The Auto-segmentation for Human Left Ventricles

Author: Pei-Yan Li (Alvin)
Contact Email: d05548014@ntu.edu.tw

Introduction

This technical document summarizes the programs (as shown in **table 1**) that have been done for automatically doing the segmentation of the left ventricles (LV's). The main idea is to utilize the designed U-net model to ultrasound images of LV's. Besides, image processes and how to generate ground truth labels are also illustrated to readers. Please read and follow the instructions to get familiar with it before move a step forward.

Other information might also be useful that can be found in this directory: /home/alvinli/Desktop/EF/Weekly_Reports.

Table 1: Developed Programs @ /home/alvinli/Desktop/EF/bin.

Program	Purpose
image_preprocess.py	Enhance images for labeling
boundary_fill.py	Get ground truth masks
data_rename.py	Rename files consistently
data2_pool.py	Pull data into a pool
data2_batched.py	Partition data to train_batch and test_batch
Unet_input.py	Create a training queue
Unet.py	The Unet model
Unet_train.py	Training the model
Unet_eval.py	Evaluating the model
volumn_calculator.py	Calculate the volume of LV

Materials

The ultrasound images of LV include two types: 2-chamber images and 4-chamber images, which were named by A2C and A4C, respectively.

This directory stores the original images without any modification: /home/alvinli/Desktop/EF/Raw_data.

From here, images would be selected for training purposes. Only the types of A2C and A4C were used and processed. An image with two screens (double A2C or A4C) should not be selected.

The selected images of A2C and A4C were respectively copied to: /home/alvinli/Desktop/EF/dataset/A2C_patients and /home/alvinli/Desktop/EF/dataset/A4C_patients.

A table called "Folder_information.ods" describing the information of data is inside both directories of A2C and A4C.

Please update this table manually if any folder is added or renamed.

Labeling process

For each patient folder in either /home/alvinli/Desktop/EF/dataset/A2C_patients or /home/alvinli/Desktop/EF/dataset/A4C_patients, please execute the program **image_preprocess.py** to enhance the images in entire folder.

Ex:

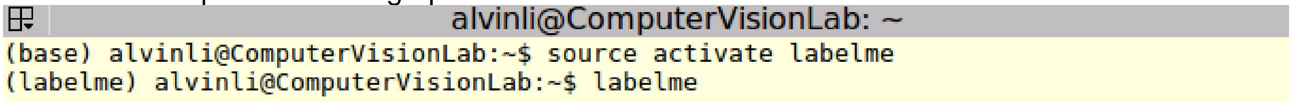
```
alvinli@ComputerVisionLab:~$ cd ~/Desktop/EF/dataset/A2C_patients/180234_20190510_1
alvinli@ComputerVisionLab:~/Desktop/EF/dataset/A2C_patients/180234_20190510_1$ python Desktop/EF/bin/ image_preprocess.py
```

Then, you get “*_enhanced.png” for each “*.png” accordingly.

The “enhanced images” are used to label the border of LV.

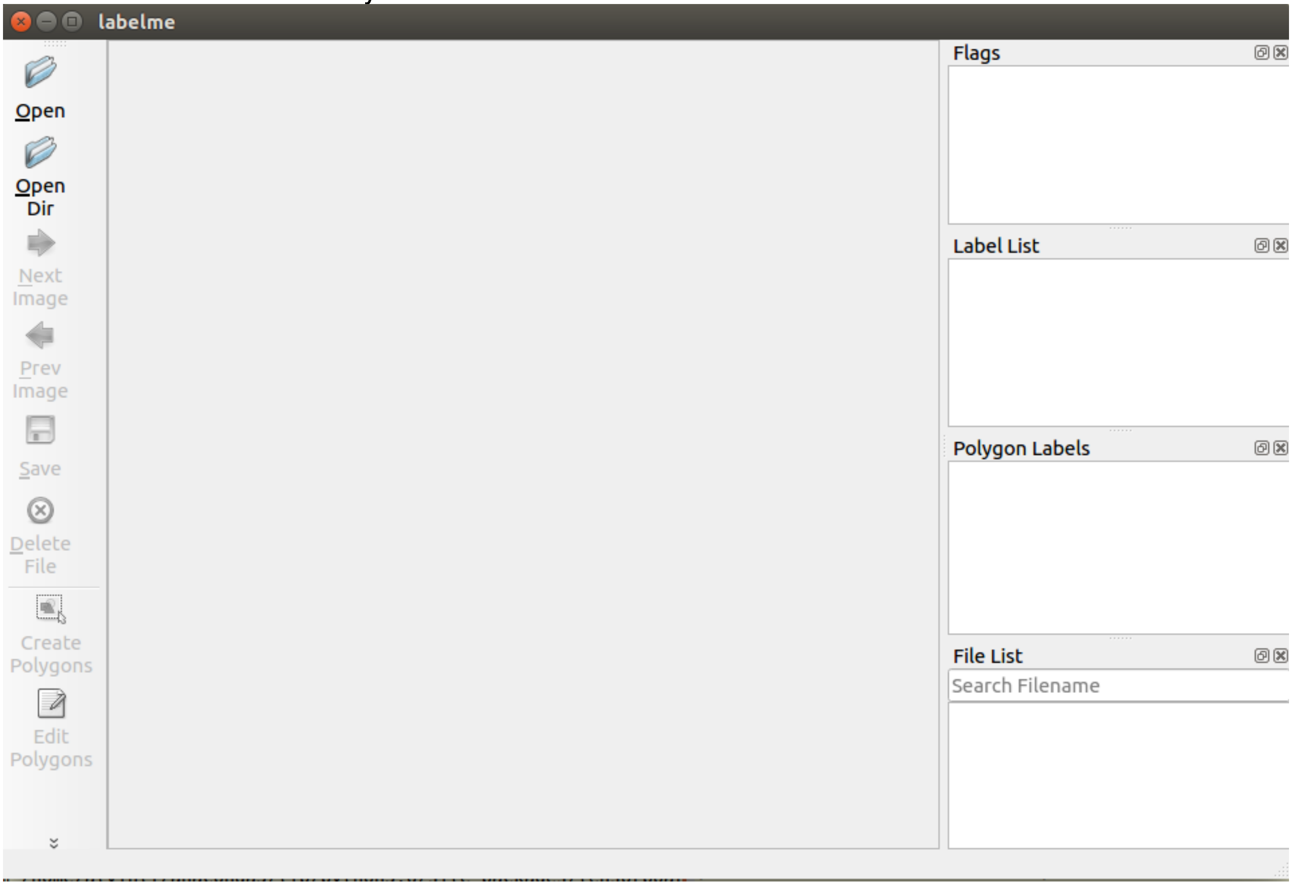
After the enhance process being done, open “labelme” app to label and save label information in the same folder.

