**本文为code0518.py说明文档**

作者：刘一帆

日期：2020年5月19日星期二

版本：1.1

目录

[第一章、运行 - 1 -](#_Toc40798616)

[1.数据集 - 1 -](#_Toc40798617)

[2.代码语言 - 1 -](#_Toc40798618)

[3.硬件条件 - 1 -](#_Toc40798619)

[4.数据预处理之图片存放 - 1 -](#_Toc40798620)

[5.超参数说明 - 2 -](#_Toc40798621)

[6.图片切割（运行代码） - 4 -](#_Toc40798622)

[7.生成标签索引 - 4 -](#_Toc40798623)

[8.划分数据集 - 4 -](#_Toc40798624)

[9.训练模型 - 5 -](#_Toc40798625)

[10.验证模型 - 5 -](#_Toc40798626)

[11.输出结果 - 6 -](#_Toc40798627)

[12.错误输入 - 6 -](#_Toc40798628)

[第二章、代码结构 - 7 -](#_Toc40798629)

[第三章、分函数细节 - 7 -](#_Toc40798630)

[1.data\_loader.MyData0518(train\_txt) - 7 -](#_Toc40798631)

[2.train() - 7 -](#_Toc40798632)

[3.evaluation() - 8 -](#_Toc40798633)

[4.test() - 9 -](#_Toc40798634)

[5. sr1\_sr2\_label\_path\_make() - 10 -](#_Toc40798635)

[6. train\_validation\_test\_division() - 10 -](#_Toc40798636)

[7.model() - 11 -](#_Toc40798637)

## 第一章、运行

### 1.数据集

数据集采用Building change detection dataset，地址：<https://study.rsgis.whu.edu.cn/pages/download/building_dataset.html>

