lesson1

1. 基本介绍

1.1 参加 ML/DL 比赛的目的

中心点：如何在简历中体现出自己的能力从而在千份挑一的简历中脱颖而出？

比赛的排名不重要 !!

重要的是一定要思路清晰的表达出自己的解决方案，兼顾创新性和性能

1.2 入门路径

中心点：如何在发现一个好的 AI 问题 并解决并且 publish 一篇论文？

Dear Authors,

ICTAI 2022 uses a double-blind review process. To facilitate this, please ensure that your names and affiliations do not appear anywhere in the submitted paper.

Best Regards!
ICTAI 2022 Organizing Committee

Important Dates – Extended

Paper Submission: ~~July 31, 2022~~ August 8, 2022
Paper Notification: ~~August 31, 2022~~ September 5, 2022
Camera Ready: ~~September 20, 2022~~ September 26, 2022

最近的会议链接：https://ictai.computer.org/2022

7:34

搜索宝贝

20

宝贝 评价 详情 推荐

2 舒适鼠标

经久耐用 多买多送

直播讲解

【不同配置供你挑选】

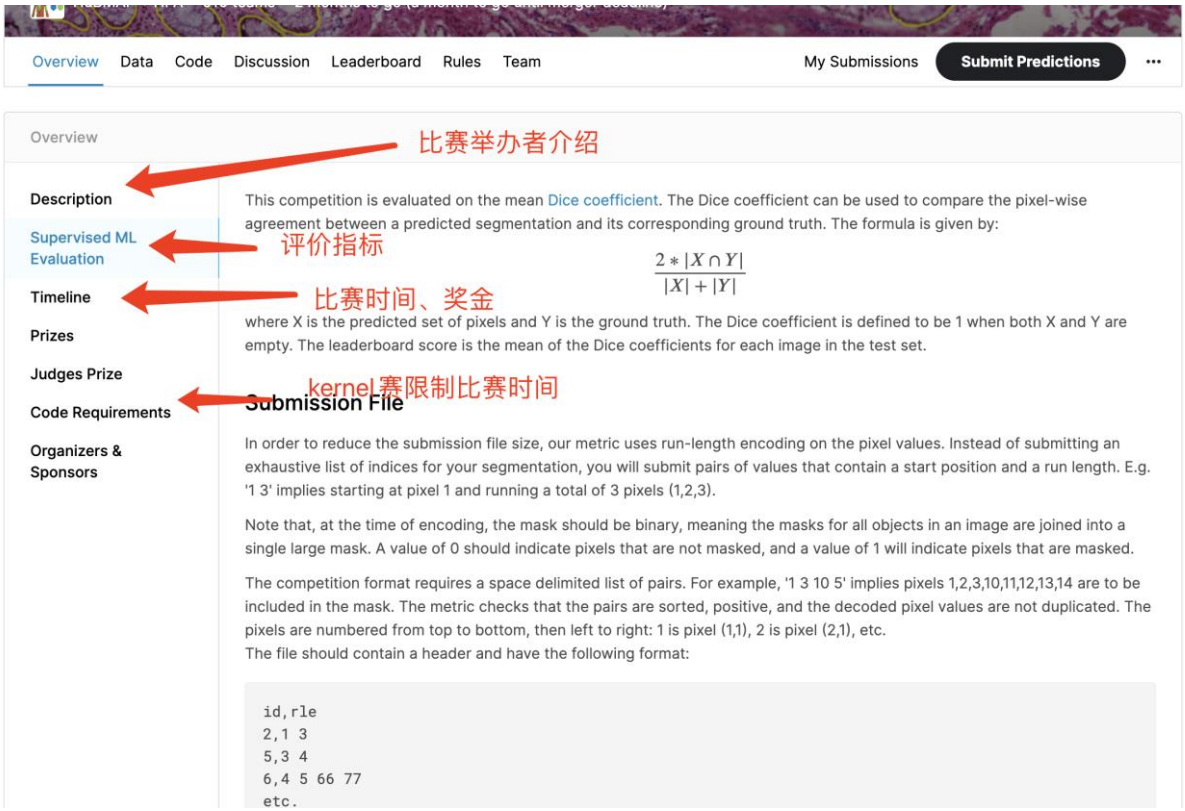
配置	处理器	显卡选择	内存+硬盘	价格
配置一	E5-2680 C2*2 (双路16核32线程)	GTX960M 4G	32G+360G SSD固态	¥2290
配置二	E5-2670 V2*2 (双路20核40线程)	GTX960 4G /GTX850M 8G	48G+360G SSD固态	¥2690
配置三	E5-2680 V2*2 (双路20核40线程)	GTX960 4G /GTX850M 8G	64G+500G NVME M.2固态	¥3190
配置四	E5-2666 V3*2 (双路20核40线程)	GTX1060 5G	64G+500G NVME M.2固态	¥3490
配置五	E5-2673 V3*2 (双路24核48线程)	GTX1060 5G	64G+500G NVME M.2固态	¥3690
配置六	E5-2680 V3*2 (双路24核48线程)	GTX1060 6G	64G+500G NVME M.2固态	¥4090
配置七	E5-2680 V4*2 (双路28核56线程)	GTX1060 6G	64G+500G NVME M.2固态	
配置八	E5-2697 V3*2 (双路28核56线程)	GTX1070 8G	64G+500G NVME M.2固态	¥4790
配置九	E5-2696 V3 *2 (双路36核72线程)	GTX1070 8G	64G+500G NVME M.2固态	¥5690
配置十	E5-2682 V4 *2 (双路32核64线程)	GTX1070 8G	64G+500G NVME M.2固态	¥5890
配置十一	E5-2673 V4 *2 (双路40核80线程)	GTX1070TI 8G	96G+1TB NVME M.2固态	¥6990
配置十二	E5-2696 V4 *2 (双路44核88线程)	RTX2060 12G	128G+1TB NVME M.2固态	¥8590
配置十三	E5-2696 V4 *2	RTX3060 12G	128G+1TB	¥9990

