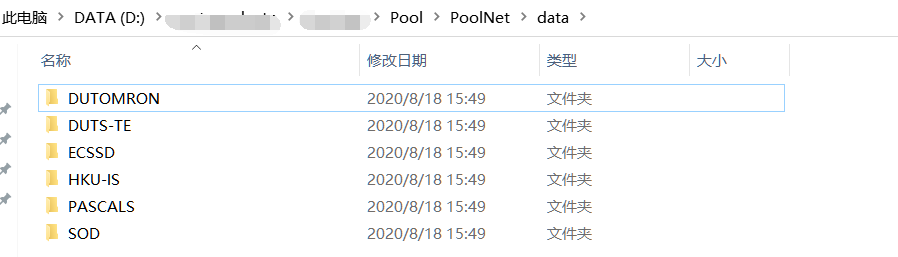
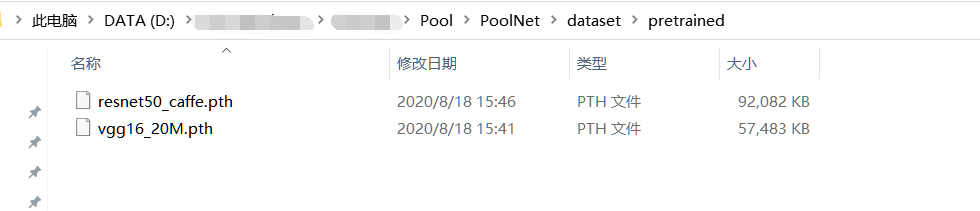
A Simple Pooling-Based Design for Real-Time Salient Object Detection复现

1. 需求分析
   1. 根据Github提供内容，本文实际代码需0.4.1版本及以上的PyTorch和相应的torchvision，对于其他库的要求并没有提及。
   2. 本文在下图六种不同数据集上进行了训练，Github上提供了缩水版的测试用数据集，可以直接下载使用。

