Report

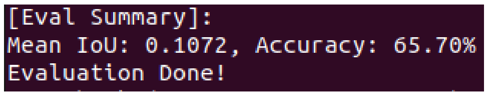
1. The content of my dataset：

Floor 1 = 115 photos

Floor 2 = 64 photos

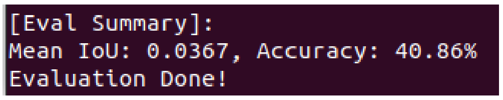
The photo below that show my mIOU in four different way：

a. Foor1 in apartment0 model



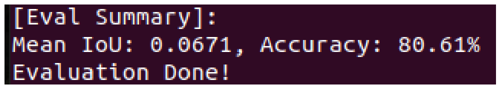
```
[Eval Summary]:
Mean IoU: 0.1072, Accuracy: 65.70%
Evaluation Done!
```

b. Foor1 in others model



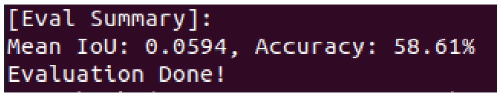
```
[Eval Summary]:
Mean IoU: 0.0367, Accuracy: 40.86%
Evaluation Done!
```

c. Floor2 in apartment0 model



```
[Eval Summary]:
Mean IoU: 0.0671, Accuracy: 80.61%
Evaluation Done!
```

d. Floor2 in others model



```
[Eval Summary]:
Mean IoU: 0.0594, Accuracy: 58.61%
Evaluation Done!
```

The accuracy and loss that I forgeted to take a screenshot when finished training, so I lost this information.😥

2. The way that how to run my program can see in the README.

- Add the code below in the load.py to save the image of the annotations

```
# 對語意分割進行截圖（相比HW1所增加的）
instance_id_to_semantic_label_id = np.array(scene_dict["id_to_label"])
semantic = instance_id_to_semantic_label_id[a[3]]
semantic_img = Image.new("L", (semantic.shape[1], semantic.shape[0]))
semantic_img.putdata(semantic.flatten())
semantic_img.save(os.path.join("./my_data/annotations"+str(input_floor)+"/apartment"+str(photo_of_number_change)+".png"))
```

my_trans.py to change the path to .odgt

- Use the code that appy in the hackmd and finetune them to get the custom.odgt

```
# my_trans.py
import os
import cv2
import json

def odgt(img_path):
    seg_path = img_path.replace('images1', 'annotations1')
    seg_path = seg_path.replace('.jpg', '.png')   # image 要jpg, annotations 要png

    if os.path.exists(seg_path):
        img = cv2.imread(img_path)
        h, w, _ = img.shape

        odgt_dic = {}
        odgt_dic["fpath_img"] = img_path
        odgt_dic["fpath_segm"] = seg_path
        odgt_dic["width"] = h
        odgt_dic["height"] = w
        return odgt_dic
    else:
        return None


save = 'data_for_reconstruct1.odgt'
dir_path = f"./my_data/images1"
img_list = os.listdir(dir_path)
img_list.sort()
print(img_list)
img_list = [os.path.join(dir_path, img) for img in img_list]

with open(f'./my_data/{save}', mode='wt', encoding='utf-8') as myodgt:
    for i, img in enumerate(img_list):
        a_odgt = odgt(img)
        if a_odgt is not None:
            myodgt.write(f'{json.dumps(a_odgt)}\n')
```

- Use the data that the data_generator generate to train the model. Put my .odgt path in the custom .config file and then use train.py to train the model that we set in the config. The model that I use is the encoder of resnet50dilated and the decoder of ppm_deepsup, the code below is my apartment.yaml

