**kerasYolov3训练自己的数据**

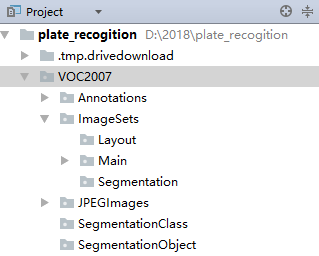
参考：

<https://blog.csdn.net/Patrick_Lxc/article/details/80615433>

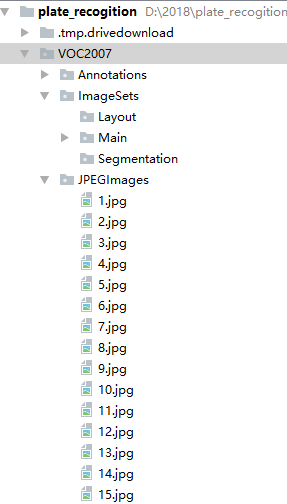
<https://blog.csdn.net/yuhq3/article/details/80281929>

**首先建立目录如下：**

工程Plate\_recogition下目录如下所示：



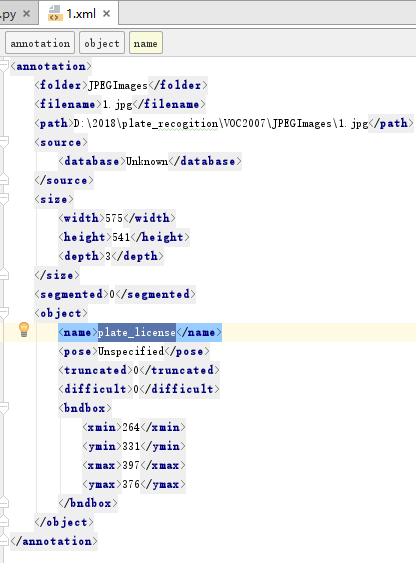
**图片都放入JEPGImages目录内：**



**生成XML文件，并放入Annotations目录内：**

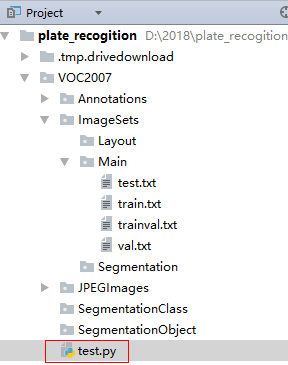
使用LabelImg工具生成xml文件，生成时和图片文件在同一目录内，都标注完成时，**将xml文件都复制到Annotations目录内**：

xml文件格式如下：



**生成Main目录的文件：**

在VOC2007目录下创建文件test.py



**test.py代码：**

import os

import random

trainval\_percent = 0.1

train\_percent = 0.9

xmlfilepath = 'Annotations'

txtsavepath = 'ImageSets\Main'

total\_xml = os.listdir(xmlfilepath)

num = len(total\_xml)

list = range(num)

tv = int(num \* trainval\_percent)

tr = int(tv \* train\_percent)

trainval = random.sample(list, tv)

train = random.sample(trainval, tr)

ftrainval = open('ImageSets/Main/trainval.txt', 'w')

ftest = open('ImageSets/Main/test.txt', 'w')

ftrain = open('ImageSets/Main/train.txt', 'w')

fval = open('ImageSets/Main/val.txt', 'w')

for i in list:

name = total\_xml[i][:-4] + '\n'

if i in trainval:

ftrainval.write(name)

if i in train:

ftest.write(name)

else:

fval.write(name)

else:

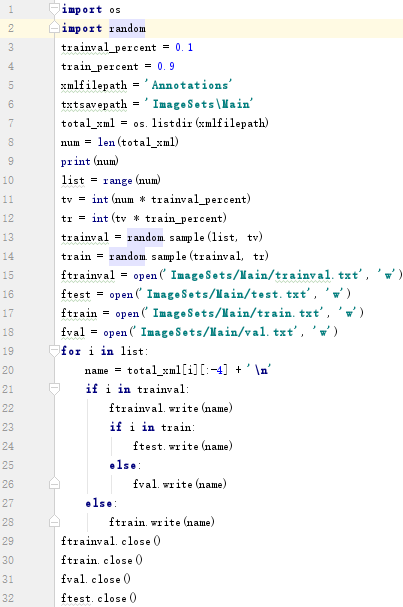
ftrain.write(name)

ftrainval.close()