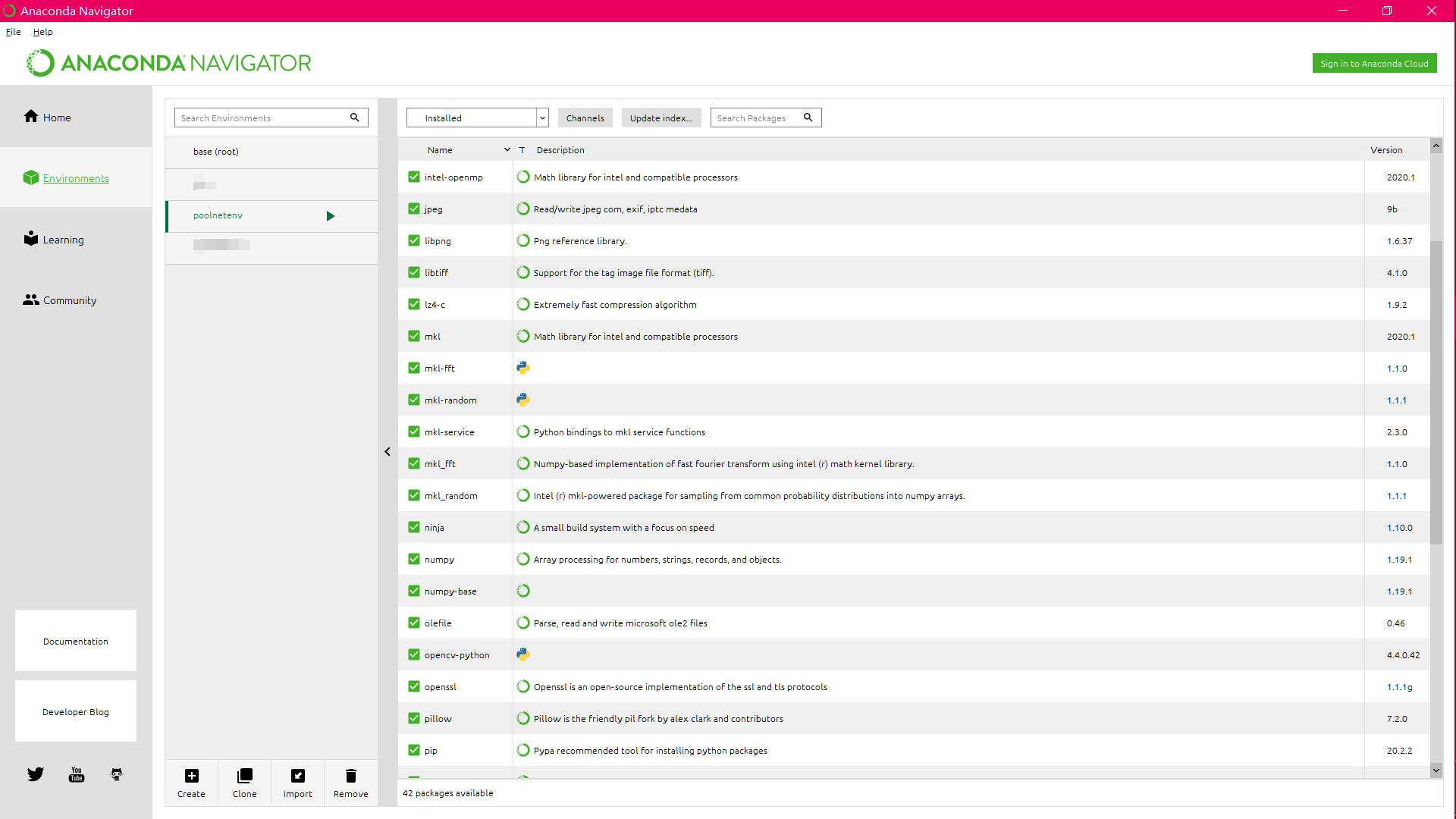


* 1. 此外，本文以训练好的Resnet50和VGG16作为骨干网络，可以直接下载使用。



