ITI108 Assignment 1 (Total: 100 marks)

Designing and implement a Deep Learning Model for Human-Instruction Controlled Autonomous Robot(Total marks 100)

1. Instructions

1. Complete all the tasks in this assignment.
2. Submission date: 31 Jan 2024 23:59
3. This is an individual assignment.

2. How to submit the answer

Your final submission should contain the following.

- The jupyter notebook ITI108(2023S2)_assg1_123456A.ipynb or ITI108(2023S2)_assg1_123456A.py with the python code
- A short video clips to demonstrate how your code interact with the robot in the simulator
- Submit the files to BrightSpace.

Submission Guidelines:

Submit a code (Python scripts) with comments and explanations.

Provide clear documentation on how to run your code and reproduce your results.

Video of the demo should not be more than 5mins.

Important Note: Plagiarism will not be tolerated. Ensure proper citation and references for all sources used in your assignment.

3. Objective

The objective of this assignment is to create a deep learning model in Python that allows an autonomous robot to be controlled using human instructions. You need to demonstrate the human interaction with the robot in the simulator with the deep learning model.

4. Task

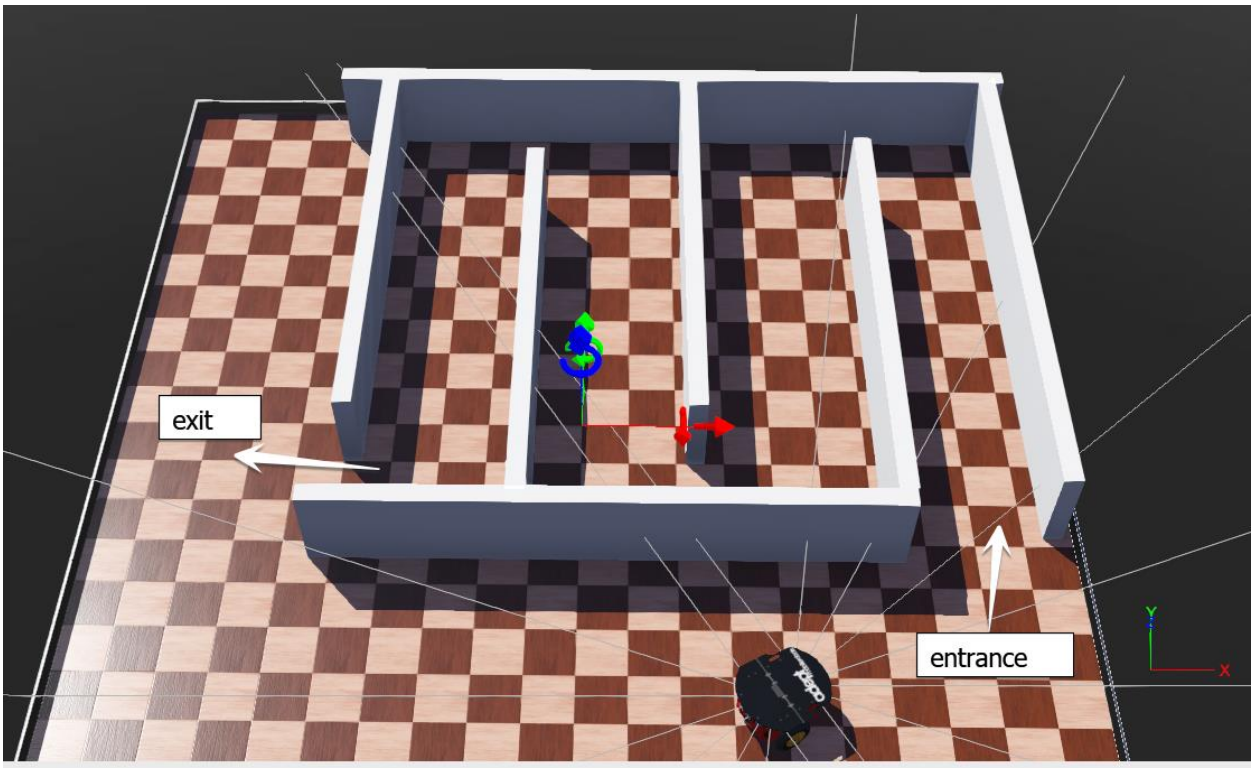
Data Collection and Preprocessing
Define the types of data you'll need for training the model, such as sensor data from the robot, audio input for human instructions, and any other relevant data sources. Describe how you plan to collect and preprocess this data to make it suitable for training.

Model Architecture
Design a deep learning architecture for the robot's control system. This architecture should take in human instructions and sensor data, and output control commands for the robot. Explain the rationale behind your design and any pre-trained models you plan to use.

Data Collection and Training
Implement data collection and training procedures for the deep learning model. Describe the data pipeline and data augmentation techniques used, and provide details on model training, including hyperparameters and training strategies.

Human-Instruction Interface
Develop an interface for human instructions. This could be in the form of voice commands, text input, or other modalities. Discuss the techniques used to understand and process human instructions.