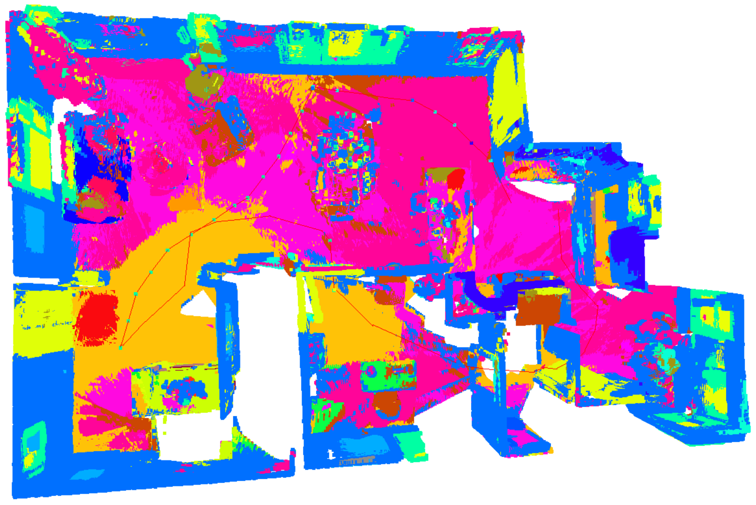
```
# custom .config
DATASET:
  root_dataset: ""
  list_train: "./data/training.odgt"    # generate by trans.py
  list_val: "./data/validation.odgt"
  num_class: 101
  imgSizes: (300, 375, 450, 525, 600)
  imgMaxSize: 1000
  padding_constant: 8
  segm_downsampling_rate: 8
  random_flip: True

MODEL:
  arch_encoder: "resnet50dilated"
  arch_decoder: "ppm_deepsup"
  fc_dim: 2048

TRAIN:
  batch_size_per_gpu: 2
  num_epoch: 20
  start_epoch: 0
  epoch_iters: 650
  optim: "SGD"
  lr_encoder: 0.02
  lr_decoder: 0.02
  lr_pow: 0.9
  beta1: 0.9
  weight_decay: 1e-4
  deep_sup_scale: 0.4
  fix_bn: False
  workers: 16
  disp_iter: 20
  seed: 304

VAL:
  visualize: False
  checkpoint: "epoch_20.pth"

TEST:
  checkpoint: "epoch_20.pth"
  result: "./"

DIR: "ckpt/apartment0"
```

mIOU in eval_multipro.py

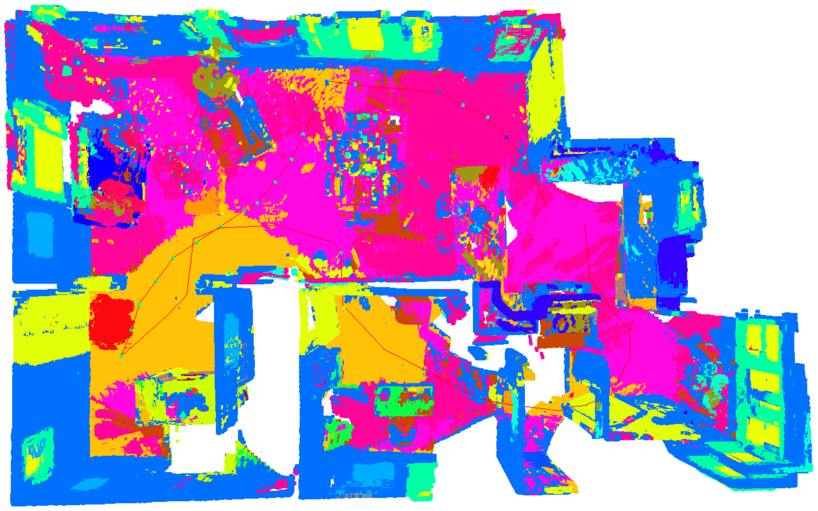
```
new_iou = []
wb = openpyxl.load_workbook('apartment0_classes.xlsx', data_only=True)
s1 = wb['Sheet1']
for i in range(2,51):
    label = s1.cell(i,1).value
    print("add  " + str(int(label)) + "  " + str(iou[int(label)]))
    new_iou.append(iou[int(label)])
wb.save('apartment0_classes.xlsx')

print('[Eval Summary]:')
print('Mean IoU: {:.4f}, Accuracy: {:.2f}%'
      .format(sum(new_iou)/len(new_iou), acc_meter.average()*100))
```

The pictures below is my result of the reconstruction.

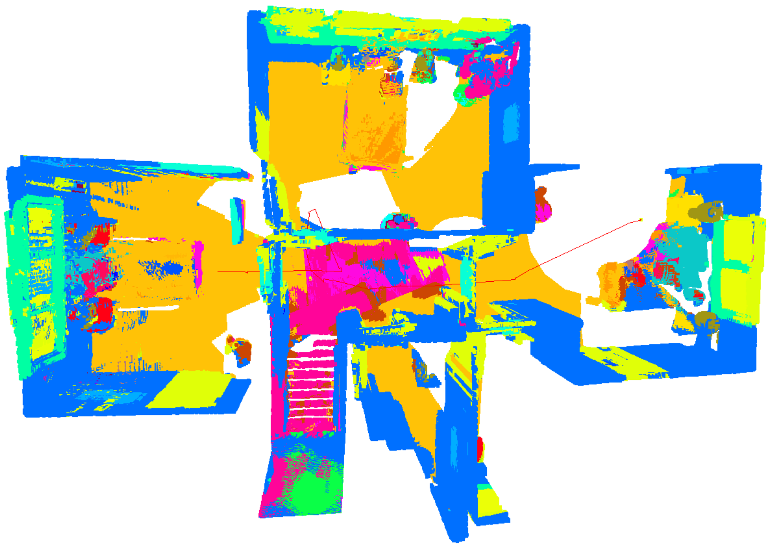
a. Foor1 in apartment0 model



b. Foor1 in others model



c. Floor2 in apartment0 model



d. Floor2 in others model