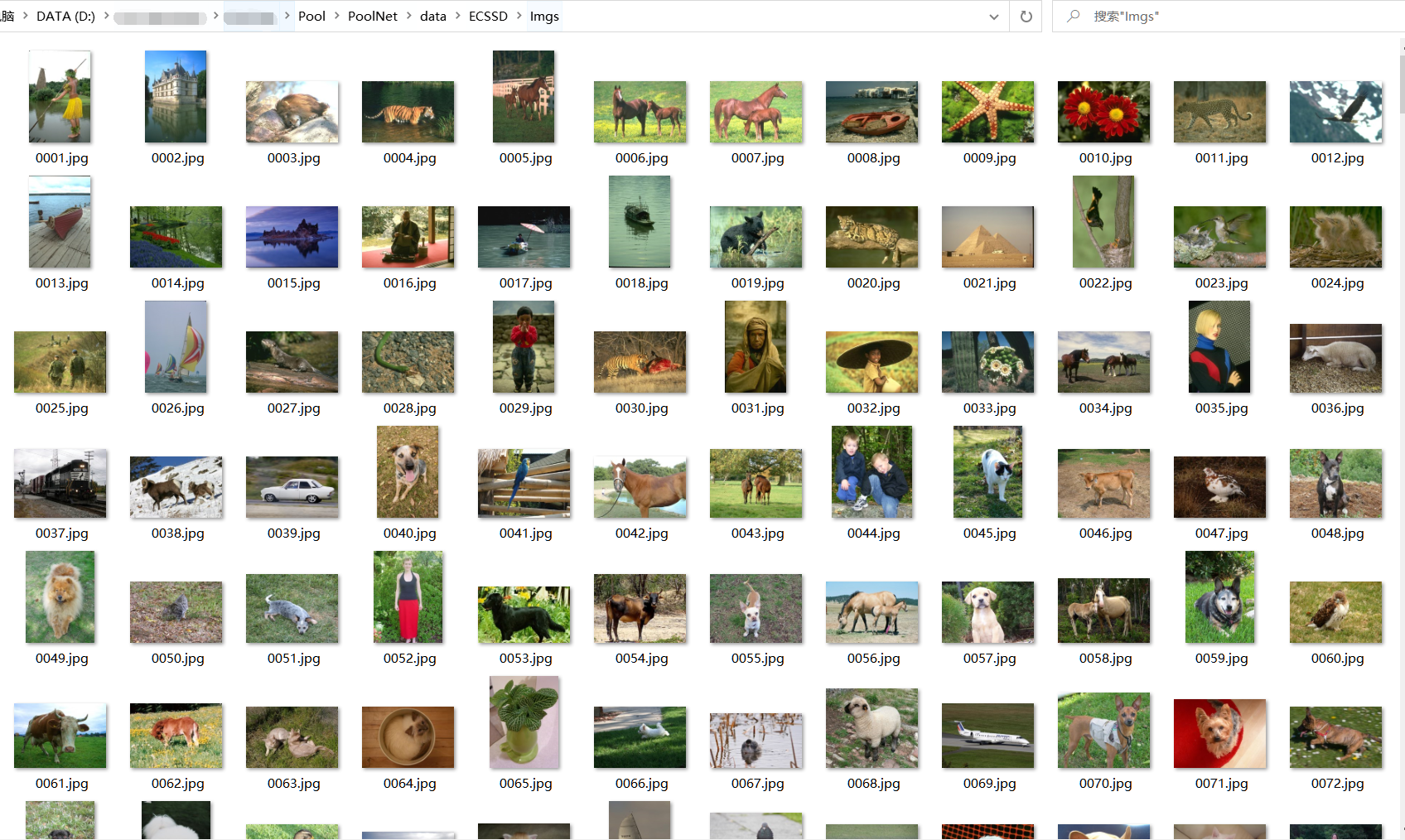
* 1. 本机GPU及相关环境为RTX2060，Cuda10.0，Cudnn7.4.2，OpenCV3.4，Python3.7.

