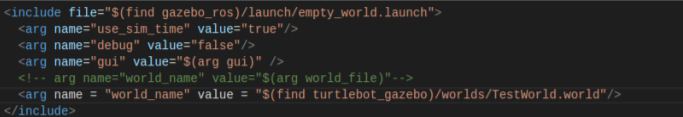
To run the function of yolo detection and move\_base tracking. Here are some steps to config the running environment.

First download the turtlebot models in your workspace. Following the steps shown in wiki website. <http://wiki.ros.org/turtlebot/Tutorials/indigo/Turtlebot%20Installation>

After download the turtlebot models, locate the turtlebot workspace, and put this my\_yolo\_track package in to the src folder under the turtlebot folder. In the my\_yolo\_track package, there is a map folder, copy the TestWorld.world into turtlebot/src/turtlebot\_simulator/turtlebot\_gazebo/worlds folder. And edit the turtlebot\_world.launch file located the /turtlebot\_gazebo/launch folder. Only change the “world\_name” value to the figure shown below.



Then every time run this launch file, the simulation world will be the TestWorld.

Install the newest version of opencv(version > 3.4.2 is ok).

In ROS kinetic, opencv3.3.1-dev is installed by default, and the python environment of ROS is only version 25 2.7 by default. After installing other versions of OpenCV through pip, ROS will call the default 3.3.1 first, so it is important to uninstall this version first:

sudo easy\_install trash-cli

sudo trash-put /opt/ros/kinetic/lib/python2.7/dist-packages/cv2.so

download this opencv by using pip command. for more information, please check the offical opencv-python website https://pypi.org/project/opencv-python/. There are all the version of opencv and users can download any version of opencv here.

pip install opencv-python

pip install opencv-contrib-python

Then download the yolov3.weights and yolov3.cfg file for yolov3 from the darknet official website <https://pjreddie.com/darknet/yolo/>,

And put the yolov3.weight file into the /my\_yolo\_track/weight folder.

Go back the workspace, and catkin\_make. Don’t forget source the terminal after catkin\_make. If everything is ok, when running the command below

roslaunch turtlebot\_gazebo turtlebot\_world.launch

there will be a turtlebot located in a coffee, and facing two different person.

And when input the command below the terminal.

rosmsg list | grep my\_yolo\_track

two types of messages should be appeared, my\_yolo\_track/TrackInfos and my\_yolo\_track/Trackinfo.

Then you can start to test the yolo detection and the tracking function.

Open a new terminal and run

roscd my\_yolo\_track

cd scrpits

python real\_time\_yolo.py

there will be a window showing on the screen, with the detected person information. Something like the figure shown below.



And then, run the command to watch the detected objects information.

rostopic echo pub\_track\_info

the objects information which includes track class, track\_id, center\_x, center\_y, distance, color\_bgr will be displayed.

Open a new terminal and run the move\_base section function by input commands.

roscd my\_yolo\_track

cd scrpits

python start\_tracking\_yolo.py

then you will need to input the class name, the track id and the desired distance the turtlebot need to keep. And then the turtlebot starts to track the target.

And don’t forget source the workspace for each time you open a new terminal