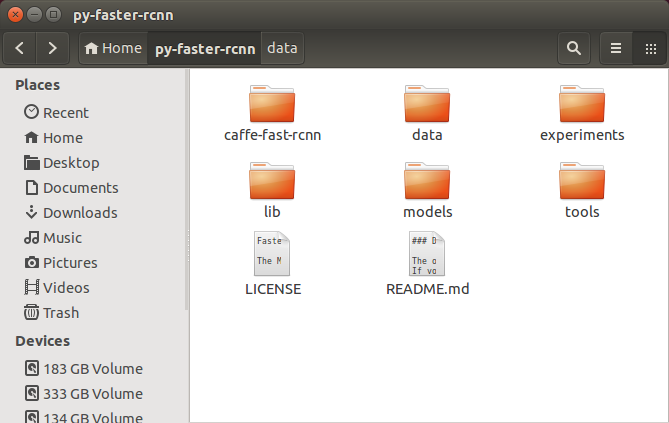
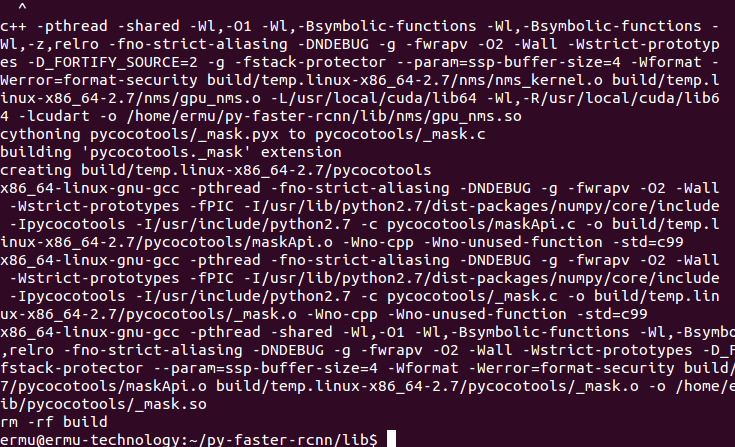
# faster rcnn 的配置和训练

## 1.下载py-faster-rcnn并解压：

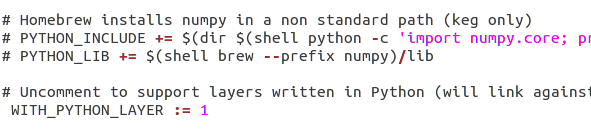


## 2.cd到lib目录下,执行 make



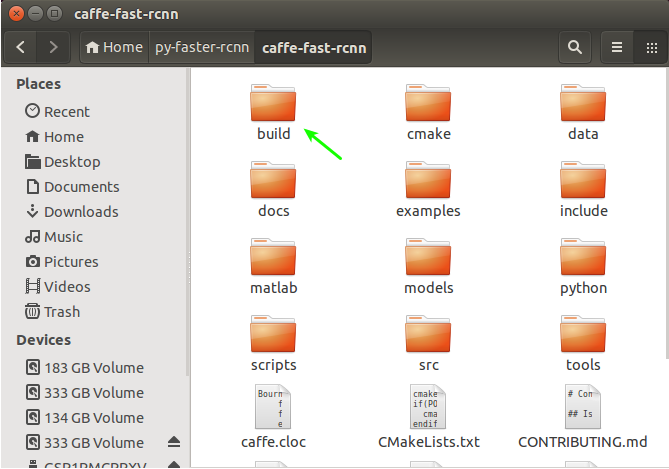
## 3.编译caffe-fast-rcnn

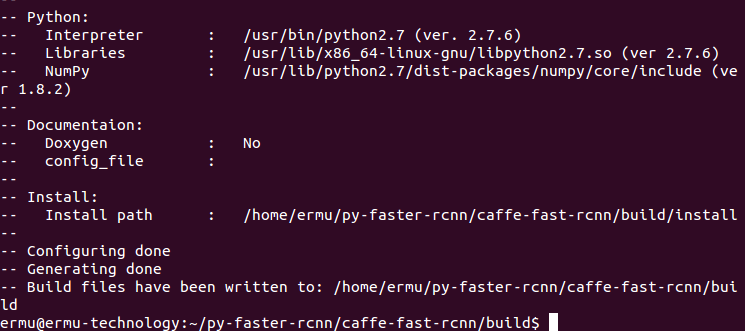
1) cd到caffe-fast-rcnn目录下，将Makefile.config.example文件重命名Makefile.config，去掉WITH\_PYTHON\_LAYER前面的#号：



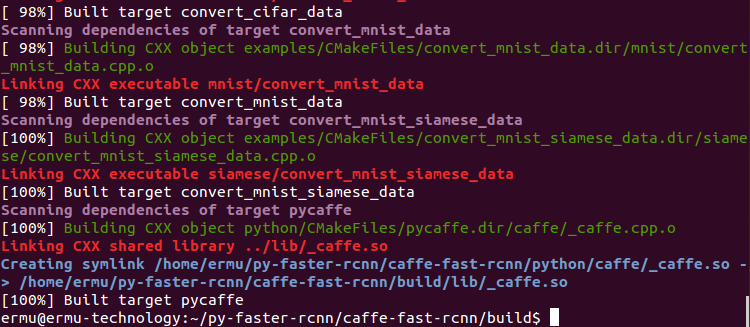
保存。

1. 在caffe-fast-rcnn目录下新建一个build文件夹, cd到该目录下，执行cmake ..

