Perception and Decision Making in Intelligent Systems
Homework 2: A Semantic Mapping Framework
Announce: 10/17, Deadline:11/7 23:59

Introduction

In this homework, you need to train a semantic segmentation model to predict the labels of all objects in the environment, and reconstruct the 3D semantic map of the apartment_0 environment of Replica dataset.

Requirements

1. Download other scenes in the Replica dataset.

2. To train the semantic segmentation model, device with GPU is required. If you don't have GPUs, Google Colab can be utilized as an alternative.

3. Make sure you have installed pytorch. Please follow the official webpage to install the suitable version based on your device.

Implementation

There are two tasks you need to implement.

Task 1. Semantic Model Training

Please refer to the semantic-segmentation-pytorch project. If you have any questions about installation or usage, you can check out the project page first. Also, you can use other semantic segmentation toolbox like mmsegmentation and so on.

1. Data Collection:

(1) Use **data_generator.py** to control agent in habitat to collect data: rgb / depth / semantic images from replica dataset. Because we want you to learn the impact of distribution shift on the performance of semantic segmentation, you need to collect two sets of training data:

- Dataset1: Collect data from apartment_1, apartment_2, frl_apartment_0, frl_apartment_1, room_0, room_1 , room_2, hotel_0, office_0, office_1.
- Dataset2: Collect data from apartment_0 (the same as testing scenario).

You should define how many images you need to collect. In Dataset1, although we list the above 10 scenes, you do not have to collect data from each scene. You can select some in these scenes for data collection.

(Note: The labels of images collected by Habitat are instance ids, and we only need to train the semantic segmentation model. Therefore, we have translated the instance to semantic labels in data_generator.py by using info_semantic.json in Replica dataset files.)

(Note: There are some errors in info_semantic.json provided by Replica dataset, so a few objects in scenes may get wrong labels. They may be colored with black (id=0).)

(2) Follow the instructions in the tutorial page to customize your dataset and create .odgt files for training.

2. Model Training:

(1) Select one model architecture and download the pre-trained checkpoint.

(2) Customize your own configuration file. Make sure the data path and all the parameters are set correctly.

(3) Run **train.py** to train **TWO semantic segmentation models**. One is trained on images collected from other scenes, the other one is trained on images collected from apartment_0.

(Note: If you use **single GPU**, you may encounter the problem like this issue. Please follow the steps to solve the problem.)

(4) Run **eval_miltipro.py** to evaluate the images you collected for HW1 for reconstruction and check the corresponding segmentation performance using the two different metrics, i.e., mIOU and accuracy. Please save and visualize the resulting images.

(Reminder: The calculation of mIOU in eval_multipro.py should be revised because the target of the evaluation dataset does not contain certain categories. Thus, these categories should not be used in the calculation of mIOU. Please check **apartment0_classes.xlsx** to calculate the mIOU of 49 categories in apartment_0. Please see the explanation here.)

To visualize the color image results, we provide the color map for 101 categories. Please use **color101.mat** file for color mapping (see the line 22-39 in eval_miltipro.py).

Task 2. 3D Semantic Map Reconstruction:

In this part, you need to use the codes and data from your hw1, and use the template code **3d_semantic_map.py** to complete the TODO sections.

1. Use the semantic images predicted by your model for coloring.

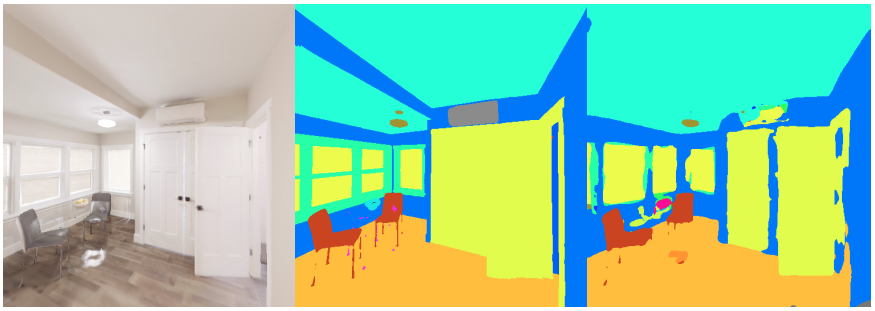
2. Implement the custom_voxel_down() function. If you average all points into one point in each voxel, it will influence the performance of the 3D semantic map if there are two or more labels within a voxel. To address the issue, we need to implement a custom_voxel_down() function to determine a label of each voxel (e.g., the label could be the majority class in a voxel).

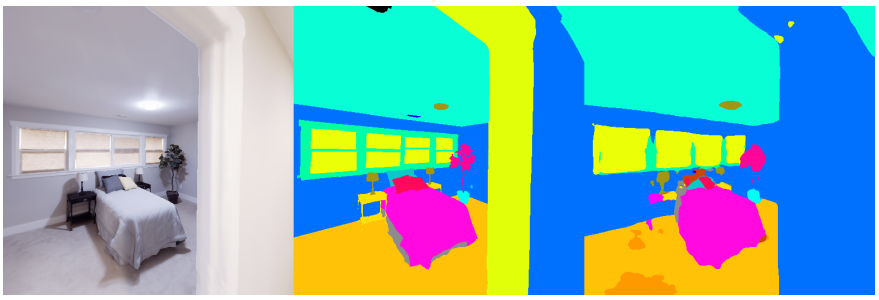
3. Reconstruct 3D semantic maps of the first and second floor with icp algorithm in hw1. (If your own ICP implementation doesn't yield satisfactory results, you can utilize open3d for the reconstruction process.)

