## 遥感影像分割模型自验报告

队伍：神之一手在哪里（yufei1900@pku.edu.cn）

# 模型简介

本模型以DeepLab V3+为基础，采用Resnet101作为主干网络，并针对给定遥感图像语义分割任务对整个网络框架和各个网络模块进行改进，如在网络框架中对分割对象进行主体和边缘解耦，引入OCR模块；在主干网络中添加attention机制等。其中DeepLab V3+是DeepLab语义分割系列网络现有性能最好的模型，相较于DeepLab V1、V2、V3，DeepLab V3+在主干网络之后连接了encoder-decoder结构来进行多尺度信息的融合，在扩大网络感受的同时获得更加高清的分割结果。也在各个权威的语义分割数据集上取得了很好的成绩。

## 网络模型结构简介：

DeepLab V3+在DeepLab V3的基础上引入了语义分割模型常用的encoder-decoder结构，其中encoder模块即为前作DeepLab V3，decoder模块将encoder部分的结果上采样四倍，并与主干网络上采样前的特征进行拼接，从而融合了多尺度信息。DeepLab V3+也在ASPP和decoder模块使用了depth-wise separable convolution，提高了编码器-解码器网络的运行速率和健壮性。此外，对DeepLab V3+的改进部分中，主体和边缘解耦操作将分割对象的主体和边缘进行解耦，让分割模型不仅需要分割出对象的主体部分，也要分割出对象的边缘部分，通过这种多任务学习来提高分割轮廓的准确率；OCR模块结合分割对象的类别信息，帮助分割网络获取更多的上下文信息，从而提高对单个像素分类的准确性；attention机制通过对不同特征赋予不同的权重，来抽取更为关键和重要的信息，有助于提升模型整体的泛化性能，减少数据差异导致的性能下降。

## 数据集：

训练数据集由大赛复赛提供的10w张数据集，其中随机选取95000张作为训练集，5000张作为验证集。

## 代码提交地址：

（代码提交地址，注意转验收时打开代码仓公开权限）

## 其它：

（其它补充信息）

# 代码目录结构说明

└──deeplabv3+

├── README.md

├── scripts

├── build\_data.sh # convert raw data to

training(1 pc)

├── ckpt # save weights

├── src

├── data

├── dataset.py # mindrecord data generator

├── build\_seg\_data.py # data preprocessing

├── loss

├── loss.py # loss definition for deeplabv3

├── nets

├── deeplab\_v3

├── deeplab\_v3.py # DeepLabV3 network structure

├── net\_factory.py # set S16 and S8 structures

├── tools

├── get\_multicards\_json.py # get rank table file

└── utils

└── learning\_rates.py # generate learning rate

├── eval.py # eval net

├── train.py # train net

└── requirements.txt

# 自验结果

## 自验环境：

MindSpore：MindSpore-1.0.0；

自验环境：Ascend-Powered-Engine 1.0 (Python3)

