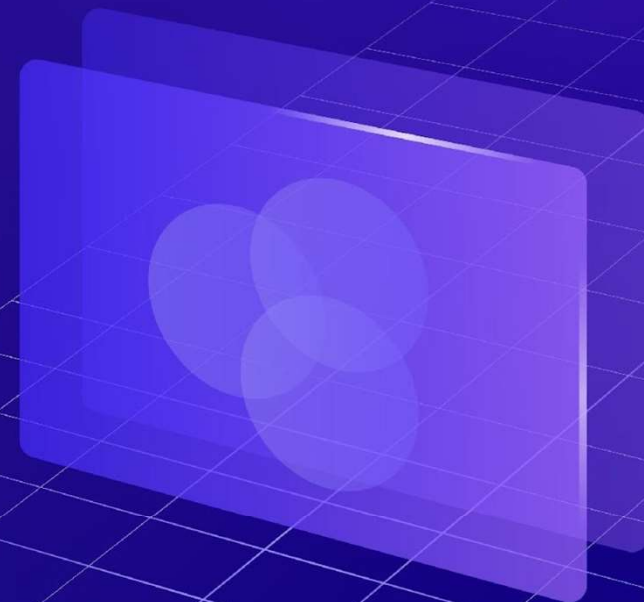


AI CUP 2025 秋季賽

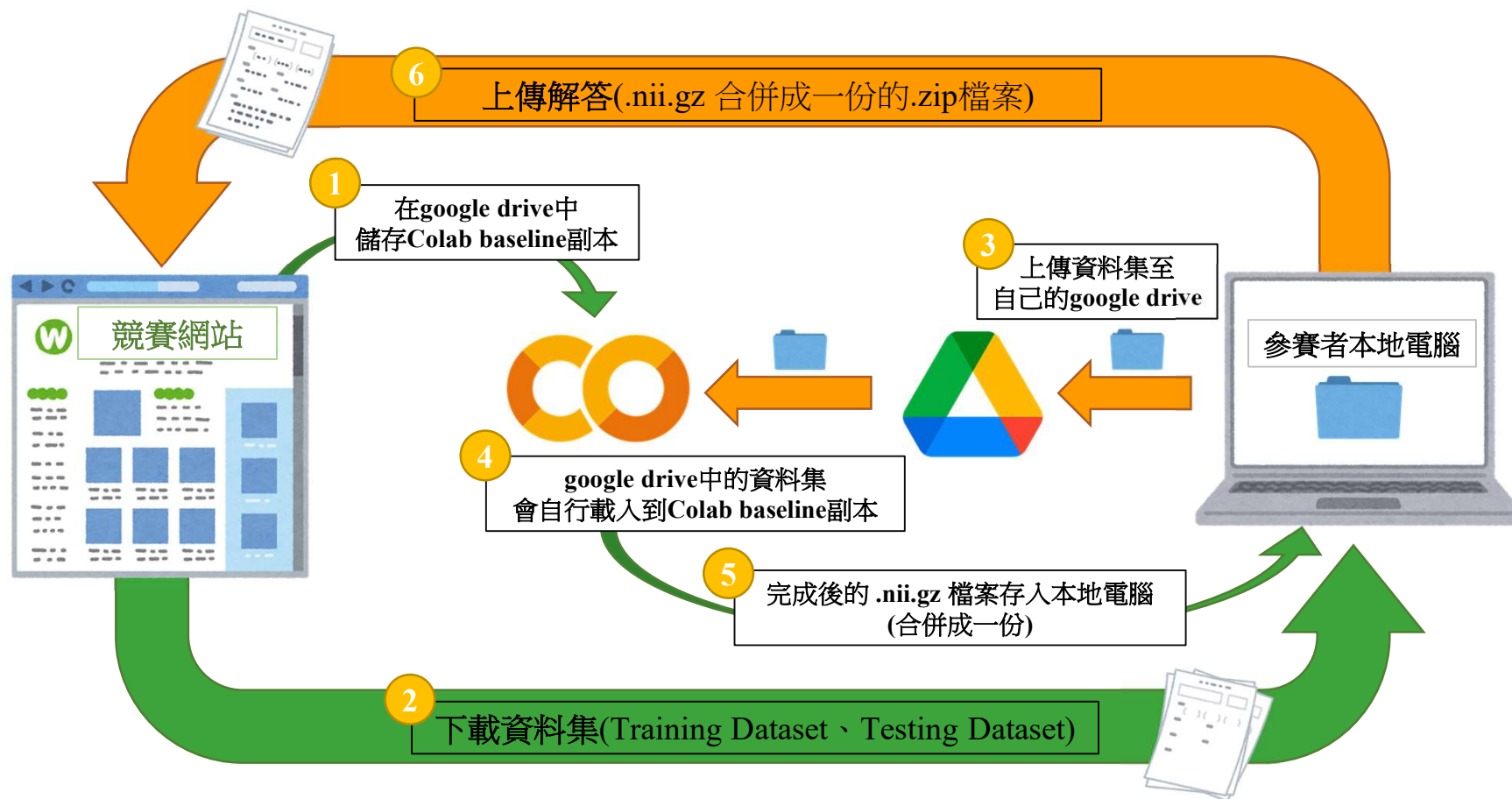
電腦斷層心臟肌肉影像分割競賽 I
心臟肌肉影像分割



目錄

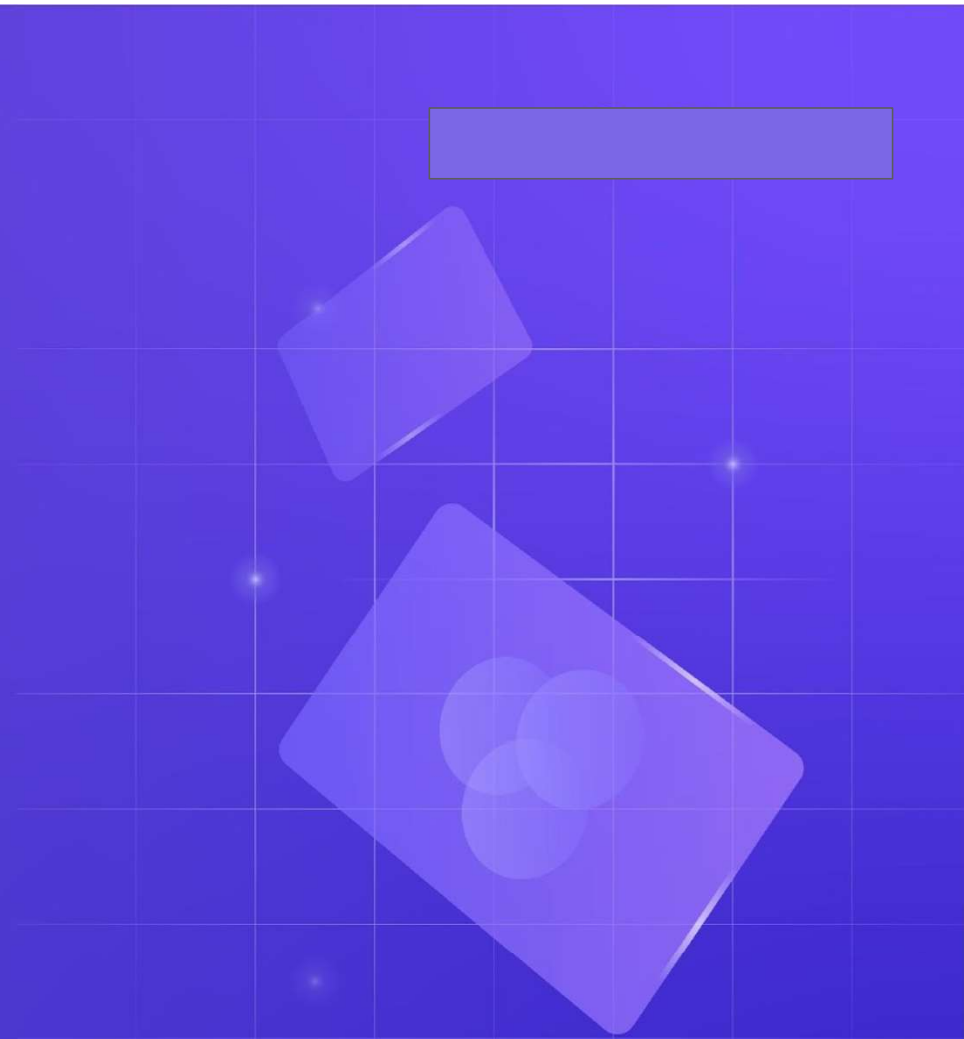
1. 資料集說明
2. 繳交檔案格式說明
3. Colab Baseline 程式說明 (train)
4. Colab Baseline 程式說明 (predict)
5. 預測成果上傳至競賽網頁

