**Human-Following Robot in a Virtual Environment**

**Author: B924007**

*READ BEFORE RUNNING CODE*

The LEARN submission code only contains the raw code for the human detection and tracking algorithms, as well as the SDF files for the different Gazebo worlds. However, it does not contain the raw files in their respective directories in order to run the codes.

To recreate the project, follow these steps:

1. Unzip folder. Move all folders into your home directory.

2. Ensure that the ‘mmdetection’ folder has the the following sub-folders: ‘coco\_image’, ‘configs’, ‘tools’, and ‘work\_dirs’.

3. Follow Appendix A to set up the virtual environment.

Note that due to Linux system updates at the university, there may be compatibility issues when running ROS or PyTorch commands.

4. If you would like to see the SDF and configuration files for the TurtleBot3 Waffle Pi and House, as well as the DAE files for the human model, these are available in the GazeboModels folder.

5. If you would like to run the Gazebo world scenarios, insert the command in Appendix B4 into the Linux terminal. Ensure you open the terminal from the HumanFollowingRobot folder to run this command.

Simply change the name of the SDF file at the end of the command to run different scenarios. For example, replace ‘walk\_straight.world’ with ‘walk\_back.world’.

Also ensure that you use CTRL+C in terminal to exit one gazebo world before running another world (as I ran into issues if the world did not close properly).

6. If you would like to access the images, go to mmdetection -> coco\_image. There will be three folders to testing, training, and validation. Each of these folders will also have a JSON file that will contain the annotations required for each image in COCO format.

7. If you would like to train the YOLOv3 model, run the following command once you open a terminal within the mmdetection folder:

python tools/train.py configs/yolo/yolov3.py

Note: The relevant configuration files that are uses are yolov3.py and yolov3\_base.py. Feel free to modify parameters in the yolov3\_base.py file such as the number of epochs (which is currently set to 300) or the type of optimizer (currently SGD) or the learning rate.

8. If you would like to test the trained yolo model, open a terminal in the mmdetection folder, and write the following command:

python tools/test.py configs/yolo/yolov3.py work\_dirs/yolo300/epoch\_100.pth

Note: Feel free to change the epoch number and run the command again to see the performance at different stages of the training process.

For example:

python tools/test.py configs/yolo/yolov3.py work\_dirs/yolo300/epoch\_50.pth

9. To test the human detection model:

Open a terminal from the HumanFollowingRobot folder and run the code for Scenario 1 or 2:

gazebo --verbose -s libgazebo\_ros\_init.so -s libgazebo\_ros\_factory.so ./{gazebo\_world\_name}

Replace the {} with ‘walk\_straight.world’ or ‘walk\_back.world’

Then open another terminal from the same file directory, and call the human detection python code you would like to call:

‘python hd.py’

Or

‘python hd\_conf.py’

10. To test the human tracking model:

First ensure Step 9 is activated so there is a running Gazebo world, and a human detection algorithm running.

Then open another terminal from the same file directory and call the python code for the human tracking algorithm you would like to use.

For example:

‘python ht.py’

With these step-by-step instructions, the project should successfully be recreated.