This is how to open “labelme” graphic user interface:

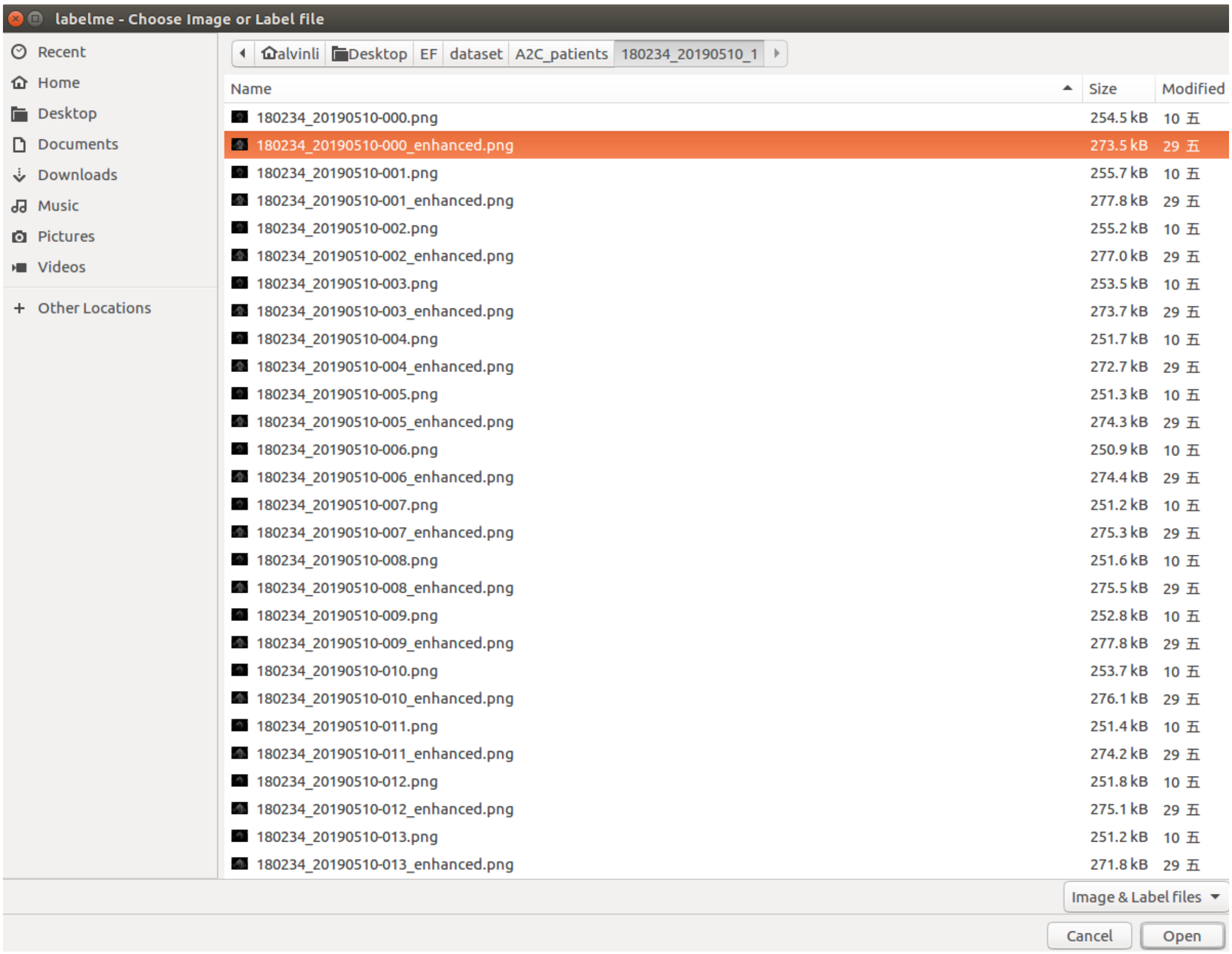


```
(base) alvinli@ComputerVisionLab:~$ source activate labelme
(labelme) alvinli@ComputerVisionLab:~$ labelme
```

You can see this window on your screen:



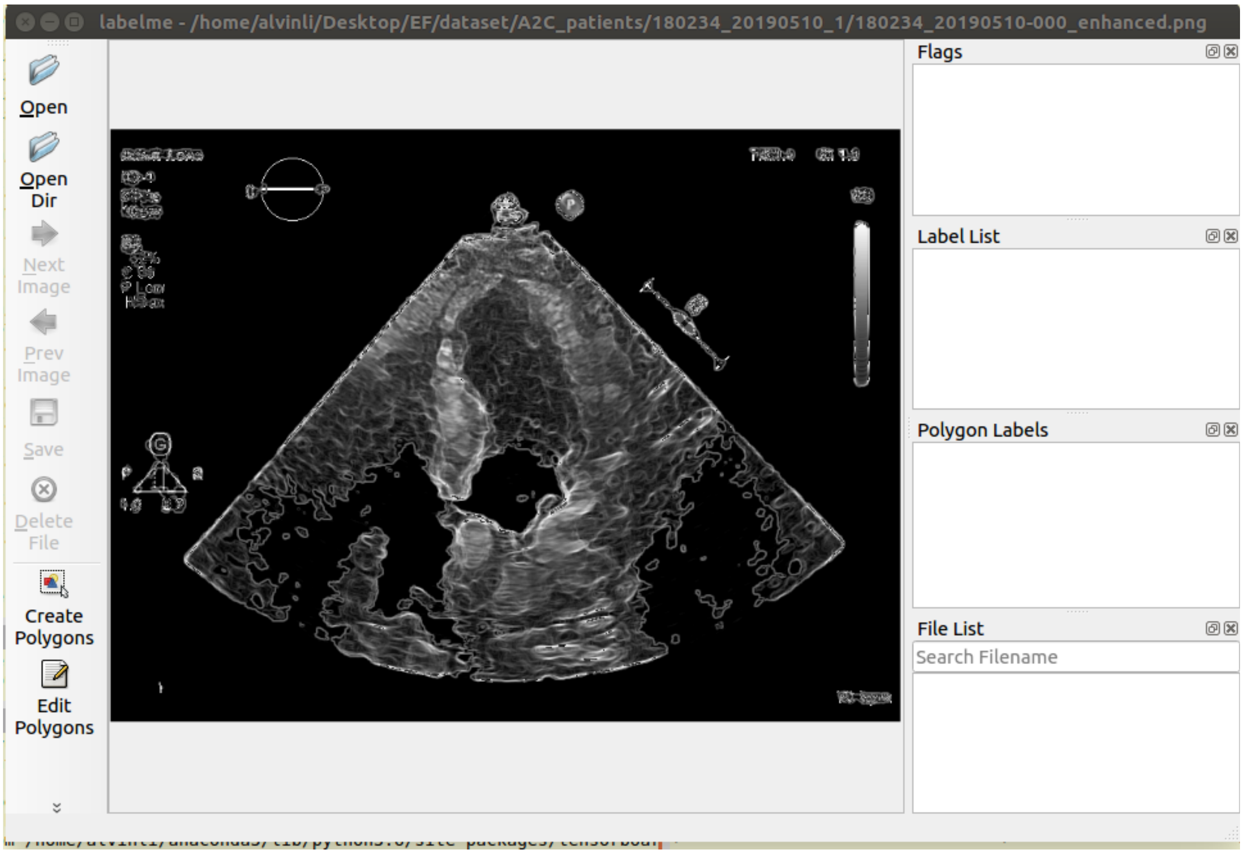
Click "Open" to select an image with the name "*_enhanced.png":