使用 Baseline 程式參加競賽流程圖



01

資料集説明



下載競賽資料集



新興智慧顯示科技
教育聯盟

- 報名完成後可至趨勢科技Tbrain競賽網站下載資料集
- 下載網址：<https://tbrain.trendmicro.com.tw/Competitions/Details/41>



點選可以看到資料集

訓練集說明

- 資料集包含兩部分Training Dataset和Testing Dataset。

Training Dataset

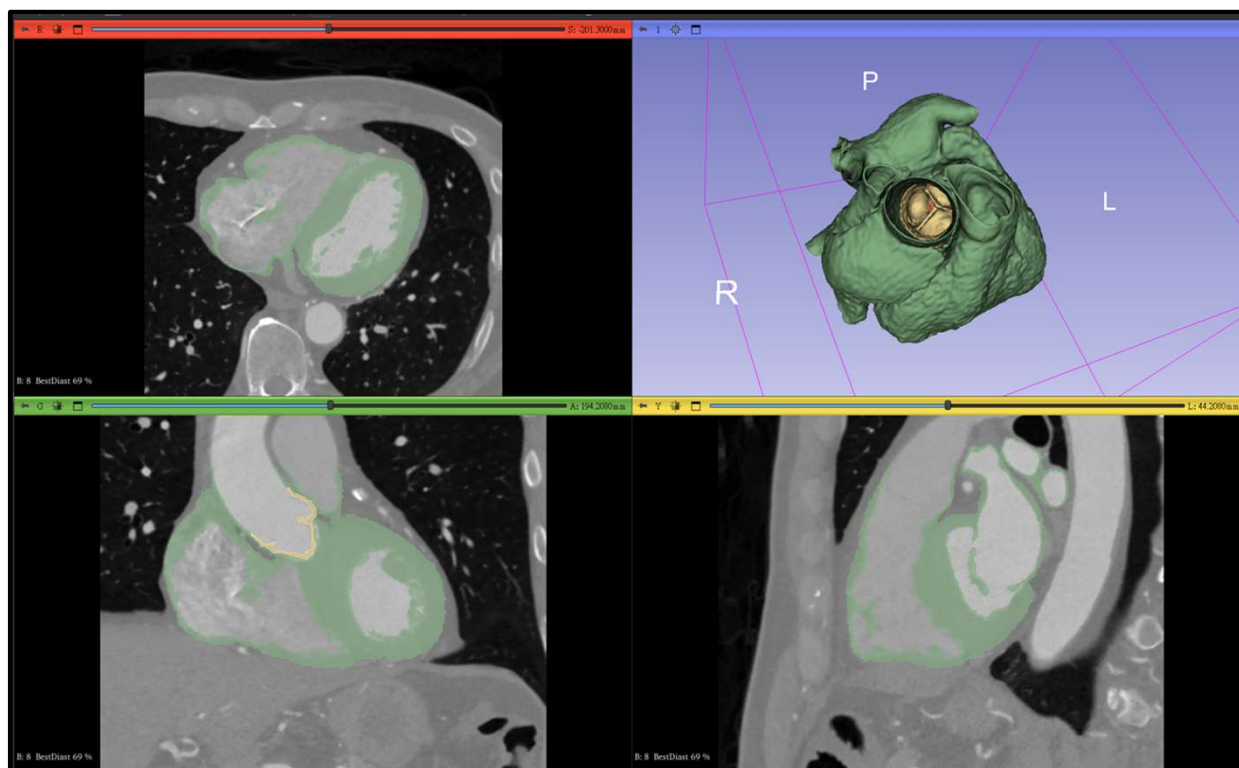
- Training Dataset 有 training_image1、2、3 和 training_label 兩種資料夾，它們裡面各自有 patient0001~patient0050 的原始CT資料，以及對應的標註檔 ground_truth(gt) 資料(如圖)。



◀ 訓練資料集

資料檢視說明-1

- training_image 資料夾內包含原始的 CT 電腦斷層影像，檔案格式為 .nii.gz（壓縮版的 NIfTI 格式）。
- training_label 資料夾內包含對應的標註資料，為經模型和人工標記過的 CT 影像，檔案格式為 .nii.gz（壓縮版的 NIfTI 格式）。



左上 左下 右下 為原始CT電腦斷層影像，
有顏色的部分為標註資料

對應在該原始CT電腦斷層影像的位置

右上為標記出來的資料所呈現出的3D模型

◀ 3D Slicer畫面

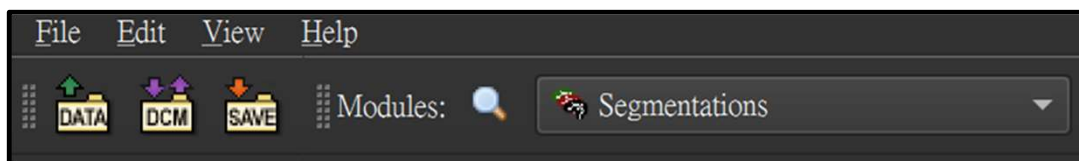
資料檢視說明-2

- 要檢視資料的話，建議使用3D Slicer(<https://www.slicer.org/>)，將資料讀取後將CT檔選用 Volume，labels 檔(gt)選用 Segmentation(如圖一)。



