Pull Requests

Fu Lian, M1
Prof. Oishi’s Lab

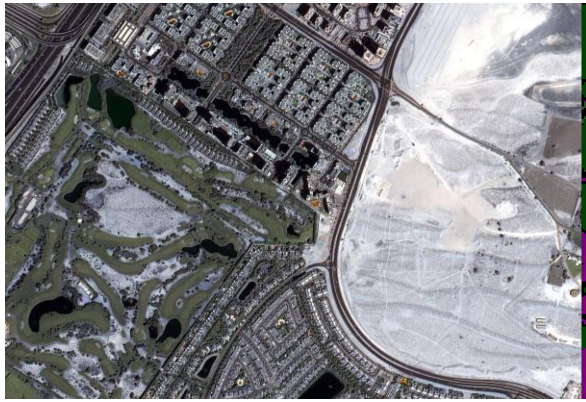
2022/07/06

Created PRs List

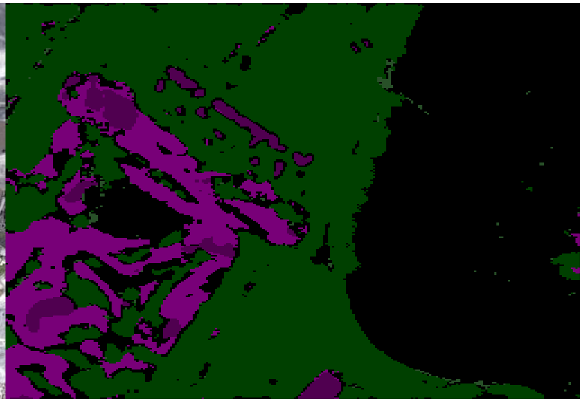
- 2022-U-Net: Add support for MBRSC dataset. Train model on this dataset.
- 2022-Flooding: Add support for tensorboard
- 2022-BackgroundMatting: Use Hydra for configuration management.

2022-U-Net

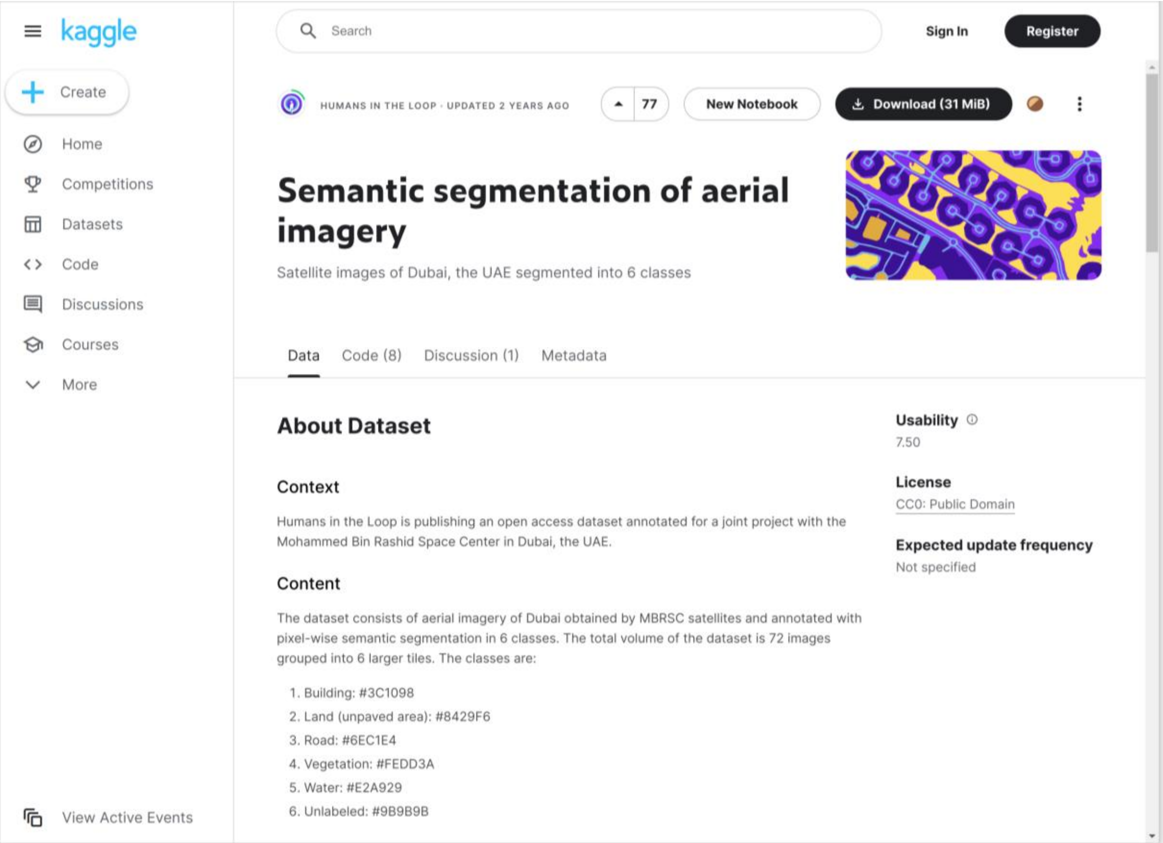
- This repo is about semantic segementation using U-Net
- Problem: Only test on KITTI dataset.
- **My PRs:**
    - **Add support for MBRSC dataset.**
        - **Data collection related code**
            - Dataset structure is different
        - **Mask image translation related code**
            - Mask image format is different
    - **Train & Test model on MBRSC dataset.**
        - Train for 50 epoch, gain test accuracy of 0.7654