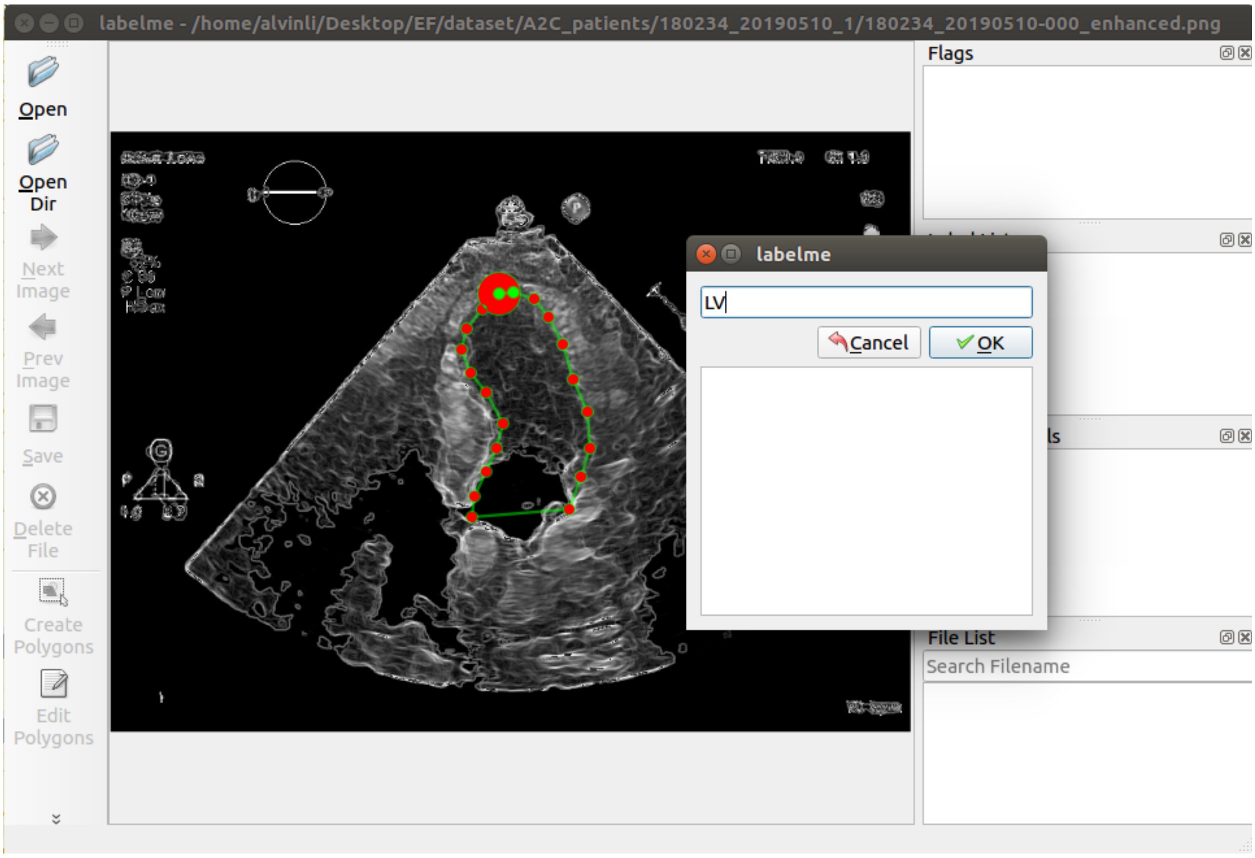
labelme - Choose Image or Label file

Name	Size	Modified
180234_20190510-000.png	254.5 kB	10 五
180234_20190510-000_enhanced.png	273.5 kB	29 五
180234_20190510-001.png	255.7 kB	10 五
180234_20190510-001_enhanced.png	277.8 kB	29 五
180234_20190510-002.png	255.2 kB	10 五
180234_20190510-002_enhanced.png	277.0 kB	29 五
180234_20190510-003.png	253.5 kB	10 五
180234_20190510-003_enhanced.png	273.7 kB	29 五
180234_20190510-004.png	251.7 kB	10 五
180234_20190510-004_enhanced.png	272.7 kB	29 五
180234_20190510-005.png	251.3 kB	10 五
180234_20190510-005_enhanced.png	274.3 kB	29 五
180234_20190510-006.png	250.9 kB	10 五
180234_20190510-006_enhanced.png	274.4 kB	29 五
180234_20190510-007.png	251.2 kB	10 五
180234_20190510-007_enhanced.png	275.3 kB	29 五
180234_20190510-008.png	251.6 kB	10 五
180234_20190510-008_enhanced.png	275.5 kB	29 五
180234_20190510-009.png	252.8 kB	10 五
180234_20190510-009_enhanced.png	277.8 kB	29 五
180234_20190510-010.png	253.7 kB	10 五
180234_20190510-010_enhanced.png	276.1 kB	29 五
180234_20190510-011.png	251.4 kB	10 五
180234_20190510-011_enhanced.png	274.2 kB	29 五
180234_20190510-012.png	251.8 kB	10 五
180234_20190510-012_enhanced.png	275.1 kB	29 五
180234_20190510-013.png	251.2 kB	10 五
180234_20190510-013_enhanced.png	271.8 kB	29 五

Click "Create Polygons" to use the label tool and store it with label index as 'LV':



labelme - /home/alvinli/Desktop/EF/dataset/A2C_patients/180234_20190510_1/180234_20190510-000_enhanced.png

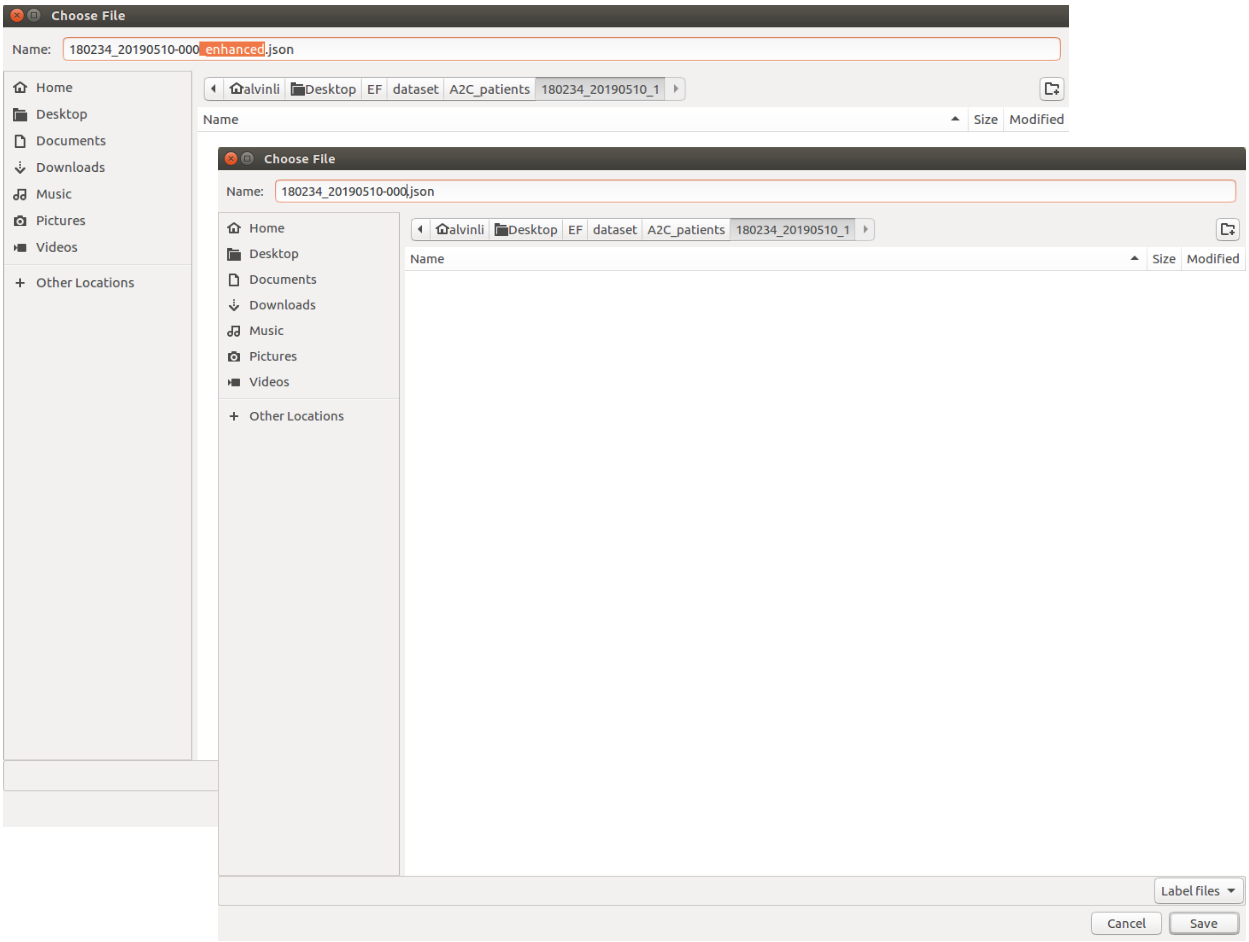


labelme - /home/alvinli/Desktop/EF/dataset/A2C_patients/180234_20190510_1/180234_20190510-000_enhanced.png

labelme

LV

Click “Save” to save it in the folder with the name of “.json” **without** “_enhanced” (*delete* it from the recommended name):