### 2.代码语言

python 3

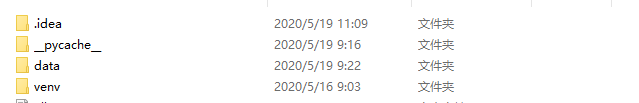
### 3.硬件条件

RTX2080Ti 11G

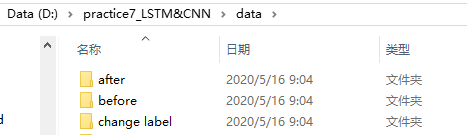
CUDA上运行

### 4.数据预处理之图片存放

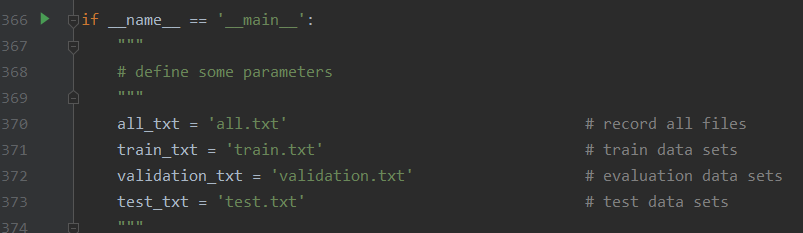
构建data文件夹，把下载好的数据集放进去



data内部



### 5.超参数说明

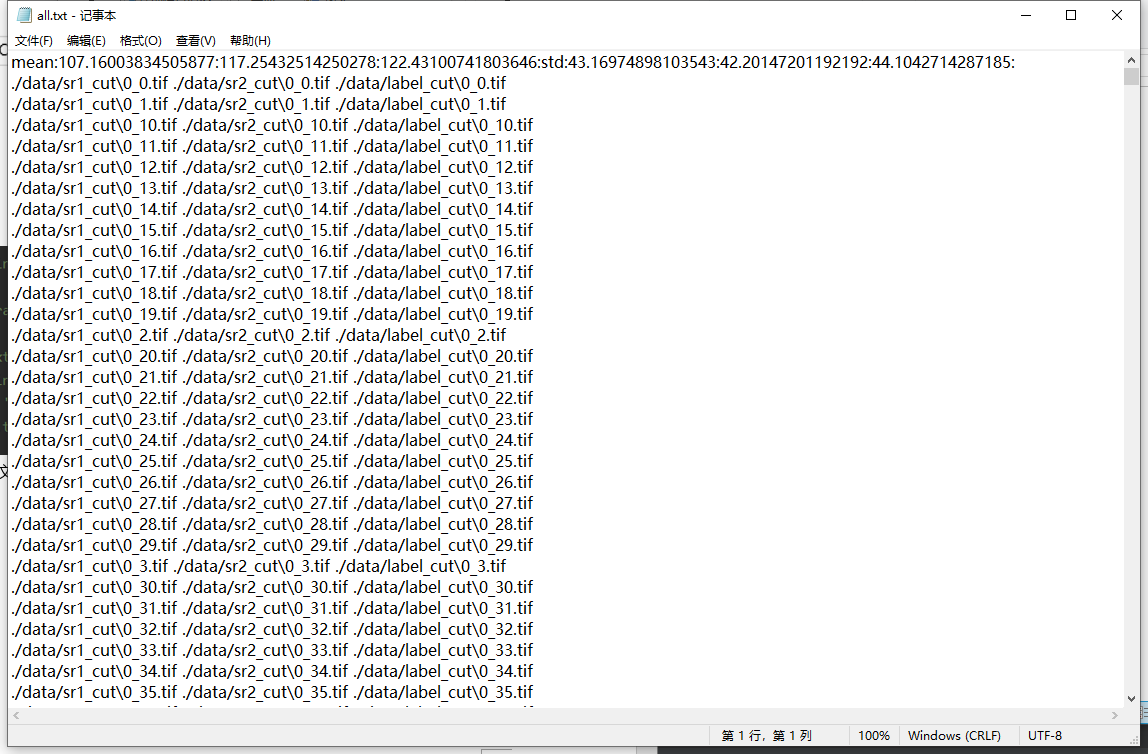


all\_txt 记录所有的切割后小块文件

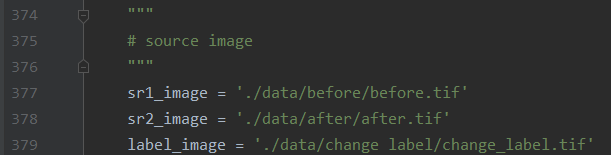
train\_txt 记录训练的标签

validation\_txt 记录验证集

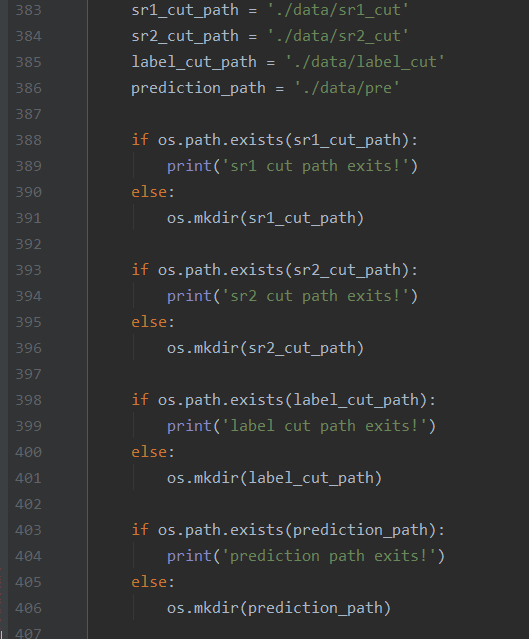
test\_txt记录测试集



all.txt内容如上，第一行记录整体图片的三通道均值以及方差，用来对数据正态化变换。

