一．下载 yolov3 工程项目

git clone https://github.com/pjreddie/darknet
cd darknet
将 makefile 的相应部分参数改为 1，
#如果使用 GPU 设置为 1，CPU 设置为 0,GPU=0
#如果使用 CUDNN 设置为 1，否则为 0,CUDNN=0
#如果使用 OPENCV 设置为 1，否则为 0,OPENCV=1
修改后保存，再 make 一下。
将官网的预训练权重先下载到 darknet 下

二．准备训练数据集

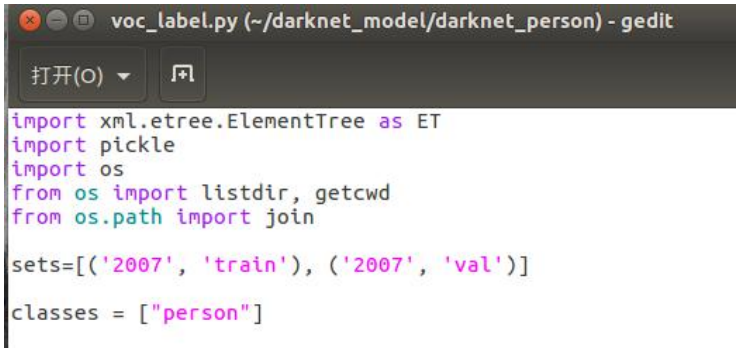
按照下面文件夹的结构，将训练数据集放到各个文件夹下面，生成 2 个文件一个 train.txt, 一个是 val.txt。这里使用到的脚本为 getfile.py，

VOCdevkit
—VOC2012
——Annotations
——ImageSets
———Layout
———Main
———Segmentation
——JPEGImages

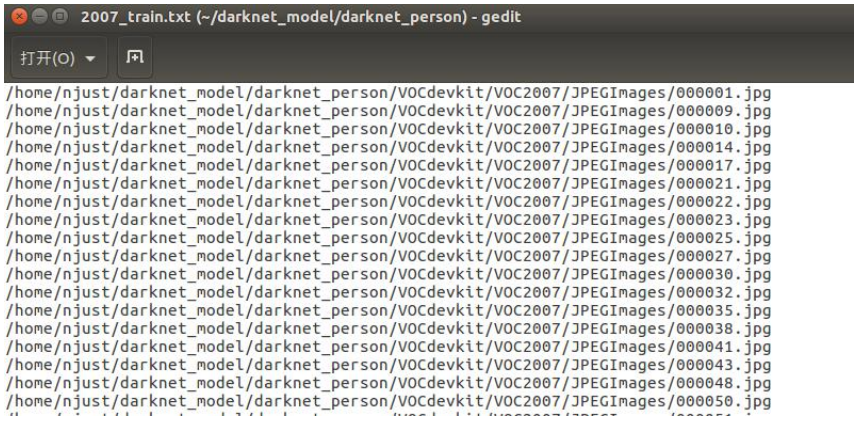
注意：其中 Annotations 中是所有的 xml 文件
JPEGImages 中是所有的训练文件
Main 中是 2 个 txt 文件：train.txt 与 val.txt 文件（前一个用来验证，后一个用来测试）

三．生成 2007_train.txt 和 2007_val.txt 文件

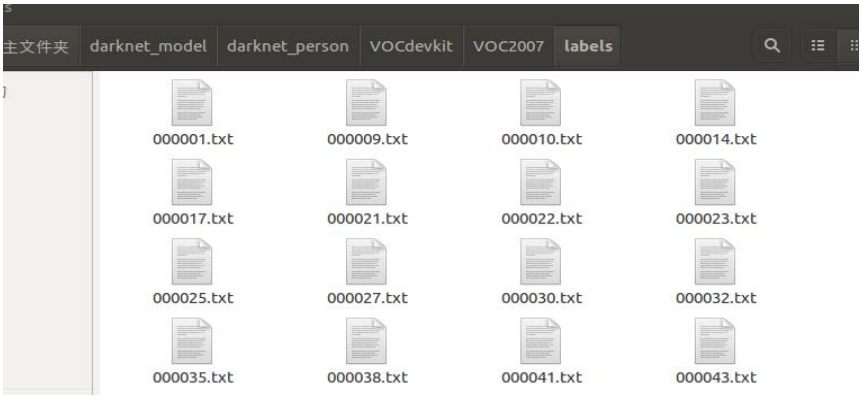
下载 voc_label.py，或者直接在 scripts 中找到 voc_labels.py 将该文件与 VOCdevkit 数据集放到同一级路径下。
首先修改 voc_label.py 里面的值：
修改 sets 为自己训练样本集的名称，以及 classes 为训练样本集的类标签