进店逛逛

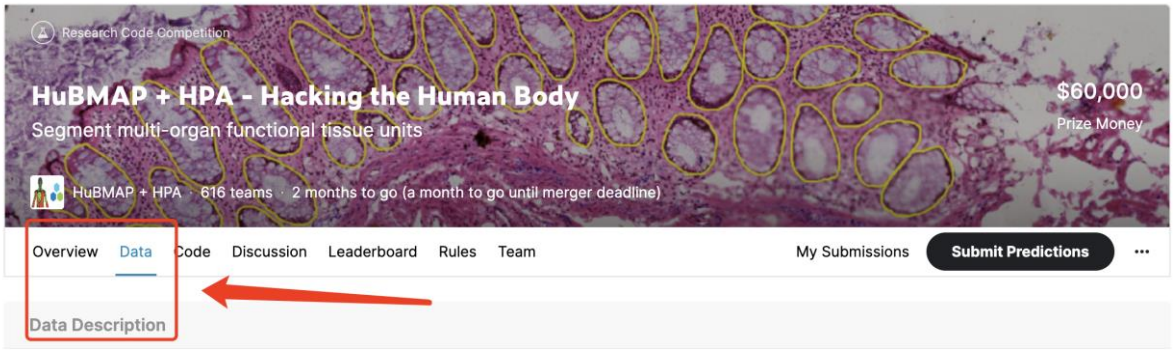
2. Kaggle 平台使用

中心点：Kaggle & 天池的基本介绍

2.1 Overview



2.2 Data



- 比赛数据描述

Files

[train/test].csv Metadata for the train/test set. Only the first few rows of the test set are available for download.

- `id` - The image ID.
- `organ` - The organ that the biopsy sample was taken from.
- `data_source` - Whether the image was provided by HuBMAP or HPA.
- `img_height` - The height of the image in pixels.
- `img_width` - The width of the image in pixels.
- `pixel_size` - The height/width of a single pixel from this image in micrometers. All HPA images have a pixel size of 0.4 µm. For HuBMAP imagery the pixel size is 0.5 µm for kidney, 0.2290 µm for large intestine, 0.7562 µm for lung, 0.4945 µm for spleen, and 6.263 µm for prostate.
- `tissue_thickness` - The thickness of the biopsy sample in micrometers. All HPA images have a thickness of 4 µm. The HuBMAP samples have tissue slice thicknesses 10 µm for kidney, 8 µm for large intestine, 4 µm for spleen, 5 µm for lung, and 5 µm for prostate.
- `rle` - The target column. A run length encoded copy of the annotations. Provided for the training set only.
- `age` - The patient's age in years. Provided for the training set only.
- `sex` - The sex of the patient. Provided for the training set only.

- 比赛如何提交

注意：每场 kaggle 比赛稍有不同

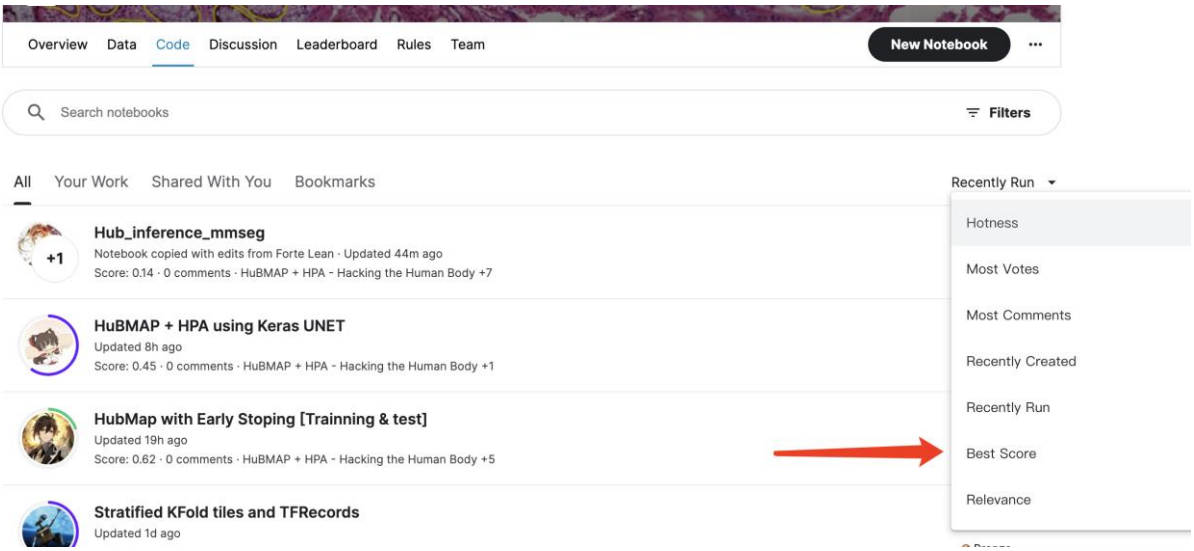
sample_submission.csv

- `id` - The image ID.
- `rle` - A run length encoded mask of the FTUs in the image.

