NATIONAL UNIVERSITY OF SINGAPORE

CEG5503: Intelligent Autonomous Robotic Systems

Project1 (30%)

Instructions

1. The assignment constitutes collaborative tasks. These tasks needs to be done in collaboration with 3 students in a group.
2. In this module, all the assignments/projects are coherent, i.e., the subsequent assignments and projects are to be added on top of the earlier ones. Students are encouraged to thoroughly search, study, and research on the tasks of your selected robot: be it an autonomous vehicle or a marine aircraft, which you find it convenient to work on for your subsequent assessments.
3. Mention your group #, matriculation numbers, and the module code on the cover page; each group is required to submit the assignment independently.
4. The minimum report length is 10 pages; 1.5 line spacing; 12-point font; 1-inch margins; upload in PDF including proper citations and references. Importantly, the report must have following **five sections**: Task-I, Task-II, and Task-III.
5. Submit the zip file including your codes, reports, etc., in Canvas in Week 10 by 6pm on 30 March 2025 (Sunday). Follow the file name convention as: Assignment_Group#.zip If you have additional materials, they can be included as appendices with your Assignment_group_appendix a, b, and c.
6. Feel free to contact the Graduate Assistant, Zhang Binjie (binjie97@u.nus.edu) or Zhang Aoqian (e1144122@u.nus.edu) should you have any queries.

Objective:

Motivated by ChatGPT, this project aspires to engineer intelligent digital entities proficient in conversational interactions, driving capabilities, and car interaction emulation using the NUS Cloud platform. The overarching objective is to cultivate an amiable ecosystem for interactive agents within a simulated setting. These digital entities may encompass virtual drivers for automobiles, robotic delivery systems tailored for deployment in hospitality, corporate settings, or manufacturing environments. Additionally, the scope extends to surface vehicles or aircraft equivalents, should relevant analogs be identified or obtained.

The secondary objective involves the integration of state-of-the-art YOLOv7 object detection capabilities into the digital entities. This integration is crucial for endowing the virtual agents with real-time intelligence in a virtual environment. The successful implementation of YOLOv7 not only enhances the project's technical sophistication but also facilitates a deeper understanding of employing cloud-based platforms for the training of deep learning models.

The final task is dedicated to the development of a dialogue system and a local Large Language Model (LLM) platform. This phase aims to refine the conversational abilities of the digital entities, fostering seamless and contextually relevant interactions. By establishing a local LLM platform, the project aims to empower the digital entities with language understanding and generation capabilities, contributing to the overall advancement of their communicative competence within the simulated environment.

Task-I: Design and Simulation in a Virtual Environment (20 points)

To show your creativity and imagination, design your own concept and future IARs. For your **desired** intelligent autonomous systems (cars or marine vessels), discuss the features and functions of your desired concept robot in the digital world, and explain their principle of operation in reality (physics) and virtual reality (graphics+physics =virtuity). You may use Unity or Unreal Engine to simulate in a virtual environment. You are allowed to use open-source CAD models if your chosen IARs architecture model is complicated.

Ensure that your model incorporate minimum basic essential sensors and physics associated with its physical entity. For example, a typical autonomous vehicle model should comprise of chassis, batteries, sensors such as binocular camera, fisheye camera, LIDAR, to scan, perception and recognize the surrounding environment, etc. The physics should ensure that collisions are not elastic, which is an ideal scenario, to mimic the real object more closely in the virtual environment. You are also encouraged to brainstorm with some exciting ideas, for example, add your preferred sensors, your designed modules, your developed structures, etc.

Note: *For tasks I, you may consider an example from the lectures that you have found inspiring. Nonetheless, you are welcome to explore in the internet as it is an open-ended task with an aim to evoke creative solutions from your end.*

Task-II: Objection Detection using YOLOv7 at NUS HPC (35 points)

This task requires you to implement object detection using YOLOv7 in both physical and virtual environments to make your robot intelligent in both worlds. The objects to be detected could include: passenger, pedestrians, trees, faces, marine animal, birds, ships, etc. Specifically, establish your own cloud environment using Singularity and Miniconda environment for deep learning training on NUS HPC. After successful training, feed the real-time video of your working concept during the deployment in a virtual environment to the deep learning network. You are required to provide a maximum of 1 min video recording of result obtained from detection.

	Task II: YOLOv7 assessment	Score Distribution
i)	Robustness: the successful implementation of the task that includes building your own dataset, training on HPC, and YOLOv7 testing in a virtual environment real-time; Estimate the robustness of your model by testing on a dataset with shifted distribution from your training dataset.*	21
ii)	Defence: incorporate numerous hard negative and positive training samples as much as possible to get the maximum	7

	precision and recall. For example, you need to consider at least two objects that resemble very closely such as fish and shark in the environment for marine project and your model prediction accuracy should remain defended with high confidence. Total number of classes (including robot) to be detected should be at least 6 and 3 for group with 3 and 2 members, respectively.*	
iii)	Attack: There are popular adversarial attacks on deep learning models such as I-FGSM, I-StepLL, C&W L2, DeepFool, and DAG. Implement any one of them on your model and evaluate the testing accuracy real-time.Discuss the reason if there is any reduction in accuracy. Implement a counter algorithm to retain the accuracy in presence of such models. (Hint: If you are using ReLU, you may try replacing it with incremental step ReLU).*	7

*The text highlighted in red indicates the group with two members—due to uneven strength of the class—are exempted from attempting the respective portion of the task in (i) and (iii). For part (ii), the # of detection classes are reduced for a two-member group.

A supplementary document implementing YOLOv7 on a VOC2007 dataset is provided for your reference. The document helps you familiarize with downloading prerequisite platforms, using NUS HPC, typical implementation of YOLOv7, and creating your own dataset of training.

Some recommendations:

- If you wish to select an **autonomous vehicle**, you can create an environment with road or traffic signs. For YOLOv7 detection, you may program your vehicle to detect the objects in the environment such as traffic/road signs, etc.
- If you are going with **marine craft** in intelligence surveillance tasks, there are various exciting environment models you may use such as marine animals, ship crafts, etc.

Task-III: Develop a dialogue system and your own Large Language Model (35 points)

Dialogue systems leverage various modes of communication, including text, speech, graphics, haptics, gestures, and other modalities, to facilitate interactions on both input and output channels. For this task, the initial step involves acquainting oneself with prevalent open-source Large Language Models (LLM) such as Llama, GLM, among others. Understanding their architecture, functionalities, and integration capabilities is imperative for the subsequent development phase. The subsequent task involves crafting code to construct a Graphical User Interface (GUI) demonstration for the dialogue system. Tkinter, a widely used Python library for creating GUI applications, is recommended for this purpose.

Task-IV: Evaluate and analyze results (10 points)

In this section, it is essential to analyze the advantages and disadvantages of trained models for Task II and Task III. You can evaluate these models using public datasets and present the metrics for a comprehensive assessment.

The assignment score will also be impacted by the complexity of the problem selected. Hence, try to be as creative as possible and work on an interesting IAR. Make sure to mention the contribution of each member in the last page of the report.

As mentioned in the instruction, your future assessment on kinematics, dynamics, map building, guidance and navigation project and lab tasks are required to be integrated in the virtual environment developed in this assignment. Select your desired IARs on a specific autonomous vehicle or marine craft after a thorough research.

Score distributions

The distribution of the scores for each item is listed below. If an item is completed by more than one person, they will share the marks of the item equally.

Task	Type	Score Distribution
Task 1	individual	20%
Task 2	group	35%
Task 3	group	35%
Task 4	individual	10%