1. 配置环境
   1. 在Anaconda3中通过命令行创建名为poolnetenv的虚拟环境，更换清华镜像站的源安装Cuda10.0对应的PyTorch，torchvision以及cudatoolkit。
   2. 执行代码时遇到了很多module error，分别安装了numpy，scipy和opencv-python，最终Anaconda Navigator中环境部分截图如下

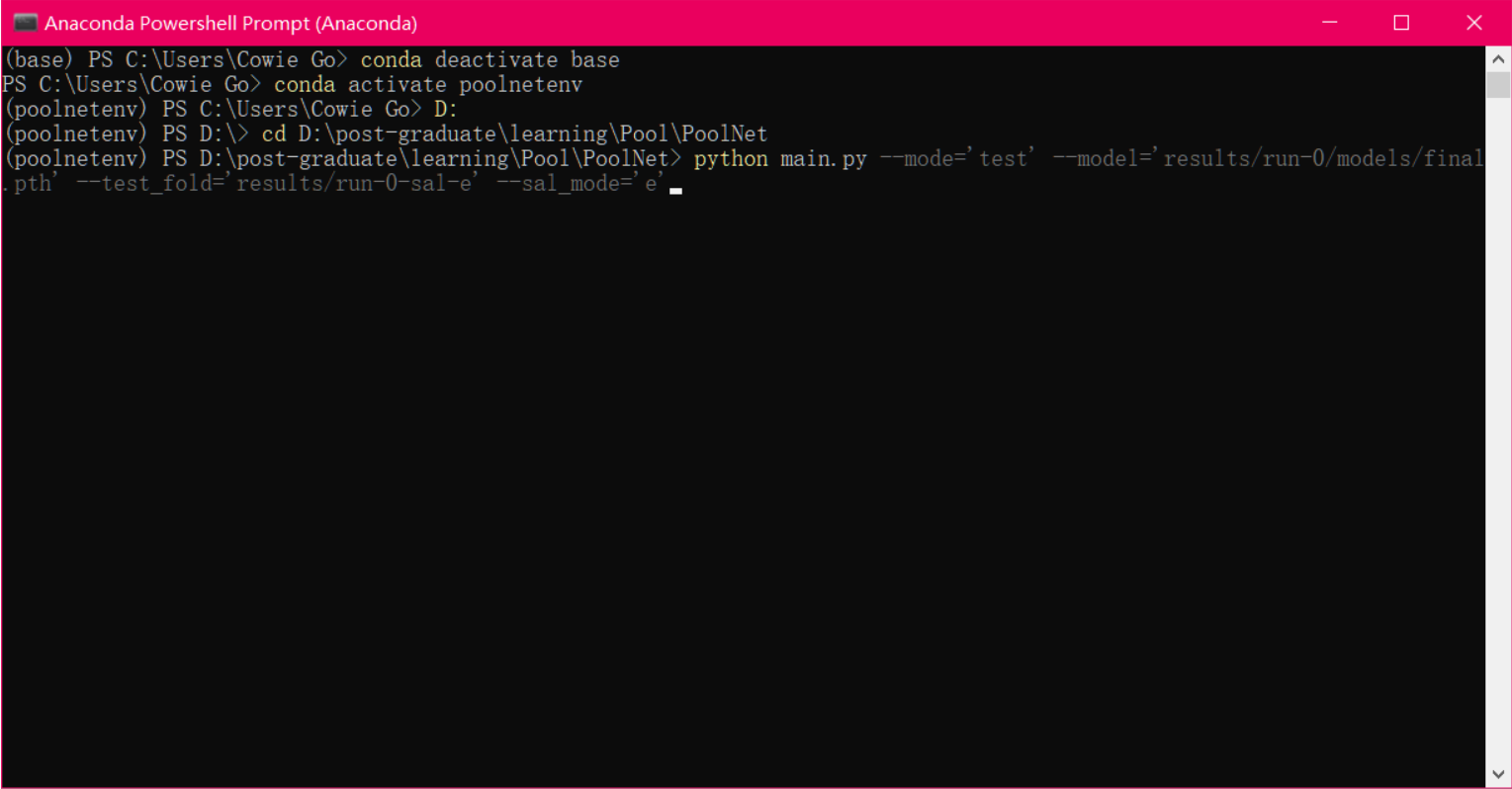