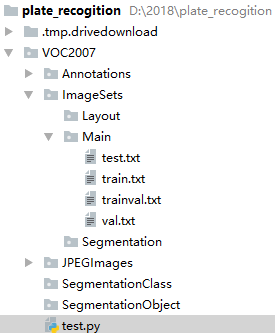
ftrain.close()

fval.close()

ftest.close()

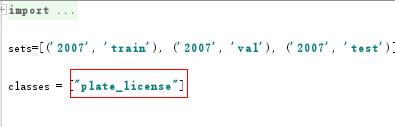


**运行后目录如下：**

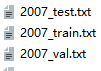


**生成yolov3所需的文件：**

先修改voc\_annotation.py文件：



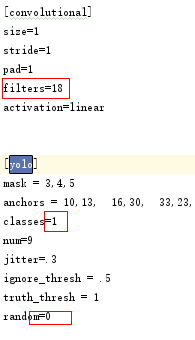
运行后在主目录会生成3个文件，修改后如下：



手动修改掉 “2007\_”

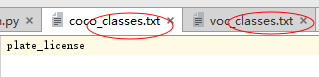
**修改yolo3.cfg文件：**

在文件中搜索yolo共有三处需要修改：



**注意：Filters= 3\*(5+classes)**

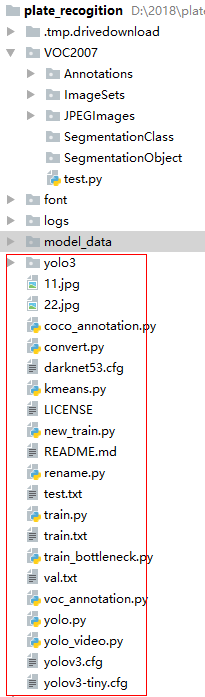
**修改model\_data目录下的文件：**



两个文件都修改。

**修改代码，准备训练：**

将keras-yolo3放入主目录，结构如下:



**注意：**

如果不想使用预训练的权重，train.py文件中，设置传给create\_model函数的参数load\_pretrained=False。

运行：train.py

报

**错误：**

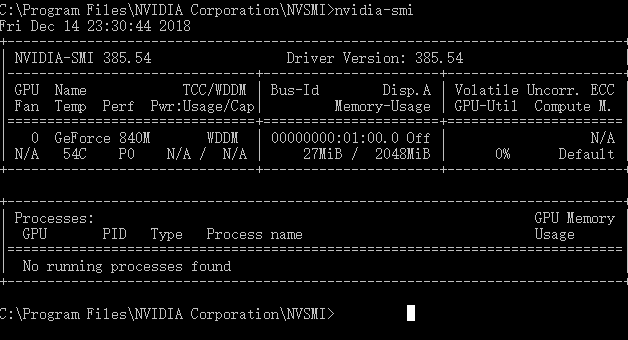
2018-12-14 22:46:14.233089: W tensorflow/core/common\_runtime/bfc\_allocator.cc:271] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\_

2018-12-14 22:46:14.297136: W tensorflow/core/framework/op\_kernel.cc:1273] OP\_REQUIRES failed at cwise\_ops\_common.h:245 : Resource exhausted: OOM when allocating tensor with shape[3,3,512,1024] and type float on /job:localhost/replica:0/task:0/device:GPU:0 by allocator GPU\_0\_bfc

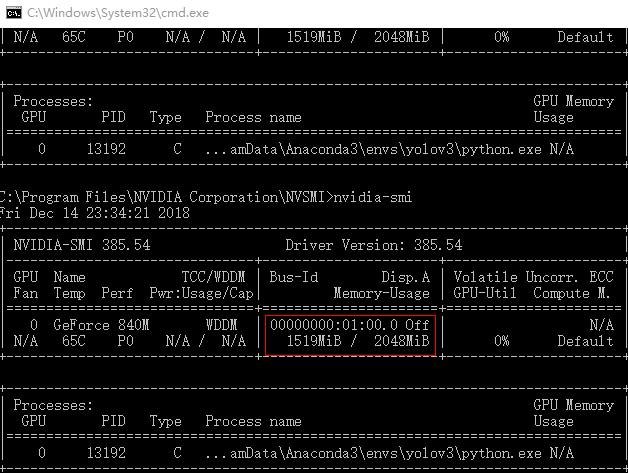
命令：nvidia-smi

将加入path系统变量；

**运行前或停止运行显示如下：**



**报错时显示GPU的内存如下：**



将batch\_size都改成了最小值1

运行仍然报错：

Create YOLOv3 model with 9 anchors and 1 classes.

