Perception and Decision Making in Intelligent Systems
Homework 2: A Semantic Mapping Framework
Announce: 10/18, Deadline:11/8 23:55

1. Introduction

In the second homework, we will train a semantic segmentation model, using data collected from the scenes (apartment_1, apartment_2, frl_apartment_0, frl_apartment_1, room_0, room_1 , room_2, hotel_0, office_0, office_1) in the **replica dataset**. We will test the model on apartment_0 (which is the same scene used in the HW1). Then, we will label the reconstructed point cloud with semantic labels obtained from your trained model. Finally, a 3D semantic map can be generated.

https://docs.google.com/document/d/1PCaJ2L7kWUCN7w7erHnxOBDoCcsuIic5/edit?usp=sharing&ouid=114222386363369914303&rtpof=true&sd=true

Replica-Dataset

There are three main tasks in HW2, i.e., **Data Collection, finetune a semantic segmentation Model, and reconstruct a 3D semantic map.**

1. Data Collection:

   In the data collection phase, we will collect two sets of training data. A set of data is similar to the distribution of testing data. The set of data is different from the distribution of testing data. The goal is to let you learn the impact of **distribution shift** on the performance of semantic segmentation.

2. Finetune a Semantic Segmentation Model:

   Please check out the link: semantic-segmentation-pytorch, which has many semantic segmentation models pretrained on the ADE20k dataset. It is free to select your model of interest.

3. Reconstruct a 3D Semantic Map:

   With predicted semantic images, the same method in HW1 can be used to generate 3D semantic maps.

2. Requirements

- OS : ubuntu 18.04
- **GPU for finetuning model**
- Python 3.6 (You can use conda to create new environment)
- opencv-contrib-python
- Open3d
- Habitat lab and sim

3. Tasks

Part 1: Data Collection:

Control the agent and collect data in Habitat: **rgb images** and **semantic images** from Replica datasets.

1. Collect from apartment_1, apartment_2, frl_apartment_0, frl_apartment_1, room_0, room_1 , room_2, hotel_0, office_0, office_1.
2. Collect from apartment_0.

See data_generator.py here. You can use this code to collect data.

- Collect **arbitrary number** of images, and the numbers of images collected from apartment_0 and others **should be the same**.
  - collect total of 1000 images form apartment_1, apartment_2, frl_apartment, and so on
  - collect 1000 images on apartment_0

- Although we list the above 10 scenes, you do not have to collect data from each scene. You can select some of these scenes for data collection.

- The labels of images collected by Habitat are instance ids. We have translated instance to semantic labels in data_generator.py by using info_semantic.json in Replica dataset files. Please check line 177, fix_semantic_observation function.

  Reference: label mapping guide link. (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) after color mapping.)

Part 2: Finetune a Semantic Segmentation Model:

1. Choose a semantic segmentation model (semantic-segmentation-pytorch).

2. Fine-tuning. It is up to you to determine the layers you would like to freeze while fine-tuning. Please check the tutorial page to customize your dataset and create .odgt files for training. Note that you will train TWO models, one is trained on images collected from other scenes, the other one is trained on images collected from apartment_0.

3. Please run **eval_miltipro.py** to evaluate images collected from HW1 for reconstruction and check the corresponding segmentation performance using the two different metrics, i.e., mIOU and accuracy. Please save the result images and check.

   (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 apartment_0_categorise to calculate the mIOU of 49 categories in apartment_0. Please see the explanation here.)

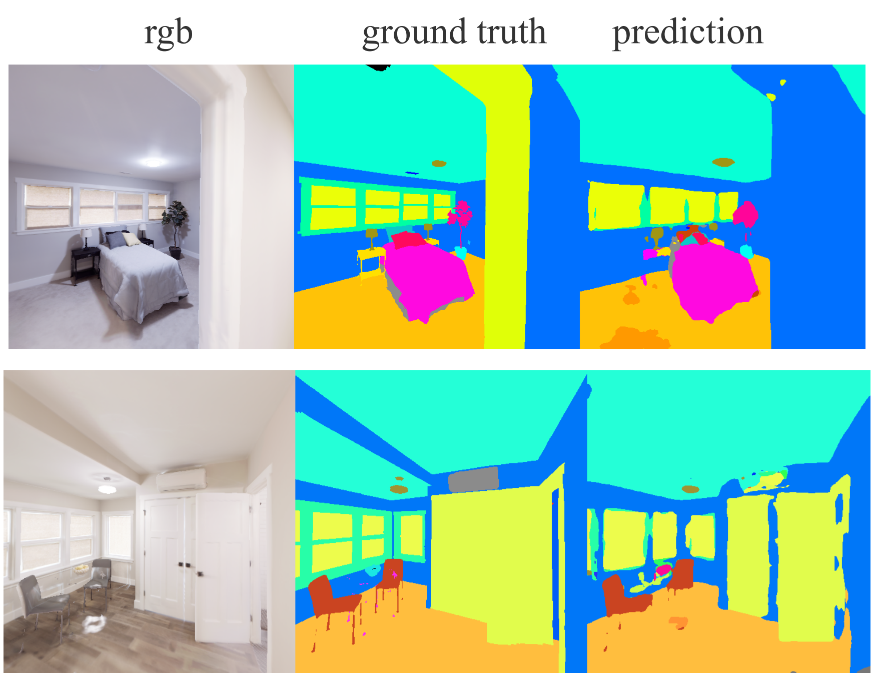
   To visualize the color image results, we provide the color map for 101 categories. Please use color101.mat file for color mapping (see the line 22-39 in eval_miltipro.py).