1. 执行测试代码

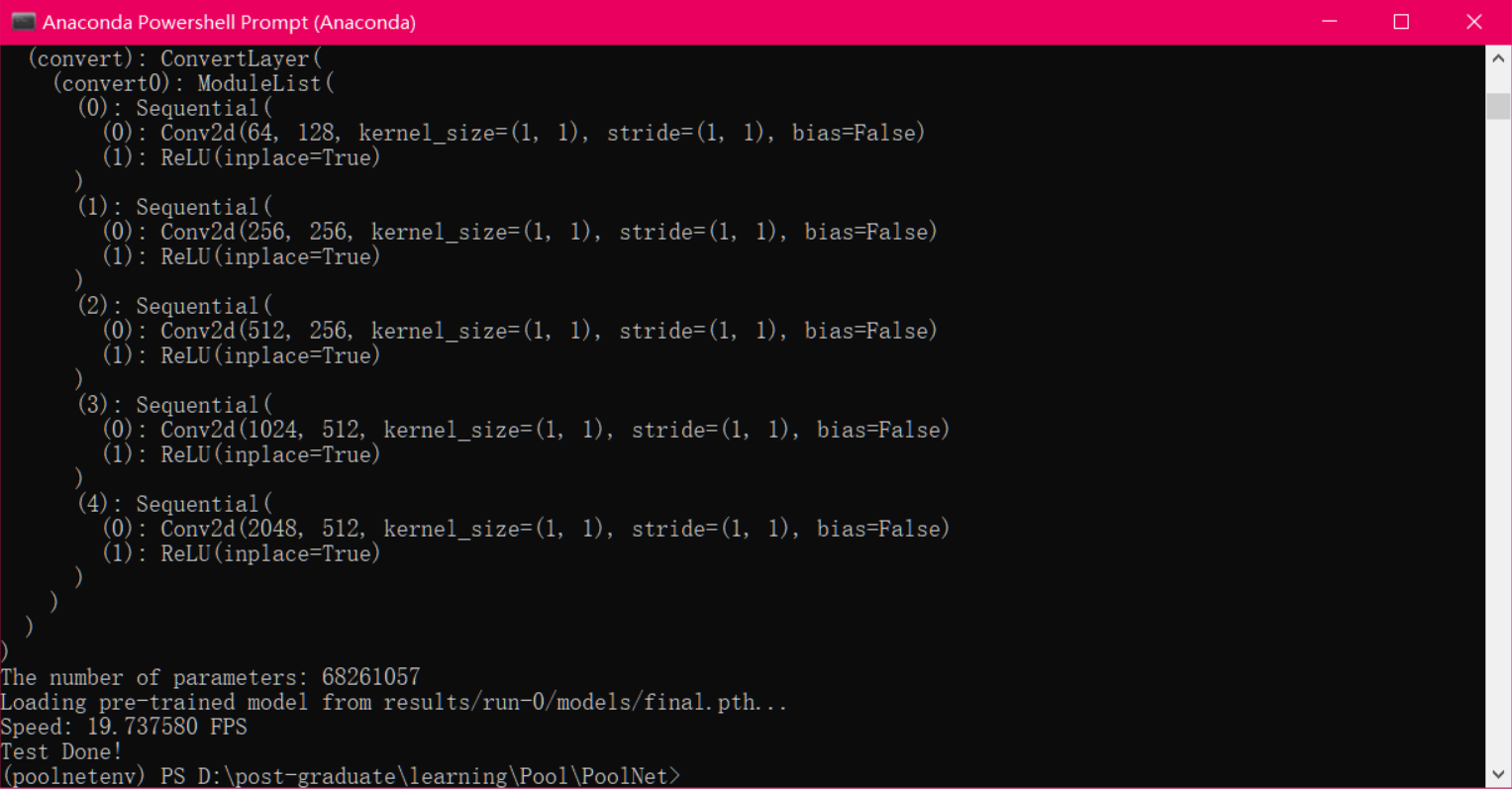
在ECSSD测试数据集上执行Github上给出的测试代码，执行前数据集如下所示



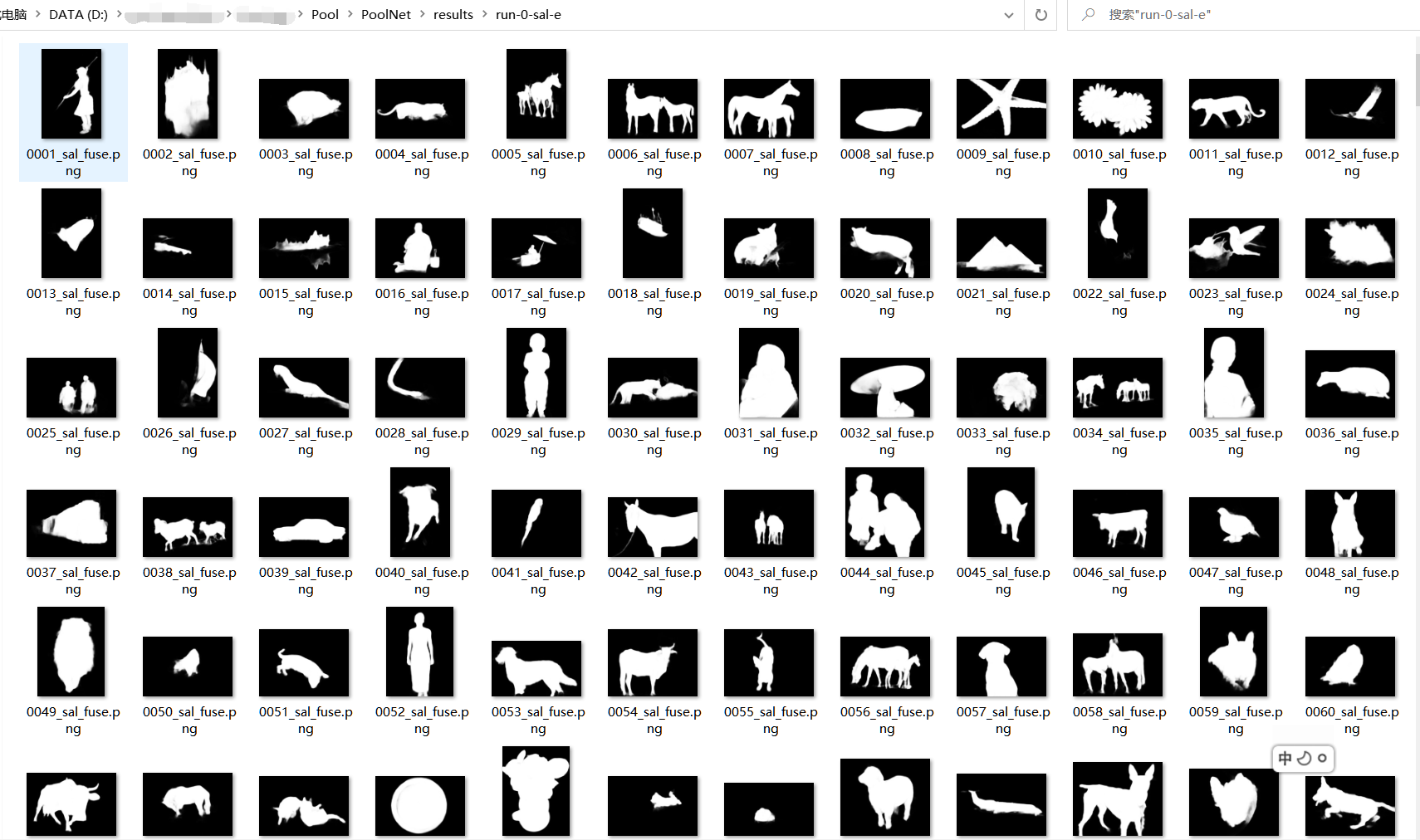
在该虚拟环境中执行test代码，run-0中为预训练模型，sal-mode为e表示该测试ECSSD测试数据集中的1000张测试图片



