System Requirement: Ubuntu 18.04
ROS Version: ROS Melodic

Step-1:
Setup the Office environment from 3DGEMS Dataset following steps given in 3DGEMS_Dataset_with_Gazebo_ROS.pdf

Step-2:
Install and setup RTAB-Map with turtlebot3 using steps given in rtabmap_ros_with_turtlebot3_in_ROS_Melodic.pdf

Step-3:
Setup Mask-RCNN with ROS and use it with gazebo and turtlebot3 with 3DGEMS Dataset following the steps given in Integrating_Mask_RCNN_with_ROS.pdf

Step-4: (After setting up all above three steps close everything and follow following step)

Launch the Office environment:

```
export TURTLEBOT3_MODEL=waffle
roslaunch turtlebot3_gazebo office_env_small.launch
```

Open a new terminal and launch RTAB-Map node:

```
export TURTLEBOT3_MODEL=waffle
roslaunch turtlebot3_slam turtlebot3_slam.launch slam_methods:=rtabmap
```

Open a new terminal and launch Mask-RCNN node:

```
roslaunch mask_rcnn_ros mrcnn_node.launch
```

Open new terminal and launch teleoperation node:

```
export TURTLEBOT3_MODEL=waffle_pi
roslaunch turtlebot3_teleop turtlebot3_teleop_key.launch
```

Now in the RVIZ of RTAB-Map , add Image and visualize topics of `/camera/rgb/image_raw` and /mask_rcnn/visualization

Start recording this data in rosbag file:
Open a new terminal and type following command:

```
rosbag record -a
```

Save this rosbag file with name “data.bag”

Also, once you terminate all the above process map will be automatically stored in the following location:

~/.ros/rtabmap.db

Get 2D Occupancy grid map via. Following steps:

Open a terminal window and type following command:

```
roscore
```

Open a terminal window, go to the folder where the .db file is there, (via. cd command)

```
rosrun rtabmap_ros rtabmap _database_path:=rtabmap.db
```

(Here, rtabmap.db is the filename , give the name of the file as per by what name you saved your file)

```
rosrun map_server map_saver map:=proj_map
```

Now, you can find 2D map files i.e, map.pgm and map.yaml in the same directory where you are now.

Convert map.pgm to map.jpg

```
sudo apt install imagemagick
```

```
convert map.pgm map.jpg
```

Step-5:

Setup and Install jsk_recognition meta package using steps given in
Using_pcl_library_for_point_cloud_segmentation.pdf

Step-6:

Go to **/catkin_ws/jsk_recognition/jsk_pcl_ros/launch/mask_image_filter.launch**

Change input `<remap from="~input/mask" to="/mask_rcnn/result" />`
And save this file.

Again record whatever process is happening via. :

```
rosbag record -a
```

Play “data.bag” file and simultaneously launch mask_image_filter node:

New terminal:

```
rosbag play data.bag
```

New terminal:

```
roslaunch jsk_pcl_ros mask_image_filter.launch
```

After this save the recorded .bag file as “data1.bag”

Hence, in “data1.bag” we have indices of segmented point clouds also which will be used in the next step for projection in the map.

Step-7:

Create package into your catkin_ws:

```
cd ~/catkin_ws/src
catkin_create_pkg projection rospy std_msgs
```

Now, download the python file **projection.py**
and save in the folder: **/catkin_ws/src/projection/src/**

Open the file and give the path of “data1.bag” in the data_path variable and the path of “map.jpg” to the map_path variable.

Then run the following command:

```
rosrun projection projection.py
```

Hence, you will get the “semantic_map.jpg” in the same path where “map.jpg” is there. Also, “labels.txt” which has pixel value(0-255) for an object and semantic category of the object.