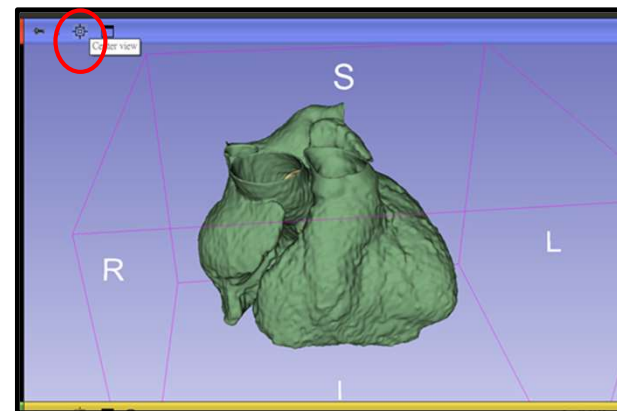
▲ 圖一

- 將上方選為 segmentation (如圖二)，之後點選 Show 3D 並點選 3D 視窗的 Center View (如圖三)。



▲ 圖二

▶ 圖三

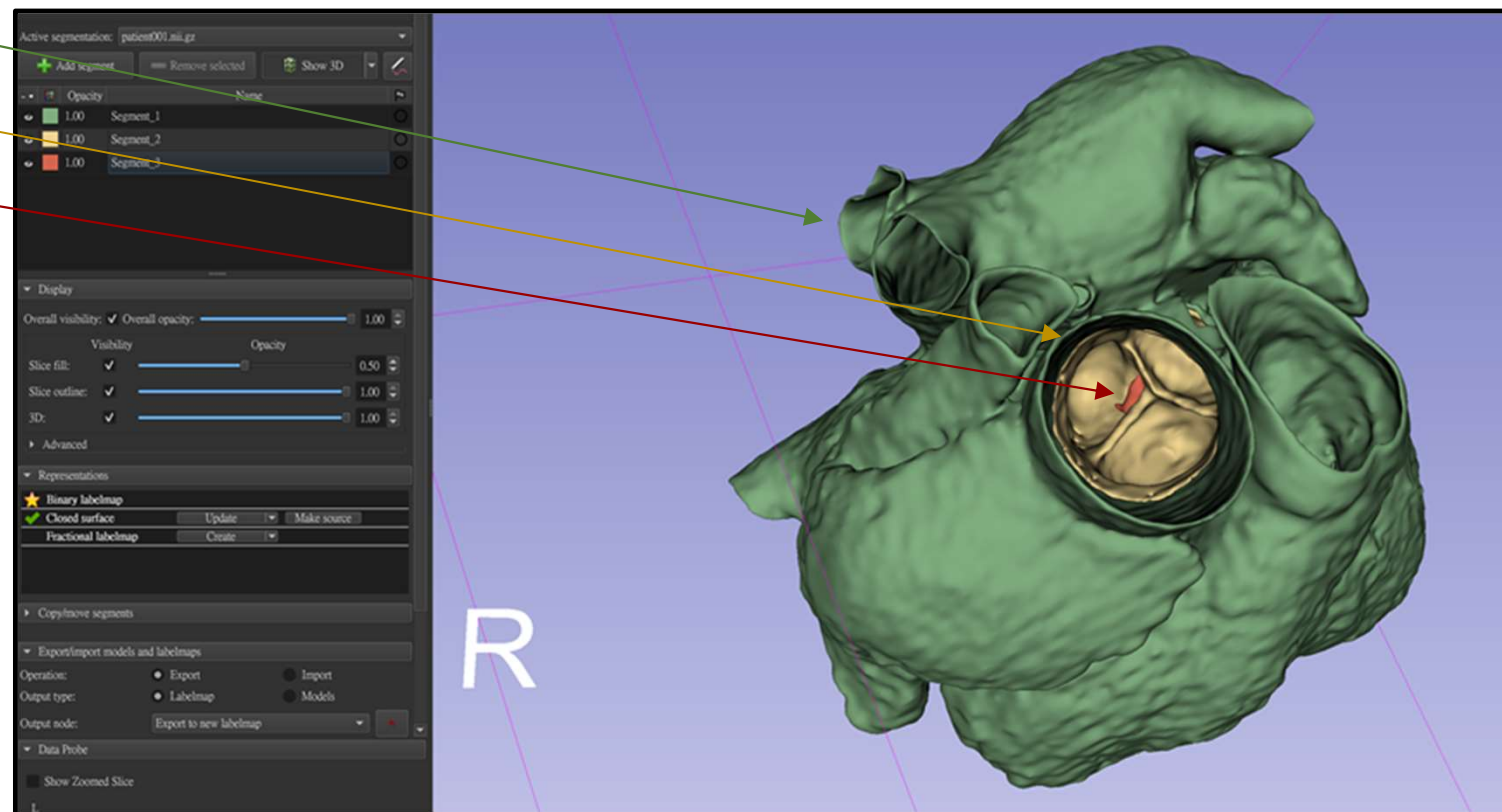


資料檢視說明-3

Segment_1(綠色)為心臟肌肉

Segment_2(黃色)為主動脈瓣膜

Segment_3(紅色)為鈣化(不一定會有)



▲ 3D Slicer畫面



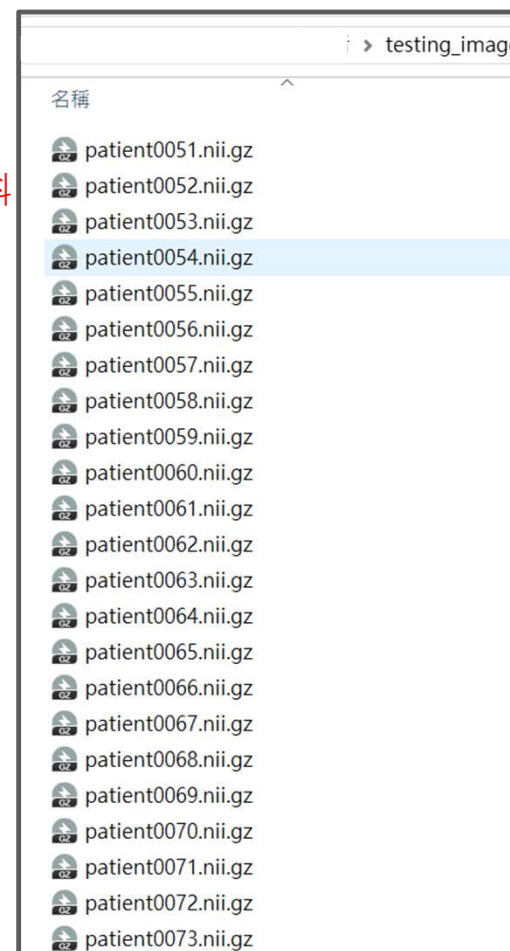
測試集說明

- Testing Dataset
- Testing Dataset · 只有patient0051~patient0100 的CT資料
- 將使用這50筆資料產出推論的.nii.gz的label檔