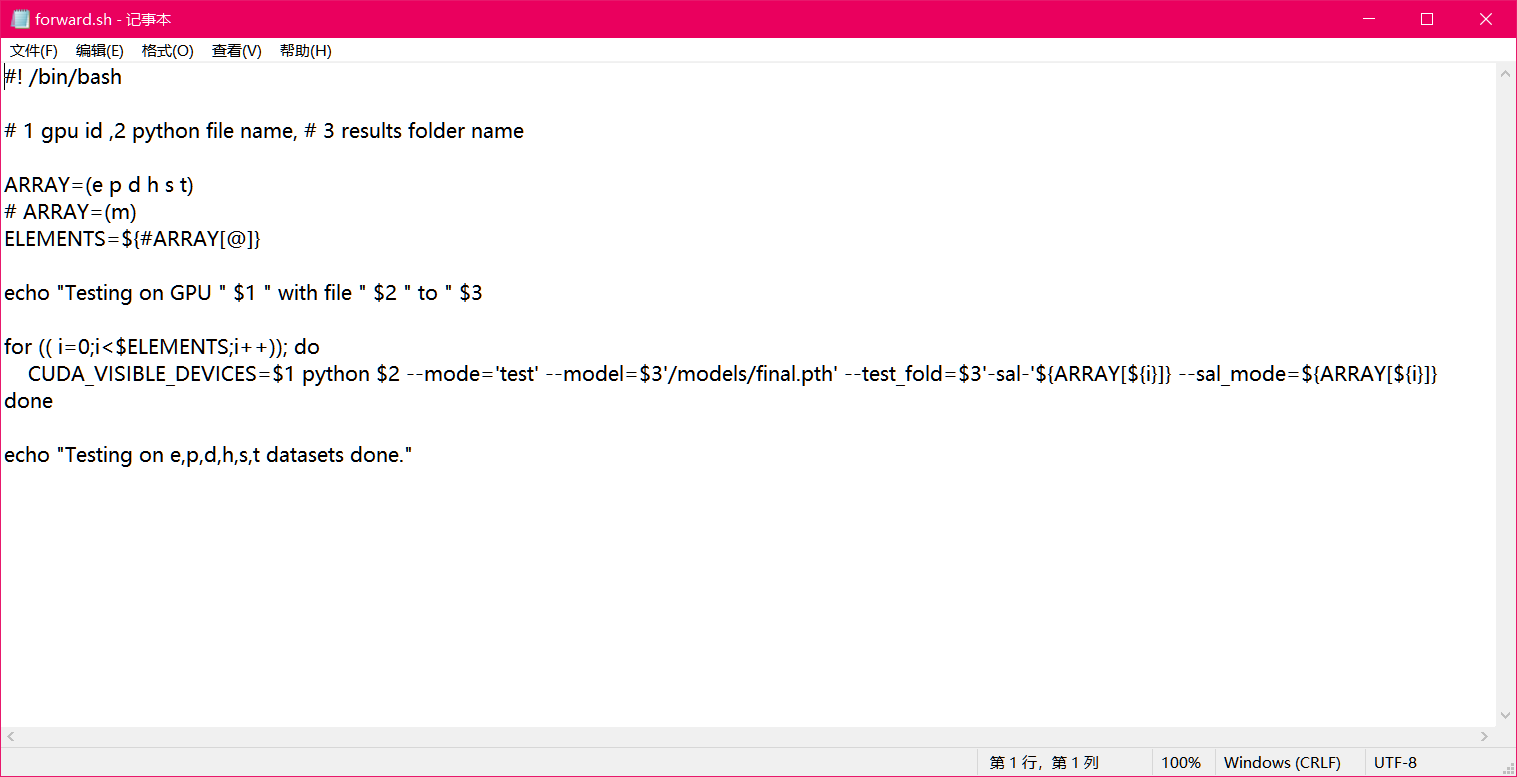
经过一段时间的测试，test指令以接近20FPS的速度完成了测试



执行后生成的文件夹中对应的图如下所示



除此之外，Github上还提供了使用GPU测试所有6个数据集的测试集的方法，具体sh文件如下

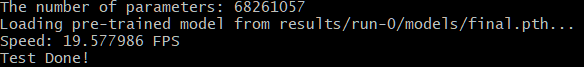


基于自己的笔记本单显卡执行，将执行指令中的2改为0代表0号GPU即唯一的显卡后，在下载好的预训练模型run-0/final.pth进行测试test。

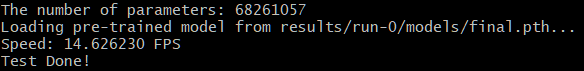


经过短暂的等待，得到了如下数据

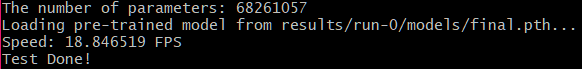
基于预训练模型final.pth对于ECSSD的测试集合的速度



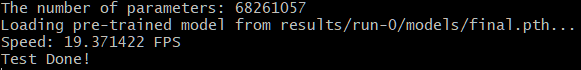
基于预训练模型final.pth对于PASCALS的测试集合的速度



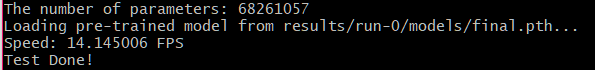
基于预训练模型final.pth对于DOTOMROM的测试集合的速度



基于预训练模型final.pth对于HKU-IS的测试集合的速度



