

Semi-Supervised Semantic Segmentation

Data Science Homework 6

阮柏愷 bkruan.ee11@nycu.edu.tw

Outline

- [Introduction](#)
- [Dataset](#)
- [Evaluation](#)
- [Baseline Code](#)
- [Submission Format](#)
- [Grading Policy](#)
- [Rules and Information](#)

Introduction - Terminology

- Only having a portion of labels



Semi-Supervised Semantic Segmentation

- Assigning pixel to each category

data



x10

label



x1

Introduction

- Training a **semantic segmentation** model by **semi-supervised learning** on:
 - 300 **labeled** images
 - 700 **unlabeled** images

Introduction

- Semi-Supervised Learning
 - [Learning Resource 1](#)
 - [Learning Resource 2](#)
 - [Recommended Paper](#)

Dataset - Introduction

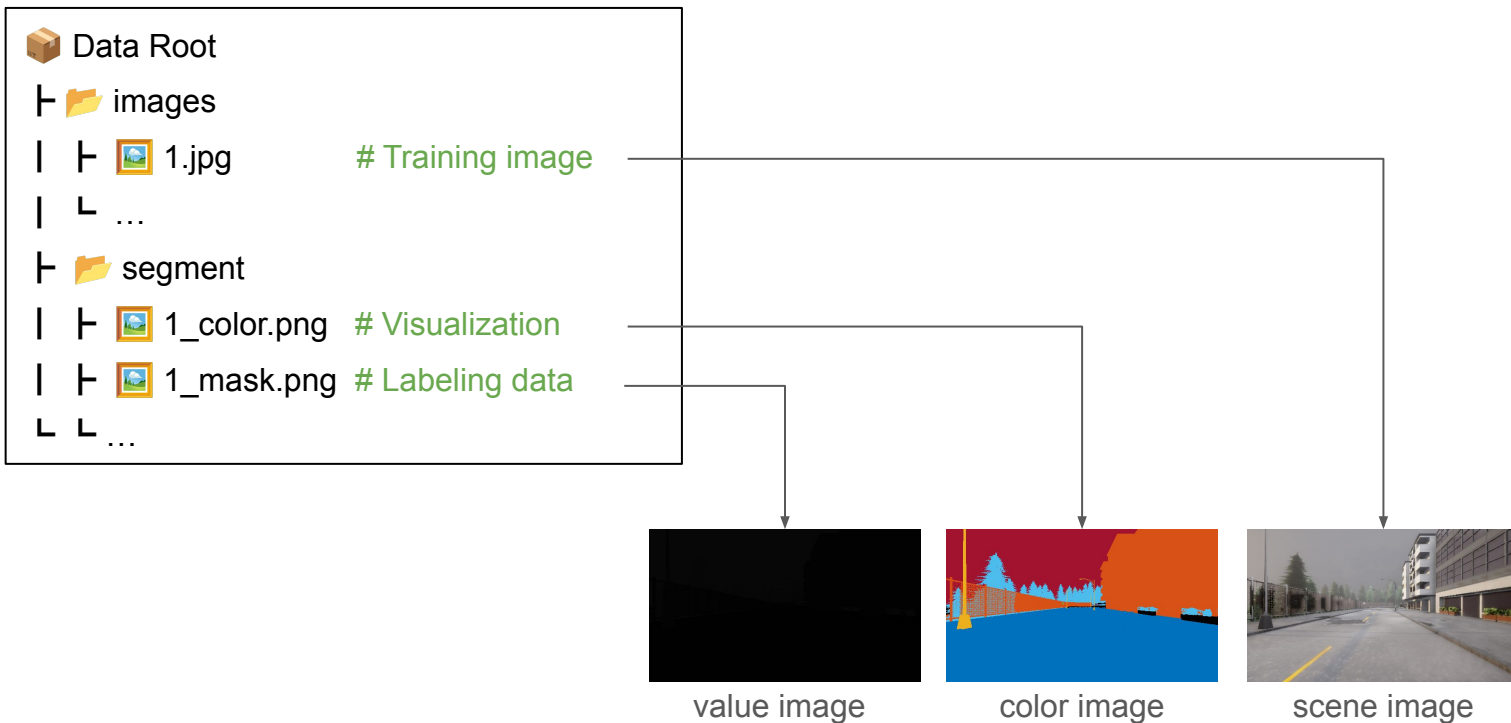


- The dataset is collected on a simulated driving environment.
- The dataset is partitioned into:
 - 300 **labeled** images
 - 700 **unlabeled** images (including 100 **test** images)
- 9 classes are included:

Value	0	1	2	3	4
Name	none	road	building	pole	traffic light

Value	5	6	7	8	9
Name	traffic sign	vegetation	sky	person	car

Dataset - Folder Structure



Evaluation

- Image's ID for evaluation: 901~1000
- We evaluate the mask with mIoU, which is computed by:

$$\text{mIoU} = \frac{1}{C} \sum_{i=1}^C \frac{\text{TP}_i}{\text{TP}_i + \text{FP}_i + \text{FN}_i},$$

where,

C : is the total number of classes.