Input RGB image



Output Segmentation Mask



Semantic segmentation of aerial imagery

Satellite images of Dubai, the UAE segmented into 6 classes

About Dataset

Context

Humans in the Loop is publishing an open access dataset annotated for a joint project with the Mohammed Bin Rashid Space Center in Dubai, the UAE.

Content

The dataset consists of aerial imagery of Dubai obtained by MBRSC satellites and annotated with pixel-wise semantic segmentation in 6 classes. The total volume of the dataset is 72 images grouped into 6 larger tiles. The classes are:

1. Building: #3C1098
2. Land (unpaved area): #8429F6
3. Road: #6EC1E4
4. Vegetation: #FEDD3A
5. Water: #E2A929
6. Unlabeled: #9B9B9B

Usability
7.50

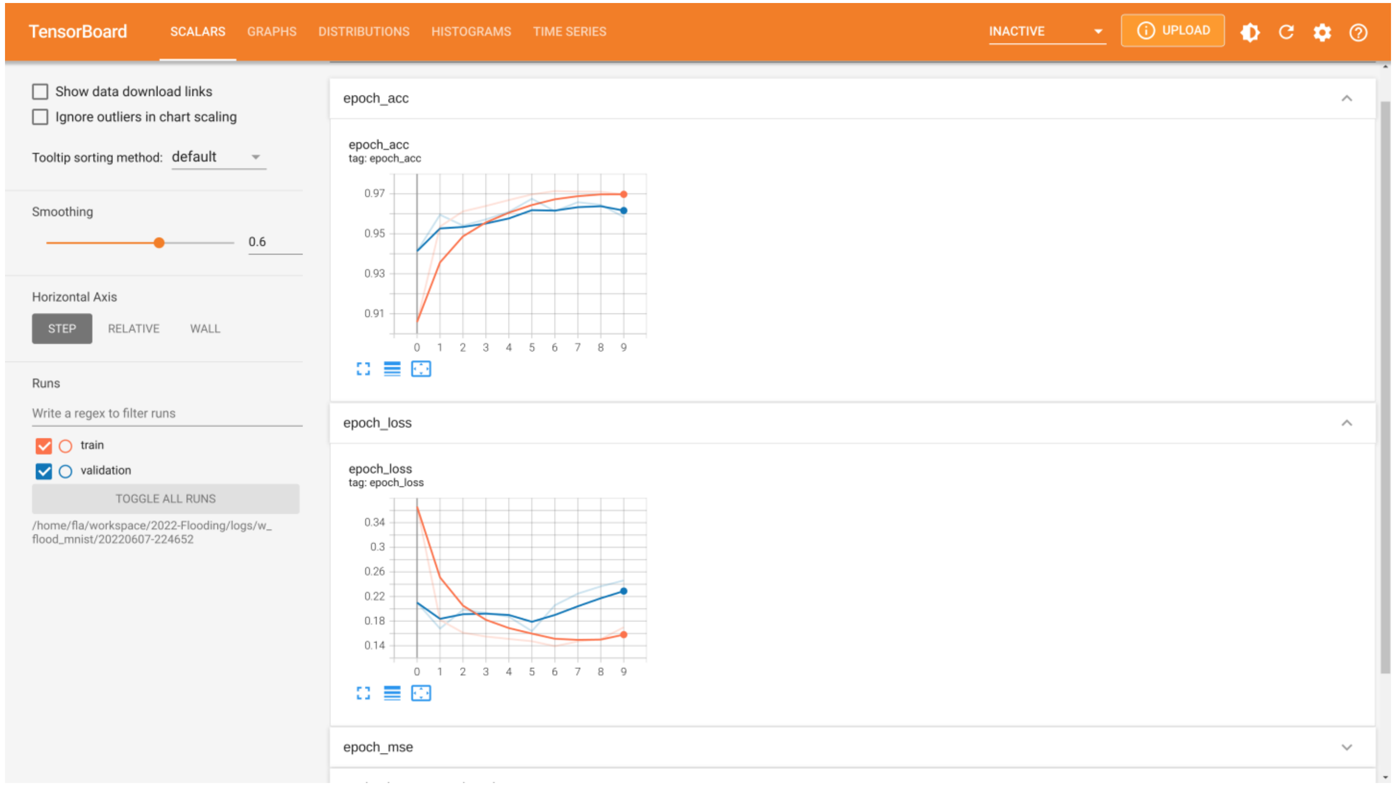
License
CC0: Public Domain

Expected update frequency
Not specified

MBRSC Dataset from Kaggle

2022-Flooding

- This repo is about using 'flooding' to improve performance of ANN.
- Problem: Training without logger.
- **My PRs:**
  - **Add support for Tensorboard.**