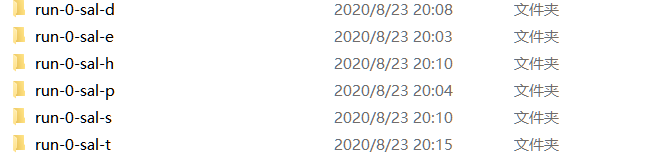
基于预训练模型final.pth对于SOD的测试集合的速度



基于预训练模型final.pth对于DUTS-TE的测试集合的速度

（由于自动关闭.sh，最后一轮测试没有得到速度截图）

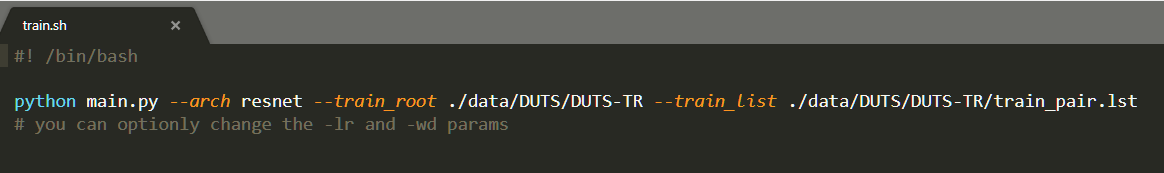
此外，在results文件夹中生成了6个测试集合的对应图片文件夹



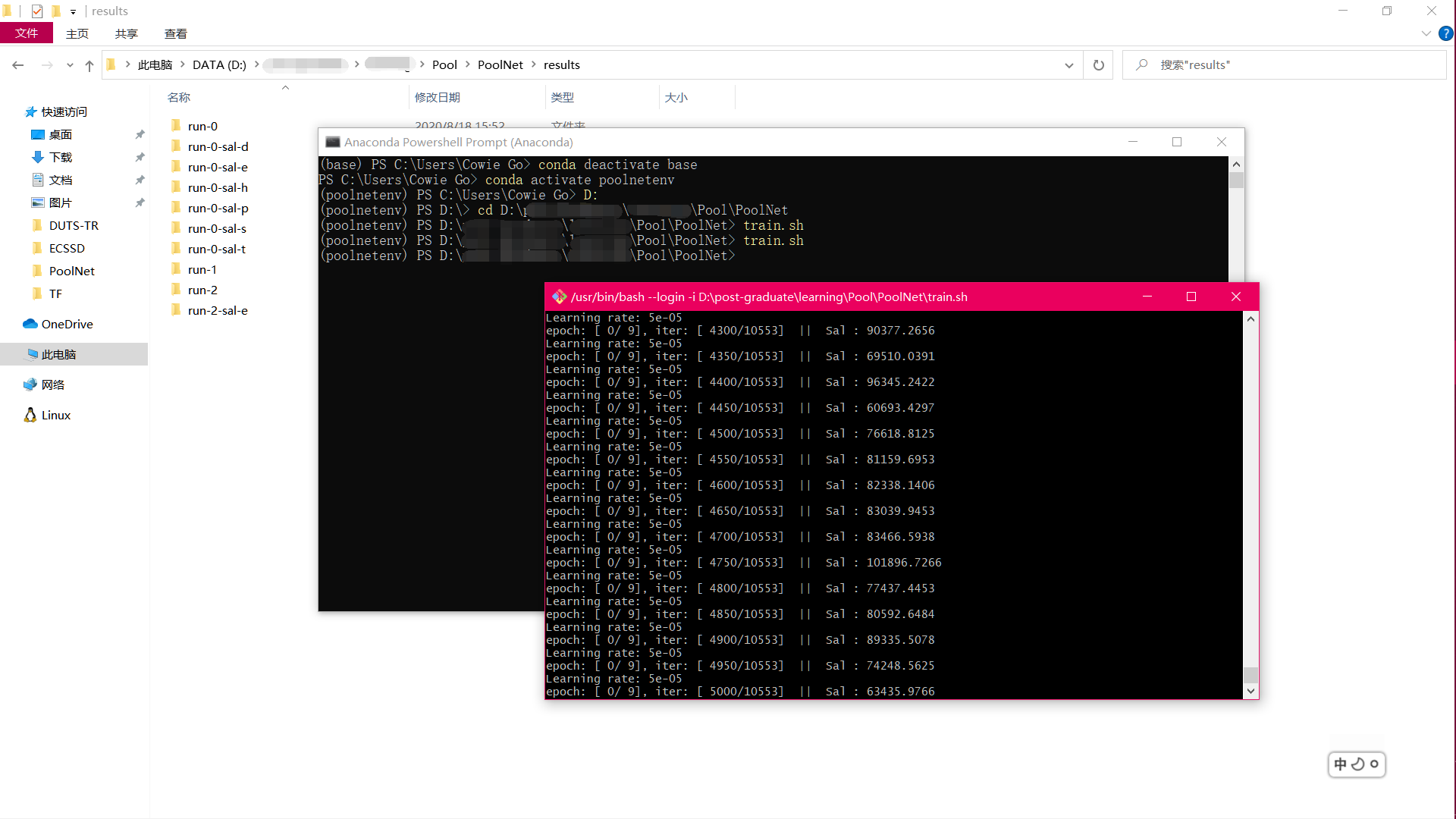
1. 训练

本文Github中提供了训练和联合训练的代码

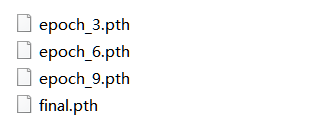
在修改训练路径和训练文件后执行train.sh文件，这里训练的是DUTS完整数据集而不是先前的测试集合，开始学习率learning rate为5e-5，batch-size为1，epoch为24。在15个epoch后学习率降为原先的1/10再进行9个epoch。



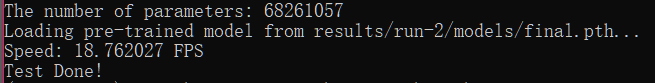
main.py中设定每3个epoch进行一次保存，最后将产生的.pth文件整合成一个单独的final.pth。由于24个epoch时间过长，在函数中将epoch次数改为小一些的3的倍数9次后进行训练。经过约4个小时，训练完成。



