MY-SEG使用说明

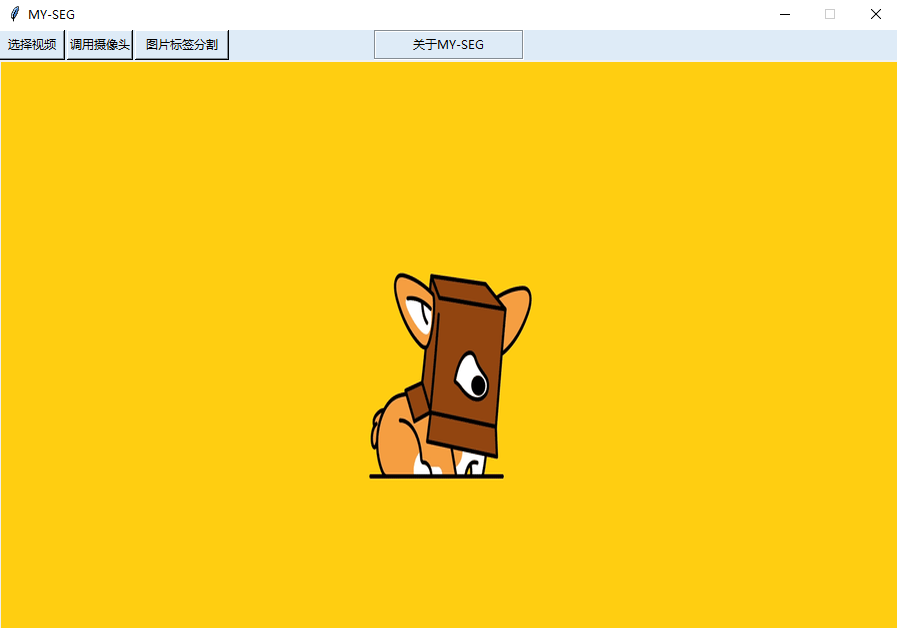
欢迎使用MY-SEG，这是一项集成了VOC数据集上的10000轮与CITYSCAPES数据集上的160K轮的训练模型的用户使用程序，该程序具有图片批量分割，视频分割和摄像头捕捉分割三大功能，下面进行使用说明介绍：

Figure 1

MY-SEG的启动路径文件为：SegFormer-master\front.py。通过运行front程序文件即可启动MY-SEG，出现如图1所示的用户界面窗口。如图中所示，该窗口具有三个按钮功能选项，分别是选择视频，调用摄像头和图片标签分割，以及一个用户反馈按钮“关于MY-SEG”，通过点击用户反馈按钮会出现作者邮箱弹窗，如有任何问题，可以随时向我反馈。

接着进行三大功能的使用介绍：

1. 选择视频功能：

通过点击该按钮，跟着指示窗引导，可以进行视频语义分割的实现，引导窗的示意图如下：

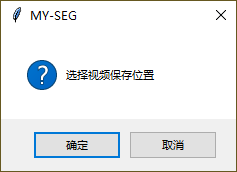
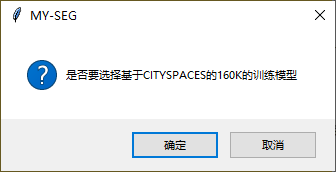
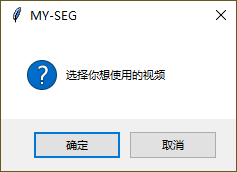
如图2所示，按从左到右的指示窗引导依次进行，在选择想要使用的视频后选择视频要保存的位置，最后进行模型的选择，默认为使用VOC数据集上训练了10000轮的模型，但若在第三个窗口中点击确定，即为选择了CITYSCAPES数据集上的160K训练模型。（大部分为家用笔记本电脑，算力不是很充足，这里建议视频大小在15~20秒左右，否则等待的时间将会较长。），作者尚未加入进入分割阶段缓冲等待的提示动画，不过一旦进入了分割阶段，主界面上的GIF动图一般会卡的不能动，执行完毕又会接着播放，可以以此为判断标准来判断是否程序开始执行分割了，所以出现窗口“未响应”的字样，不必担心，很大概率上是你的电脑正在执行视频分割命令。