□ Show data download links
□ Ignore outliers in chart scaling

Tooltip sorting method: default

Smoothing 0.6

Horizontal Axis STEP RELATIVE WALL

Runs

Write a regex to filter runs

☑ ○ train
☑ ○ validation

TOGGLE ALL RUNS

/home/fla/workspace/2022-Flooding/logs/w_flood_mnist/20220607-224652

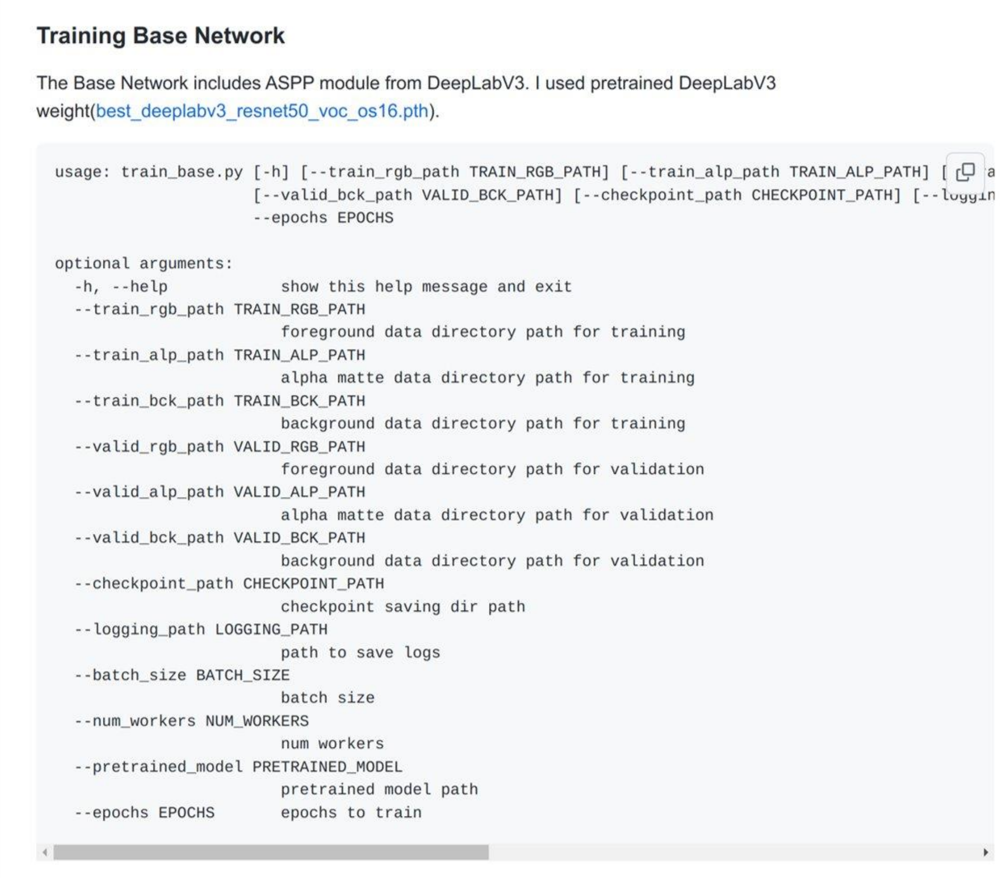
Tensorboard

2022-BackgroundMatting

- This repo is about background matting, reimplemented using Pytorch.