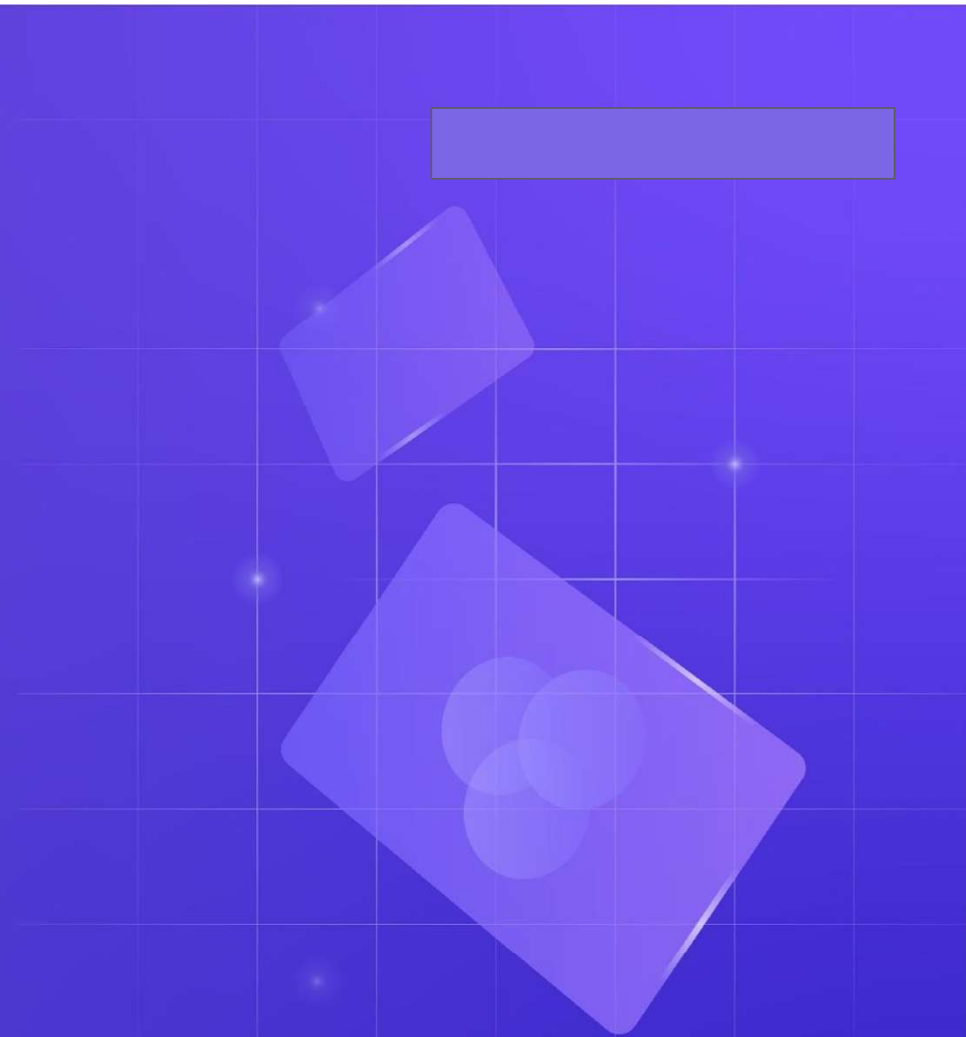
本次競賽試題！

► 測試資料集



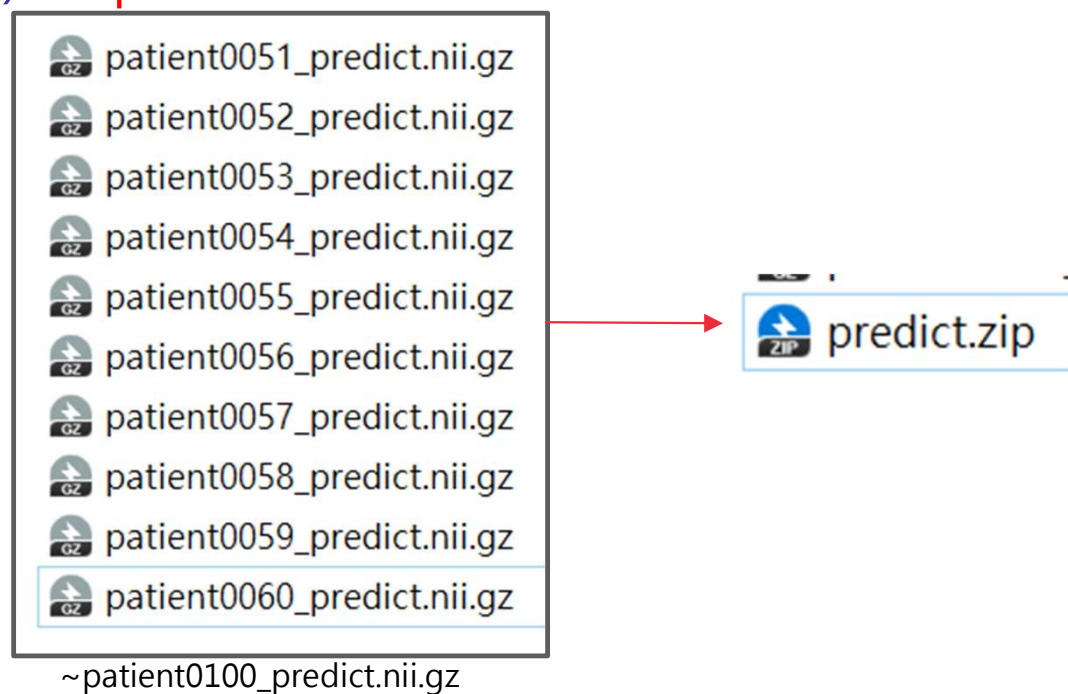
02

繳交檔案格式說明



繳交檔案格式說明

- 參賽者只需要繳交**共一份**合併後包含patient0051~patient0100推論過資料(格式：**.nii.gz**)的**zip**檔案



▲ 繳交範例

03

Colab Baseline 程式說明 (train)



打開瀏覽器



新興智慧顯示科技
教育聯盟

Baseline 會透過瀏覽器打開 Colab 完成模型訓練。



3D U-Net training (Easy Baseline)



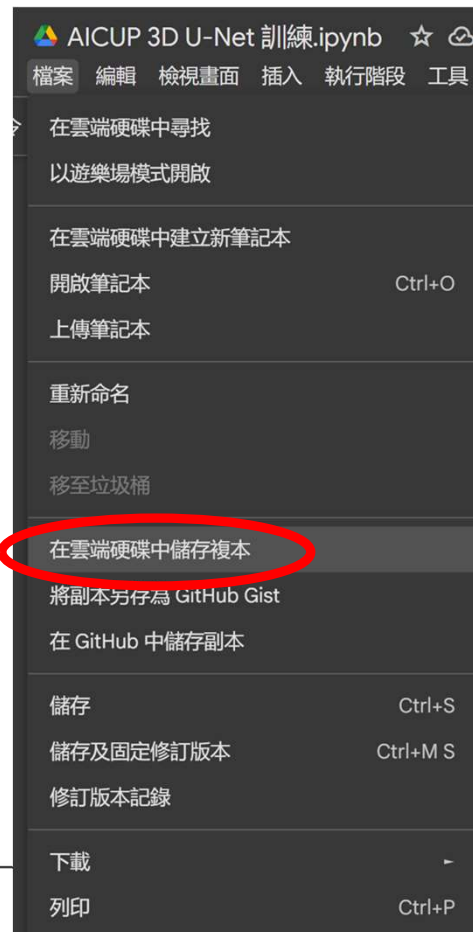
colab連結：

https://colab.research.google.com/drive/1iC7i_EWCZsCr5T-7jDD77V8dt_simGsn?usp=sharing

3D U-Net (Easy Baseline)



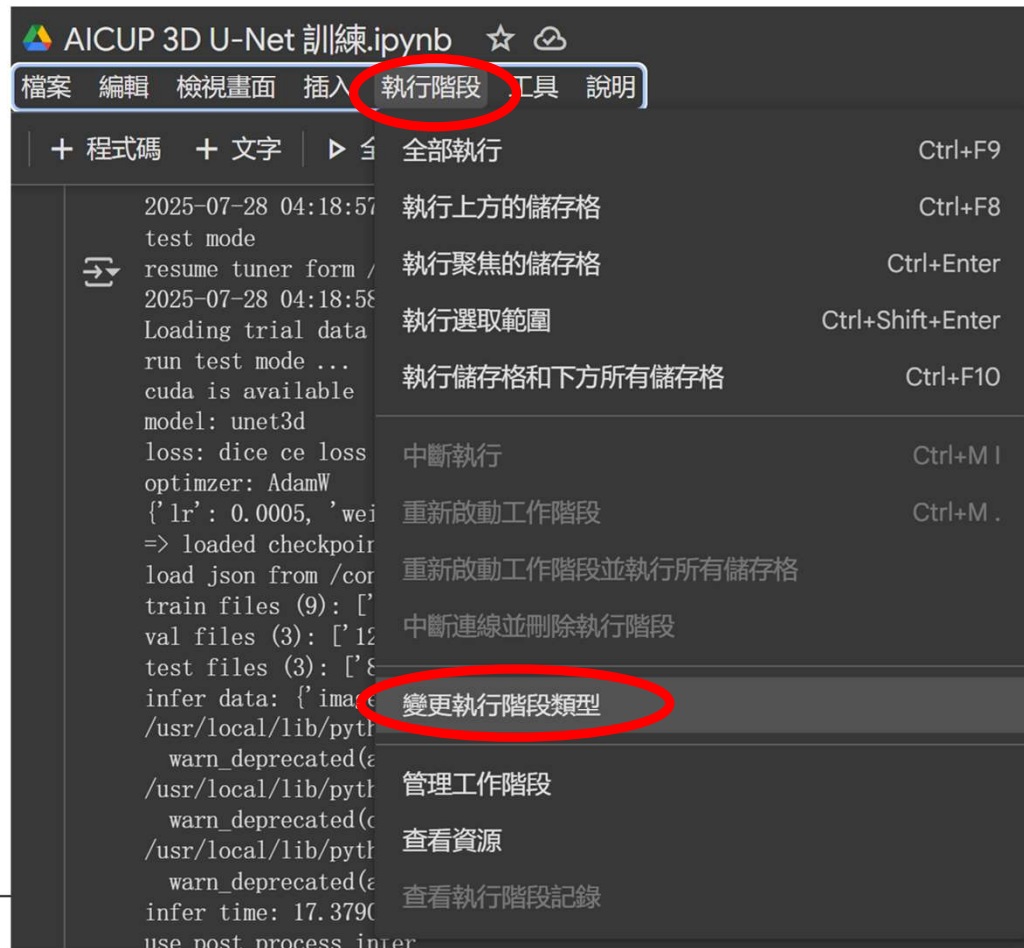
打開colab連結，並將這份colab儲存一份副本到自己的colab內



3D U-Net (Easy Baseline)



點選「執行階段」->「變更執行階段類型」



3D U-Net (Easy Baseline)



點選「T4 GPU」、執行階段版本選擇2025.07 -> 儲存

變更執行階段類型

執行階段類型

Python 3

硬體加速器 ?