Figure 2

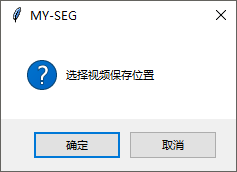
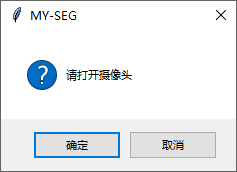
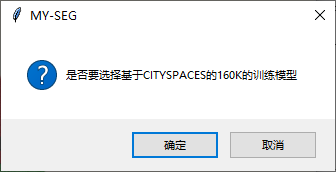
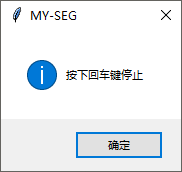
1. 调用摄像头功能：

Figure 3

该部分的指示窗引导序列图如图3所示，选择是否打开摄像头，选择视频保存位置后出现提示用户操作窗口，即用户需通过按下回车键来随时停止摄像头视频捕捉，最后进入模型选择阶段，最后便可开始进行分割。

1. 图片标签分割功能：

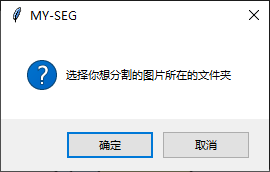
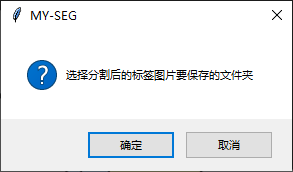
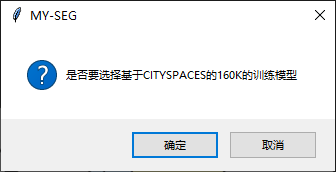
该部分指示窗引导序列图如图4所示，选择想分割的批量图片所在的文件夹，选择分割后的图片要保存的文件夹（这里建议选不一样的文件夹，分割后的图片名称和分割前的图片名称是相同的！），最后进行模型的选择。

Figure 4

至此，MY-SEG功能介绍完毕，祝您有愉快的分割体验！

SEGFORMER源码使用说明

若是您想体验自行利用SEGFORMER训练数据集，接下来将带你一步步解析源码的使用方法。

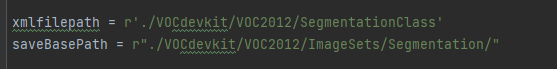
首先是数据集的处理，这里以COCO数据集和VOC数据集为例进行说明：

首先是VOC数据集，（推荐使用，轻量化，大部分笔记本的GPU都能带动！）：

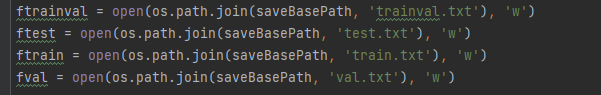
官网下载连接：<http://host.robots.ox.ac.uk/pascal/VOC/voc2012/VOCtrainval_11-May-2012.tar> 复制该连接即可唤醒下载引擎进行下载。

下载后解压的页面如图所示：

这里建议将下载地址选择为：SegFormer-master\tools\data下，下载完毕后打开该目录下的python文件voc.py，运行程序，就会自动为你分割好训练集、测试集和验证集。若您的voc数据集不巧没有下载在这个页面，也不需要担心，同样点开voc.py，如图所示：



将这两个路径文件修改为您所下载的路径即可，处理后的数据集的名称按下图中所示保存在相应的txt文件中，若您没有做任何修改，文件保存路径默认为：VOCdevkit\VOC2012\ImageSets\Segmentation

