Model transformation and application

Note: This course is only applicable to the R2L car and Yahboom's autopilot map.

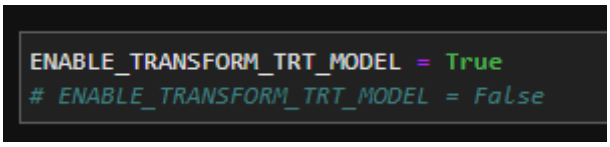
If you use other models or other maps, you need to develop and debug the code yourself, and the code provided in this chapter cannot be used directly.

1. Development environment

This training model needs to use TensorFlow and other related tools. Due to the use of many tools and complex deployment, the ROSMASTER car factory image has deployed the development environment of the training model, which can be used directly without repeated construction.

2. Code analysis

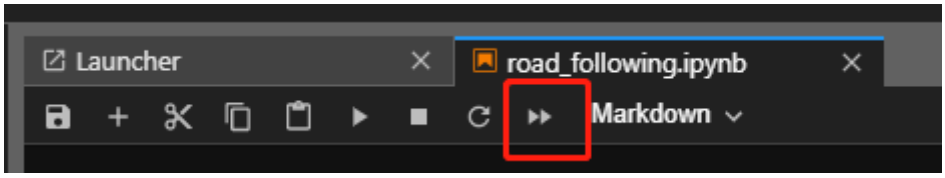
Please check the specific code: Rosmaster/auto_drive/road_following.ipynb



The ENABLE_TRANSFORM_TRT_MODEL parameter is the flag of the conversion model. If it is True, it will be converted to a trt model. If it is False, the model will not be converted (it is considered that there is already a trt model), and the trt model will be loaded directly.

3. Conversion mode

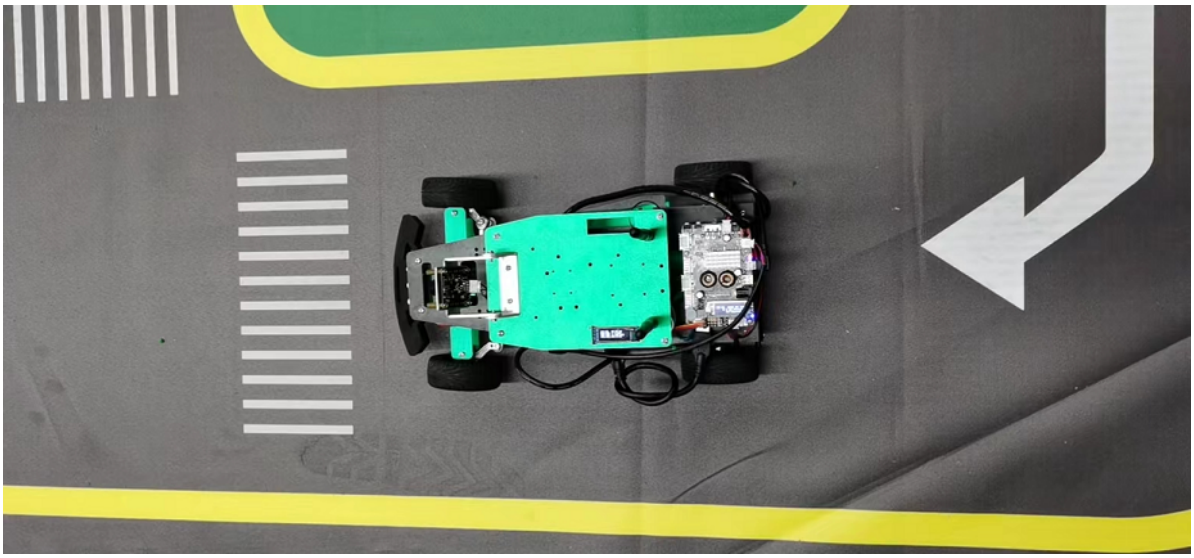
Open jupyter lab with a browser, find the road_following.ipynb file and open it, click the button to run all cells, wait for all the cells to run, and the buzzer will sound a 'beep'.



At this time, put the car on the track, move the position of the car, and you can see that the front wheel servo will change. Since the speed of the car is 0 at this time, it will not run on the track. We need to pass the speed to the car to make the car move.

As for why the speed is not directly set for the car, the reason is that it takes a certain amount of time to start the model transformation, and if the car starts to move after starting, sometimes the position is not on the track, or there is a problem with the trained model, the car will run around. Therefore, it is recommended to use the handle to control the speed of the trolley, which is

convenient for debugging.



4. Handle control car

The code path of the car handle control: Rosmaster/auto_drive/joystick_R2L.py

Insert the handle receiver into the USB port of the main control board, and then turn on the power of the wireless handle.

Open the terminal and run the following command to open the handle control program

```
cd ~/Rosmaster/auto_drive/
python3 joystick_R2L.py
```

At this time, press the START button of the wireless handle, and you can hear the buzzer to indicate that the connection is successful. Release the START button and the buzzer will turn off.

The functions of the wireless handle are as follows:

Handle	**Effect**
left stick up/down	Car forward, backward
Right stick up/down	The front wheel of the car turns left and right
Arrow key forward	The car moves forward (release without stopping)
Arrow key backward	The car backs up (release to stop)
L1 key	Adjust speed to 0.5
R1 key	Adjust speed to 0.3
L2/R2 key	Stop the car
"START" key	Control buzzer/end sleep
X key	Front wheel fine adjustment to the left
B key	Front wheel fine adjustment to the right

The default handle speed is 0.3, you can press the L1 key or R1 key to modify the speed. Different speed values may require re-collecting the data set and training the model to achieve the optimal effect. Then press the forward key of the direction key, and the car starts to move. If the car runs out of the track, you can press L2 or R2 to stop, or you can press the direction keys to go back, and release the direction keys to go back and stop automatically.

5. End the kernel

Since the camera is used for this training, the kernel needs to be shut down after the program runs.

Click on the kernel management bar, find road_following.ipynb and click the X on the right to end.