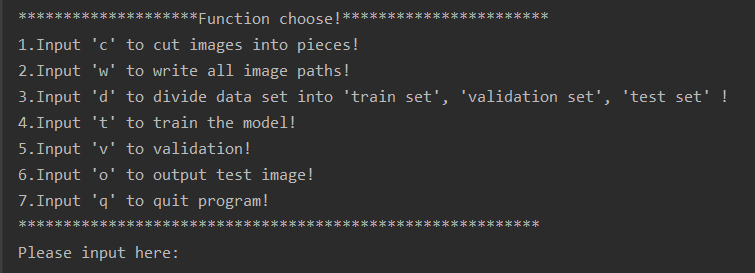


sr1\_image,sr2\_image,label\_image用来指定第一期，第二期，标签影像名称。因为数据太大，所以需要进行切割，这里指定第几期。



sr1\_cut\_path, sr2\_cut\_path, label\_cut\_path,prediction\_path指定输出的文件夹，切割输出文件夹。如果不存在运行程序将会创建。

### 6.图片切割（运行代码）



运行程序，输入c切割图片，将图片切割成512\*512小块，两块之间没有重叠部分（指定stride=512，步长与图片大小相同）。边界部分不足的统统补充纯黑色，下图为示例。



### 7.生成标签索引



输入w进入标签索引txt文档，将生成all.txt文档

all.txt内部，三张图片以“空格”进行分割。

### 8.划分数据集



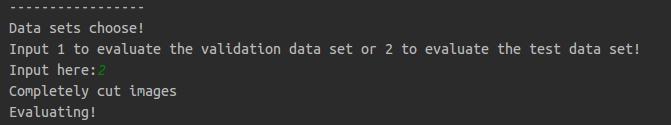
输入d随机划分训练集（占比0.7）、验证集（占比0.1）和测试集（占比0.2）.然后保存成文档。