☐ CPU ☒ T4 GPU ☐ A100 GPU ☐ L4 GPU

☐ v5e-1 TPU ☐ v2-8 TPU ☐ v6e-1 TPU

想要使用付費 GPU 嗎? [購買額外運算單元](#)

執行階段版本 ?

2025.07

取消 儲存

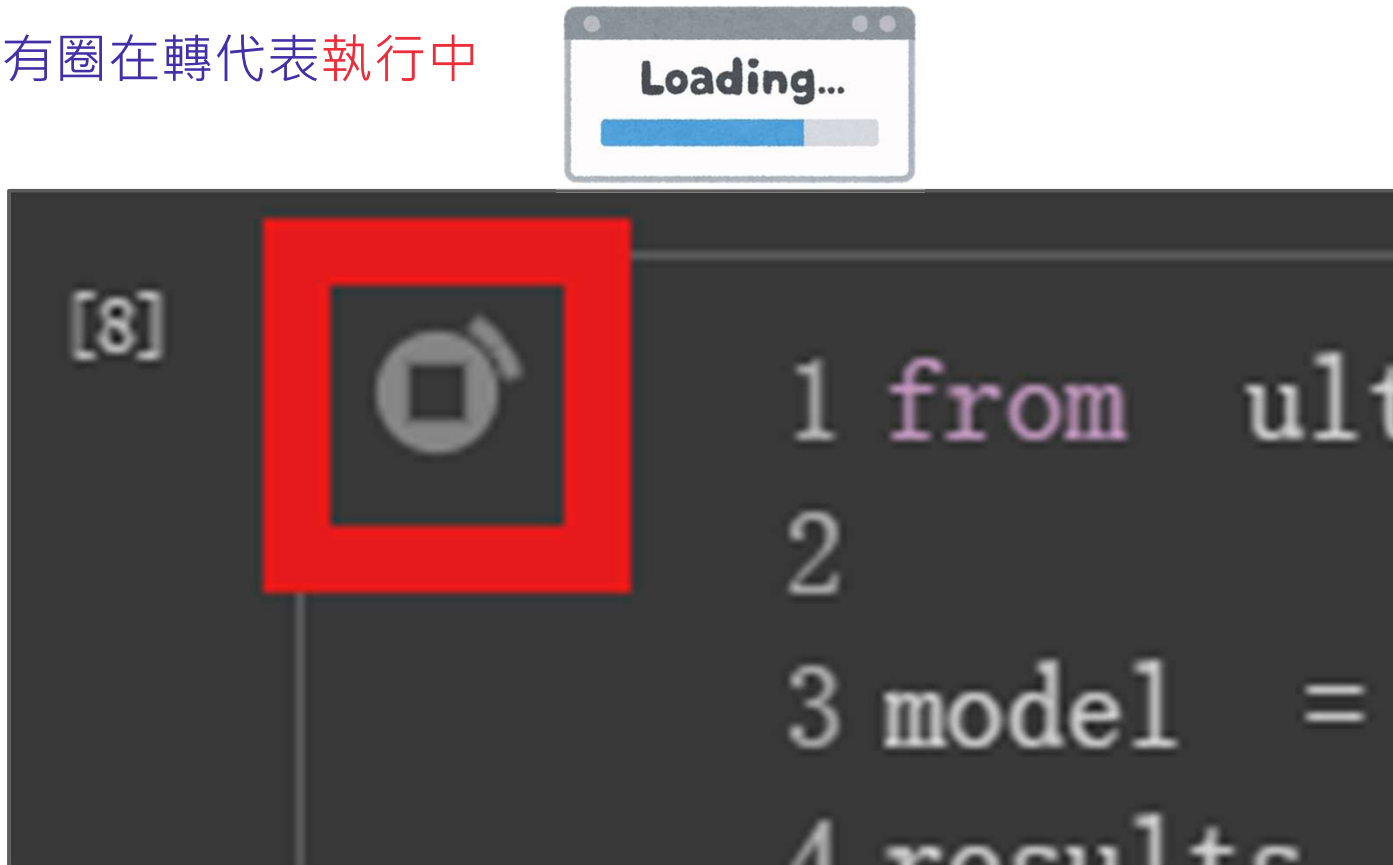
Colab運行情況(成功)

Colab區塊左方有綠色勾代表執行完成且成功



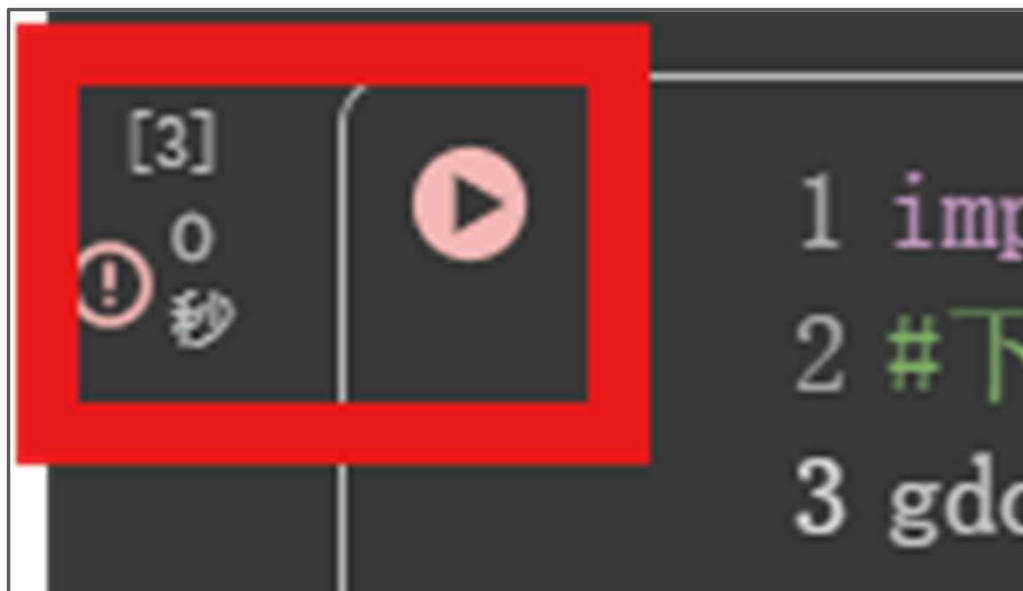
Colab運行情況(執行中)

區塊左方有圈在轉代表執行中



Colab運行情況(運行錯誤)

區塊左方有紅色驚嘆號代表運行錯誤



STEP 1: 環境設定



執行每一行，會出現紅字錯誤是正常的，但最後必須要看到輸出：

```
Successfully installed [套件]
```

```
import sys
!{sys.executable} --version
!{sys.executable} -m pip install -U numpy==1.26.4
import os
os._exit(0)
```

顯示隱藏的輸出內容

```
import sys
!{sys.executable} -m pip install -U ray==2.5.0
import ray
print(ray.__version__)
```

顯示隱藏的輸出內容

```
import sys
!{sys.executable} -m pip install -U monai==1.2.0
import monai
print(monai.__version__)
```

顯示隱藏的輸出內容

```
!{sys.executable} -m pip install -U ml_collections
```

顯示隱藏的輸出內容

```
!git clone https://github.com/kairaun/Sam.git
```

顯示隱藏的輸出內容

```
%cd ./Sam
! ls
! head pyproject.toml
!pip install -e .
import os
os._exit(0)
```