- Problem: Multi-Model & Multi-experiment involved, complex parameter configuration.
- **My PRs:**
  - **Using Hydra to handle configuration management.**
  - **Modify Action codes & README.md correspondingly**

2022-BackgroundMatting

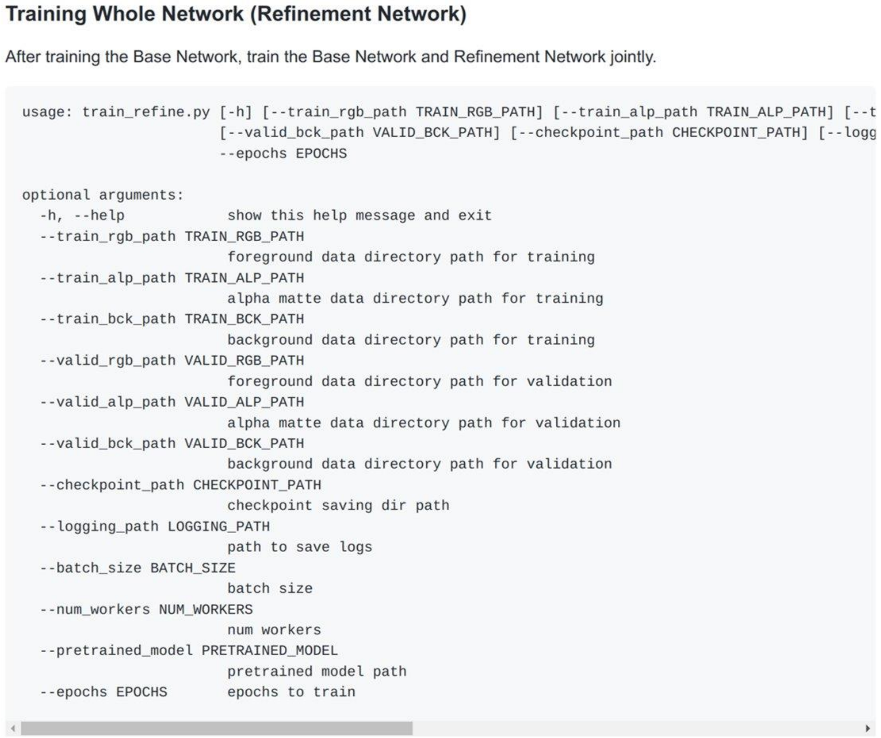
Orginal: Using Command Line for configuration of each experiment



Training Base Network

The Base Network includes ASPP module from DeepLabV3. I used pretrained DeepLabV3 weight(best_deeplabv3_resnet50_voc_os16.pth).