TP_i : True Positives for class i , which is the number of correctly predicted pixels of class i .

FP_i : False Positives for class i , which is the number of pixels incorrectly predicted as class i .

FN_i : False Negatives for class i , which is the number of actual class i that were not predicted as class i

Baseline Code



- Setup

```
git clone https://github.com/Justin900429/semi-supervised-segmentation.git  
pip install -r requirements.txt  
gdown 1seiemd2silpWHIfbRDVaEhOfkh2rk2G6
```

- Training

```
# single-gpu  
python train.py --config configs/default.yaml  
  
# multi_gpu  
accelerate launch --multi_gpu --num_processes=2 train.py --config configs/default.yaml
```

Baseline Code



- Prediction

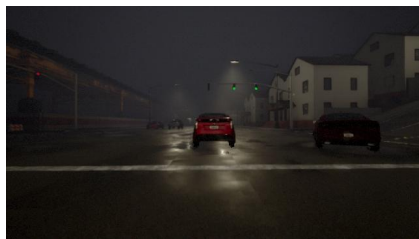
```
# {PLACE_TO_SAVE} is the place for saving the prediction results  
# {PATH_TO_CHECKPOINT} is the saved model weights  
python train.py --test --config configs/default.yaml \  
    --save-path {PLACE_TO_SAVE} \  
    --opts MODEL.CHECKPOINT {PATH_TO_CHECKPOINT}
```

- Create submission

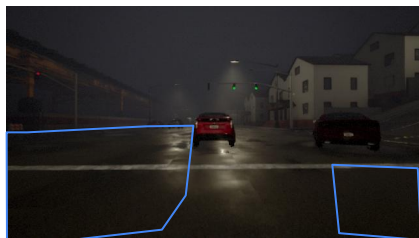
```
# {PLACE_TO_PREDICTION} is the same as {PLACE_TO_SAVE} above  
# {SAVED_FILE_NAME} is the csv name for submission  
python create_submission.py --pred {PLACE_TO_PREDICTION} --save-file {SAVED_FILE_NAME}
```

How about using SAM?

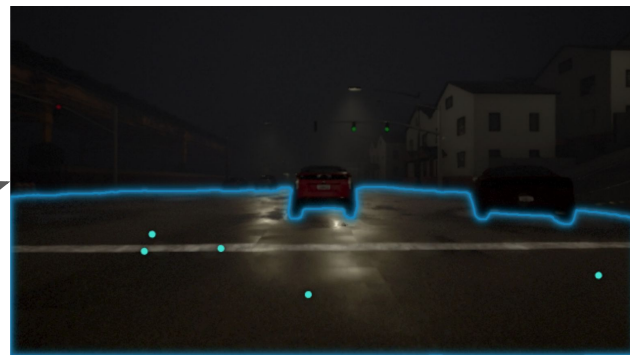
- [SemiSAM: Exploring SAM for Enhancing Semi-Supervised Medical Image Segmentation with Extremely Limited Annotations](#)



↓ prediction



Mask refinement
with points



Submission Format - Kaggle

- Submit the **csv** file to the Kaggle
 - <https://www.kaggle.com/t/54f6ced56b5347c68858ce6ce84a7da3>
- The output of each prediction should be in size of **640x360**
- The prediction should be flattened into a csv row:

```
img_id,label  
901,0 1 1 2 ...  
902,2 7 7 9 ...  
903,5 1 1 7 ...  
...
```

Submission Format - New E3

- Submit your zipped source code {student_id}.zip to E3. The zip file should contain a folder:

```
📁 {student_id}
├── 📁 / 📁 other files
├── 📄 requirements.txt # Dependices
├── 📄 train.sh         # Script to train the model
├── 📄 test.sh          # Script to generate the prediction
└── 📄 model.pth        # Weight for your model
```

Grading Policy

- Top 10%: 100 points
- Top 25%: 90 points
- Top 50%: 80 points
- Top 75%: 75 points
- Over the baseline: 70 points
- Bellow the baseline (shown in leaderboard): 0 point
- Public 50%, Private 50%

Rules and Information

- Use your **student ID** as the team name on Kaggle.
- A maximum of **5** submissions per day is allowed on Kaggle.
- Do not use additional accounts to get more submission quota.
- Do not plagiarize. Write your own codes. (Discuss is encouraged)
- You can **only** use the dataset provided in this competition to train your model.
- You can use any model architecture **but pretrained weight is not allowed**.
(Only SAM is allowed to use the pretrained weight)
- **Firm Deadline: 2024/06/22, 23:59**
- Please email to TA for scheduling an appointment if you need.