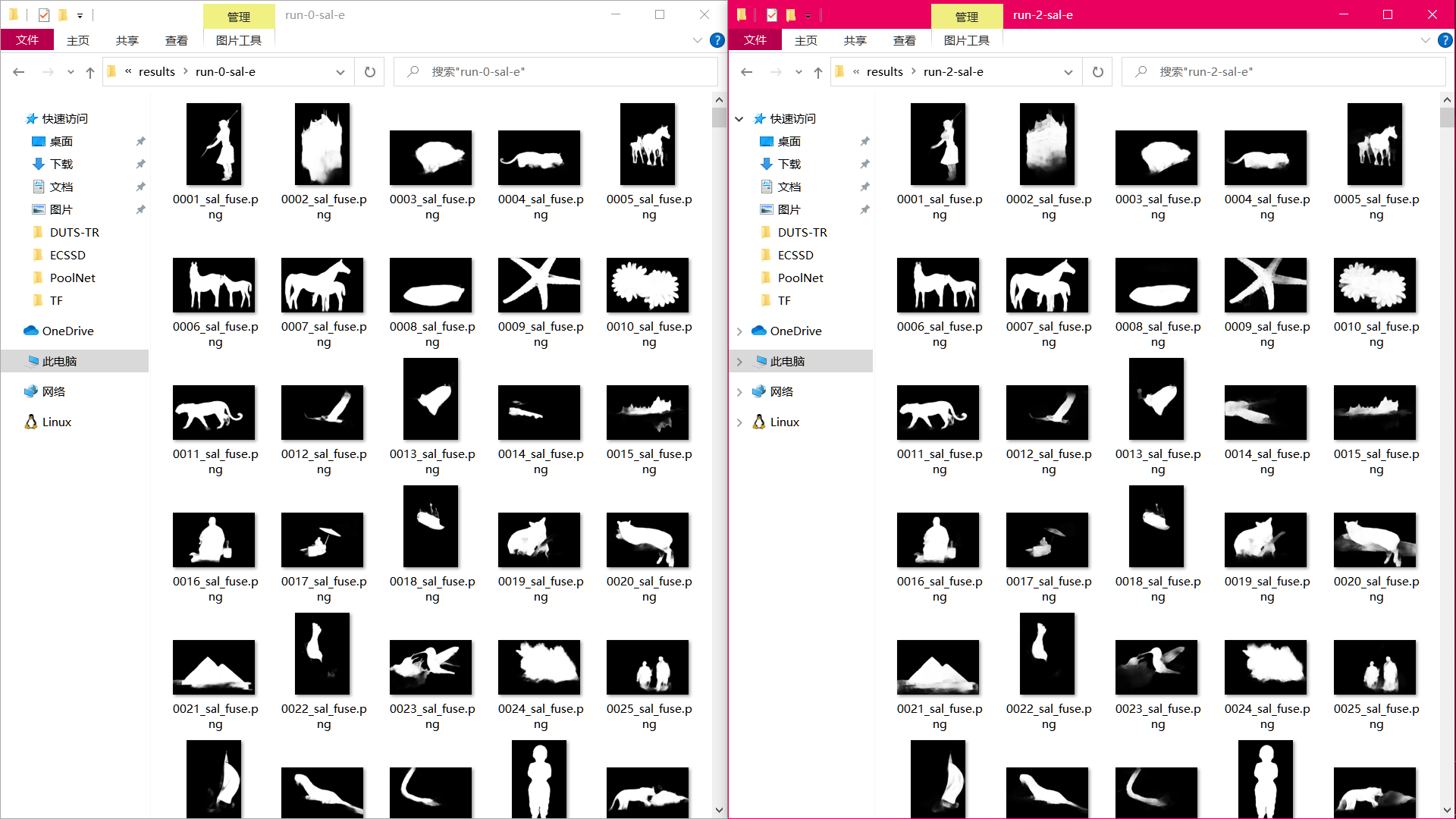
文件夹中产生的.pth模型目录如下



最后对训练的该模型进行测试，测试基于ECSSD的测试集合，包含1000张图片。



将先前从Github上下载的经过24个epoch预训练模型在相同配置，环境下对同一数据集的结果与在本机上进行9个epoch后产生的模型进行测试的结果进行对比。



可以看到，9个epoch与24个epoch训练出的模型还是有不少肉眼可见的差距。

1. TODO

探索joint-training以及GPU与CPU在本实验中的运用和表现结果。

1. Reference
   1. 本文Github地址：<https://github.com/backseason/PoolNet>
   2. PyTorch及相应torchvision在windows平台Anaconda虚拟环境中的安装参考链接 <https://blog.csdn.net/qq_31736627/article/details/99415460>