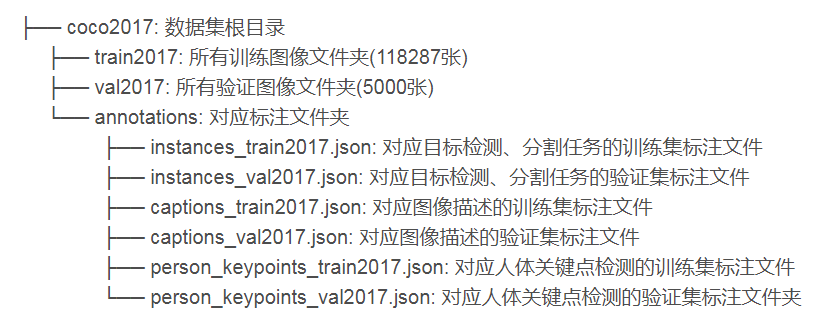


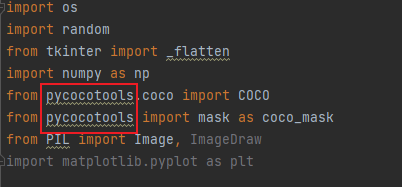
至此voc数据集处理完毕。

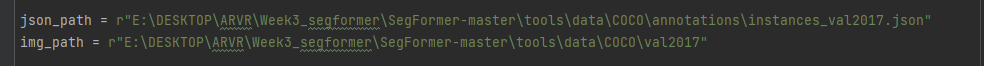
下面介绍COCO数据集的处理（不推荐，数据集过于庞大，一般的笔记本电脑带不动，若是您对自己的配置有信心，可以接着往下看）：

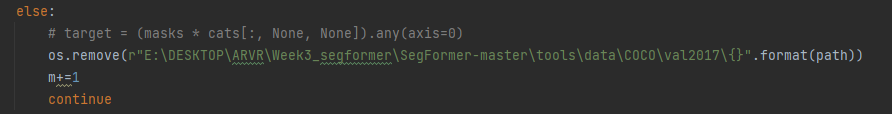
由于官网打开下载较慢，这里参考一篇博客：

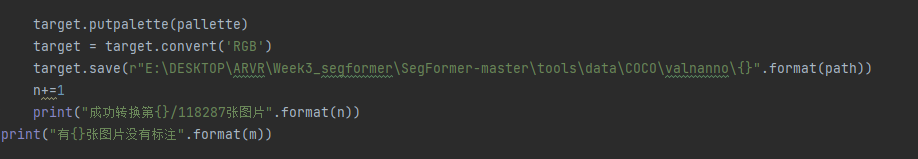
[https://blog.csdn.net/ji\_meng/article/details/124959983?ops\_request\_misc=%257B%2522request%255Fid%2522%253A%2522168739963616800188572815%2522%252C%2522scm%2522%253A%252220140713.130102334..%2522%257D&request\_id=168739963616800188572815&biz\_id=0&utm\_medium=distribute.pc\_search\_result.none-task-blog-2~all~top\_positive~default-1-124959983-null-null.142^v88^insert\_down1,239^v2^insert\_chatgpt&utm\_term=coco%E6%95%B0%E6%8D%AE%E9%9B%86%E4%B8%8B%E8%BD%BD&spm=1018.2226.3001.4187](https://blog.csdn.net/ji_meng/article/details/124959983?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522168739963616800188572815%2522%252C%2522scm%2522%253A%252220140713.130102334..%2522%257D&request_id=168739963616800188572815&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~top_positive~default-1-124959983-null-null.142%5ev88%5einsert_down1,239%5ev2%5einsert_chatgpt&utm_term=coco%E6%95%B0%E6%8D%AE%E9%9B%86%E4%B8%8B%E8%BD%BD&spm=1018.2226.3001.4187) 读者可通过该大佬提供的下载链接自行进行下载。下载后的文件目录如下图所示：

