

摘要

近期，深度学习的发展可是说是已经相对成熟，计算机视觉领域更是发展迅猛，自从FCN网络的问世，卷积神经网络加图像语义分割也取得了不错的成果，但是高光谱影响分类一直因为数据集多样，数据量大，发展的速度赶不上传统三通道的图像语义分割的速度，并且精度反而没有信息量更少的语义分割效果好，同时训练起来也是非常的麻烦。

基于训练精度的问题，我想通过形态学，对结果进行二次处理，对比是否形态学可以调高训练精度；对于难以训练的问题，我使用Django写出后端接口的方式，让训练更加简单，同时使用到了异步技术和tensorboard。

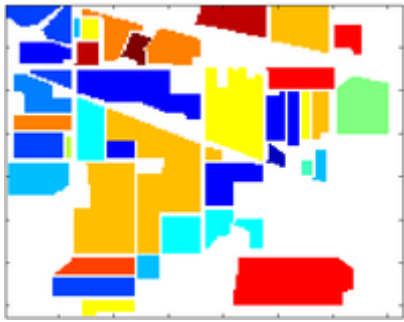
主体

1 绪论

在数字图像的领域和计算机视觉的领域，所谓语义分割就是将一整幅照片分割为多个碎片的过程，每一个碎片可以代表不同的含义。语义分割一直有一个非常难以解决的问题，那就是背景往往比实际要分割出的对象多太多，以至于如果将多有一整张图片全部分割为背景，在数字评价中，往往精度也会非常高，但是这样的分割显然是没有任何意义的，这就需要我们认真的调整参数，才有可能取得不错的成果。

本次设计使用到两个数据集，其一是印度松，她是使用AVIRIS传感器在印第安纳州西北部的印第安纳州西北部的印第安纳松树测试点上拍摄的，整个图像有145*145像素和224个通道。它包含着三分之一的森林和其他天然多年生植物、三分之二的农业、有两条高速公路和一条铁路和低密度的建筑等。这些地面被详细划分为16个等级，通过删除覆盖吸水区的条带，实际可以使用的波段实际有200个。