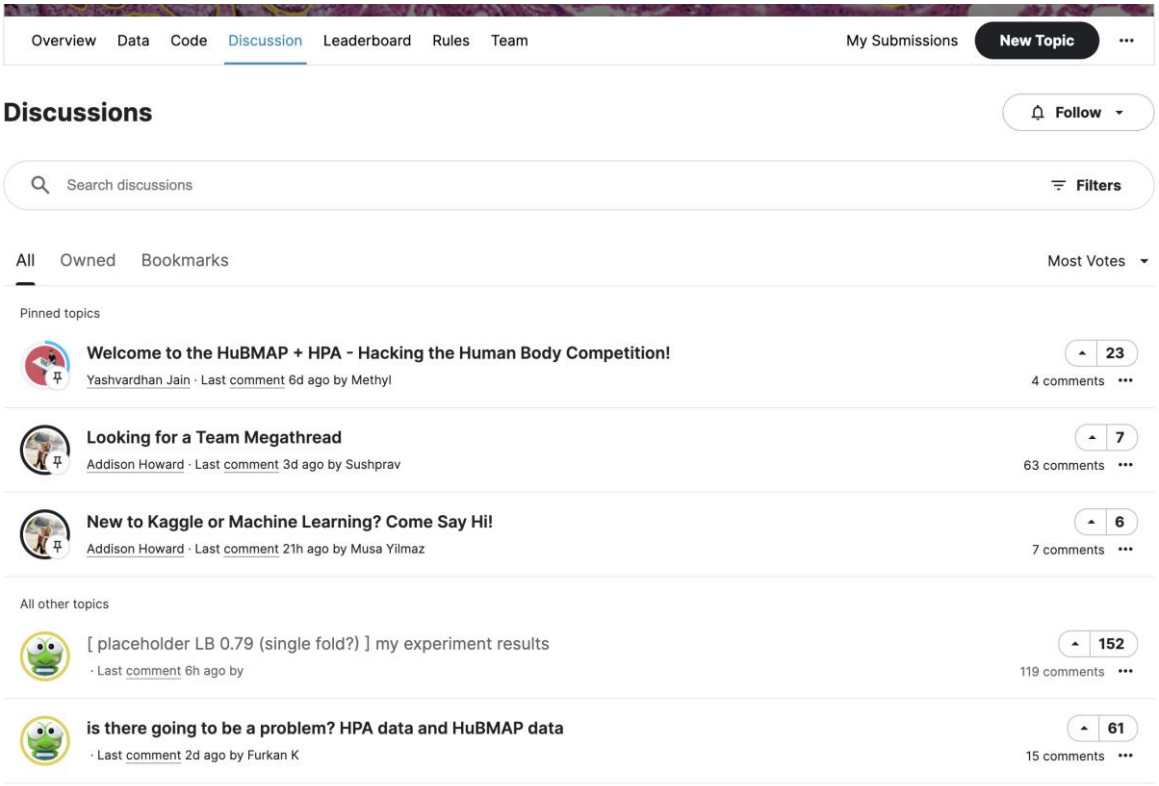
[train/test]_images/ The images. Expect roughly 550 images in the hidden test set. All HPA images are 3000 × 3000 pixels with a tissue area within the image around 2500 × 2500 pixels. The HuBMAP images range in size from 4500×4500 down to 160×160 pixels. HPA samples were stained with antibodies visualized with 3,3'-diaminobenzidine (DAB) and counterstained with hematoxylin. HuBMAP images were prepared using Periodic acid-Schiff (PAS)/hematoxylin and eosin (H&E) stains. All images used have at least one FTU. All tissue data used in this competition is from healthy donors that pathologists identified as pathologically unremarkable tissue.

train_annotations/ The annotations provided in the format of points that define the boundaries of the polygon masks of the FTUs.