## 训练超参数：

batch\_size：64

epoch：200

learning rate：0.01

loss function：交叉熵损失（OHEM交叉熵损失）

optimizer：SGD

## 训练：

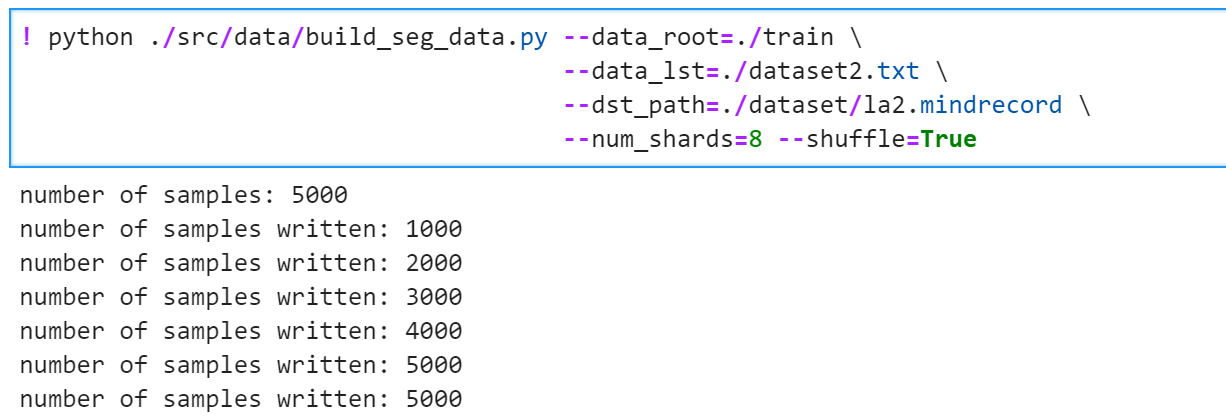
模型训练

### 如何启动训练脚本：

1. 将convert.py放在训练数据集dataset目录下，运行convert.py将 tif文件转换成jpg格式；
2. 生成如下格式的图片路径文档；



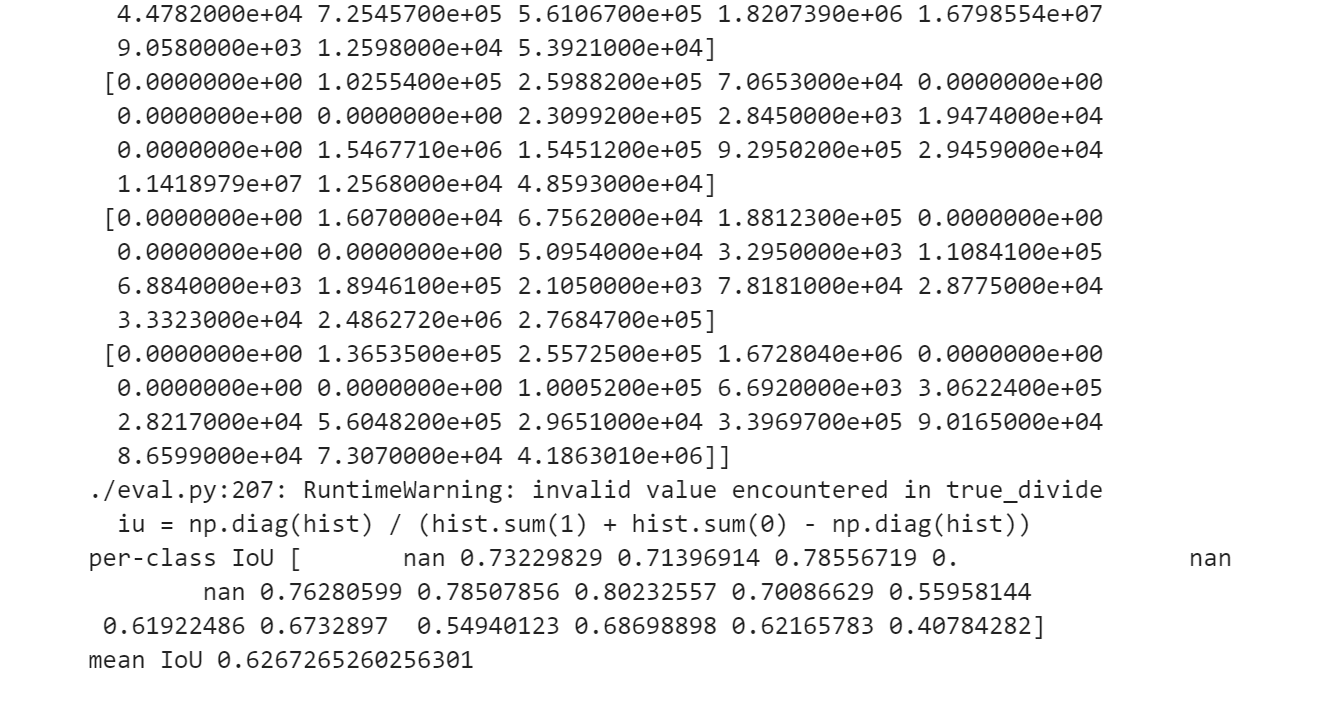