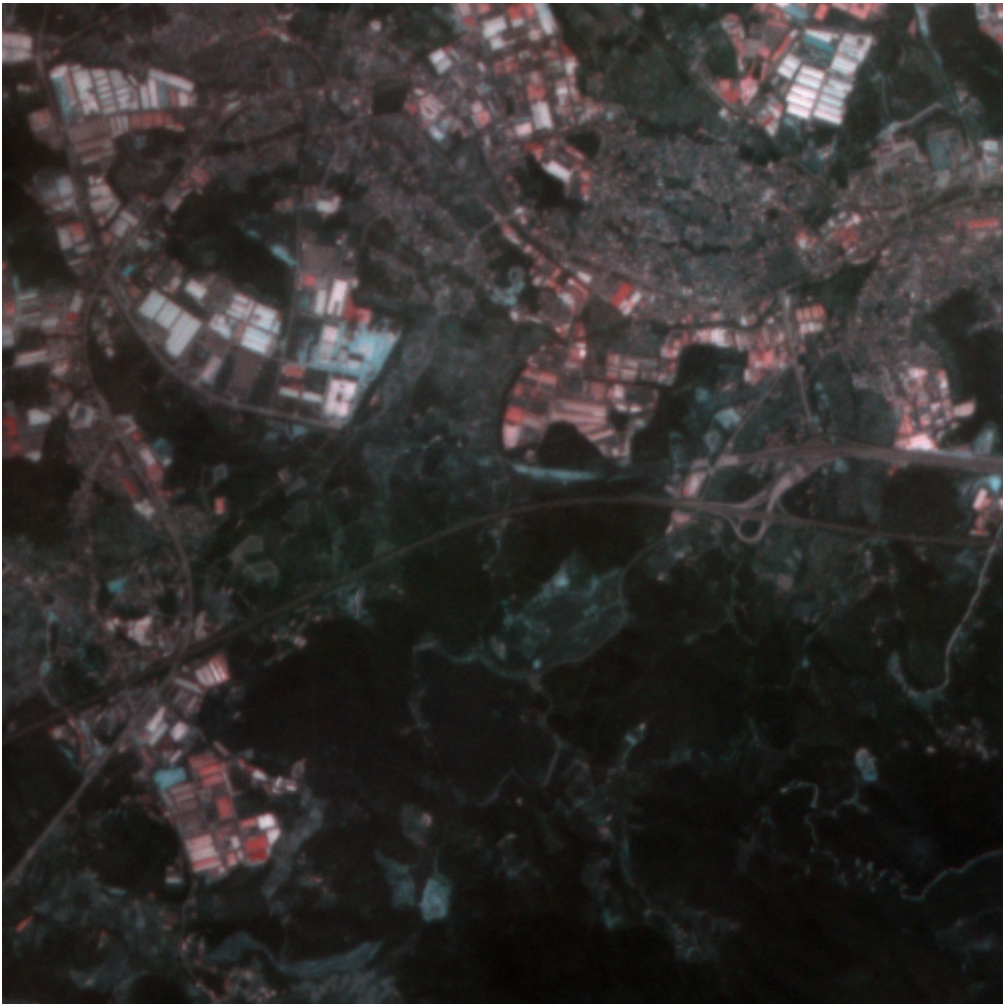
Indian Pines场景的Groundtruth类及其各自的样本数

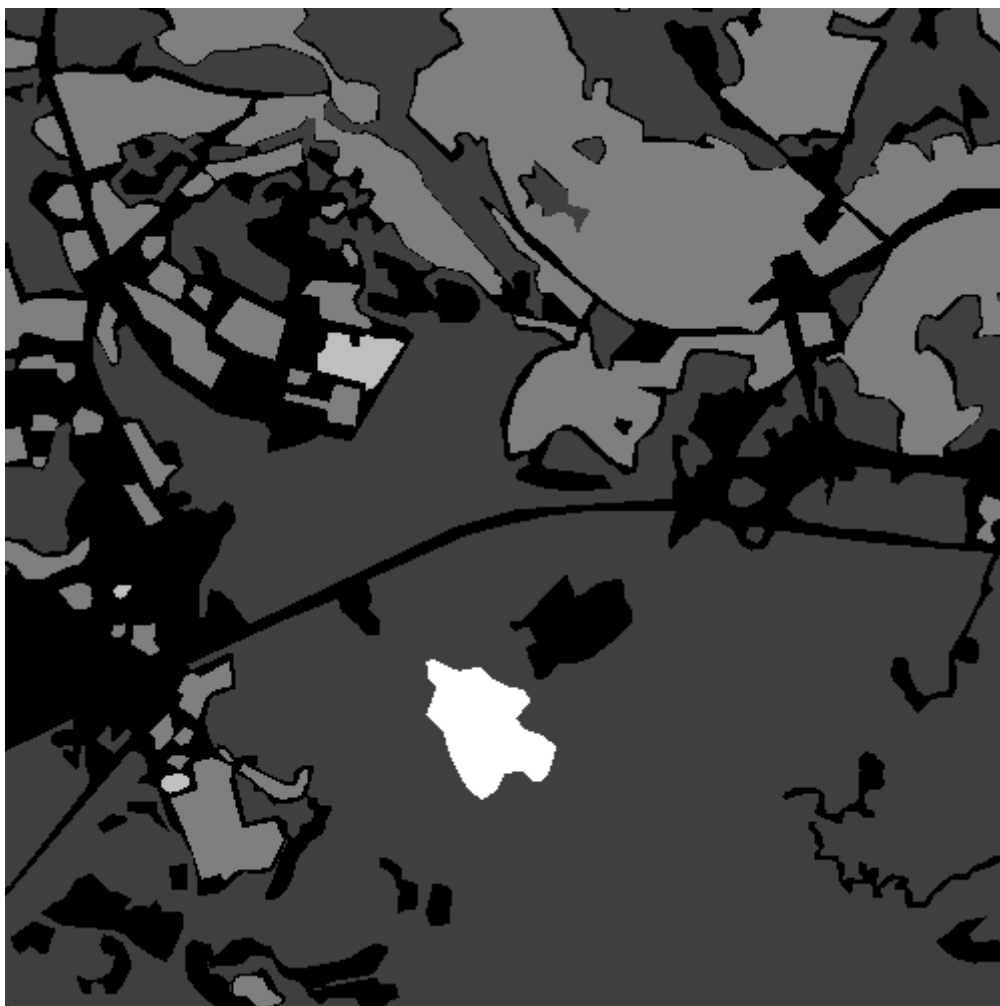
#	类	样品
1	紫花苜蓿	46
2	玉米not	1428
3	玉米薄荷	830
4	玉米	237
5	草场	483
6	草树	730
7	割草草	28
8	干草堆	478
9	燕麦	20
10	大豆芥末	972
11	大豆薄荷	2455
12	大豆清洁	593
13	小麦	205
14	树木	1265
15	建筑物-草木-树木驱动器	386
16	石钢塔	93