顯示隱藏的輸出內容

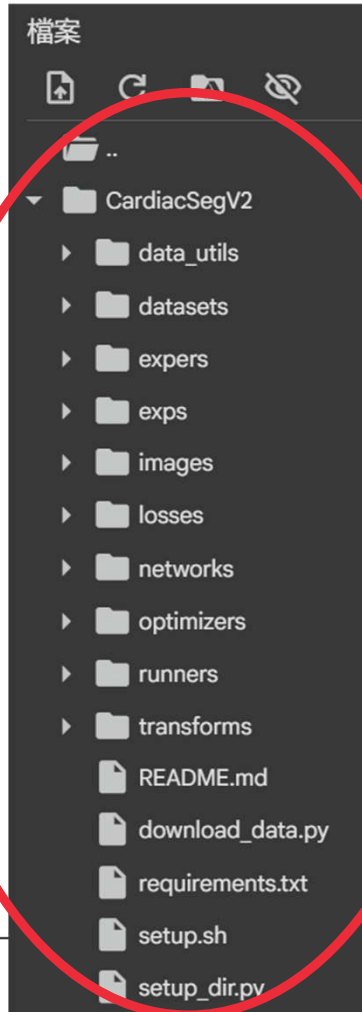
```
import sys
!{sys.executable} -m pip install -U --no-deps monailabel
import monailabel
print(monailabel.__version__)
```

STEP 1: 環境設定

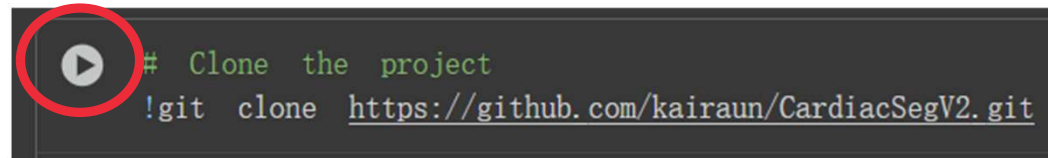
將需要的檔案透過gitclone複製至colab暫存
左方會出現clone好的檔案



2. 新增了
CardiacSegV2
資料夾



1. 點選執行



```
# Clone the project
!git clone https://github.com/kairaun/CardiacSegV2.git
```

STEP 1: 環境設定

依序執行cell，安裝並調整好執行環境
遇到一些**pip error**可以不理他請繼續執行



```
# Install the Environment
! sudo chmod 700 /content/CardiacSegV2/setup.sh
! /content/CardiacSegV2/setup.sh
! pip install gdown==4.6.0
```

顯示隱藏的輸出內容

```
# Modify the code
! sed -i '8c \\\          os.makedirs("/content/CardiacSegV2/"+data_dir, exist_ok=True)' /content/CardiacSegV2/setup_dir.py
! sed -i '20c \\\          os.makedirs("/content/CardiacSegV2/"+model_exp_dir, exist_ok=True)' /content/CardiacSegV2/setup_dir.py
! sed -i '29c \\\          os.makedirs("/content/CardiacSegV2/"+model_exp_dir, exist_ok=True)' /content/CardiacSegV2/setup_dir.py
! sed -i '3c sys.path.append("/content/CardiacSegV2/")' /content/CardiacSegV2/expers/infer.py
! sed -i '3c sys.path.append("/content/CardiacSegV2/")' /content/CardiacSegV2/expers/tune.py
! sed -i '36c ray.init(runtime_env={"working_dir": \"/content/CardiacSegV2/"})' /content/CardiacSegV2/expers/tune.py
```

```
! python /content/CardiacSegV2/setup_dir.py
```

顯示隱藏的輸出內容

STEP 2: Download Dataset



下載



下載主辦方提供之資料集至自己的colab
(google 雲端連結->colab)

下載訓練資料集

```
!ls
```

CardiacSegV2 Sam sample_data

```
import gdown
import os

gdown.download("https://drive.google.com/u/0/uc?id=1-PrUraWQ38qsSGqZpisvIe1R6N8thYld&export=download", "/content/CardiacSegV2/dataset/chgh/AICUP_training.zip")
! unzip "/content/CardiacSegV2/dataset/chgh/AICUP_training.zip" -d "/content/CardiacSegV2/dataset/chgh/"
```

顯示隱藏的輸出內容

```
!pip install "pydantic<2.0"
```

顯示隱藏的輸出內容

```
gdown.download("https://drive.google.com/u/0/uc?id=1uYoYs7pCPhNjn38SYyexUeT5YooCuimR&export=download", "/content/CardiacSegV2/exps/data_dicts/chgh/AICUP_training.json")
```

Downloading...

```
From: https://drive.google.com/u/0/uc?id=1uYoYs7pCPhNjn38SYyexUeT5YooCuimR&export=download
To: /content/CardiacSegV2/exps/data_dicts/chgh/AICUP_training.json
100%|██████████| 1.76k/1.76k [00:00<00:00, 6.03MB/s]
'/content/CardiacSegV2/exps/data_dicts/chgh/AICUP_training.json'
```

STEP 2: 更改訓練集

在colab內找CardiacSegV2/exps/data_dicts/chgh/AICUP_training.json
並自行修改成想用的data

EX :

測試

```
1 {
2   "test": [
3     {
4       "image": "patient001.nii.gz",
5       "label": "patient001_gt.nii.gz"
6     },
7     {
8       "image": "patient002.nii.gz",
9       "label": "patient002_gt.nii.gz"
10    },
11    {
12      "image": "patient003.nii.gz",
13      "label": "patient003_gt.nii.gz"
14    }
15  ],
```

訓練

```
16   "train": [
17     {
18       "image": "patient011.nii.gz",
19       "label": "patient011_gt.nii.gz"
20     },
21     {
22       "image": "patient012.nii.gz",
23       "label": "patient012_gt.nii.gz"
24     },
25     {
26       "image": "patient013.nii.gz",
27       "label": "patient013_gt.nii.gz"
28     },
29     {
30       "image": "patient014.nii.gz",
31       "label": "patient014_gt.nii.gz"
32     },
33     {
34       "image": "patient015.nii.gz",
35       "label": "patient015_gt.nii.gz"
36     },
37     {
38       "image": "patient016.nii.gz",
39       "label": "patient016_gt.nii.gz"
40     },
41     {
42       "image": "patient017.nii.gz",
43       "label": "patient017_gt.nii.gz"
44     },
45     {
46       "image": "patient018.nii.gz",
47       "label": "patient018_gt.nii.gz"
48     },
49     {
50       "image": "patient019.nii.gz",
51       "label": "patient019_gt.nii.gz"
52     }
53  ],
```

驗證

```
54   "val": [
55     {
56       "image": "patient048.nii.gz",
57       "label": "patient048_gt.nii.gz"
58     },
59     {
60       "image": "patient049.nii.gz",
61       "label": "patient049_gt.nii.gz"
62     },
63     {
64       "image": "patient050.nii.gz",
65       "label": "patient050_gt.nii.gz"
66     }
67   ]
68 }
```