Choose File

Name: 180234_20190510-000_enhanced.json

Choose File

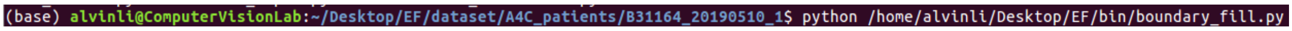
Name: 180234_20190510-000.json

Do it for every “*_enhanced.png” to get the “*.json” accordingly.

Generate ground truth masks

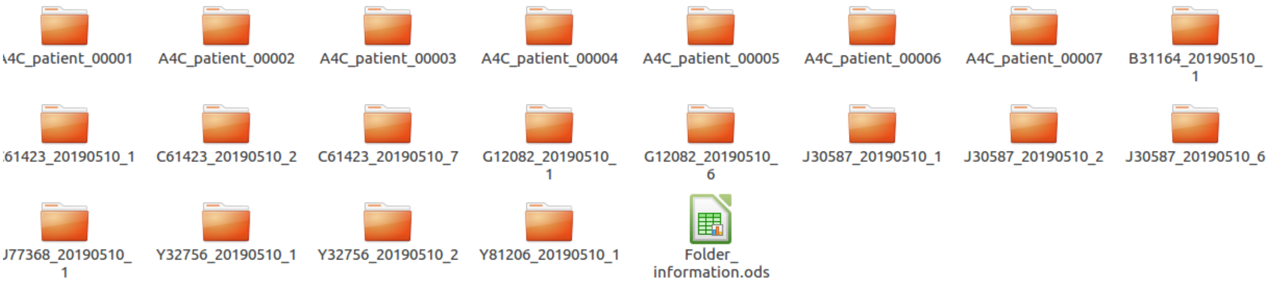
After you have labeled all the enhanced images in the same folder, please execute the program **boundary_fill.py** at the same folder to get the ground truth mask with the name of “*_ROI_mask.png”.

Ex:



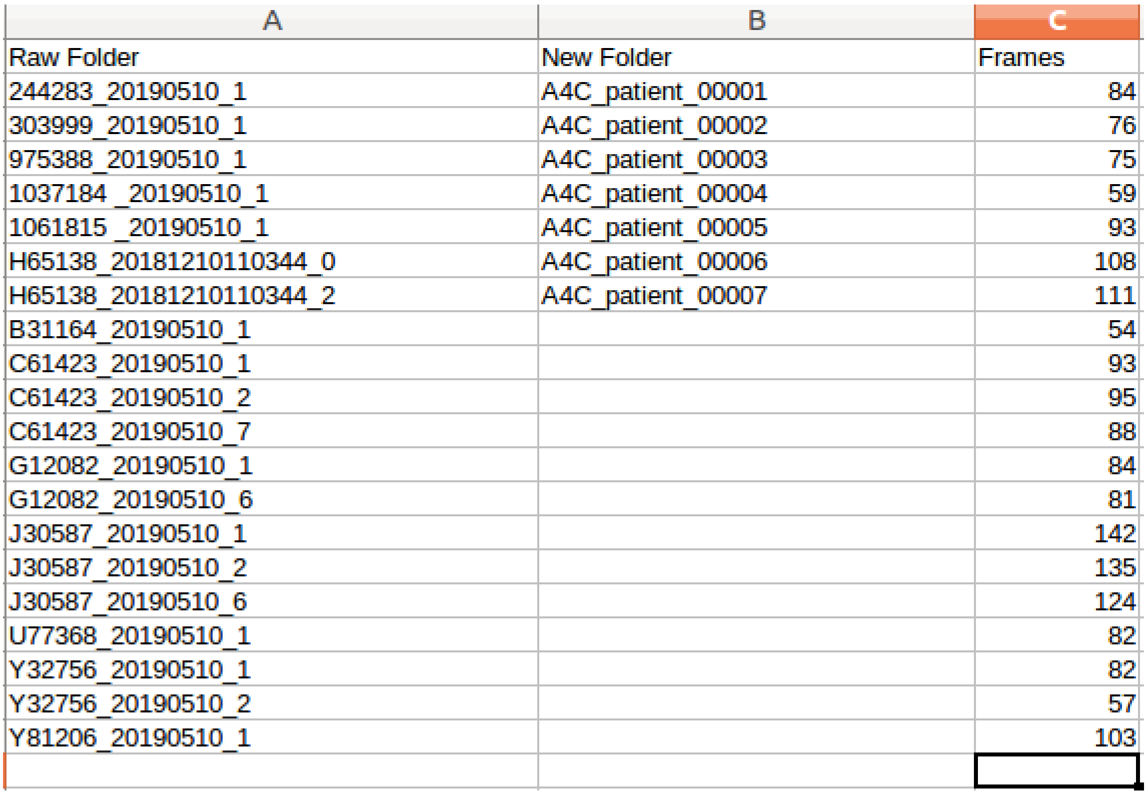
```
(base) alvinli@ComputerVisionLab:~/Desktop/EF/dataset/A4C_patients/B31164_20190510_1$ python /home/alvinli/Desktop/EF/bin/boundary_fill.py
```

Then, please rename the folder manually to this form “A2C_patient_xxxxx” or “A4C_patient_xxxxx” like:



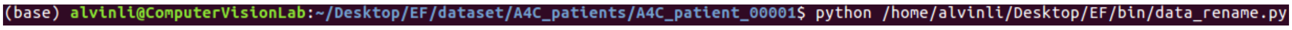
4C_patient_00001
A4C_patient_00002
A4C_patient_00003
A4C_patient_00004
A4C_patient_00005
A4C_patient_00006
A4C_patient_00007
B31164_20190510_1
61423_20190510_1
C61423_20190510_2
C61423_20190510_7
G12082_20190510_1
G12082_20190510_6
J30587_20190510_1
J30587_20190510_2
J30587_20190510_6
J77368_20190510_1
Y32756_20190510_1
Y32756_20190510_2
Y81206_20190510_1
Folder_information.ods

This will make the folder names in a consistent way for further processes. However, please **only** rename the folder after you have labeled all the inside enhanced-images. This can make a difference to distinguish the folders with or without labeling. And please update “Folder_information.ods” to make a record for this change.