Example results

Outputs of Segmentation Model

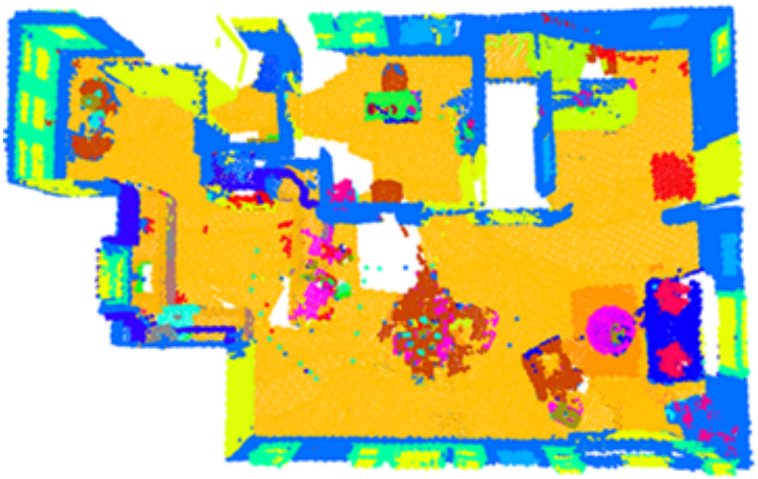
rgb ground truth prediction



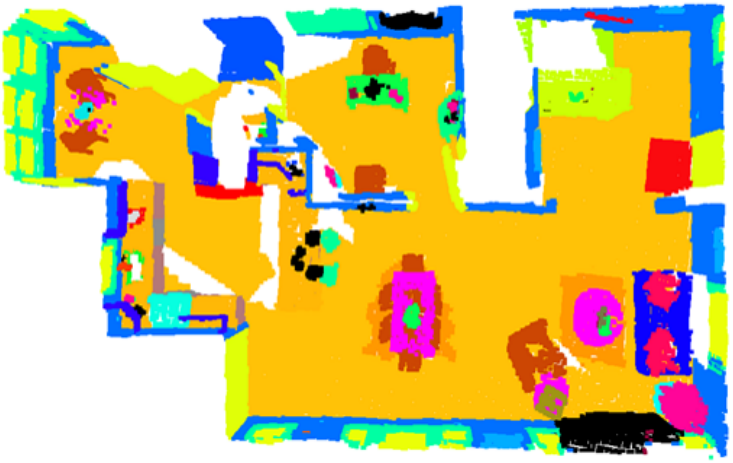


Semantic map (trained on other scenes)

Semantic map (trained on apartment_0)



Semantic map (ground truth)



(objects with black color are wrong outputs after color mapping by info_semantic.json file)

Grading

Online Demo 30%

1. Show the reconstruction result on your computer.
2. Briefly describe the training process and explain voxel down function.

Report 70%

Your report should include the following content:

1. **Implementation (40%)**

   Task1:

   a. **Code**
   Detailed explanation of your implementation. For example, how many images do you collect, which model do you use, and how do you define the training parameters.

   b. **Result and Discussion**
      i. Result of your segmentation model in the **first and second floor**. Visualize the results (like the examples above) and show the mIOU and accuracy.
      ii. Anything you want to discuss
      iii. Any reference you take

   Task2:

a. **Code**
   Detailed explanation of your implementation. For example, how you implement the voxel down function.
b. **Result and Discussion**
   i. Result of your 3D semantic map in the **first and second floor**. Visualize the results (like the examples above) with BEV.
   ii. Anything you want to discuss.
   iii. Any reference you take

2. Questions (30%)

a. From the results, we can clearly see the impact of domain shift. In your opinion, besides collecting training data directly under the same scene, are there any other methods to improve performance?
b. List and explain all the tricks you have tried to speed up the voxelization, and compare the results.

Submission

Due Date: 2023/10/31 23:59

Please directly compress all your code files and report (.pdf) into {STUDENT ID}_hw2.zip and submit it to the New E3 System.

The file structure should look like:

```
{student_id}_hw2.zip
├ 3d_semantic_map.py                # custom voxel down funcion
├ report.pdf
├ README.md
└ ckpt
  └ model_others                    # model trained on other scenes
    ├ encoder_epoch_(epoch num).pth
    ├ decoder_epoch_(epoch num).pth
    └ config.yaml
  └ model_apartment_0               # model trained on apartment_0 scene
    ├ encoder_epoch_(epoch num).pth
    ├ decoder_epoch_(epoch num).pth
    └ config.yaml
```

Wrong submission format leads to -10 point

Late submission leads to -20 points per day