STEP 3: 3D U-Net Training

設置好訓練所需路徑



設定路徑

```
import os

workspace_dir = '/content/CardiacSegV2'
model_name = 'unet3d'
data_name = 'chgh'
exp_name = 'exp_b7_9'
data_dict_file_name = 'AICUP_training.json'

# set exp dir
root_exp_dir = os.path.join(workspace_dir, 'exps', 'exps', model_name, data_name, 'tune_results')

# set data dir
root_data_dir = os.path.join(workspace_dir, 'dataset', data_name)
data_dir = os.path.join(root_data_dir)
# data dict json path
data_dicts_json = os.path.join(workspace_dir, 'exps', 'data_dicts', data_name, data_dict_file_name)

# set model, log, eval dir
model_dir = os.path.join('.', 'models')
log_dir = os.path.join('.', 'logs')
eval_dir = os.path.join('.', 'evals')
os.makedirs(model_dir, exist_ok=True)
os.makedirs(log_dir, exist_ok=True)
os.makedirs(eval_dir, exist_ok=True)

# model path
best_checkpoint = os.path.join(model_dir, 'best_model.pth')
final_checkpoint = os.path.join(model_dir, 'final_model.pth')

# mkdir root exp dir
os.makedirs(root_exp_dir, exist_ok=True)
```

model_name : 模型名稱

data_name : 資料夾位置

exp_name : 實驗名稱

data_dict_file_name : 實驗用json檔

STEP 3: 3D U-Net Training

執行訓練，需等約20~30分鐘



正式訓練

```
# training
!python /content/CardiacSegV2/expers/tune.py \
--tune_mode="train" \
--exp_name={exp_name} \
--data_name={data_name} \
--data_dir={data_dir} \
--root_exp_dir={root_exp_dir} \
--model_name={model_name} \
--model_dir={model_dir} \
--log_dir={log_dir} \
--eval_dir={eval_dir} \
--start_epoch=0 \
--val_every=5 \
--max_early_stop_count=2 \
--max_epoch=20 \
--data_dicts_json={data_dicts_json} \
--pin_memory \
--out_channels=4 \
--patch_size=4 \
--feature_size=48 \
--drop_rate=0.1 \
--depths 3 3 9 3 \
--kernel_size 7 \
--exp_rate 4 \
--norm_name='layer' \
--a_min=-42 \
--a_max=423 \
--space_x=0.7 \
--space_y=0.7 \
--space_z=1.0 \
--roi_x=128 \
--roi_y=128 \
--roi_z=128 \
--optim="AdamW" \
--lr=5e-4 \
--weight_decay=5e-4 \
--checkpoint={final_checkpoint} \
--use_init_weights \
--infer_post_process
```

STEP 4: 3D U-Net Testing

測試訓練結果，需要等約3分鐘



測試訓練結果

```
! sudo chmod 700 /content/CardiacSegV2/expers/tune.py
```

```
# testing
!python /content/CardiacSegV2/expers/tune.py \
--tune_mode="test" \
--exp_name={exp_name} \
--data_name={data_name} \
--data_dir={data_dir} \
--root_exp_dir={root_exp_dir} \
--model_name={model_name} \
--model_dir={model_dir} \
--log_dir={log_dir} \
--eval_dir={eval_dir} \
--data_dicts_json={data_dicts_json} \
--pin_memory \
--out_channels=4 \
--patch_size=4 \
--feature_size=48 \
--drop_rate=0.1 \
--depths 3 3 9 3 \
--kernel_size 7 \
--exp_rate 4 \
--norm_name='layer' \
--a_min=-42 \
--a_max=423 \
--space_x=0.7 \
--space_y=0.7 \
--space_z=1.0 \
--roi_x=128 \
--roi_y=128 \
--roi_z=128 \
--optim="AdamW" \
--lr=5e-4 \
--weight_decay=5e-4 \
--checkpoint={final_checkpoint} \
--use_init_weights \
--infer_post_process \
--resume_tuner \
--save_eval_csv \
--test_mode
```

STEP 4: Download weights

可將前面訓練好的權重(暫存於Colab)
下載到你的電腦裝置



下載權重檔

```
!zip -r /content/CardiacSegV2/exps/exps/unet3d/chgh/tune_results/exp_b7_9.zip /content/CardiacSegV2/exps/exps/unet3d/chgh/tune_results/exp_b7_9
```

顯示隱藏的輸出內容

```
from google.colab import files  
files.download("/content/CardiacSegV2/exps/exps/unet3d/chgh/tune_results/exp_b7_9.zip")
```

04

Colab Baseline 程式說明 (predict)

3D U-Net infer (Easy Baseline)



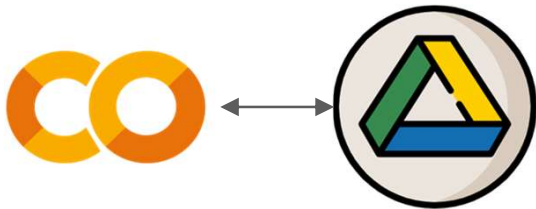
colab連結：

https://colab.research.google.com/drive/1LtvjHWXEJxWIPBvm-kKsTpWprs_qUy-J?usp=sharing

*環境設定和前面訓練一樣

STEP 5(1): 3D U-Net Inferring - Connect Google Drive

將自己的google雲端
掛載至Colab
使資料能夠被複製到個人雲端



```
# Connect Google Drive
from google.colab import drive
drive.mount('/content/drive')
```

要允許這個筆記本存取你的 Google 雲端硬碟檔案嗎？

這個筆記本要求存取你的 Google 雲端硬碟檔案。獲得 Google 雲端硬碟存取權後，筆記本中執行的程式碼將可修改 Google 雲端硬碟的檔案。請務必在允許這項存取權前，謹慎審查筆記本中的程式碼。

不用了，謝謝

連線至 Google 雲端硬碟

✓
36
秒

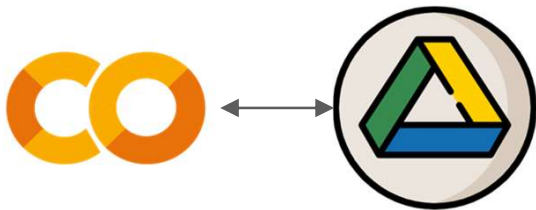
```
[12] # Connect Google Drive
      from google.colab import drive
      drive.mount('/content/drive')
```



Mounted at /content/drive

STEP 5(1): 3D U-Net Inferring - Connect Google Drive

並在自己的google drive中
創建myo_pred/chgh資料夾
和image、infer子資料夾



執行推論 (儲存於Google Drive上)

```
from google.colab import drive
drive.mount('/content/drive')
```

Mounted at /content/drive

```
import os
# create folder in drive
os.makedirs('/content/drive/MyDrive/myo_pred/chgh/image', exist_ok=True)
os.makedirs('/content/drive/MyDrive/myo_pred/chgh/infer', exist_ok=True)
```

STEP 5(1): 3D U-Net Inferring - Store in Colab

設定找tune_model的函式



```
workspace_dir = './content/CardiacSegV2'
import sys
sys.path.append(workspace_dir)
import os
import importlib
from pathlib import PurePath

import pandas as pd
from ray import tune
from ray.train.trainer import BaseTrainer

#from datasets.chgh_dataset import get_data_dicts
#from data_utils.utils import get_pids_by_data_dicts

def get_tune_model_dir(root_exp_dir, exp_name):
    experiment_path = os.path.join(root_exp_dir, exp_name)

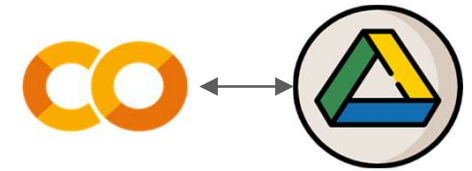
    print(f>Loading results from {experiment_path}...)

    # restored_tuner = tune.Tuner.restore(
    #     experiment_path, trainable=BaseTrainer, resume_errored=True)
    # 727 fix
    restored_tuner = tune.Tuner.restore(experiment_path, trainable='main')
    result_grid = restored_tuner.get_results()

    best_result = result_grid.get_best_result(metric='tt_dice', mode='max')
    print(f"\nBest trial {best_result.metrics['trial_id']}: ")
    print('config:', best_result.metrics['config'])
    print('tt_dice:', best_result.metrics['tt_dice'])
    print('tt_hd95:', best_result.metrics['tt_hd95'])
    if 'esc' in best_result.metrics:
        print('esc:', best_result.metrics['esc'])
    print(f'best log dir:', best_result.log_dir)
    model_dir = os.path.join(best_result.log_dir, 'models')
    return model_dir

def get_tune_dir(exp_dir):
    restored_tuner = tune.Tuner.restore(exp_dir)
    result_grid = restored_tuner.get_results()
    best_result = result_grid.get_best_result(metric='tt_dice', mode='max')
    return best_result.log_dir
```

STEP 5(1): 3D U-Net Inferring - Store in Colab



放置要推論的CT檔在自己的雲端硬碟中
myo_pred/chgh/image內

... > chgh > image ▾

類型 ▾

使用者 ▾

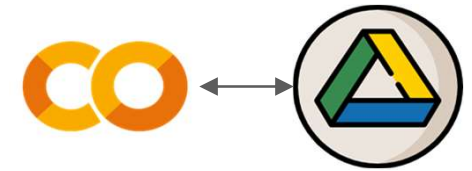
上次修改日期 ▾

來源 ▾

名稱 ↑

patient0001.nii.gz

STEP 5(1): 3D U-Net Inferring - Store in Colab



確認image資料夾內的CT檔

確定Google Drive內容

```
import os
pred_img = []
for root, dirs, files in os.walk("/content/drive/MyDrive/myo_pred/chgh/image", topdown=False):
    for name in files:
        pred_img.append(os.path.join(root, name))
        print(os.path.join(root, name))
```

STEP 5(2): 使用自己訓練好的權重進行推論...