### 9.训练模型

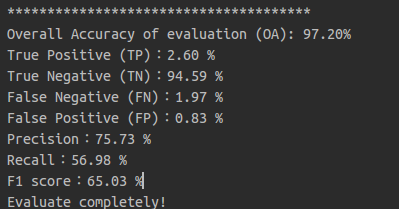
输入t进行训练

### 10.验证模型

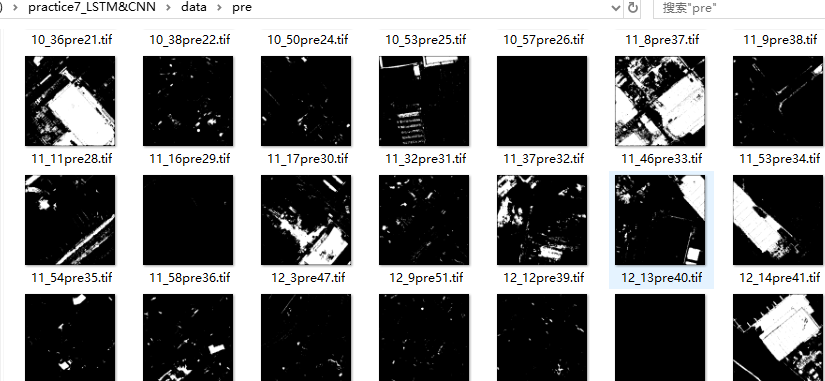


输入v验证，进入后输入1在validation.txt上验证，输入2在test.txt上验证。

输出OA,TP,TN,FN,FP数据。这里有点问题，每次我加载模型验证得到的数据都不一样。



### 11.输出结果

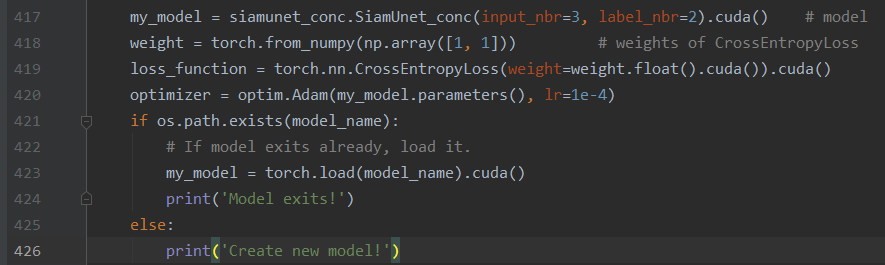


输入o输出test.txt文档中的图片的验证结果

### 12.错误输入

输入除以上选项的其它选项将重新进行选择。

## 第二章、代码结构

1.模型，模型采用siamunet网络中conc模型，指定图片输入通道数，输出标签数。

2.损失函数，采用交叉熵，权重暂时定为1：1

3.优化函数，暴力Adam

4.如果同名模型存在那么加载进去。

## 第三章、分函数细节

### 1.data\_loader.MyData0518(train\_txt)

#### 1.1初始化

该函数读取文档首行关于整个数据集的三通达均值与方差，然后将每一行的数据索引拆分存储到self列表中。

#### 1.2加载数据

将item所指向的图片加载，然后进行三通道变成标准正态分布，返回时将通道数提升到前面，标签数据除以255.

### 2.train()

先给定一个之前误差pre\_loss，当当前误差cur\_loss小于pre\_loss时保存模型，并且把cur\_loss赋值给pre\_loss。程序每20次打印出一下当前误差值。

### 3.evaluation()

评估函数，评估正确的像素点数以及其他评估指标。

|  |  |
| --- | --- |
| **1** | **0** |
| **0** | **1** |

|  |  |
| --- | --- |
| **1** | **1** |
| **0** | **0** |

如上两图为例说明如何计算正确错误的像素点数，假定左侧为输出的预测图像，右图为实际的标签图像。预测正确的点为左图左侧两点，在这两点中，上侧为预测变换标签变换的点，下册为预测不变，标签不变的点。这两点为正确的分类。



正确的分类点数获得：用预测图与标签图相加，用numpy统计数值为1的点数即为错误点数{因为（0+0=0，1+1=2）正确，（0+1=1，1+0=1）错误}，总数减去错误点数即可得到正确点数。



正确点数中两者区分：用标签图像与预测图像相乘，只有预测变换（1）且标签变换（1）相乘才为1，其他相乘都为0。这样统计结果中为1的点数即可得到。

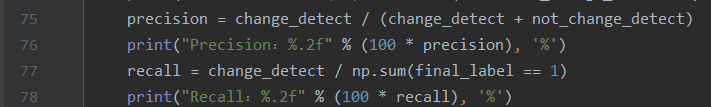


错误点数中两者区分：将预测结果加1与标签数据相乘，这样只有预测不变标签不变的相乘得1（1\*1=1，其他情况2\*1=2，1\*0=0，2\*0=0），然后统计数据即可区分。

|  |  |
| --- | --- |
| **2** | **1** |
| **1** | **2** |

|  |  |
| --- | --- |
| **1** | **1** |
| **0** | **0** |

准确度：略



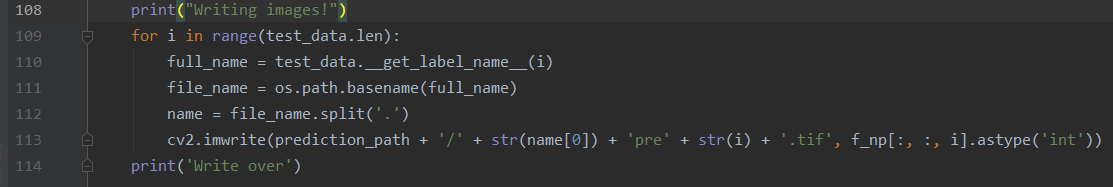
精确度Precision:分类正确的正样本个数占分类器判定为正样本的样本比例的个数。