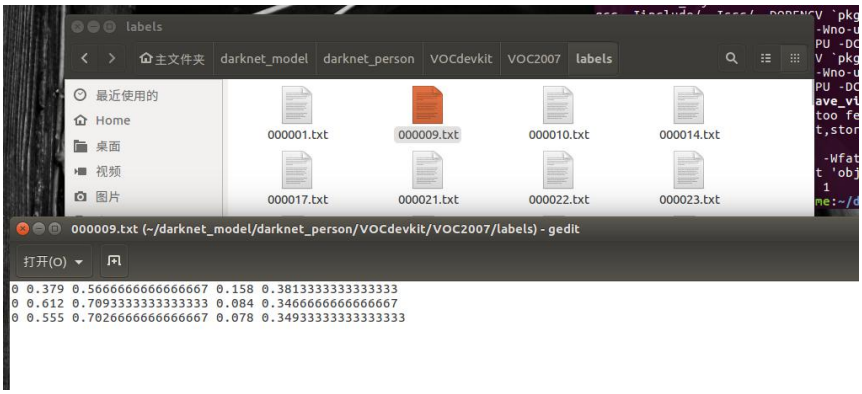


文件最后两行注释掉。
在当前终端运行 python voc_label.py
生成
① 训练和验证的文件列表（2007_train.txt 和 2007_val.txt）主要存储的是图片位置信息；



② 生成与 Annotations 等同级的 label 文件，label 里面的文件是每张图片位置坐标以及类别的标签信息。





四．修改 cfg/voc.data 文件

#classes 为训练样本集的类别总数：classes= 1

#train 的路径为训练样本集所在的路径

train = /home/njust/darknet_model/darknet_person/2007_train.txt

#valid 的路径为验证样本集所在的路径

valid = /home/njust/darknet_model/darknet_person/2007_val.txt

#names 的路径为 data/voc.names 文件所在的路径

names = data/voc.names

backup = backup

```
classes= 1
train  = /home/njust/darknet_model/darknet_person/2007_train.txt
valid  = /home/njust/darknet_model/darknet_person/2007_val.txt
names = data/voc.names
backup = backup
```

五．在 darknet 文件夹下面新建文件夹 backup
六．修改 voc.name 为样本集的标签名
七．修改 cfg/yolov3-voc.cfg
　　这里我只是以一类目标检测为例，主要有 4 处地方调整：

```
[net]
# Testing
# batch=1
# subdivisions=1
# Training
  batch=64

  subdivisions=16

......

[convolutional]
size=1
stride=1
pad=1
filters=18###75
activation=linear

[yolo]
mask = 6,7,8
anchors = 10,13,  16,30,  33,23,  30,61,  62,45,  59,119,  116,90,  156,198,  373,326
classes=1###20
num=9
jitter=.3
ignore_thresh = .5
truth_thresh = 1
random=0###1
......

[convolutional]
size=1
stride=1
pad=1
filters=18###75
activation=linear

[yolo]
mask = 3,4,5
anchors = 10,13,  16,30,  33,23,  30,61,  62,45,  59,119,  116,90,  156,198,  373,326
```

```
classes=1###20
num=9
jitter=.3
ignore_thresh = .5
truth_thresh = 1
random=0###1
......

[convolutional]
size=1
stride=1
pad=1
filters=18###75
activation=linear

[yolo]
mask = 0,1,2
anchors = 10,13,  16,30,  33,23,  30,61,  62,45,  59,119,  116,90,  156,198,  373,326
classes=1###20
num=9
jitter=.3
ignore_thresh = .5
truth_thresh = 1
random=0###1
```

A.filters 数目是怎么计算的：3x(classes 数目+5)，和聚类数目分布有关，论文中有说明；

B.如果想修改默认 anchors 数值，使用 k-means 即可；

C.如果显存很小，将 random 设置为 0，关闭多尺度训练；

D.其他参数基本与 V2 一致，不再说明;

E.前 100 次迭代 loss 较大，后面会很快收敛；

八．开始训练

```
./darknet detector train cfg/voc.data cfg/yolov3-voc.cfg darknet53.conv.74
```

九．识别

```
./darnket detector test cfg/voc.data cfg/yolov3-voc.cfg backup/yolov3-voc.backup data/test.jpg
```

常见问题解决：

1、什么时候停止训练：

通常每个类（对象）有足够的 2000 次迭代。但是，当你应该停止训练时，为了更准确的定义，请使用以下指标：

Region Avg IOU: 0.798363, Class: 0.893232, Obj: 0.700808, No Obj: 0.004567, Avg Recall: 1.000000, count: 8 Region Avg IOU: 0.800677, Class: 0.892181, Obj: 0.701590, No Obj: 0.004574, Avg Recall: 1.000000, count: 8

9002: 0.211667, **0.060730 avg**, 0.001000 rate, 3.868000 seconds, 576128 images Loaded: 0.000000 seconds

9002：是迭代次数

0.060730：是平均损失，当看到平均损失 0.xxxxx avg 在许多吃迭代不再减少时，应该停止训练。一旦停止训练，可以得到最后的.weights 文件，可以在 backup 里面找到最佳的文件，也可以使用 yolov3-voc.backup 这个文件。

2、暂停训练 ctrl+z、停止训练 ctrl+c 、暂停回来按住 fg

安芯　20181120