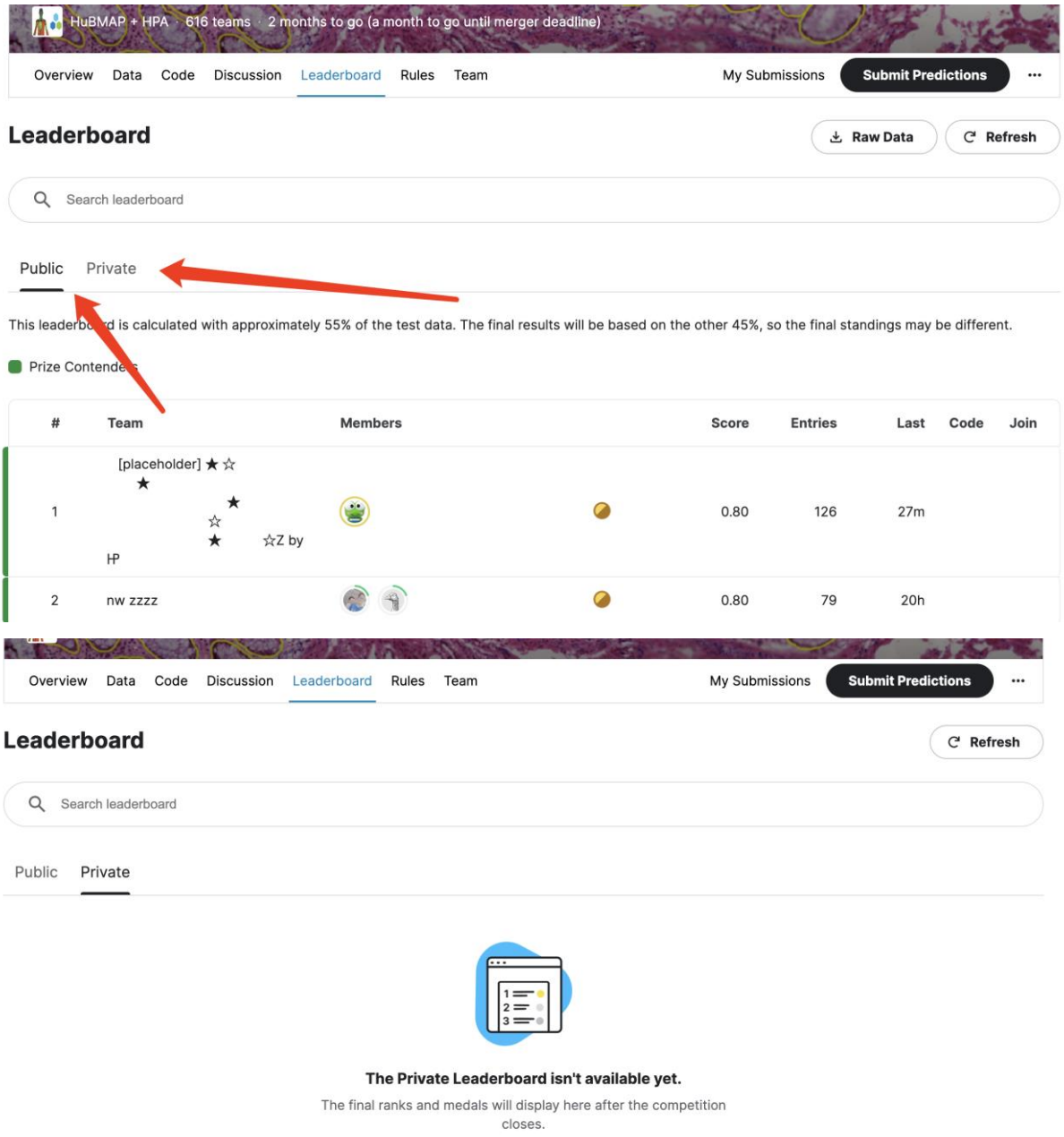
2.3 Code



2.4 Dicussion



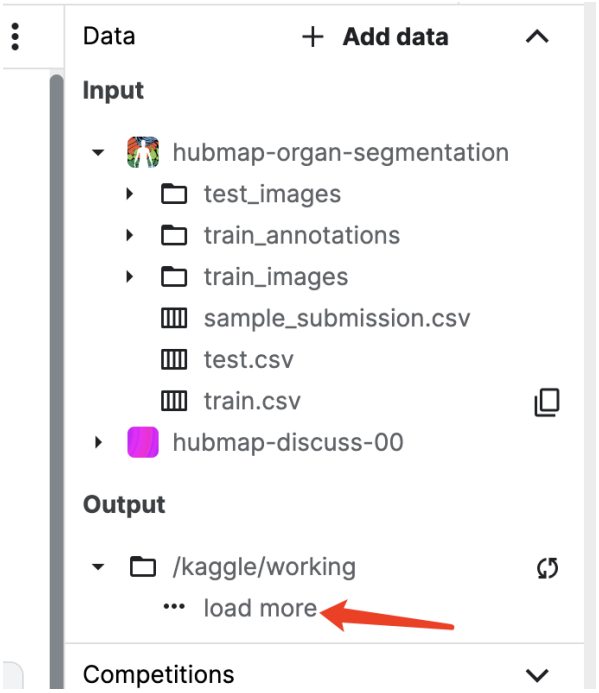
2.5 LeaderBoard



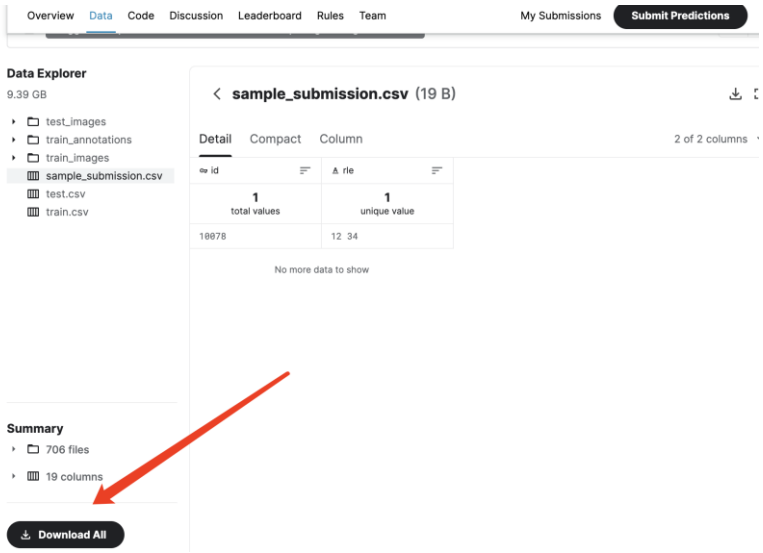
3. Baseline

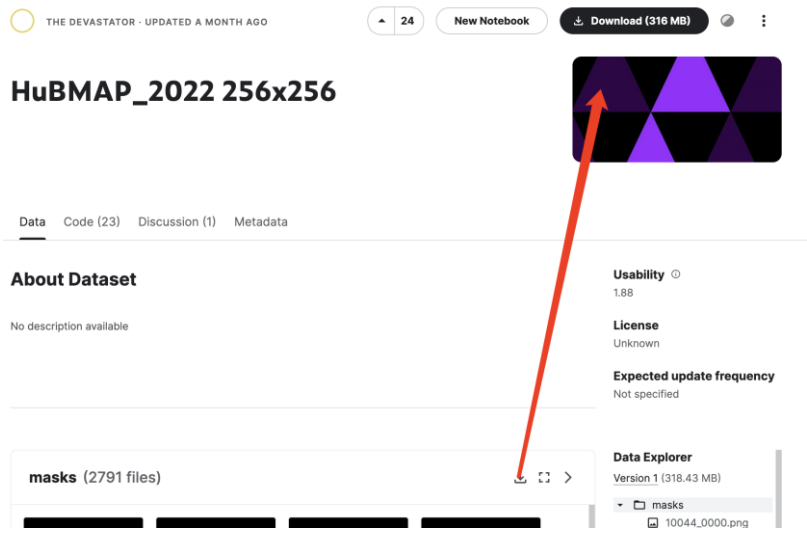
中心点：本次比赛的任务目的. & Baseline 基本结构

3.1 基本结构



本地构造的目录结构





https://www.kaggle.com/datasets/thedevastator/hubmap-2022-256x256

```
(base) ➜  input ls
ckpt-frank            hubmap-2022-256x256.zip    hubmap-organ-segmentation.zip
hubmap-2022-256x256   hubmap-organ-segmentation   tmp
```

```
(base) ➜  working git:(main) ✗ ls
apex  logs    note     submission.csv          train_resnext101.py  utils
data  models  scripts  submit_ensembel_tta.py  train_resnext50.py
```