Train on 49 samples, val on 5 samples, with batch size 1.

Epoch 1/10

2018-12-15 00:15:30.160351: W tensorflow/core/common\_runtime/bfc\_allocator.cc:211] Allocator (GPU\_0\_bfc) ran out of memory trying to allocate 831.81MiB. The caller indicates that this is not a failure, but may mean that there could be performance gains if more memory were available.

2018-12-15 00:15:40.164825: W tensorflow/core/common\_runtime/bfc\_allocator.cc:267] Allocator (GPU\_0\_bfc) ran out of memory trying to allocate 4.50MiB. Current allocation summary follows.

2018-12-15 00:15:40.165628: I tensorflow/core/common\_runtime/bfc\_allocator.cc:597] Bin (256): Total Chunks: 1655, Chunks in use: 1654. 413.8KiB allocated for chunks. 413.5KiB in use in bin. 28.3KiB client-requested in use in bin.

2018-12-15 00:15:40.166510: I tensorflow/core/common\_runtime/bfc\_allocator.cc:597] Bin (512): Total Chunks: 242, Chunks in use: 242. 121.0KiB allocated for chunks. 121.0KiB in use in bin. 121.0KiB client-requested in use in bin.

.......

.......

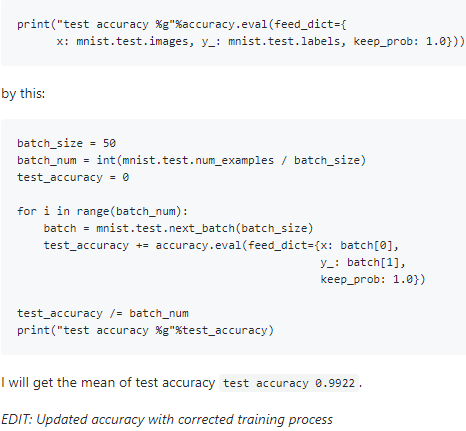
018-12-15 00:15:41.847415: W tensorflow/core/common\_runtime/bfc\_allocator.cc:271] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

2018-12-15 00:15:41.847838: W tensorflow/core/framework/op\_kernel.cc:1273] OP\_REQUIRES failed at cwise\_ops\_common.h:245 : Resource exhausted: OOM when allocating tensor with shape[3,3,256,512] and type float on /job:localhost/replica:0/task:0/device:GPU:0 by allocator GPU\_0\_bfc

查资料排除错误：

<https://github.com/tensorflow/tensorflow/issues/136>

其中有个分批喂



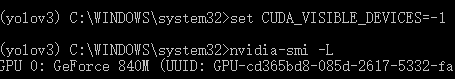
总之就是提示Gpu内存不够用，就用CPU来训练吧，慢就慢点吧

**设置禁用GPU：**

查看可用的GPU：nvidia-smi -L



**Set CUDA\_VISIBLE\_DEVICES=-1 好像没起作用。**



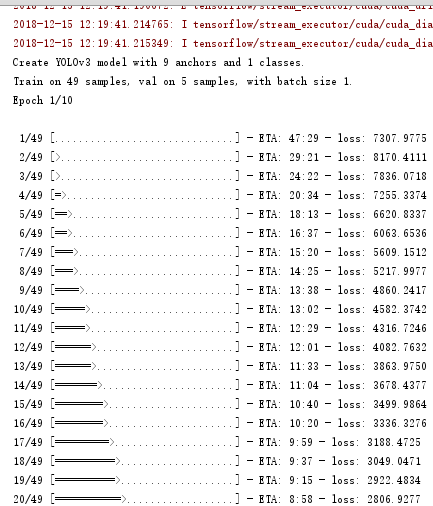
Set CUDA\_VISIBLE\_DEVICES=n ； n 分别是0，1，2 的时候，则分别代表该编号的GPU可以用，如果同时设置这3个数，则代表这三个同时可用。

代码里设置吧：



代码里加入如下：

**import** os  
os.environ[**"CUDA\_VISIBLE\_DEVICES"**] = **"-1"**



设置GPU禁用，使用CPU训练正常。

**测试训练效果（60张车牌训练）：**

C:\ProgramData\Anaconda3\envs\yolov3\python.exe yolo\_video.py --image --input ‘’

命令行状态下提示如下：

**Input image filename:**4.jpg （红色为输入的图片，图片放根目录）

(416, 416, 3)

...





**200张车牌训练后的效果：**