另一个数据集是欧比特数据，更详细的说是使用珠海一号高光谱卫星数据，该数据90幅500*500* 32的高光谱图像，将地表可以分为如下表：

#	类	标签值
1	植被	1
2	建筑	2

#	类	标签值
3	裸土	3
4	水体	4
5	背景	255





形态学是生物学的分支，主要是研究生物体结构和他们的结构之间性质，在图像领域是指一系列处理图像形状特征的图像处理技术。它的基本思想是用特殊的结构元来提取或者测量输入图形中相应的形状或者特征，以便于进一步对图像进行分析和目标识别。

为了能够让训练过程更加友好，使用Django+Celery+sqlite+tensorboard的方式，将训练函数通过接口的方式暴露给用户，将训练过程通过Celery异步处理，防止堵塞Django；并且将用户的每一次训练过程都存入数据库；在训练过程中使用tensorboard展示给用户。

2 论文主体

2.1 欧比特珠海一号卫星数据

2.1.1 修改网络结构

因为传统的网络使用的都是32通道，多以需要将传统网络进行修改，修改为32通道。同时传统的网络并没有加入空洞卷积，但是空洞卷积对图像语义分割问题中下采样带来信息丢失，图像分辨率降低问题有显著弥补效果，所以使用了空洞卷积。

使用的网络有FCN, LR-ASPP和DeepLabv3。其中FCN和DeepLabv3使用的backbone是resnet152, resnet101, resnet50，LR-ASPP使用的backbone是mobile-net.

2.1.2 数据预处理

欧比特珠海一号卫星实际只有90幅影像，为了增加数据，选择了裁剪的方式，将数据裁剪为4幅256*256的影像，并拿出20帕数据作为验证集；原始数据是tiff格式，但是在python中对tiff格式的处理速度远远没有numpy中adarray类数据快，所以将数据预转换为numpy格式。

但是，并没有使用降维或者去噪的预处理，因为卷积神经网络应该容许数据有误差的存在，这是深度学习的特点，而且如果非要做预处理的话，反而起不到加速网络训练和测试的速度，因为使用GPU进行训练的速度比PCA等方法预处理图片的速度如今并不慢。

2.1.3 训练卷积模型

1. 使用的Adam优化器。Adam优化器在pytorch中的计算如下

$$\begin{aligned} \text{input} : & \gamma, \beta_1, \beta_2, \theta_0, f(\theta), \lambda, \text{amsgrad}, \text{maximize} \\ \text{outout} : & m_0 \leftarrow 0, v_0 \leftarrow 0, \hat{v}_0^{max} \leftarrow 0 \\ & \text{fort} \leftarrow 1 \dots n \\ & \quad \text{if max} \end{aligned}$$

2. 使用cross entropy作为评价指标，它的计算过程简单如下：

$$C^{**}E(p,q) = -i=1 \sum C p i l o^{**} g(q^{**}i)$$

3 一开始使用的是固定的学习率，但是效果并不怎么好，选择的是学习率递减的方式，依旧不怎么理想，最后选择了热启动（warm up）的方式，既能增加学习的速度，也能提高精度。

