1. 配置环境

* 方法一，直接运行程序，缺少哪些包，直接pip install就可以
* 方法二，在程序中提供了unext.yaml，里面包含了所需的环境，你可以根据该文件进行包的载入。
* 方法三，我们也给出了相应的环境压缩包unext.zip,你可以解压该文件夹，将里面的文件粘贴到${conda的根路径}\ envs的文件夹中。

1. 使用程序之前，先删除data文件夹中的文件。可以运行delete\_data\_file.py完成此任务
2. 半自动化语义分割标注——方法一

图示, 日程表

描述已自动生成

**第一步，由三个部分组成：**

* **config/config\_v1.py文件中Config类的all\_sample\_reprocess属性设为False，表示每一轮循环无需重新预处理所有文件，只需处理其中新加入的文件。**
* **从病理图像的原始文件夹中挑选出一部分病理图像（jpg文件），将其导入到QuPath之中进行标注。标注完成之后，将标注结果导出，保存为Json文件。**
* **将被挑选出的病理图像所在的文件夹的绝对路径赋值给config/config\_v1.py文件中Config类的img\_dir属性；将被挑选出的病理图像对应的标注文件（json文件）所在的文件夹的绝对路径赋值给config/config\_v1.py文件中Config类的json\_dir属性；**

**第二步，运行run.py文件，即可以等待模型训练完成。训练完成之后，其训练好的模型权重被保存在config/config\_v1.py文件中Config类的exp\_name属性所指定的地址之中。在该文件夹之中，首先是一个时间文件夹，记录开始训练的时间。打开此文件夹，会有五个子文件夹，分别代表五折交叉实验中每一次的模型训练结果**。

第二步详解：由三个部分组成：

* 运行run.py文件之后，首先会调用utils/from\_maskJson\_to\_maskJpg.py中的from\_maskJson\_to\_maskJpg方法。该方法可以将json标注文件转化成jpg标注文件，并保存到config/config\_v1.py文件中Config类的jpg\_save\_dir属性所指定的地址之中。
* 然后会调用utils/split\_wsi\_to\_patch.py中的split\_wsi\_to\_patch方法。该方法可以将训练样本从wsi级别分割成patch级别，分割方式为网格分割，分割结果分别保存到config/config\_v1.py文件中Config类的IMG\_SAVE\_DIR属性（保存分割之后的病理图像png文件）和MASK\_SAVE\_DIR属性（保存分割之后的标注npy文件）所指定的地址之中。
* 最后调用train.py文件中的train方法，该方法利用给定的标注样本，对模型进行训练，并将训练好的模型进行存储。（config/config\_v1.py文件中Config类中有许多关于训练的超参数，可以对训练进行调整）

**第三步，从病理图像库中选出几张未被标注的病理图像，将这几张未被标注的病理图像单独放在一个文件夹，然后该文件夹的绝对路径赋值给config/config\_v1.py文件中Config类的predict\_img\_dir属性。**

**第四步，由两个部分组成：**

* **将第二步训练好的模型权重所在路径赋值给config/config\_v1.py文件中Config类的predict\_ckpt\_path属性。**
* **运行predict.py文件，进行语义分割预测。预测分割结果分别被保存到config/config\_v1.py文件中Config类的predict\_json\_save\_dir属性（存储预测的语义分割结果json文件的所在文件夹位置）和predict\_jpg\_save\_dir属性（存储预测的语义分割结果jpg文件的所在文件夹位置）所指定的地址之中。**

**第五步，将模型预测语义分割结果json文件与被预测的病理图像导入到QuPath之中，对预测语义分割结果进行判定与修正：**

* **如果预测语义分割结果无需人工进行修正，说明模型已经训练完成。**
* **如果预测语义分割结果需要人工进行修正，则将修正结果导出为json文件，并且将其放到config/config\_v1.py文件中Config类的json\_dir属性所指定的文件夹之中。同时将被预测的病理图像放到config/config\_v1.py文件中Config类的img\_dir属性所指定的文件夹之中。返回到第二步。**

1. 半自动化语义分割标注——方法二

**第一步，由三个部分组成：**

* **config/config\_v1.py文件中Config类的all\_sample\_reprocess属性设为True，表示每一轮循环需要重新预处理所有文件。**
* **将病理图像导入到QuPath之中进行粗略标注。标注完成之后，将标注结果导出，保存为Json文件。**
* **将被挑选出的病理图像所在的文件夹的绝对路径赋值给config/config\_v1.py文件中Config类的img\_dir属性；将被挑选出的病理图像对应的标注文件（json文件）所在的文件夹的绝对路径赋值给config/config\_v1.py文件中Config类的json\_dir属性；**

**第二步，运行run.py文件，即可以等待模型训练完成。训练完成之后，其训练好的模型权重被保存在config/config\_v1.py文件中Config类的exp\_name属性所指定的地址之中。在该文件夹之中，首先是一个时间文件夹，记录开始训练的时间。打开此文件夹，会有五个子文件夹，分别代表五折交叉实验中每一次的模型训练结果**。

第二步详解：由三个部分组成：

* 运行run.py文件之后，首先会调用utils/from\_maskJson\_to\_maskJpg.py中的from\_maskJson\_to\_maskJpg方法。该方法可以将json标注文件转化成jpg标注文件，并保存到config/config\_v1.py文件中Config类的jpg\_save\_dir属性所指定的地址之中。
* 然后会调用utils/split\_wsi\_to\_patch.py中的split\_wsi\_to\_patch方法。该方法可以将训练样本从wsi级别分割成patch级别，分割方式为网格分割，分割结果分别保存到config/config\_v1.py文件中Config类的IMG\_SAVE\_DIR属性（保存分割之后的病理图像png文件）和MASK\_SAVE\_DIR属性（保存分割之后的标注npy文件）所指定的地址之中。
* 最后调用train.py文件中的train方法，该方法利用给定的标注样本，对模型进行训练，并将训练好的模型进行存储。（config/config\_v1.py文件中Config类中有许多关于训练的超参数，可以对训练进行调整）

**第三步，将病理图像库所在文件夹的绝对路径赋值给config/config\_v1.py文件中Config类的predict\_img\_dir属性。**

**第四步，由两个部分组成：**

* **将第二步训练好的模型权重所在路径赋值给config/config\_v1.py文件中Config类的predict\_ckpt\_path属性。**
* **运行predict.py文件，进行语义分割预测。预测分割结果分别被保存到config/config\_v1.py文件中Config类的predict\_json\_save\_dir属性（存储预测的语义分割结果json文件的所在文件夹位置）和predict\_jpg\_save\_dir属性（存储预测的语义分割结果jpg文件的所在文件夹位置）所指定的地址之中。**

**第五步，将模型预测语义分割结果json文件与病理图像导入到QuPath之中，对预测语义分割结果进行判定与粗略修正：**

* **如果预测语义分割结果无需人工进行修正，说明模型已经训练完成。**
* **如果预测语义分割结果需要人工进行修正，则将粗略修正结果导出为json文件，并且将其放到config/config\_v1.py文件中Config类的json\_dir属性所指定的文件夹之中。返回到第二步。**