# Instruction

**Lin, Jing-Siang**

**2020.10.26**

## 1. Introduction

This instruction is try helping the user to easily execute the code of my thesis - Prior guiding based multiple organ segmentation (基於先驗知識引導之多器官切割) which focus on using the prior based on clinical knowledge to get a more robust and accurate organ segmentation (See Fig. 1). Our work report an average Dice score of 86.00% on the validation set of MICCAI2015 challenge “Multi- Atlas Labeling Beyond the Cranial Vault”.


Figure 1. The multi-organ segmentation results in axial view.

## 2. Prerequisites

Numpy=1.16.3
Opencv-python=3.1.0
Matplotlib=2.2.3
Nibabel=3.0.1
Simple ITK=1.2.4
Tensorflow=1.14.0
Scipy=1.4.1
Sklearn

## 3. Environment

The code is developed using gcc7.5.0, python 3.6+, cuda10.0+ on Ubuntu 16.04. NVIDIA GPUs are needed. The code is tested using 1 x NVIDIA 1080ti GPUS cards. All the experiments are tested on Tensorflow 1.14.0.

## 4. Modules

The file structure is illustrated in Fig. 2.

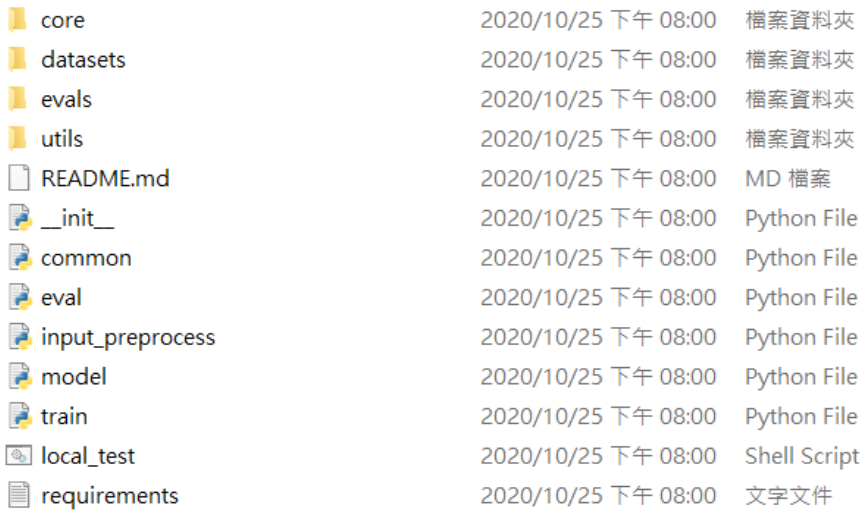


Figure 2. The file structure.

- evals

  Evaluation tools and metrics.

- core

  All the submodules and dependencies for model.py.

- datasets

  Building tfrecord from raw data.

  Building prior from raw data. (See Fig. 3)

  Building data sample for training and evaluation.

- utils

  Training tools and losses.


Figure 3. CHAOS Challenge MR sample prior built from training label.

## 5. Dataset

Two multi-organ segmentation datasets are used in this work, including **MICCAI2015 challenge "Multi- Atlas Labeling Beyond the Cranial Vault"** and **Combined (CT-MR) Healthy Abdominal Organ Segmentation (CHAOS challenge)** (See Fig. 4). Please check the references for more detail.