为了更加有好的显示训练过程，每次训练完，都会使用验证集做验证，分别是，对验证集不作处理，形态学腐蚀，形态学膨胀，形态学开处理，形态学闭处理

2.2 Django 将接口暴露给用户

2.2.1 Django 和 Django rest framework

Django 是一个高度集成的Python web框架，它主要可以用来快速的开发一个网站后端，它已经预先处理好了很多web开发比较麻烦的事情，所以用户只需要注重自己的需求就可以，而不用重复早轮子。

Django rest framework 是Django能够在如今前后端分离趋势下，依旧能够保持热度的原因所在，使用它可以轻松的完成序列化和反序列化。

2.2.2 设计模型

主要的实体有，用户，模型。

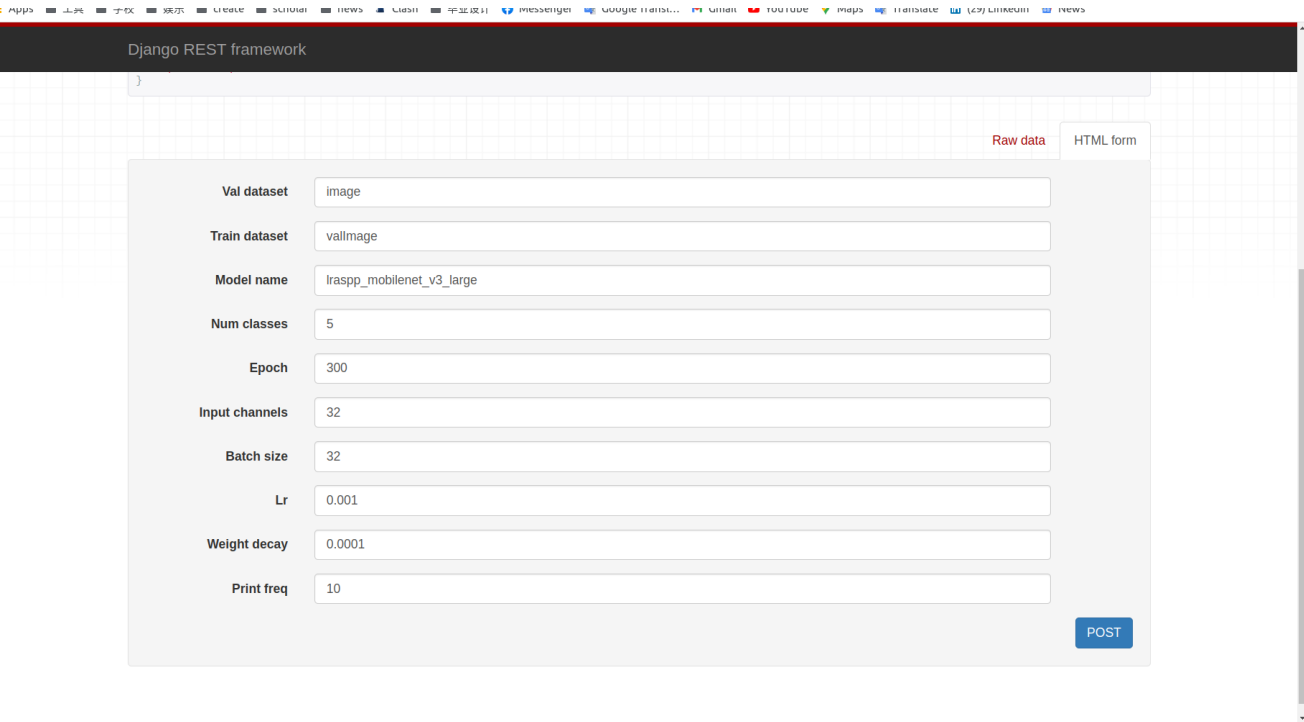
其中用户类有 姓名，手机号，邮箱，是否有权限等；模型类有模型名字，训练数据，验证数据，类别总数，输入通道数，batch size, epoch, 学习率， decay rate, 保存模型名字的前缀等。设计的实体图如下：