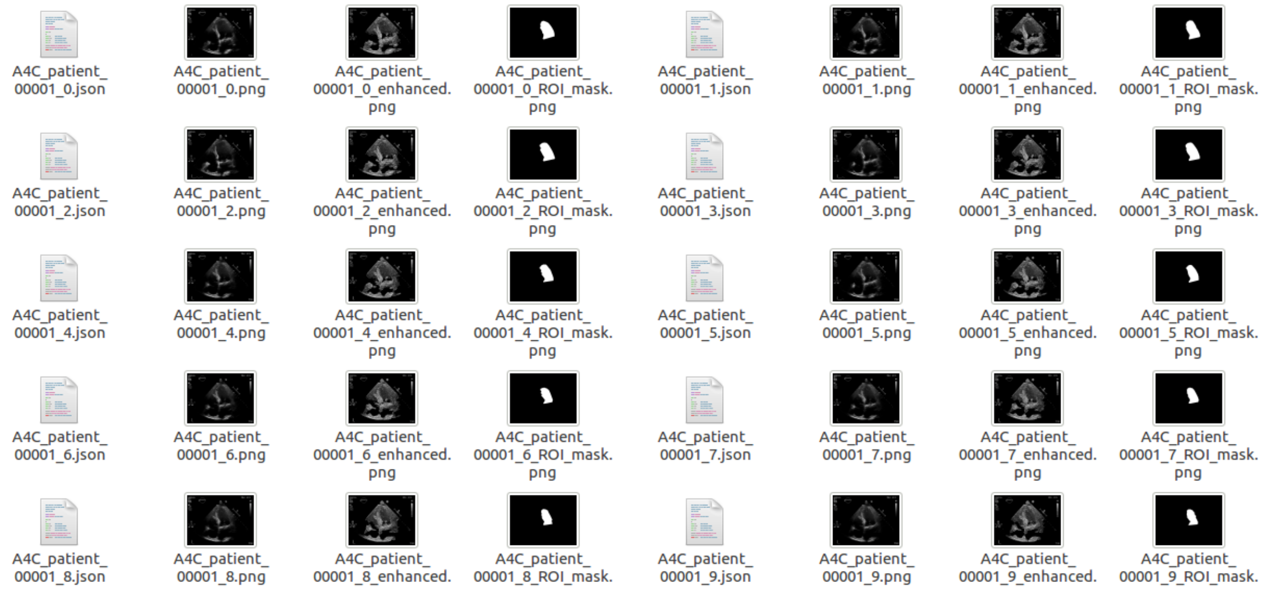
A	B	C
Raw Folder	New Folder	Frames
244283_20190510_1	A4C_patient_00001	84
303999_20190510_1	A4C_patient_00002	76
975388_20190510_1	A4C_patient_00003	75
1037184 _20190510_1	A4C_patient_00004	59
1061815 _20190510_1	A4C_patient_00005	93
H65138_20181210110344_0	A4C_patient_00006	108
H65138_20181210110344_2	A4C_patient_00007	111
B31164_20190510_1		54
C61423_20190510_1		93
C61423_20190510_2		95
C61423_20190510_7		88
G12082_20190510_1		84
G12082_20190510_6		81
J30587_20190510_1		142
J30587_20190510_2		135
J30587_20190510_6		124
U77368_20190510_1		82
Y32756_20190510_1		82
Y32756_20190510_2		57
Y81206_20190510_1		103

Then, please execute the program **data_rename.py** to rename all the files in the folder:



```
(base) alvinli@ComputerVisionLab:~/Desktop/EF/dataset/A4C_patients/A4C_patient_00001$ python /home/alvinli/Desktop/EF/bin/data_rename.py
```

This will make the files' name in a consistent way:



```
A4C_patient_00001_0.json   A4C_patient_00001_0.png   A4C_patient_00001_0_enhanced.png   A4C_patient_00001_0_ROI_mask.png   A4C_patient_00001_1.json   A4C_patient_00001_1.png   A4C_patient_00001_1_enhanced.png   A4C_patient_00001_1_ROI_mask.png
A4C_patient_00001_2.json   A4C_patient_00001_2.png   A4C_patient_00001_2_enhanced.png   A4C_patient_00001_2_ROI_mask.png   A4C_patient_00001_3.json   A4C_patient_00001_3.png   A4C_patient_00001_3_enhanced.png   A4C_patient_00001_3_ROI_mask.png
A4C_patient_00001_4.json   A4C_patient_00001_4.png   A4C_patient_00001_4_enhanced.png   A4C_patient_00001_4_ROI_mask.png   A4C_patient_00001_5.json   A4C_patient_00001_5.png   A4C_patient_00001_5_enhanced.png   A4C_patient_00001_5_ROI_mask.png
A4C_patient_00001_6.json   A4C_patient_00001_6.png   A4C_patient_00001_6_enhanced.png   A4C_patient_00001_6_ROI_mask.png   A4C_patient_00001_7.json   A4C_patient_00001_7.png   A4C_patient_00001_7_enhanced.png   A4C_patient_00001_7_ROI_mask.png
A4C_patient_00001_8.json   A4C_patient_00001_8.png   A4C_patient_00001_8_enhanced.   A4C_patient_00001_8_ROI_mask.   A4C_patient_00001_9.json   A4C_patient_00001_9.png   A4C_patient_00001_9_enhanced.   A4C_patient_00001_9_ROI_mask.
```