COCO数据集的处理比较复杂，他不像VOC那样图片文件和标签文件是分开的两个文件夹，且保存形式均为jpg或png的形式。COCO数据集的原图片位置在train2017文件夹中，其中有118287张训练图片，但这些训练图片的对应标签文件并不是以文件夹+图片的形式保存的，而是以json文件保存的，所以我们要先把标签图片读取出来。下载完成后，打开SegFormer-master\tools\data路径下的coco.py，执行程序即可实现标签图片的读取，该过程执行时间较长，耐心等待。若是执行报错，下面进行详细介绍如何修改使用coco.py。



如图所示，最常见的报错是没有安装对应的库，图中红框的pycocotools是专门为了读取COCO数据集所创建的第三方库，使用pip install下载即可。

如上图所示，读者可自行修改路径，json\_path路径即为标签文件json路径，img\_path路径为与标签相对应的图片文件路径。如图中所示，val2017是图片文件夹的名称，它所对应的标签保存在instances\_val2017.json文件中，需要把它提取出来。

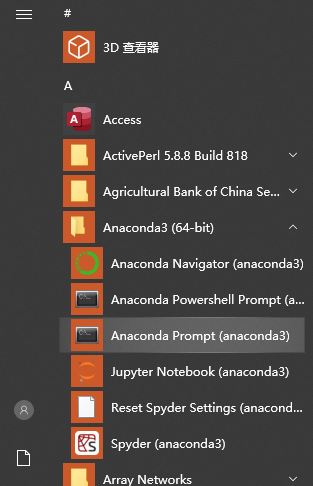
如上图所示，这里操作的含义是对原图片文件夹中没有对应标签的图片去除，这也是COCO数据集比较麻烦的地方，它的原图片中不是每一张图片都有对应的标签，比如train2017数据集中的118287张训练图片大概有几千张是没有标签的，所以这里做了去除处理。

读者可以通过修改这里的路径，自行选择建立标签文件夹的位置。

通过这一步骤的处理后，COCO数据集还是不能正常运行，经过我的相关查询，语义分割的标签图片的位深度需要为8，而上一步中处理后的图片位深度为24。所以我们打开SegFormer-master\tools\data路径下的COCO3.py文件运行处理位深度，至此，标签图片文件才算生成完毕。（坚持一下，虽然真的很不容易）

最后打开SegFormer-master\tools\data路径下的COCO2.py文件，执行程序，这个文件的原理与voc.py的原理相同，即进行训练集，测试集，验证集的划分并保存为txt文件。至此COCO数据集处理完毕。

下面介绍SEGFORMER的使用说明，这里以VOC数据集为例：

首先进行环境的配置，强烈建议按着接下来的步骤一步步走，配置一个新的环境！

打开电脑上的anaconda，按照下列步骤配置新的虚拟环境（这里参考了网上一篇大佬的博客，跟着步骤走！一般不会报错！）：

conda create -n segformer python=3.8

conda activate segformer

conda install pytorch=1.8 torchvision cudatoolkit=10.2 -c pytorch

pip install timm==0.3.2 IPython cython matplotlib opencv-python==4.5.1.48

接下来要安装mmcv库，近年来才刚推出了支持windows的mmcv库下载安装，且版本有限，若是您按照上面的步骤创建环境，cuda的版本为10.2，torch的版本为1.8，即可点击下面的链接：<https://download.openmmlab.com/mmcv/dist/cu102/torch1.8.0/index.html>

进行windows版mmcv库的安装文件下载，我这里使用的是mmcv\_full-1.4.5-cp38-cp38-win\_amd64.whl文件，即mmcv的版本号为1.4.5，建议在这左右即可，太高的版本反而可能无法在segformer源码中运行（segformer原作者推荐1.2.7但我并没有找到对应的安装包，1.4.5也能使用。）

如果你的cuda的版本和torch的版本与上文不一致，没有按照上述命令来配置，也可以修改网址<https://download.openmmlab.com/mmcv/dist/cu102/torch1.8.0/index.htm>