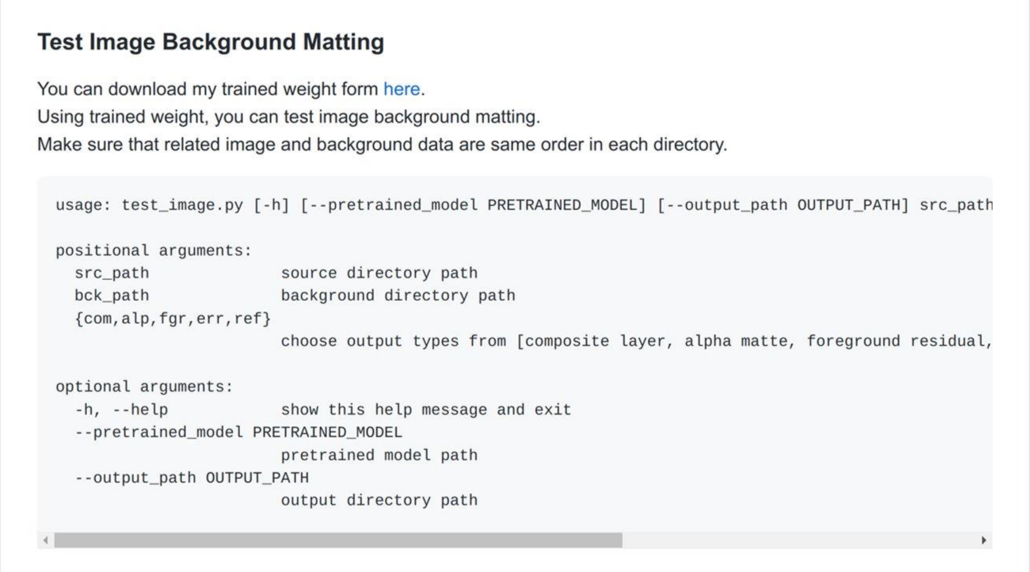
```
usage: train_base.py [-h] [--train_rgb_path TRAIN_RGB_PATH] [--train_alp_path TRAIN_ALP_PATH] [   a
                     [--valid_bck_path VALID_BCK_PATH] [--checkpoint_path CHECKPOINT_PATH] [--loggin
                     --epochs EPOCHS

optional arguments:
  -h, --help            show this help message and exit
  --train_rgb_path TRAIN_RGB_PATH
                        foreground data directory path for training
  --train_alp_path TRAIN_ALP_PATH
                        alpha matte data directory path for training
  --train_bck_path TRAIN_BCK_PATH
                        background data directory path for training
  --valid_rgb_path VALID_RGB_PATH
                        foreground data directory path for validation
  --valid_alp_path VALID_ALP_PATH
                        alpha matte data directory path for validation
  --valid_bck_path VALID_BCK_PATH
                        background data directory path for validation
  --checkpoint_path CHECKPOINT_PATH
                        checkpoint saving dir path
  --logging_path LOGGING_PATH
                        path to save logs
  --batch_size BATCH_SIZE
                        batch size
  --num_workers NUM_WORKERS
                        num workers
  --pretrained_model PRETRAINED_MODEL
                        pretrained model path
  --epochs EPOCHS       epochs to train
```



Training Whole Network (Refinement Network)

After training the Base Network, train the Base Network and Refinement Network jointly.

```
usage: train_refine.py [-h] [--train_rgb_path TRAIN_RGB_PATH] [--train_alp_path TRAIN_ALP_PATH] [--t
                       [--valid_bck_path VALID_BCK_PATH] [--checkpoint_path CHECKPOINT_PATH] [--logg
                       --epochs EPOCHS

optional arguments:
  -h, --help            show this help message and exit
  --train_rgb_path TRAIN_RGB_PATH
                        foreground data directory path for training
  --train_alp_path TRAIN_ALP_PATH
                        alpha matte data directory path for training
  --train_bck_path TRAIN_BCK_PATH
                        background data directory path for training
  --valid_rgb_path VALID_RGB_PATH
                        foreground data directory path for validation
  --valid_alp_path VALID_ALP_PATH
                        alpha matte data directory path for validation
  --valid_bck_path VALID_BCK_PATH
                        background data directory path for validation
  --checkpoint_path CHECKPOINT_PATH
                        checkpoint saving dir path
  --logging_path LOGGING_PATH
                        path to save logs
  --batch_size BATCH_SIZE
                        batch size
  --num_workers NUM_WORKERS
                        num workers
  --pretrained_model PRETRAINED_MODEL
                        pretrained model path
  --epochs EPOCHS       epochs to train
```



