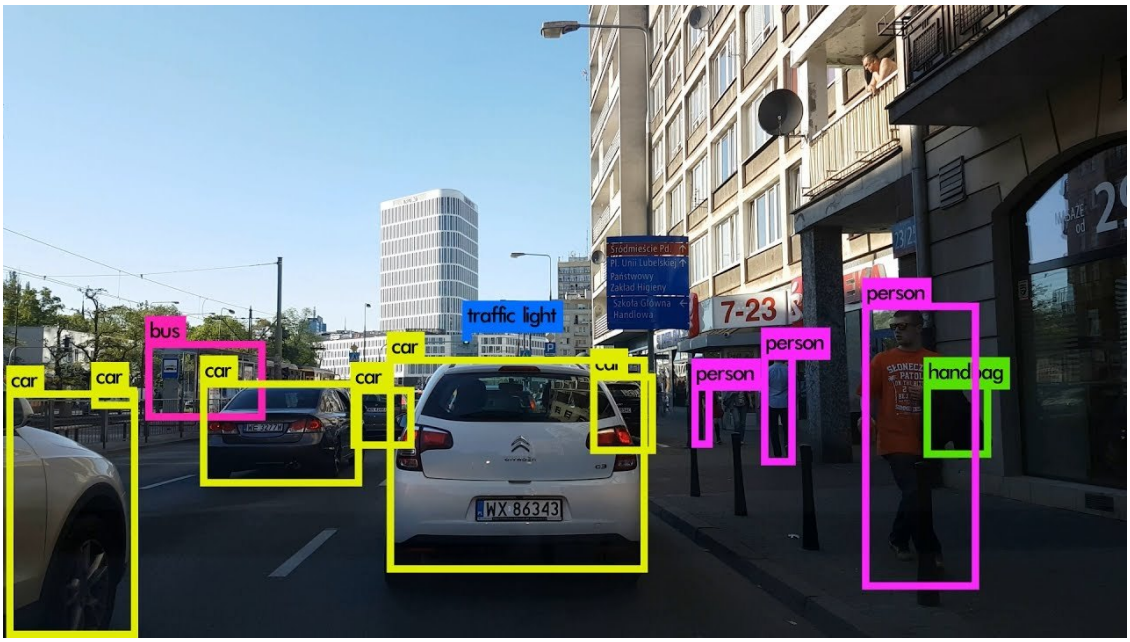

Figure 8: traffic object detection

Task Description: In this project, you are required to automatically frame the location and classify the ten types of targets in these pictures by using computer vision methods. You are able to use the checkpoints pretrained on COCO dataset and ImageNet, but you can only use the dataset provided (**Jbox link**) which includes 4000 labelled images and 4000 unlabelled images for training.

Method Description: Traffic object detection methods can be divided in classical computer vision approaches and deep learning methods. Traditional computer vision techniques include methods like Histogram of Oriented Gradients (HOG), and feature-based object detection. Deep learning has revolutionized object detection with the introduction of Convolutional Neural Networks (CNNs) and Transformer. Deep learning methods can be categorized into two-stage, one-stage and end-to-end detectors. Two-stage detectors, like Faster R-CNN (**GitHub link**), typically involve region proposal networks to identify candidate object regions before classifying and refining their positions. One-stage detectors, such as YOLO series (**GitHub link**) and SSD (**GitHub link**), directly predict object bounding boxes and class labels in a single pass. End-to-end detectors, like DETR (**GitHub link**), can produce high-quality and non-overlapping bounding box predictions for each image without the need for post-processing techniques like Non-Maximum Suppression (NMS). Employing a deep learning model, such as YOLOv8, is advisable for the training process. You are also encouraged to use semi-supervised training methods like semi-DETR (**GitHub link**) to make full use of the 4000 unlabelled images.

Performance Evaluation: Model performance will be evaluated based on both the accuracy and the complexity of your model. Model performance will be ranked by average rankings of model accuracy and complexity, and teams

with equal average rankings will be ranked by model accuracy.

- **Accuracy**:
  Model accuracy will be ranked by mAP@0.5:0.95 indicator(↑) which you can evaluated by using cocoapi.
- **Complexity**:
  The complexity of the model will be represented by the number of giga floating-point operations, GFLOPs(↓) for short, which is the lower the better. You are suggested to use the thop python library.

References

[1] Tomasi, C. (n.d.). Histograms of Oriented Gradients.

[2] Ren, S., He, K., Girshick, R. & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems, 28.

[3] Wang, C. Y., Bochkovskiy, A., & Liao, H. Y. M. (2023). YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 7464-7475)

[4] Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., & Zagoruyko, S. (2020, August). End-to-End Object Detection with Transformers. In European conference on computer vision (pp. 213-2999). Cham: Springer International Publishing.

[5] Zhang, J., Lin, X., Zhang, W., Wang, K., Tan, X., Han, J., Ding, E., Wang, J., & Li, G. (2023). Semi-DETR: Semi-Supervised Object Detection with Detection Transformers. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 23809-23818).

[6] Code Base: YOLOv8 (**GitHub link**)

[7] YOLO series Demo (**YouTube link**)