Part 3: Reconstruct a 3D Semantic Map

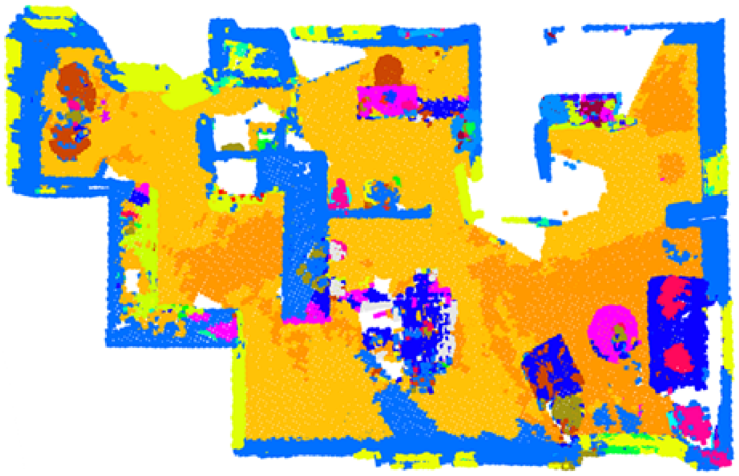
In this part, you should **use the codes and data from your hw1** and **modify the following 2 parts**:

1. Replace the rgb images to semantic images predicted by your model for coloring.
2. Open3d function: **voxel_down_sample()** averages all points into one point in each voxel, it will influence the performance of a 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).
   Finally, a 3D semantic map can be generated!
3. Visualize your results in BEV

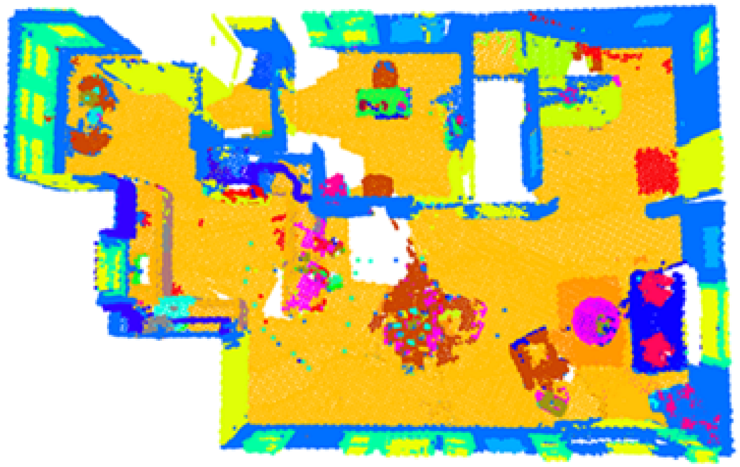
4. Example results



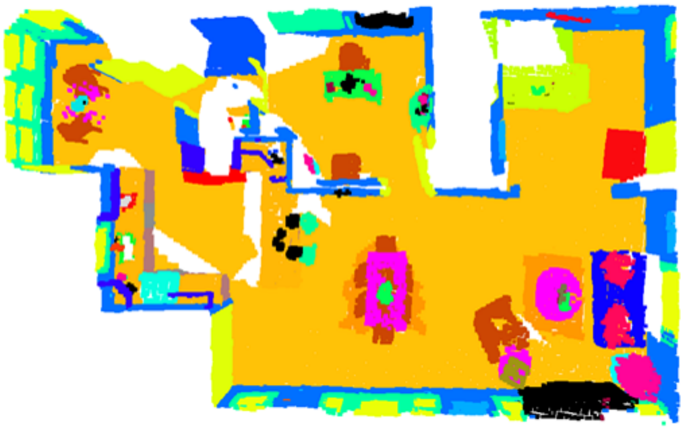
Semantic map (trained on other scenes)



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)

5. Submission

1. **File Formate:**

```
studentID_Name_hw2
                    | --- README
                    | --- report.pdf
                    | --- ckpt
                              | — model_others
                                        | --- encoder_epoch_(epoch num).pth
                                        | --- decoder_epoch_(epoch num).pth
                                        | --- config.yaml
                              | — model_apartment_0
                                        | --- encoder_epoch_(epoch num).pth
                                        | --- decoder_epoch_(epoch num).pth
                                        | --- config.yaml
                    | --- 3d_semantic_map.py (custom voxel down funcion)
```

   README: Please describe how to run your program.
   Please put all the files into folder **studentID_name_hw2** and compress your folder into **one "zip" file**. Submit it to New E3 System with the format **studentID_name_hw2.zip.**

2. The **deadline** for this homework is **11/8 23:55.**
3. Late submission leads to **-20 points per day**.
4. **Wrong submission format leads to -10 point**

6. Grading

1. Online Questions (describe the training process and explain voxel down function) 20%
2. Online Demo (reconstruct semantic map) 50%
3. Report of you implementation 30%
   a. Code
      i. Detailed explanation of your implementation.
         1. Training segmentation model:
            The model you used, the content of the dataset (number of images), training setting (parameters in .yaml file), training result (final step accuracy and loss, or just paste the screenshot), validation

result (mIOU and accuracy), and other relevant information.

2. 3D semantic map reconstruction: Describe how to run your program, and how you implement and accelerate the process of custom voxel down.

b. Result and Discussion
    i. Result of your reconstruction(Floor 1 and Floor2, Both open3d implementation and your own implementation)
    ii. Mean L2 distance between ground truth and estimated trajectory.
    iii. Anything you want to discuss, such as comparing the performance of two implementations.
    iv. Any reference you take