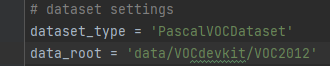
中的cu102和torch1.8.0来搜索是否有对应版本的mmcv下载链接。

将下载后的mmcv\_full-1.4.5-cp38-cp38-win\_amd64.whl文件放在主目录下（SegFormer-master\），打开python进入terminal，执行如下图所示的命令：

注意是否是在segformer虚拟环境下配置，pip install的文件名称即为下载的文件名称。

至此mmcv库安装完毕。

接着进行如何训练voc数据集的步骤介绍：

1、local\_configs/\_base\_/datasets/pascal\_voc12.py修改data\_root

如图所示，dataset\_type不要乱改，后面会详细介绍。

2、mmseg/datasets/voc.py修改 CLASSES PALETTE，CLASSES为类别名称，PALETTE为类别对应的色盘，这里已经帮你修改好了，建议不要乱改。

3. mmseg/models/decode\_heads/segformer\_head.py修改第59行 SyncBN->BN

4、tool/train修改--config --work-dir（这里是训练日志以及模型输出的保存位置）。

5、local\_config/\_base\_/models文件夹下对应的segformer.py修改norm\_cfg SyncBN->BN

修改num\_classes（VOC数据集共有21类，就修改为21）。

6、local\_config/segformer/B0文件夹下对应的在train设置的--config模型

如图所示为我所选择的config文件，读者可以自行修改，具体路径为SegFormer-master\local\_configs\segformer\下，共有B0-B5多种选择，B0的参数量最小，最适合实时分割，B5的参数量最大，分割效果也最好。

修改norm\_cfg SyncBN->BN

修改所有num\_classes

至此基本配置完毕，模型可以开始训练。

当然也可能不是那么顺利，下面介绍几种常见的报错以及解决办法：

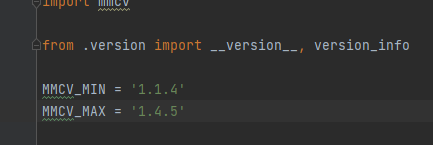
**报错1：**

**No module named ‘mmcv.\_ext‘**

出现这个报错就说明你没有按上面的配置一步一步进行，老老实实回去配置，该报错出现的原因是因为你的mmcv库和torch以及cuda版本不兼容。

**报错2：**

**版本超出限制报错**

打开路径SegFormer-master\mmseg\\_\_init\_\_.py，可以看到以下页面：

在这里修改最小版本和最大版本限制，包含你所下载的mmcv版本即可。

**报错3：**

**ImportError: DLL load failed while importing \_imaging: 找不到指定的模块。**

这个报错产生的原因是安装包的冲突，依次执行以下命令即可解决：

pip install Pillow

pip uninstall Pillow

pip install Pillow

即重新安装Pillow库，即可解决。

**报错4：**

**Cuda out of memory**

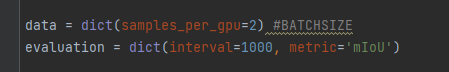
GPU的容量不够啦，后面将详细介绍训练参数的设置。

若是还有其它未涉及的报错，[可以发邮件至1509435185@qq.com](mailto:可以发邮件至1509435185@qq.com)，尽量帮你解决哈。

训练参数配置

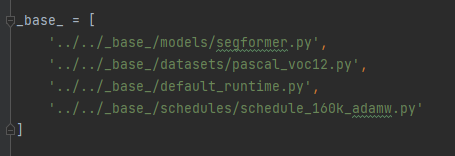
打开之前训练选择的config文件

SegFormer-ster\local\_configs\segformer\B0\segformer.b0.512x512.ade.160k.py

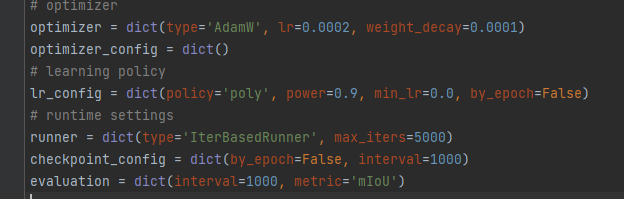
在第50行，如下图所示

samples\_per\_gpu即为batchsize，若是GPU的容量不够报错，修改这里即可，建议为2或4，如果改为1都运行不了，那么很遗憾，电脑配置不行就是运行不了；对自己电脑配置有信心的可以适当提高。