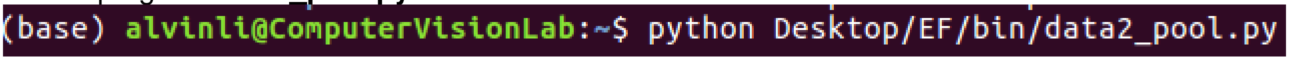
Partition data to train_batch (*80%*) and test_batch (*20%*)

Step 1:

Create a folder called 'EF-training-Pool' at the path like this:
'/home/alvinli/Desktop/EF/dataset/EF-training-Pool'

Step 2:

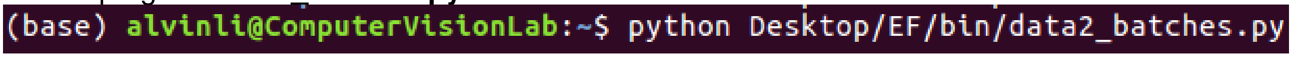
run the program **data2_pool.py** :



```
(base) alvinli@ComputerVisionLab:~$ python Desktop/EF/bin/data2_pool.py
```

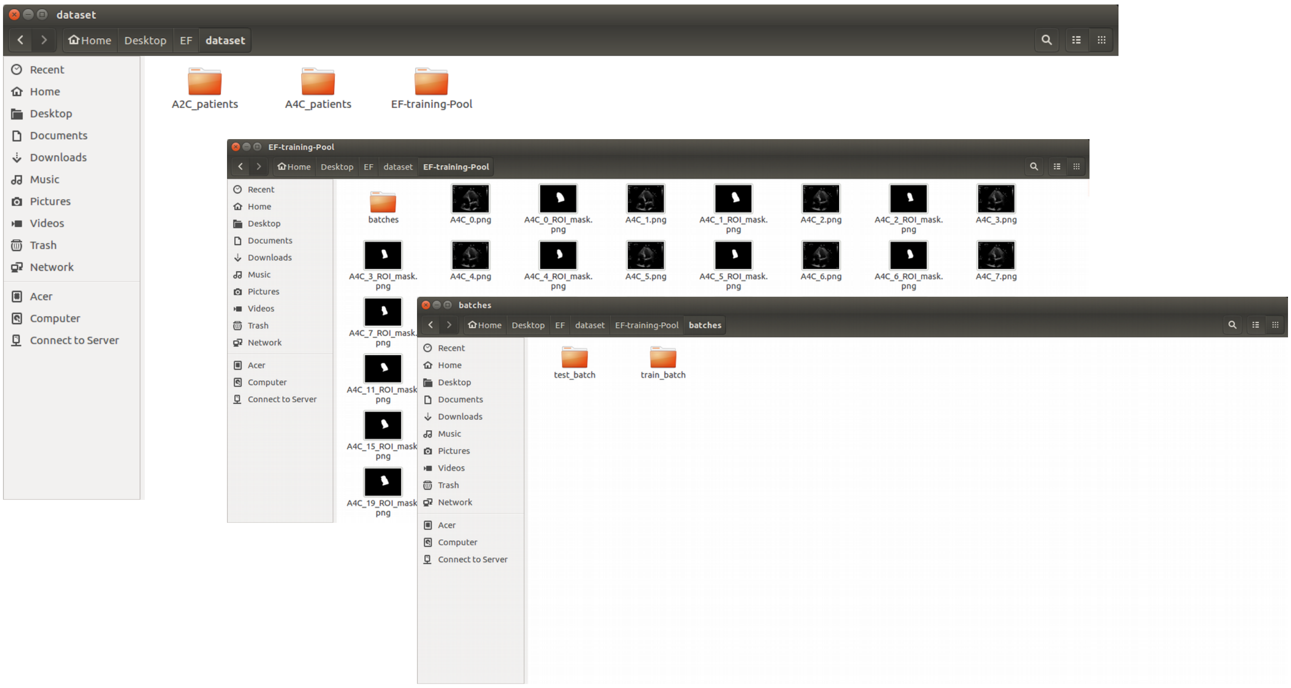
Step 3:

run the program **data2_batches.py** :



```
(base) alvinli@ComputerVisionLab:~$ python Desktop/EF/bin/data2_batches.py
```

Then you should get this:



Train model

~$ python Unet_train.py

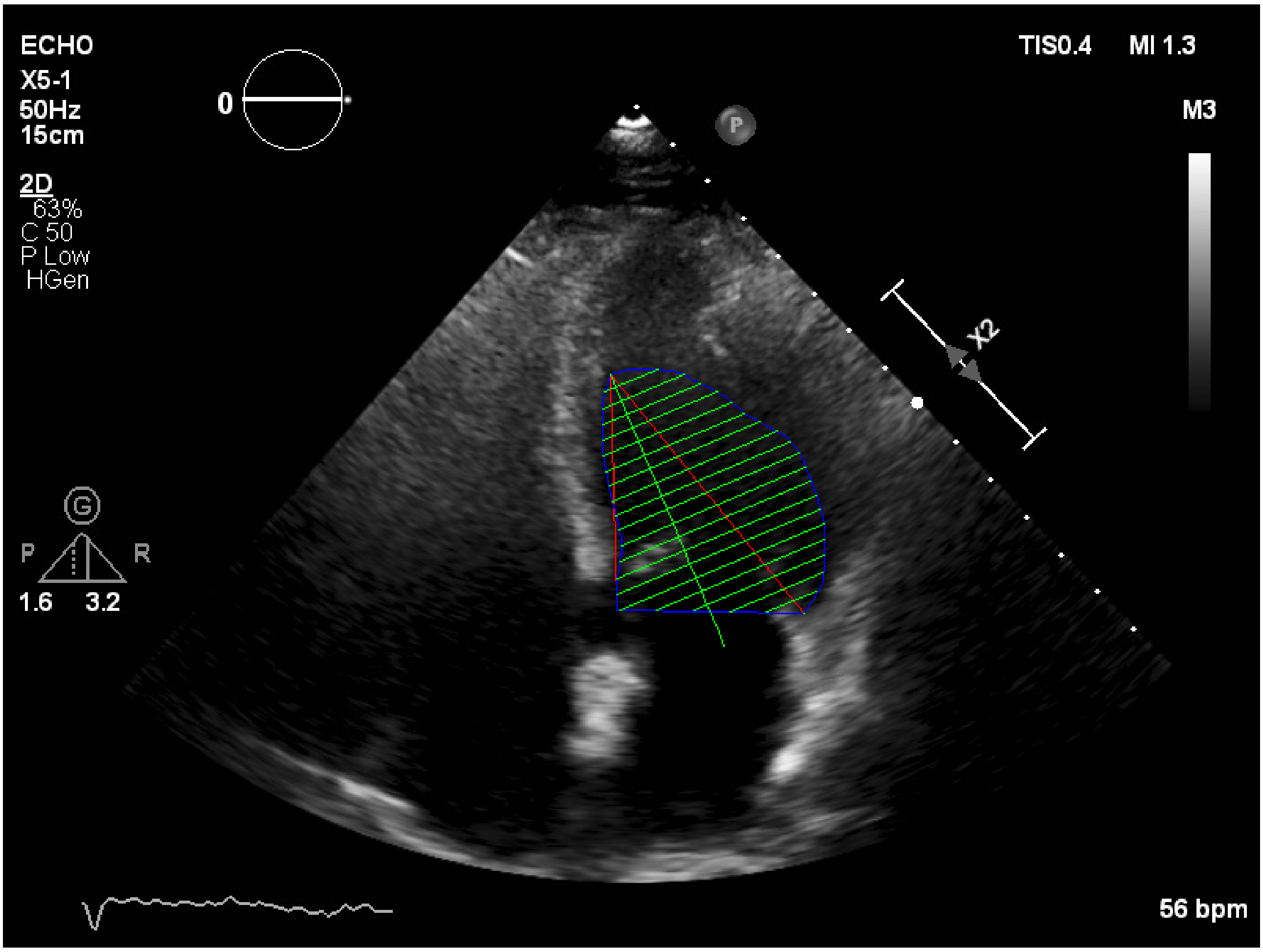
Evaluate model

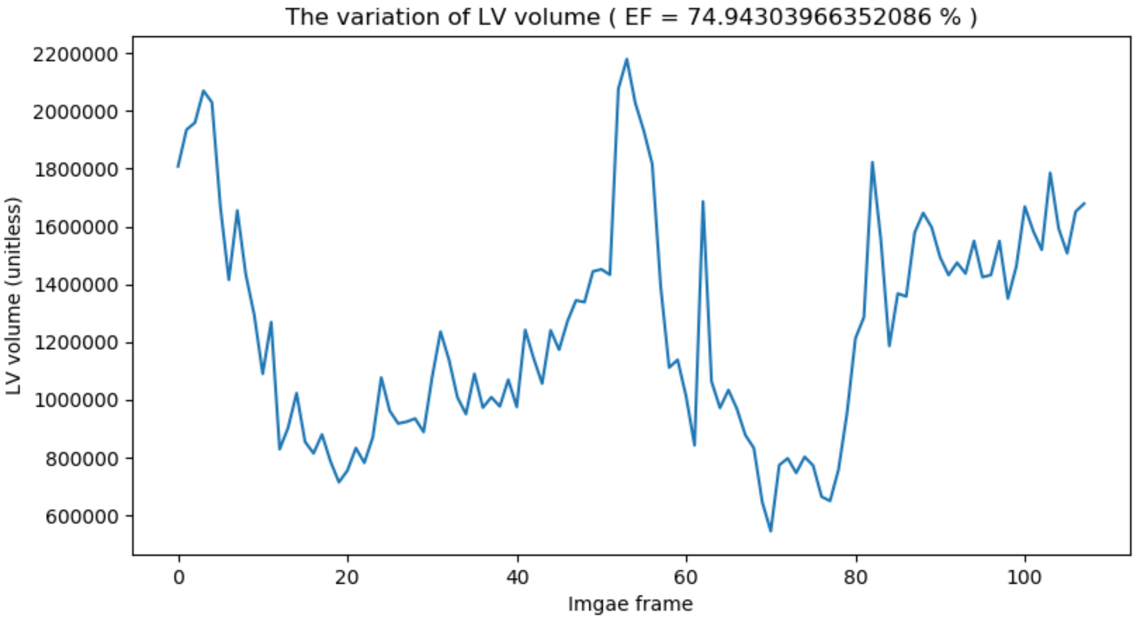
~$ python Unet_eval.py

About the model structure, please refer to the working notes:
/home/alvinli/Desktop/EF/Weekly_Reports

volumn_calculator.py

→ calculate the volume of LV by the modified Simpson's Method.
The input file is a *.json file storing the discrete labels.
The "nothing" function can output the ejection fraction with a heart pumping circle chart.


ECHO
X5-1
50Hz
15cm
2D
63%
C 50
P Low
HGen
0
P
TIS0.4
MI 1.3
M3
X2
G
P
R
1.6
3.2
56 bpm


The variation of LV volume (EF = 74.94303966352086 %)
LV volume (unitless)
2200000
2000000
1800000
1600000
1400000
1200000
1000000
800000
600000
0
20
40
60
80
100
Imgae frame