4、利用build\_seg\_data.py生成mindrecord：



5、训练数据：



### 训练性能结果：



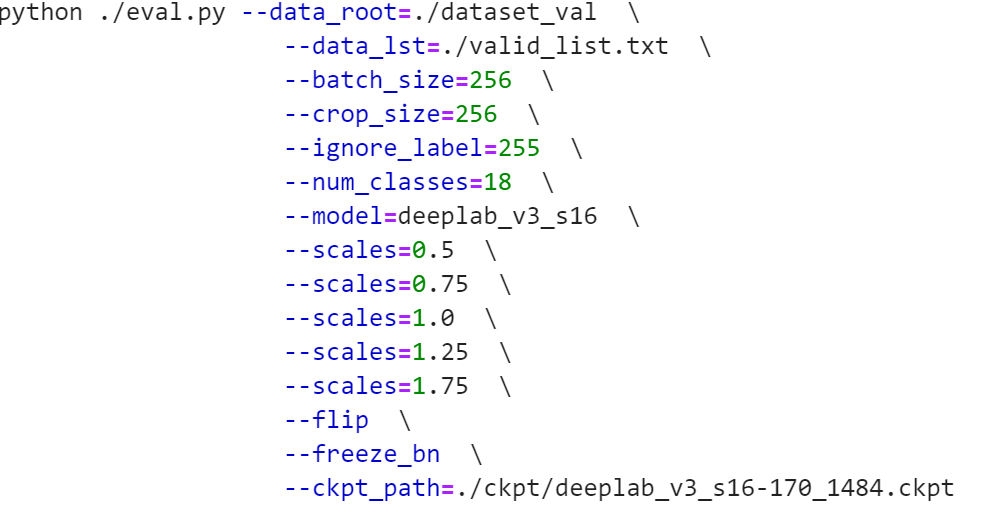
## 推理：

### 如何启动推理脚本：

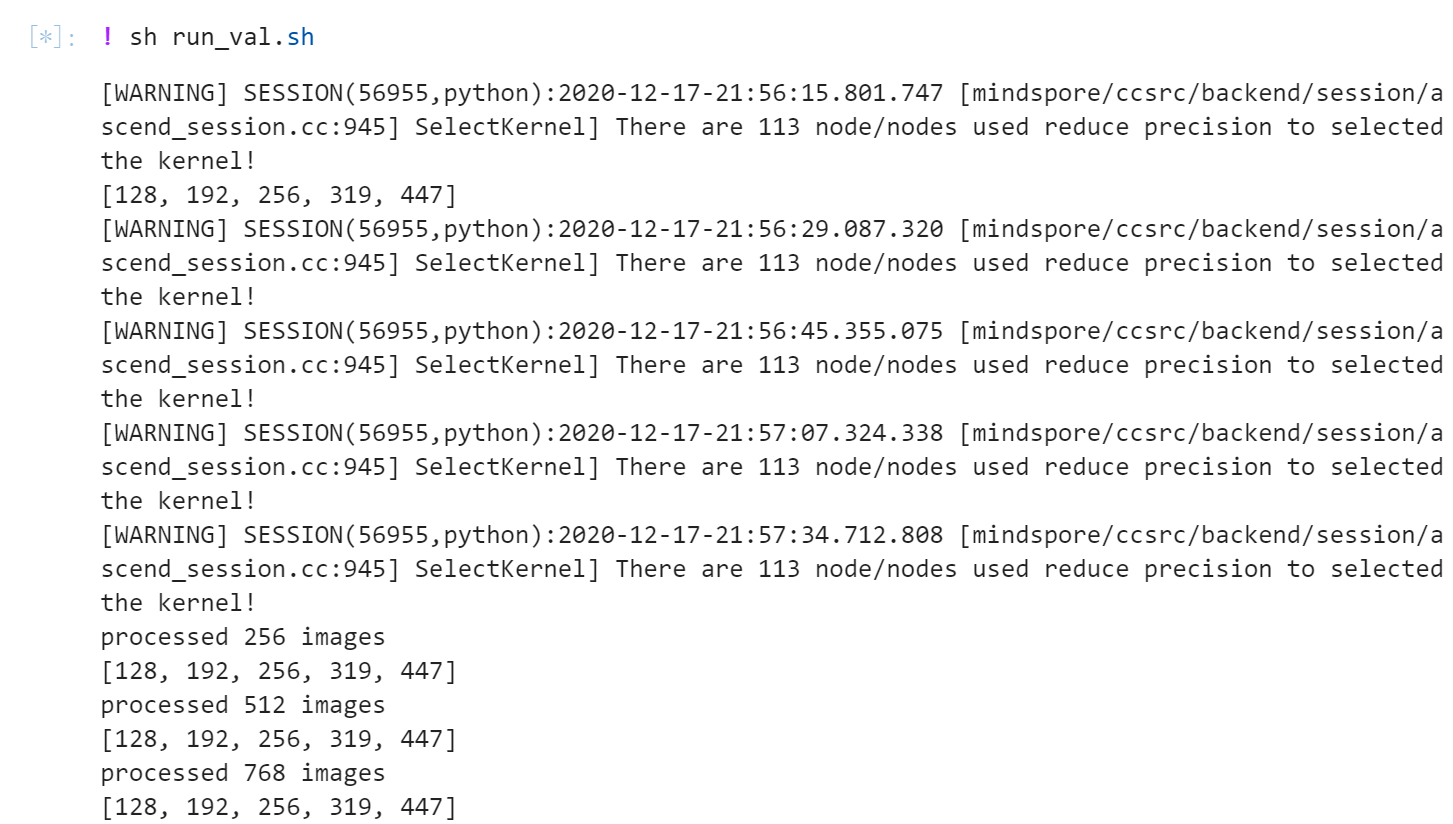
infer.py如何启动

! python infer.py --data\_path=./dataset\_val/images

eval.py如何启动



run\_eval.sh如何启动：



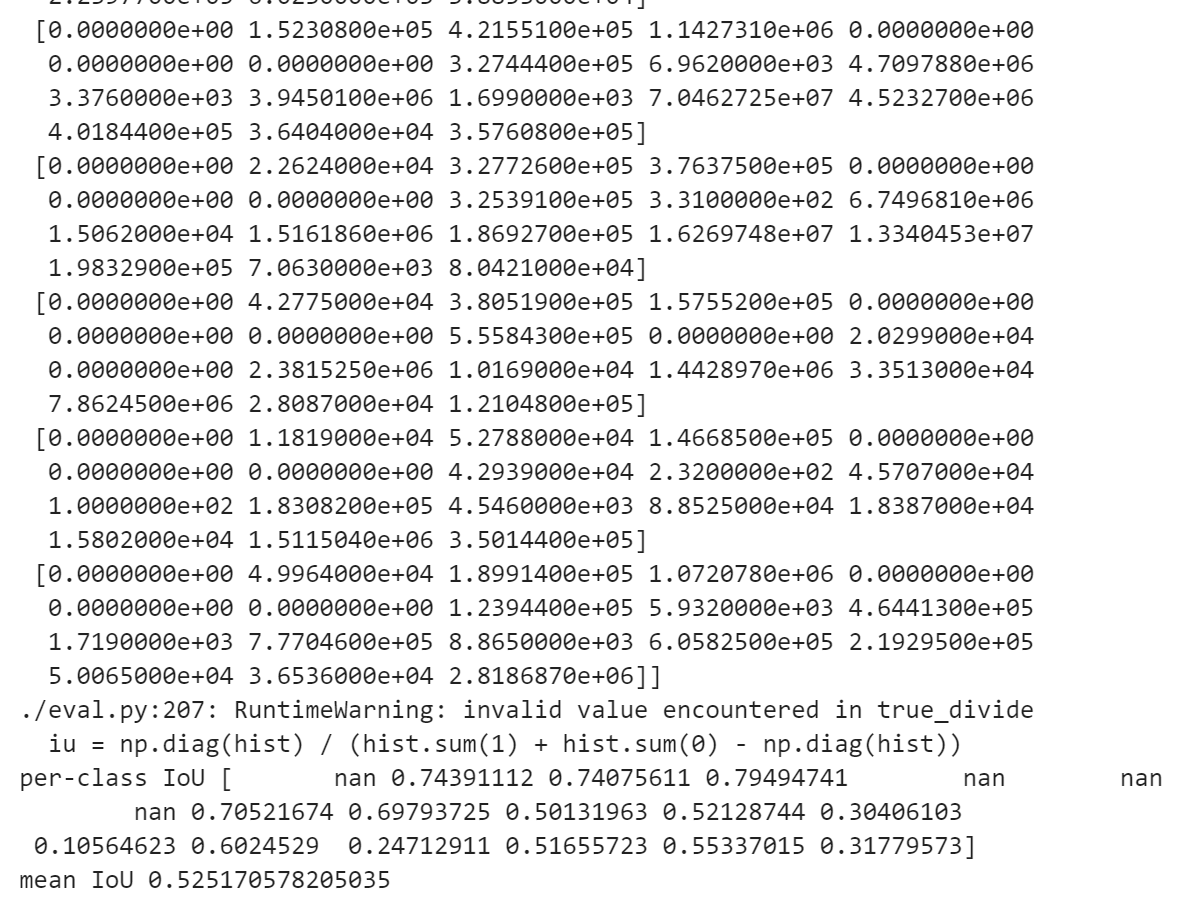
### 推理精度结果：

per-class IoU [ nan 0.74391112 0.74075611 0.79494741 nan nan

nan 0.70521674 0.69793725 0.50131963 0.52128744 0.30406103

0.10564623 0.6024529 0.24712911 0.51655723 0.55337015 0.31779573]

mean IoU 0.525170578205035



## 参考论文：

参考论文

## 参考git项目：

参考Tensorflow, pytorch, caffe等项目网址