以后补充

2.2.3 使用celery异步进行模型训练

Celery 是一种简单、灵活、可靠的分布式系统，用来处理大量的信息，同时他也提供操作和维护这样一个系统所需要的工具。

因为训练的过程是比较消耗时间的，如果不适用异步功能，Django将完全堵塞在训练过程中，用户必须等待，这是非常不友好的，选择使用celery框架，当用户启动了训练过程后，就可以继续其他功能了。

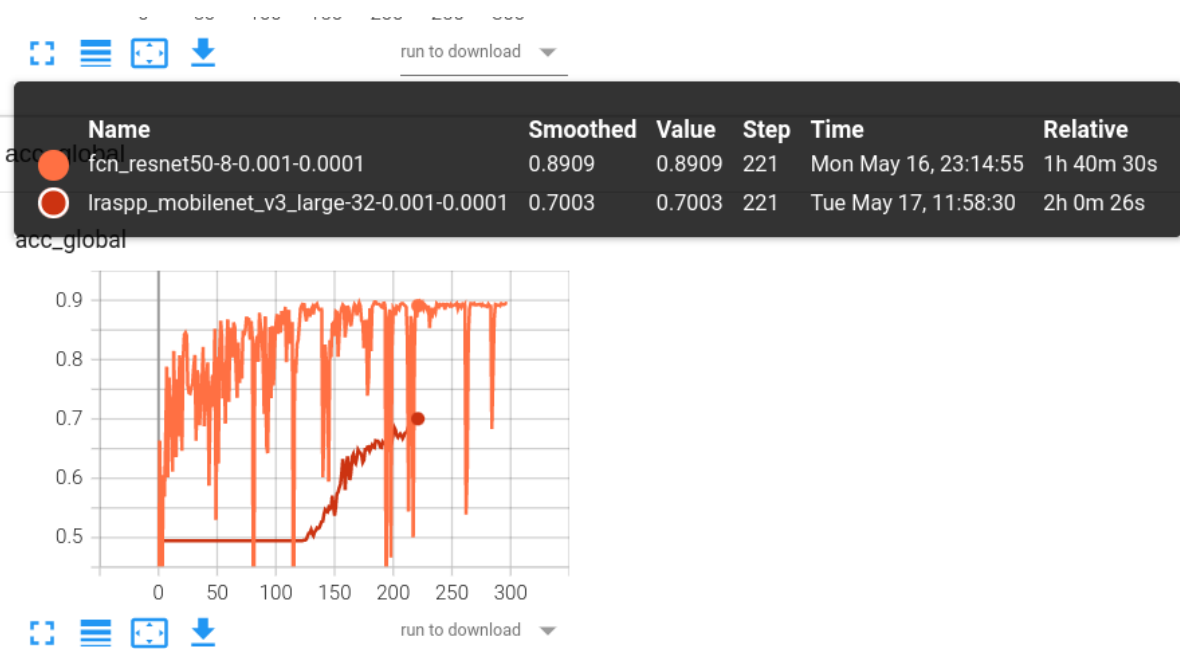


2.3 tensor board 数据可视化

TensorBoard是Google提供的机器学习的可视化工具，他可以跟踪可视化损失和准确率等指标，可以可视化模型，可以显示图片等。

我使用了tensorboard用来跟踪模型的训练过程，主要是训练的loss损失，验证集的预测精度和经过形态学处理后的精度等。

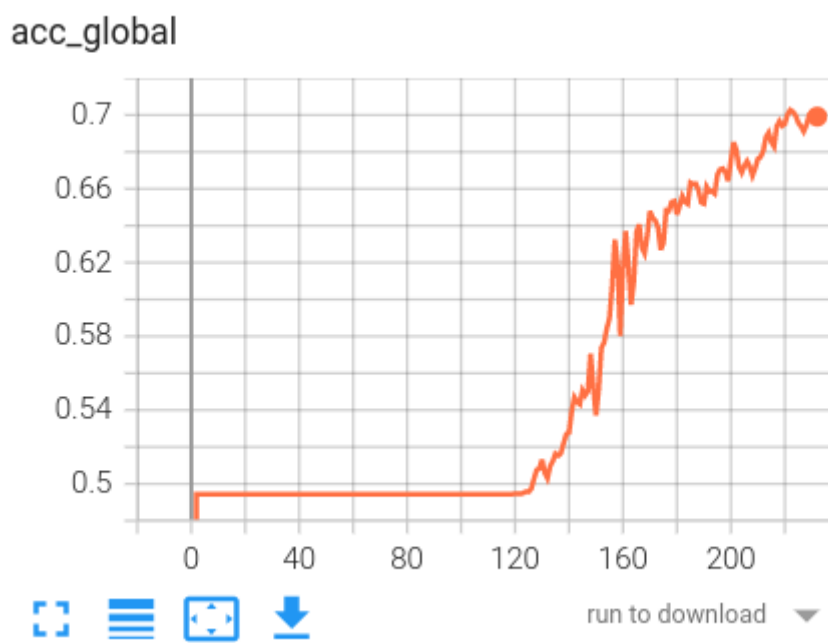
比如对比fcn和lrapp



欧比特数据形态学处理后

对比模型 lrassp:

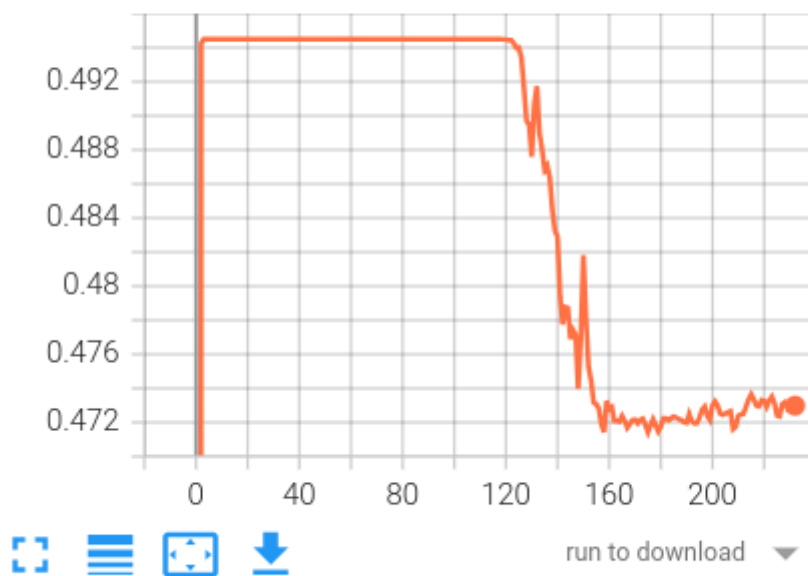
验证集精度，不做处理：



验证集精度，使用形态学处理：

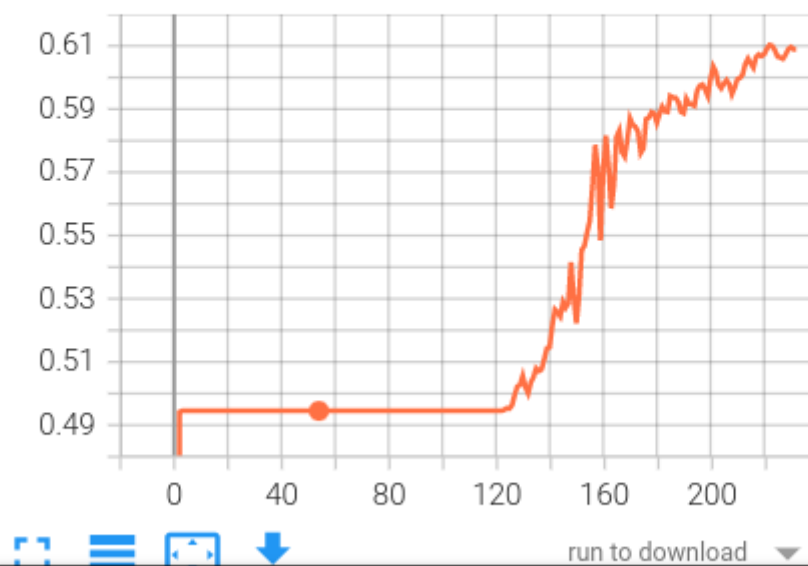
acc_globalclose

acc_globalclose



acc_globaldilate

acc_globaldilate

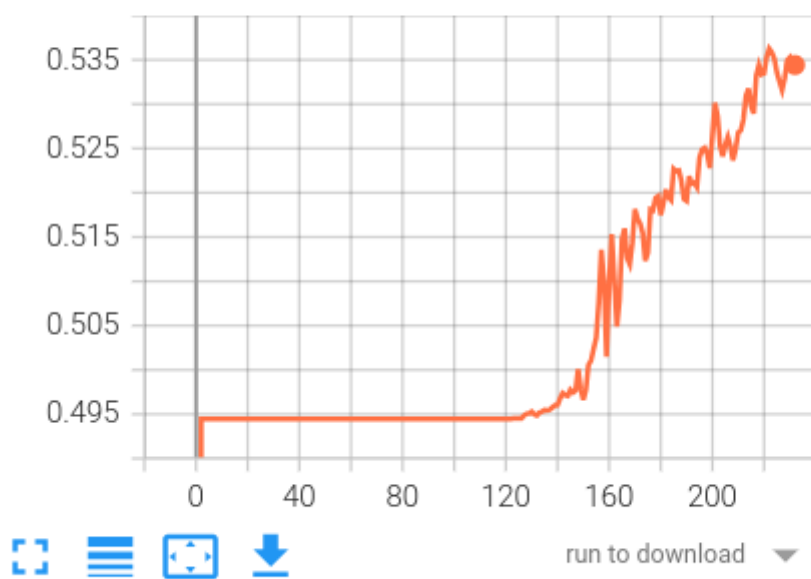


Name

Smoothed

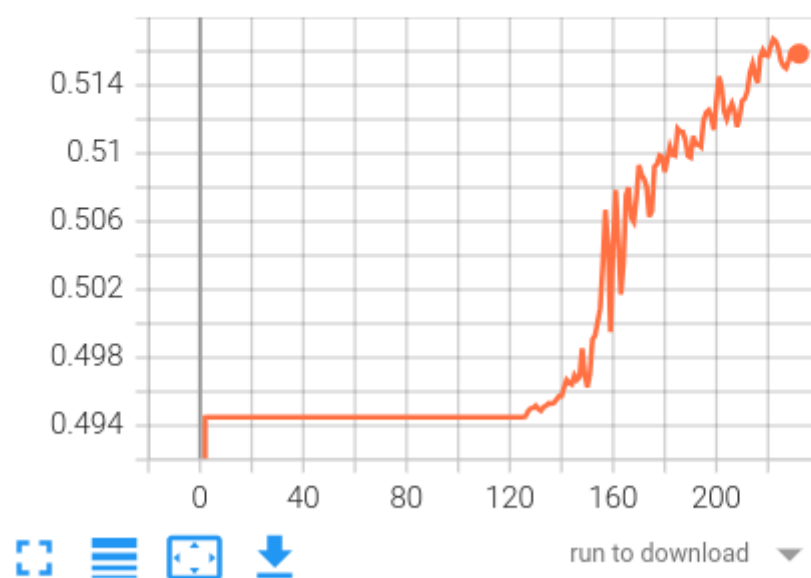
acc_globalerode

acc_globalerode



acc_globalopen

acc_globalopen



反而结果下降了。

结果分析：

为什么在印度松树上能提高呢？

印度松树使用的是patch classification, 它只是关注到了图像的部分信息，导致结果是有很多椒盐噪声的，所以使用形态学可以起到去除噪声的作用，

但是FCN和之后的网络，是关注了全局的信息，分割后的结果没有椒盐噪声，而是相对大块处理错误，所以我认为形态学只能应用在传统方法，而不能用在现有方法。

FCN网络本身就是使用一个函数预测正确结果，这个函数已经很接近正确结果了，而形态学仅仅粗暴的处理，所以实验的结果是合理的，唯一不合理的地方就是形态学就像玄学在预测结果做一次随机变换，但是居然找不到一次可以有比不做处理的要好的。