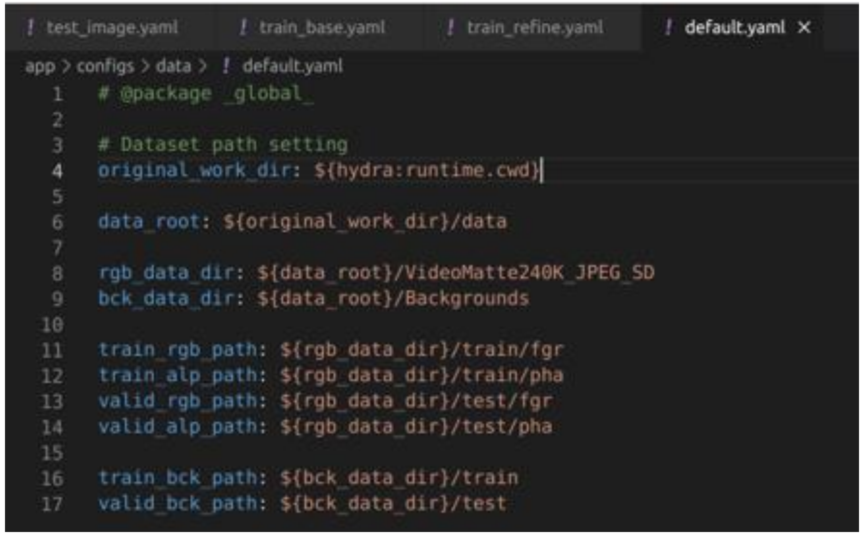
Test Image Background Matting

You can download my trained weight form here.
Using trained weight, you can test image background matting.
Make sure that related image and background data are same order in each directory.

```
usage: test_image.py [-h] [--pretrained_model PRETRAINED_MODEL] [--output_path OUTPUT_PATH] src_path

positional arguments:
  src_path              source directory path
  bck_path              background directory path
  {com,alp,fgr,err,ref}
                        choose output types from [composite layer, alpha matte, foreground residual,

optional arguments:
  -h, --help            show this help message and exit
  --pretrained_model PRETRAINED_MODEL
                        pretrained model path
  --output_path OUTPUT_PATH
                        output directory path
```

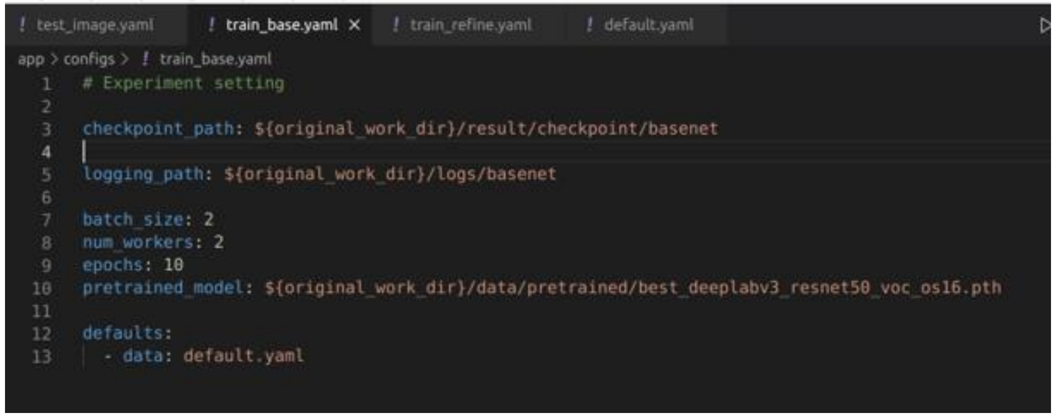
2022-BackgroundMatting

Now: Using configuration files.