Robot Control and Evaluation
Integrate the trained model with the robot and evaluate its performance. Test the robot's ability to understand and follow human instructions in various scenarios. Use the simulation environment (Webot) to demonstrate the human control of the robot through the use of the trained model.
The following illustrates the image of the simulation environment. It consists of the autonomous robot that can be controlled via human instruction using the trained model. Use the human instruction to navigate the robot to move from the entrance of the maze to exit.



5. Assessment Rubric

Assignment Rubric: Designing a Deep Learning Model for Human-Instruction Controlled Autonomous Robot

Task	Excellent	Good	Satisfactory	Needs Improvement
	(10.0 -7.5 marks)	(7.4 – 5.0 marks)	(4.9 – 2.5 marks)	(2.4 – 0.0 marks)
Data Collection and Preprocessing (10marks)	Thoroughly defined data types, well-structured data collection plan, and effective preprocessing strategies.	Clearly defined data types and collection plan, but preprocessing may have some gaps.	Data types and collection plan are defined, but some preprocessing details are missing.	Incomplete or unclear data collection and preprocessing strategies.
	(20.0 -15.0 marks)	(14.9 – 10.0 marks)	(9 - 5.0 marks)	(4.9 – 0.0 marks)
Model Architecture(20 marks)	A well-justified and innovative deep learning architecture that effectively	A well-designed architecture with	A good architecture is	

Task	Excellent	Good	Satisfactory	Needs Improvement
	combines human instructions and sensor data.	clear justification, but may lack innovation.	presented with some justification.	Architecture is not well-justified or lacks essential components.
	(20.0 -15.0 marks)	(14.9 – 10.0 marks)	(9 - 5.0 marks)	(4.9 – 0.0 marks)
Data Collection and Training(20marks)	Detailed implementation of data collection and training procedures with effective data augmentation and well-documented training strategies.	Clear implementation of data collection and training, but some details may be lacking.	Implementation is present but lacks details and thorough documentation.	Incomplete or unclear implementation of data collection and training procedures.
	(20.0 -15.0 marks)	(14.9 – 10.0 marks)	(9 - 5.0 marks)	(4.9 – 0.0 marks)
Human-Instruction Interface(20 marks)	Innovative and effective interface design for human instructions with robust with human interface sensory.	A well-implemented interface with effective human sensory, but it may lack innovation.	A functional interface is implemented with human sensory but may have some limitations.	Interface design and human sensory implementation are incomplete or ineffective.
	(20.0 -15.0 marks)	(14.9 – 10.0 marks)	(9 - 5.0 marks)	(4.9 – 0.0 marks)
Robot Control and Evaluation(20 marks)	The robot's control and evaluation process are well-integrated and effectively demonstrate the model's performance.	Integration and evaluation are well-executed but may have minor shortcomings.	Integration and evaluation are present, but there may be significant issues or gaps.	Integration and evaluation are incomplete or ineffective.
	(10.0 -7.5 marks)	(7.4 – 5.0 marks)	(4.9 – 2.5 marks)	(2.4 – 0.0 marks)
Code Documentation and Citations(10 marks)	Proper documentation of code and methodologies with comprehensive citations.	Documentation of code and citations are mostly proper but may have some omissions.	Documentation of code and citations are basic but may have significant omissions.	Poor or inadequate documentation of code and citations.

6. Simulation Environment-Webot

Install the webot simulator(refer to Autonomous System Practical for more detail). Use the given project world for the simulation and integrate your Human interface deep learning model into the python robot controller code.

```
"""my_controller controller."""

# You may need to import some classes of the controller module. Ex:
#  from controller import Robot, Motor, DistanceSensor
from controller import Robot
from controller import Keyboard

CRUISING_SPEED= 5.0
TURN_SPEED = CRUISING_SPEED/2.0
TIME_STEP = 64

# create the Robot instance.
robot = Robot()
left_wheel = robot.getDevice('left wheel')
right_wheel = robot.getDevice('right wheel')
left_wheel.setPosition(float('inf'))
right_wheel.setPosition(float('inf'))
left_wheel.setVelocity(0.0)
right_wheel.setVelocity(0.0)


keyboard = Keyboard()
keyboard.enable(TIME_STEP)



def command_motor(cmd):
        left_wheel.setVelocity(cmd[0])
        right_wheel.setVelocity(cmd[1])

while robot.step(TIME_STEP) != -1:
        key = keyboard.getKey()
        if(key == ord('W')):
                left_wheel.setVelocity(CRUISING_SPEED)
                right_wheel.setVelocity(CRUISING_SPEED)
        elif key == ord('S'):
                left_wheel.setVelocity(-CRUISING_SPEED)
                right_wheel.setVelocity(-CRUISING_SPEED)
```

```
        elif (key ==  ord('A')):
                left_wheel.setVelocity(-TURN_SPEED)
                right_wheel.setVelocity(TURN_SPEED)
        elif (key ==  ord('D')):
                left_wheel.setVelocity(TURN_SPEED)
                right_wheel.setVelocity(-TURN_SPEED)
        elif (key ==  ord('E')):
                left_wheel.setVelocity(0.0)
                right_wheel.setVelocity(0.0)
        else:
                pass
```

==========End of the Assignment==========