先將自己的權重檔上傳到google drive內

☰ AICUP_training權重.zip 〰️

下午1:19

⋮

共用->共用->一般存取權：選知道連結的任何人->檢視者

一般存取權



知道連結的任何人 ▾

任何知道這個連結的網際網路使用者都能查看

檢視者 ▾

🔗 複製連結

完成

點選複製連結會得到

[https://drive.google.com/file/d/\[位置亂碼\]/view?usp=sharing](https://drive.google.com/file/d/[位置亂碼]/view?usp=sharing)

請把[位置亂碼]複製下來

STEP 5(2): 直接使用訓練好的權重進行推論...

請在[位置亂碼]處放上你剛複製的權重檔位置

下載預訓練檔

```
import gdown
os.makedirs('/content/CardiacSegV2/exps/exps/unet3d/chgh/tune_results/exp_b7_9', exist_ok=True)

#請在[位置亂碼]處放上你的權重檔位置
gdown.download("https://drive.google.com/u/0/uc?id=[位置亂碼]&export=download", "/content/CardiacSegV2/exps/exps/unet3d/chgh/tune_results/exp_b7_9/exp_b7_9.zip")
#或使用我們提供的測試用權重檔
#gdown.download("https://drive.google.com/u/0/uc?id=1CFgvAlHoVgdyDfgKJT232HjkRK04vxBy&export=download", "/content/CardiacSegV2/exps/exps/unet3d/chgh/tune_results/exp_b7_9/exp_b7_9.zip")
! unzip "/content/CardiacSegV2/exps/exps/unet3d/chgh/tune_results/exp_b7_9/exp_b7_9.zip" -d "/content/CardiacSegV2/exps/exps/unet3d/chgh/tune_results/exp_b7_9/"
```

或使用我們提供的測試用權重檔

STEP 5(2): 3D U-Net Inferring - Store in Colab

設定參數



推論設定

```
import os

workspace_dir = '/content/CardiacSegV2'
model_name = 'unet3d' #unet3d
data_name = 'chgh'
#sub_data_dir_name = 'dataset_3'
exp_name = 'exp_b7_9'
data_dict_file_name = 'AICUP_training.json'

# set exp dir
root_exp_dir = os.path.join(workspace_dir, 'exps', 'exps', model_name, data_name, 'tune_results')

# set data dir
root_data_dir = os.path.join(workspace_dir, 'dataset', data_name)
#data_dir = os.path.join(root_data_dir, sub_data_dir_name)
data_dir = os.path.join(root_data_dir)
# data dict json path
data_dicts_json = os.path.join(workspace_dir, 'exps', 'data_dicts', data_name, data_dict_file_name)

# set model, log, eval dir
model_dir = os.path.join('.', 'models')
log_dir = os.path.join('.', 'logs')
eval_dir = os.path.join('.', 'evals')
os.makedirs(model_dir, exist_ok=True)
os.makedirs(log_dir, exist_ok=True)
os.makedirs(eval_dir, exist_ok=True)

# model path
best_checkpoint = os.path.join(model_dir, 'best_model.pth')
final_checkpoint = os.path.join(model_dir, 'final_model.pth')

# mkdir root_exp_dir
os.makedirs(root_exp_dir, exist_ok=True)

import sys
workspace_dir = '/content/CardiacSegV2'
sys.path.append(workspace_dir)

import os
#from experts.infer_utils import get_tune_model_dir, get_data_path, get_pred_path

model_name = 'unet3d' #unet3d
data_name = 'chgh'
exp_name = 'exp_b7_9'

root_exp_dir = os.path.join(workspace_dir, 'exps', 'exps', model_name, data_name, 'tune_results')
model_dir = get_tune_model_dir(root_exp_dir, exp_name)
best_checkpoint = os.path.join(model_dir, 'best_model.pth')
infer_dir = os.path.join('/content/drive/MyDrive/myo_pred', 'chgh', 'infer')
#
```


STEP 5(2): 3D U-Net Inferring - Store in Colab

執行推論

(此時推論結果儲存於自己的雲端硬碟中myo_pred/chgh/infer內)

```
for img_pth in pred_img:
    !python /content/CardiacSegV2/expers/infer.py \
    --model_name={model_name} \
    --data_name={data_name} \
    --data_dir={data_dir} \
    --model_dir={model_dir} \
    --infer_dir={infer_dir} \
    --checkpoint={best_checkpoint} \
    --img_pth={img_pth} \
    --out_channels=2 \
    --patch_size=2 \
    --feature_size=48 \
    --drop_rate=0.1 \
    --depths 3 3 9 3 \
    --kernel_size 7 \
    --exp_rate 4 \
    --norm_name='layer' \
    --a_min=-42 \
    --a_max=423 \
    --space_x=0.7 \
    --space_y=0.7 \
    --space_z=1.0 \
    --roi_x=128 \
    --roi_y=128 \
    --roi_z=128 \
    --infer_post_process
```

... > infer ▾

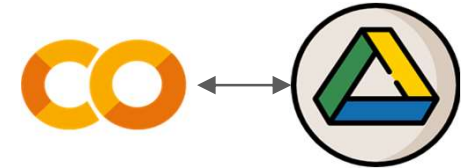
類型 ▾

使用者 ▾

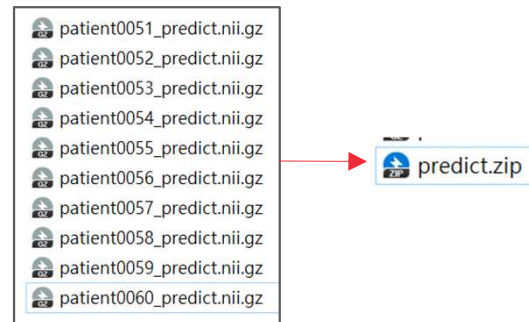
上次修改日期

名稱 ↑

patient0001.nii.gz



把patient0051~patient0100推論完後得到的結果自行儲存成在一個資料夾內，並壓縮成zip



05

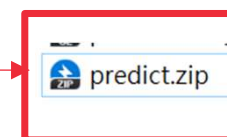
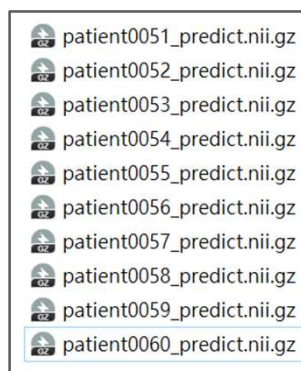
預測成果上傳至競賽網頁

提交檔案



新興智慧顯示科技
教育聯盟

將 merge 完的 predict.zip 檔案提交至趨勢科技 Tbrain 競賽網站。



相關連結

- 1. AICUP網站 : <https://www.aicup.tw/>
- 1. 趨勢科技競賽網站 : <https://tbrain.trendmicro.com.tw/>
- 1. 提問表單 : <https://forms.gle/iPmop91raJNcAUvUA>

END