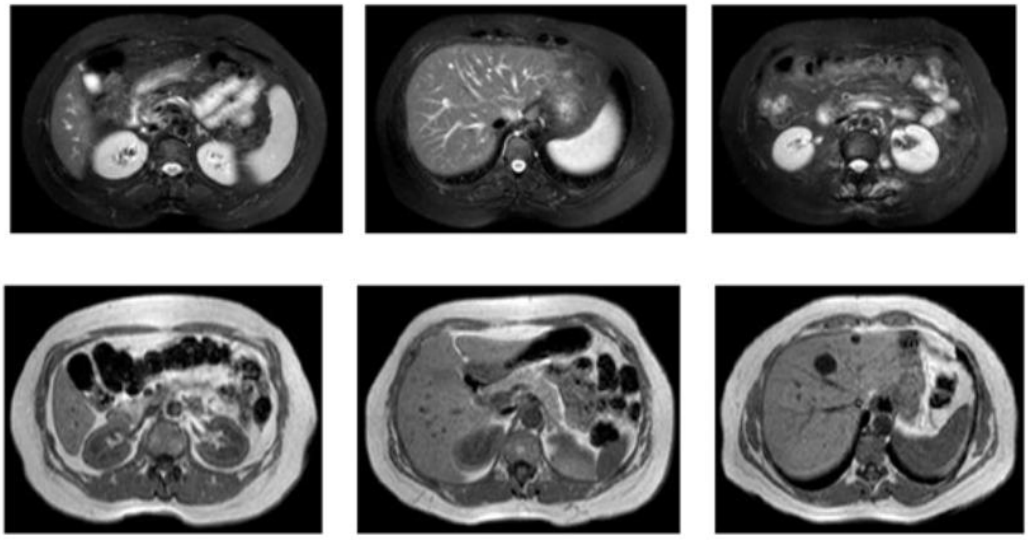

Figure 4. CHAOS challenge dataset sample

## 6. How to build up environment

**0. Create conda environment to manage python libraries**

To check existing conda environment

```
conda env list
```

To create conda environment

```
conda create –n "environment name" python=3.6
```

To activate conda environment

conda activate ***"environment name"***

To deactivate conda environment

conda deactivate

1. **Access the code**

   unzip ***"project name"***

   cd ***"project name"***
2. **Install required libraries**

   pip install -r requirements.txt
3. **Set up path**

   common.py
   - LOGGGING_PATH
   - BASE_DATA_DIR

   build_btcv_data.sh, build_chaos_data.sh
   - WORK_DIR
4. **Set up raw data**

   Upload your data according to the dataset directory structure
5. **Convert raw data to tfrecord and prior**

   sh build datasets/build_btcv_data.sh

   sh build datasets/build_chaos_data.sh
6. **Start training**

   sh local_test.sh

## 7. How to run the code?

You can simply run python train.py and set the parameters that you need.

python train.py --dataset_name 2019_ISBI_CHAOS_MR_T1 --batch_size=16

And if you need to run the training file multiple times. You can set up all the parameters you need in one single shell script, e.g., local_test.sh, and run the command:

sh local_test.sh

```
# DATASET_NAME = ['2013_MICCAI_Abdominal']
# DATASET_NAME = ['2019_ISBI_CHAOS_MR_T1', '2019_ISBI_CHAOS_MR_T2']
# DATASET_NAME = ['2019_ISBI_CHAOS_CT']
gpu_ids=1


CUDA_VISIBLE_DEVICES=$gpu_ids python train.py \
    --dataset_name 2015_MICCAI_Abdominal \
    --batch_size=4 \
    --seq_length=3 \
    --train_split train \
    --guid_fuse mean \
    --seg_loss_name softmax_dice_loss \
    --guid_loss_name sigmoid_cross_entropy \
    --stage_pred_loss_name sigmoid_cross_entropy \
    --validation_steps=150 \
    --training_number_of_steps=200000 \
    --save_checkpoint_steps=150 \
    --guid_encoder=image_only \
    --prior_num_subject 24 \
    --fusions guid_uni guid_uni guid_uni guid_uni guid_uni \
    --weight_decay=0.001 \
    --out_node=32 \
    --guid_conv_nums=2 \
```

Figure 5. Shell script example for training

## 8. License

Apache License 2.0 [5]

## 9. References


[1] **Deeplab:** https://github.com/tensorflow/models/tree/master/research/deeplab

[2] **MICCAI2015 challenge:** https://www.synapse.org/#!Synapse:syn3193805/wiki/217789

[3] **CHAOS challenge:** https://chaos.grand-challenge.org/Combined_Healthy_Abdominal_Organ_Segmentation/

[4] P. Hu, G. Wang, X. Kong, J. Kuen, and Y.-P. Tan. Motion-guided cascaded refinement network for video object segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 1400–1409, 2018.

[5] Apache License: http://www.apache.org/licenses/LICENSE-2.0