3.2 训练

```
# step0: 连接服务器
~ ssh frank@10.19.124.245 -p 2022

# step1: 查看 cuda 版本
~ nvcc -V
nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2019 NVIDIA Corporation
Built on Wed_Oct_23_19:24:38_PDT_2019
Cuda compilation tools, release 10.2, V10.2.89

~ ll /usr/local
lrwxrwxrwx  1 root root    9 5月  14 15:14 cuda -> cuda-10.2
```

```
# step2：创建 anaconda 环境 & pytorch
# https://pytorch.org/get-started/previous-versions/
conda create --name python3.8_torch1.8_cuda10.2 python=3.8
conda install pytorch==1.8.1 torchvision==0.9.1 torchaudio==0.8.1
cudatoolkit=10.2 -c pytorch

# step5: 安装 sklearn
# https://scikit-
learn.org/stable/auto_examples/release_highlights/plot_release_highligh
ts_1_1_0.html?highlight=install
pip install --upgrade scikit-learn # 安装最新版

# step7: 安装 opencv
pip install opencv-python
```

缺什么包就安装什么包

(python3.9_torch1.11_cuda11.3) ➜ working git:(main) ✗ sh scripts/run_resnext50.sh

训练好的模型保存在：

```
(python3.9_torch1.11_cuda11.3) ➜  working git:(main) ✗ ll ../input/ckpt-frank/resnext50
总用量 689M
-rw-rw-r-- 1 wanglsh wanglsh 138M 8月   3 11:26 fold0.pth
-rw-rw-r-- 1 wanglsh wanglsh 138M 8月   3 10:37 fold1.pth
-rw-rw-r-- 1 wanglsh wanglsh 138M 8月   3 10:37 fold2.pth
-rw-rw-r-- 1 wanglsh wanglsh 138M 8月   3 10:37 fold3.pth
-rw-rw-r-- 1 wanglsh wanglsh 138M 8月   3 10:37 fold4.pth
```