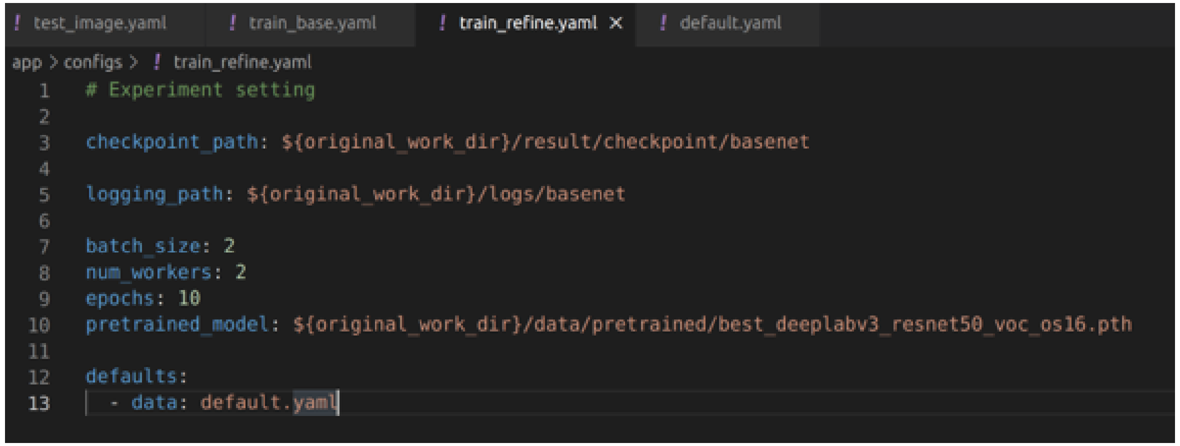
```yaml
# @package _global_

# Dataset path setting
original_work_dir: ${hydra:runtime.cwd}

data_root: ${original_work_dir}/data

rgb_data_dir: ${data_root}/VideoMatte240K_JPEG_SD
bck_data_dir: ${data_root}/Backgrounds

train_rgb_path: ${rgb_data_dir}/train/fgr
train_alp_path: ${rgb_data_dir}/train/pha
valid_rgb_path: ${rgb_data_dir}/test/fgr
valid_alp_path: ${rgb_data_dir}/test/pha

train_bck_path: ${bck_data_dir}/train
valid_bck_path: ${bck_data_dir}/test
```

Dataset path setting



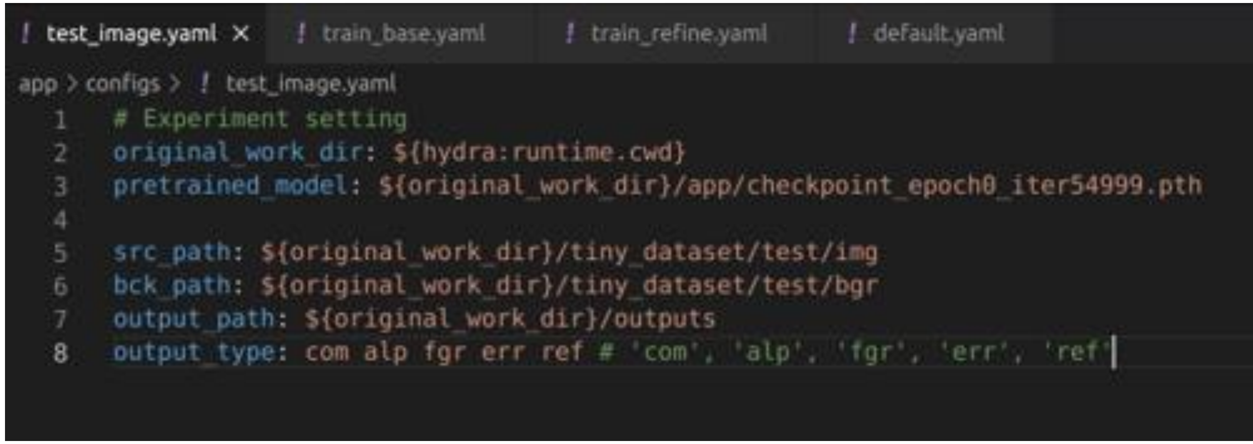
```yaml
# Experiment setting

checkpoint_path: ${original_work_dir}/result/checkpoint/basenet

logging_path: ${original_work_dir}/logs/basenet

batch_size: 2
num_workers: 2
epochs: 10
pretrained_model: ${original_work_dir}/data/pretrained/best_deeplabv3_resnet50_voc_os16.pth

defaults:
  - data: default.yaml
```