召回率Recall:分类正确的正样本个数占真正样本个数的比例。



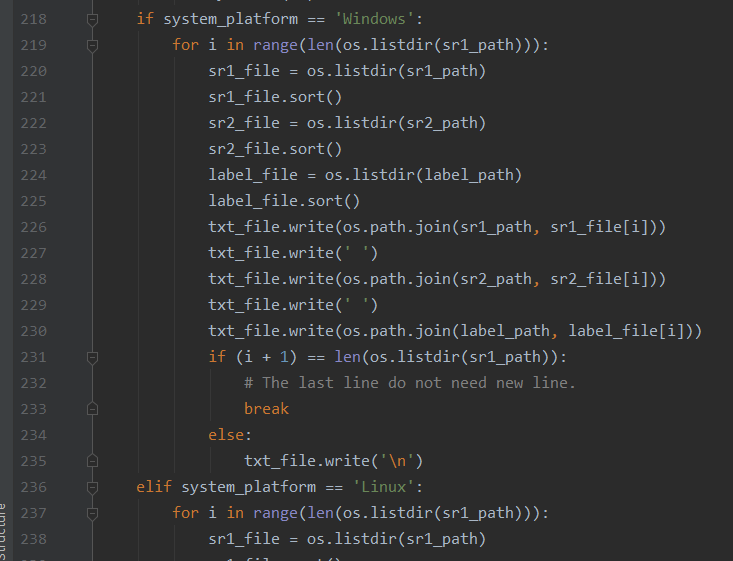
F1 score:

### 4.test()



大致与valiation相同只是最后获得test的标签名称，加上pre与数字i后保存，所以**test加载不能shuffle！**

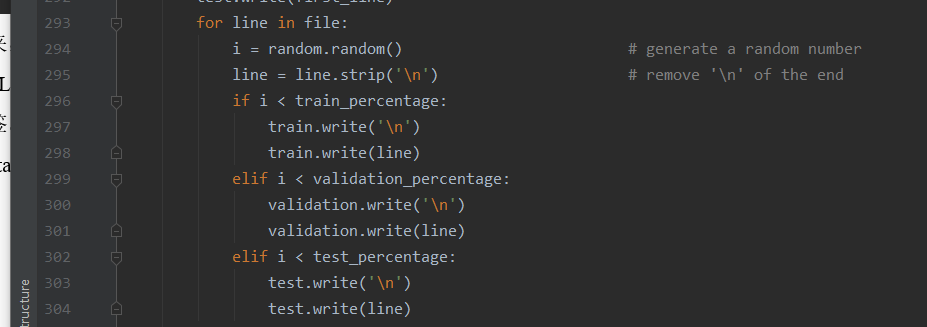
### 5. sr1\_sr2\_label\_path\_make()



这里给定文件夹，如果没有给定生成在何种系统下运行，程序会判断当前是在Windows还是Linux环境下运行，然后生成相应标签。当然也可以指定生成特定环境下的标签。Windows环境下格式 为”./data/sr1\_cut\test.tif“，Linux环境下格式为”./data/sr1\_cut/test.tif”。所以要区分生成标签。

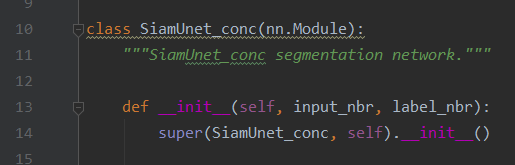
### 6. train\_validation\_test\_division()





数据集划分，这里利用随机数生成函数，生成一个[0~1]的浮点数，当浮点数小于0.7时将这条数据划归训练集，当浮点数大于0.7小于0.8时划归验证集，当浮点数大于0.8时划归测试集。这样整个数据70%训练集，10%验证机，20%测试集。

### 7.model()



模型初始化时候需要指定每张图片的通道数，供第一次卷积使用。指定为1时候是灰度图像，我这里输入3通道彩色图像，故指定3。输出类数，这里只有两类（变-不变）故这里初始化为2。