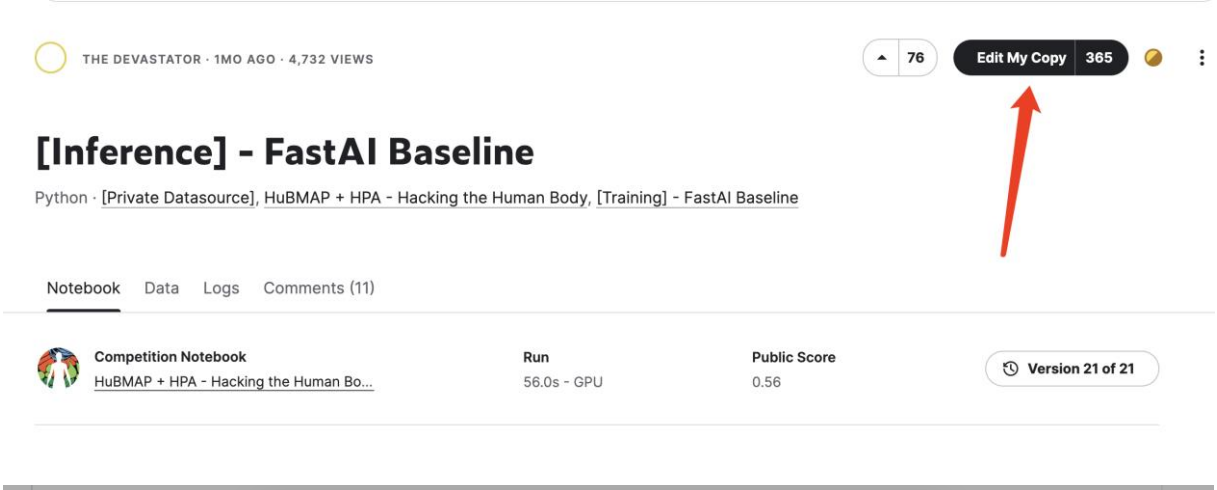
3.3 提交

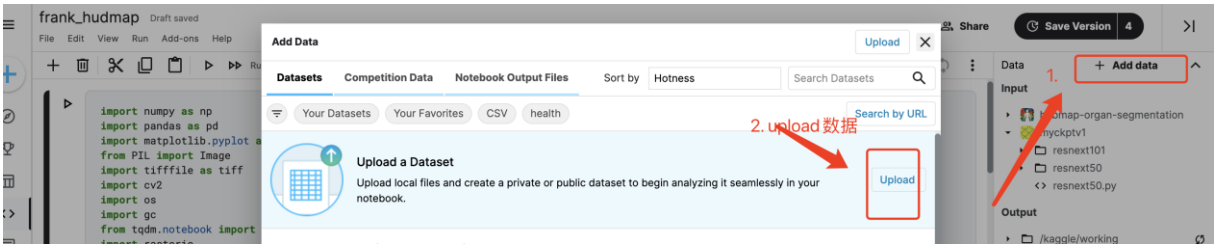
线下验证是否正常运行：

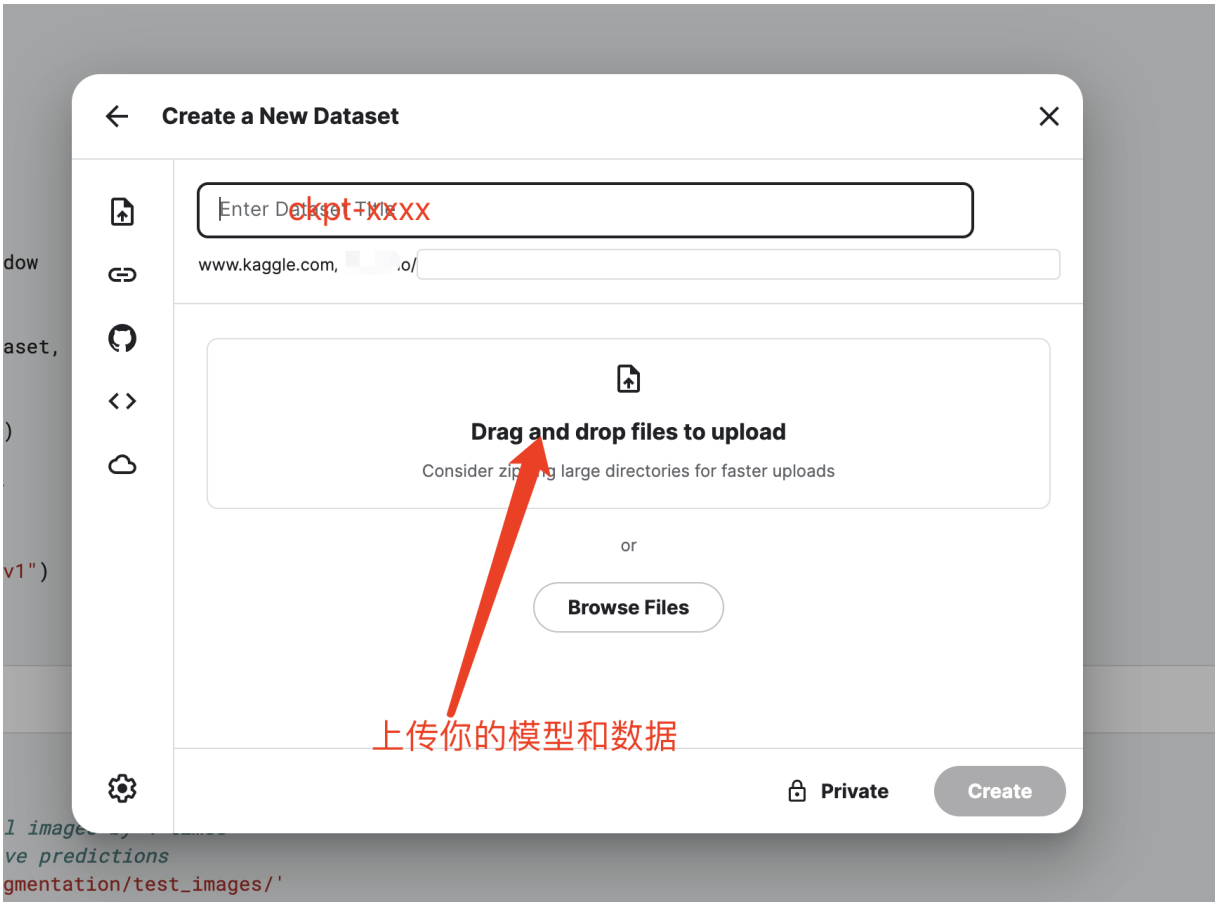
直接 python **submit_ensembel_tta.py .**

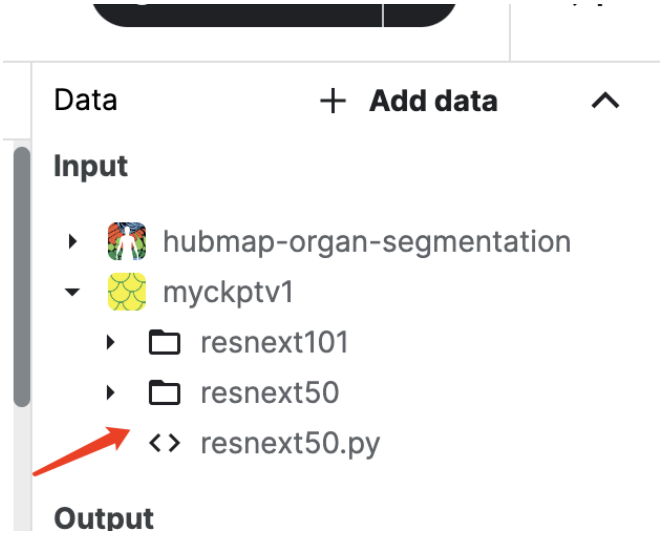
快速提交方法：找到一个类似提交 notebook: 如：

https://www.kaggle.com/code/thedevastator/inference-fastai-baseline


THE DEVASTATOR · 1MO AGO · 4,732 VIEWS
76
Edit My Copy 365
[Inference] - FastAI Baseline
Python · [Private Datasource], HuBMAP + HPA - Hacking the Human Body, [Training] - FastAI Baseline
Notebook
Data
Logs
Comments (11)
Competition Notebook
HuBMAP + HPA - Hacking the Human Bo...
Run
56.0s - GPU
Public Score
0.56
Version 21 of 21


frank_hudmap
Add Data
Upload
Datasets
Competition Data
Notebook Output Files
Sort by
Hotness
Search Datasets
Your Datasets
Your Favorites
CSV
health
Search by URL
2. upload数据
Upload a Dataset
Upload local files and create a private or public dataset to begin analyzing it seamlessly in your notebook.
Upload
Share
Save Version 4
Data
Add data
1.
Input
Output


Create a New Dataset
Enter Dataset Title
okpt-xxxx
www.kaggle.com,
.o/
Drag and drop files to upload
Consider zipping large directories for faster uploads
or
Browse Files
上传你的模型和数据
Private
Create



右边在 input 目录下就有你的各种文件.