Evaluation的interval是设置按照一定的轮数间隔输出目标模型，这里设置为1000，表示每1000轮输出一次训练模型。

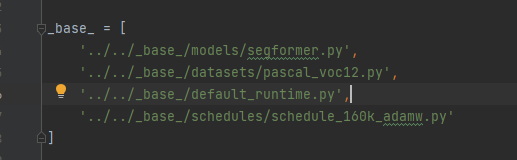


同样的在该config页面，可以看到如上图所示的base配置，按照上面的路径打开schedule\_160k\_adamw.py文件。

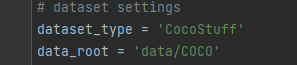


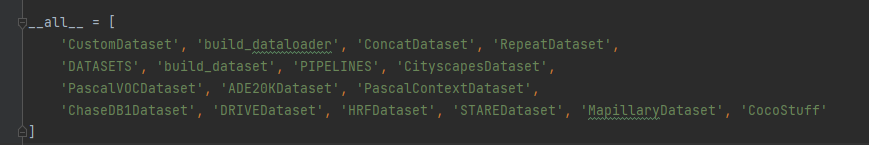
如图所示，max\_iters表示训练轮数，这里为5000轮，读者可以自行修改。对于optimizer中的数值也可以根据需要自行修改，lr表示学习率。

若是想要使用别的数据集进行训练要怎么修改呢？



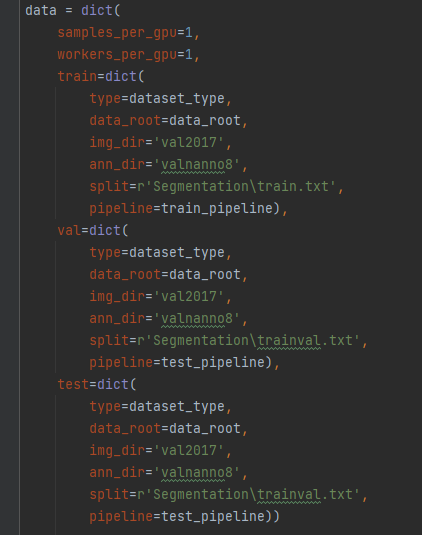
首先挑选config文件，并进入进行base的修改，需要修改datasets的文件名称，若是没有想要的数据集文件，自行增改即可，照猫画虎就行，以coco数据集为例，在SegFormer-master\local\_configs\\_base\_\datasets\路径下创建coco.py。



修改dataset\_type和data\_root即可，对于dataset\_type，如下图所示

源码中目前支持图中数据集的训练，图中有COCOSTUFF，所以可以使用，文件路径在SegFormer-master\mmseg\datasets\\_\_init\_\_.py中。

回到coco.py页面进行下列修改：



如图所示，在train、val、test、的文件dir中分别修改为之前处理过的文件夹路径，img\_dir表示原图片文件夹，ann\_dir表示标签文件夹，split为处理后生成的txt文件读取路径，即根据该txt文件进行数据集，测试集，验证集的划分。

色盘的修改路径在SegFormer-master\mmseg\datasets下，例如我要修改coco数据集的类别和色盘就进入该路径下的cocostuff.py文件进行修改即可。

至此，介绍完毕，祝您有愉快的训练体验，如有任何疑问，[请咨询1509435185@qq.com](mailto:请咨询1509435185@qq.com)。