Experiment: Base-Model Training



```yaml
# Experiment setting

checkpoint_path: ${original_work_dir}/result/checkpoint/basenet

logging_path: ${original_work_dir}/logs/basenet

batch_size: 2
num_workers: 2
epochs: 10
pretrained_model: ${original_work_dir}/data/pretrained/best_deeplabv3_resnet50_voc_os16.pth

defaults:
  - data: default.yaml
```

Experiment: Refine-Model Training



```yaml
# Experiment setting
original_work_dir: ${hydra:runtime.cwd}
pretrained_model: ${original_work_dir}/app/checkpoint_epoch0_iter54999.pth

src_path: ${original_work_dir}/tiny_dataset/test/img
bck_path: ${original_work_dir}/tiny_dataset/test/bgr
output_path: ${original_work_dir}/outputs
output_type: com alp fgr err ref # 'com', 'alp', 'fgr', 'err', 'ref'
```

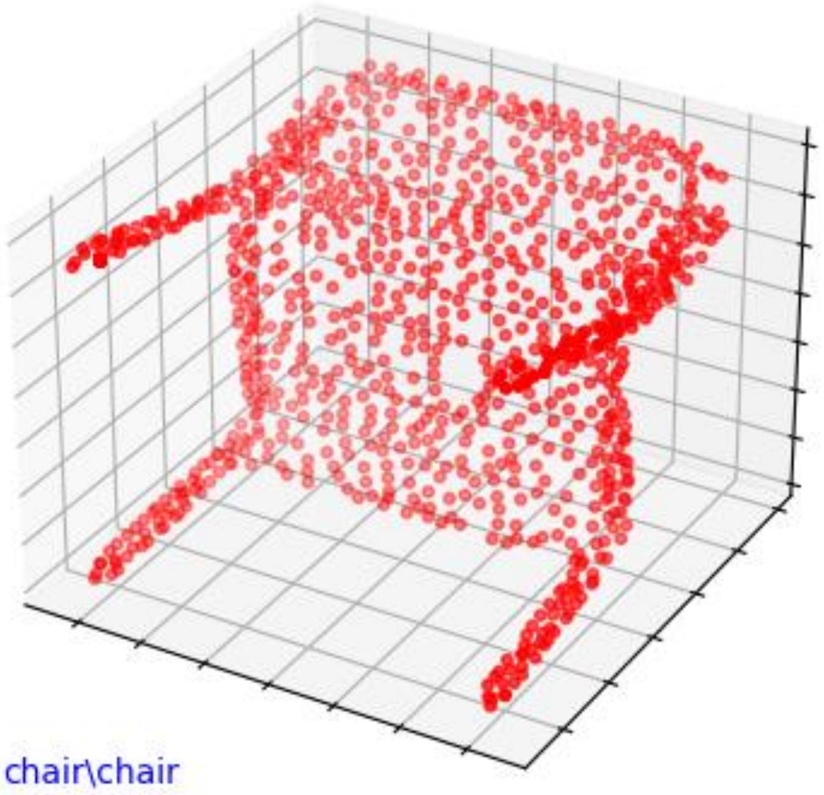
Experiment: Model Testing

Reviewed PR

- 2022-PCT-Lightning: Alternative Visualization from @Yoharol

2022-PCT-Lightning

- This repo is about Point Cloud Classfication
- Problem: Visualization result is not so clear. Using C++ to render, need compiling.
- **Recieved PR:**
  - **Using Matplotlib to visualize the result.**


chair\chair