此时，粘贴 codebase 里面的 ***submit_ensembel_tta.py 文件***

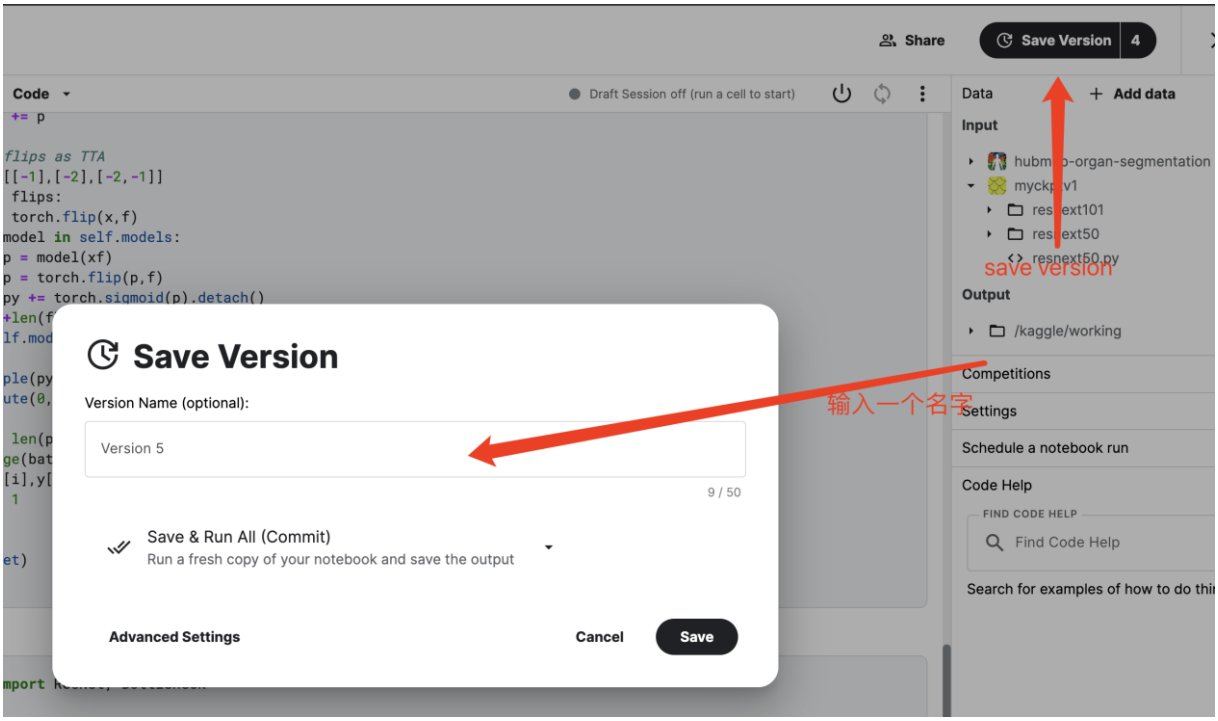
下面全部粘贴即可，但是要注意模型的路径

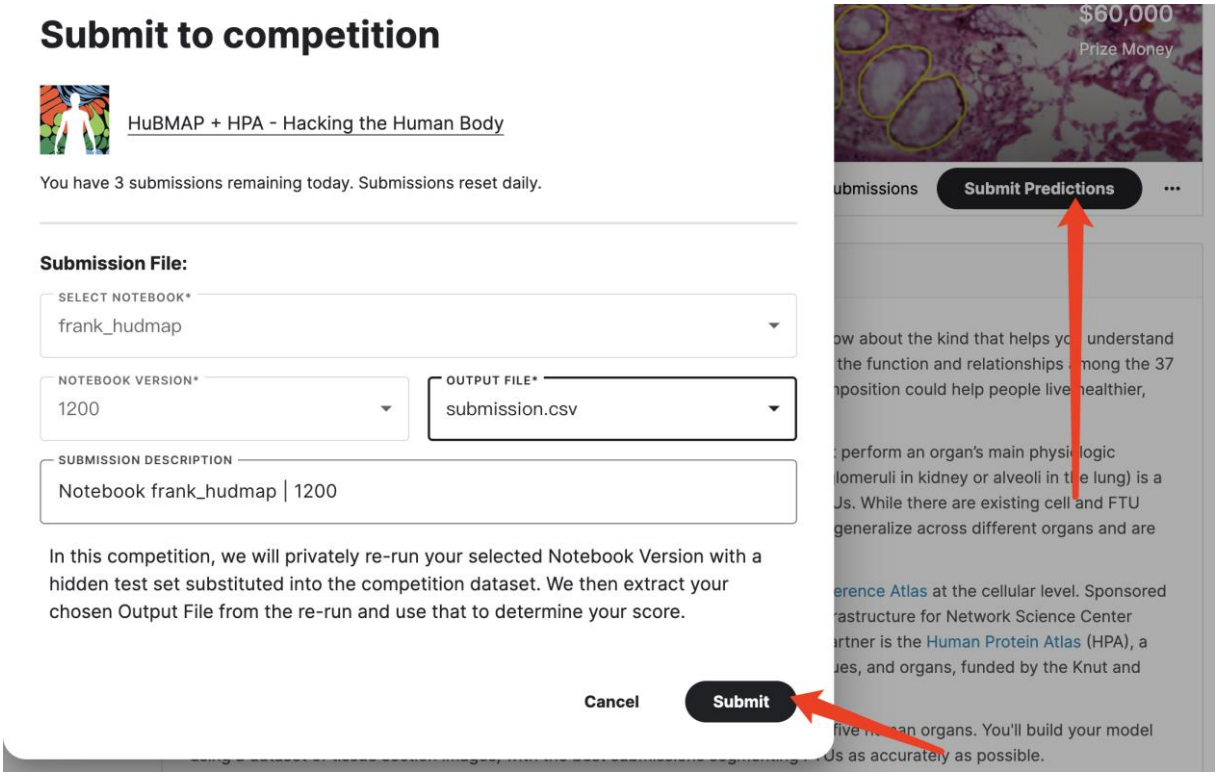
```
[ ]:
bs = 64
sz = 256    # the size of tiles
reduce = 4  # reduce the original images by 4 times
TH = 0.4  # threshold for positive predictions
DATA = '../input/hubmap-organ-segmentation/test_images/'
MODELS_rsxt50 = [f'../input/myckptv1/resnext50/fold{i}.pth' for i in range(5)]
MODELS_rsxt101 = [f'../input/myckptv1/resnext101/fold{i}.pth' for i in range(5)]

df_sample = pd.read_csv('../input/hubmap-organ-segmentation/sample_submission.csv')
device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')


# functions to convert encoding to mask and mask to encoding
def enc2mask(encs, shape):
    img = np.zeros(shape[0]*shape[1], dtype=np.uint8)
    for m,enc in enumerate(encs):
        if isinstance(enc,np.float) and np.isnan(enc): continue
        s = enc.split()
        for i in range(len(s)//2):
            start = int(s[2*i]) - 1
            length = int(s[2*i+1])
            img[start:start+length] = 1 + m
    return img.reshape(shape).T

def mask2enc(mask, n=1):
    pixels = mask.T.flatten()
    encs = []
    for i in range(1,n+1):
        p = (pixels == i).astype(np.int8)
        if p.sum() == 0: encs.append(np.nan)
```





点击 submit 即可，等待成绩即可.

3.2 trick 总结

见 UW lesson 总结 - 当成 wiki 来用

4. To do list

中心点：基本的 Training - 修改模型的能力

4.1 替换 backbone-resnext50/101 为 efficientNet

*参考：Ucode HuBMAP*https://www.kaggle.com/code/yuliknormanowen/ucode-hubmap-efficientnet-submission

Ensemble EfficientNet Results and Submitting The Result

Training

Two ways: train on local machine or on Kaggle

Local Machine

Use dataset HuBMAP (EfficientNet) Train_Local

1. Download dataset, run: bash download.sh
2. Training for fold 0, run: python3 train.py -f 0 Ensemble all 4 folds

Kaggle

Use notebook HuBMAP (EfficientNet) Train

Reference

- HuBMAP fast.ai starter (EfficientNet) by wangkui
- Converting to 256×256 by The Devastator
- [Training] - FastAI Baseline by The Devastator
- [Inference] - FastAI Baseline by The Devastator

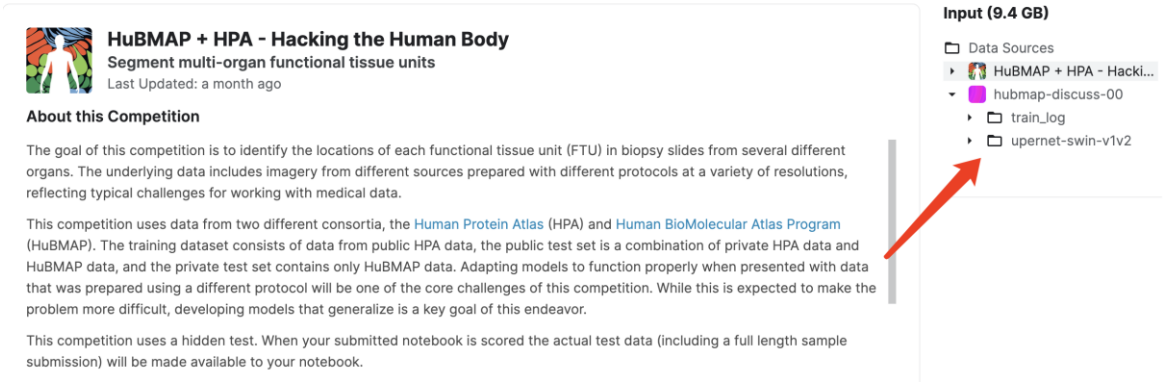
4.2 替换 backbone-resnext50/101 为 swintransformer

参考：[placeholder LB 0.79 (single fold?)] my experiment results

https://www.kaggle.com/competitions/hubmap-organ-segmentation/discussion/332941

https://www.kaggle.com/code/hengck23/lb-0-75-variable-size-swin-transformer-v1-and-v2/data

Data



5. 下节课预告

数据讲解

6.答疑

1

~~1.机器配置~~
~~colab pro 可以完成本次比赛吗，需要什么方法能加速模型训练，调参如何调呢~~
~~2.检测跟分割用的 backbone 一样吗？分割跟农业方向可以怎么结合？需不需要别的技术？~~
~~3.目前的高分方案，是否都是基于 transfomer？efficientnet 还有一战之力吗~~
~~4.colab 国内需要翻墙才能用吗~~
5.enc2mask(encs, shape)这个函数能讲解一下吗

2. 病理图片很大，怎么分 patch 来做了，一直不清楚，目前实际的工作中不分 patch 可能损失信息更多，比如缩小图片大小等，所以老师讲讲 patch 这个思路吧

3. 问题 1.目前分割 sota 前三是什么，大概讲下优缺点或者创新点 2. 打分割比赛常用 tricks

~~4. 有没有深度学习也可以借鉴和使用的特征提取方式，因为之前有类似的比赛纯靠提特征+lgbm 也前三。~~

~~5. mask2former 是否适合本场比赛，为什么，咱们的代码可以修改为使用这个模型吗~~

~~6. 医学图像第一次做，问下常用的图像预处理都有哪些，需要做怎么评估和判断(从效果和理论上都参考)。另外对于不同的图像数据和不同的下层特征提取网络或者任务，选取预处理的方法有什么不一样的地方？~~

~~7. 之前胃肠道的比赛看到有人用 MONAI 分数也不错，老师可以讲一下吗~~

~~8. kaggle 的平台如何查看报名成功，如何组队，后期如何提交任务老师可以讲一下吗~~

~~9. 讲一下 main729 中导入的包在程序中哪个地方用了~~

~~10. 模型融合的代码可以详细讲解下吗，比如 resnet50 和 efficiennetb5 融合~~

~~11. github 开源的语义分割模型众多，每个作者的项目结构都不一样，请问如何有效整合不同的模型，来方便自己使用呢？~~

~~12. 可以讲一下如何修改 smp 那个包里封装的 unet 模型嘛？比如如何引入 ASPP、FPN 等。~~

~~13. 可以讲一下这个课程大体流程吗？比如什么时候以怎样的形式组队等~~

14. 请老师讲一下如何用 HPA pixel_size=0.4 的图像去造 HubMAP 的数据？pixel_size 的变化仅仅通过 resize 大小就可以实现吗？

15. SegFormer 训练太长了，如何加速收敛呢？