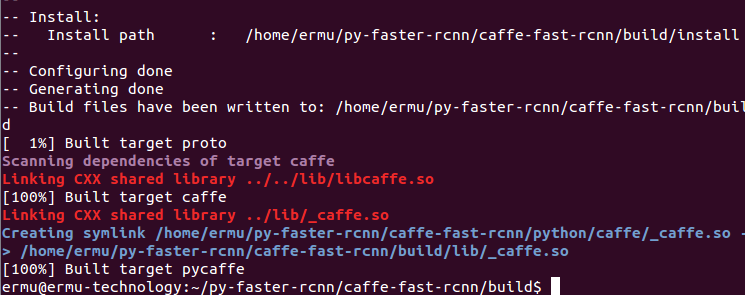




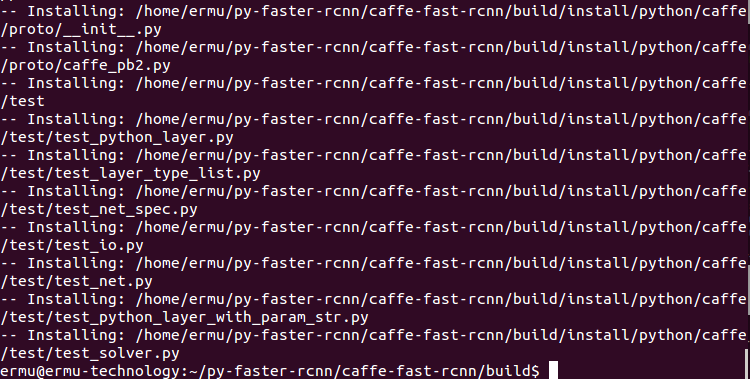
1. 执行make all



1. 执行make pycaffe

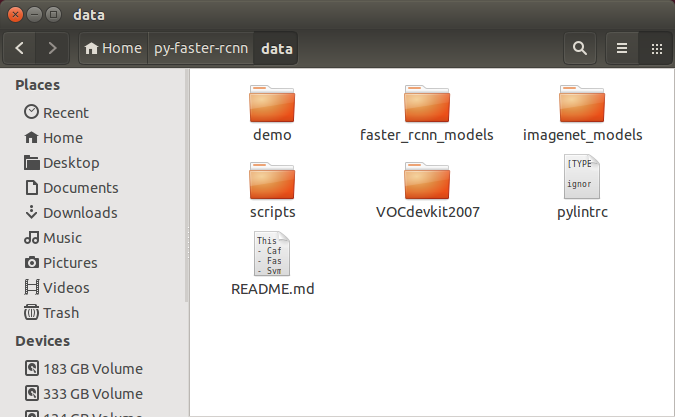


1. 执行sudo make install

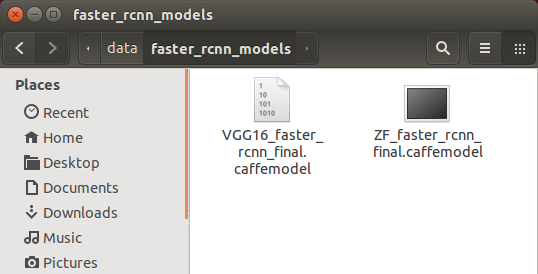


## 下载faster\_rcnn\_models,imagenet\_models和VOCdevkit2007

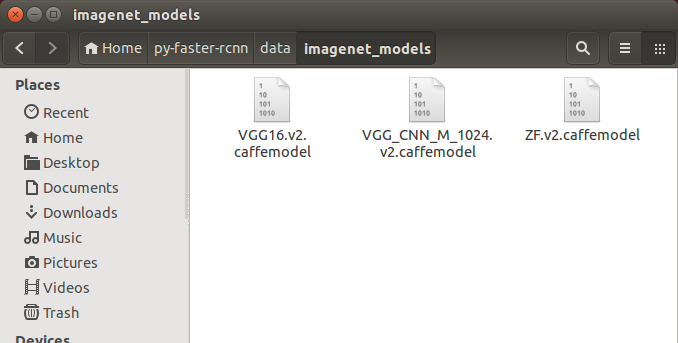
这一步，我是在windows上下载，然后复制到放置到py-faster-rcnn/data目录下的



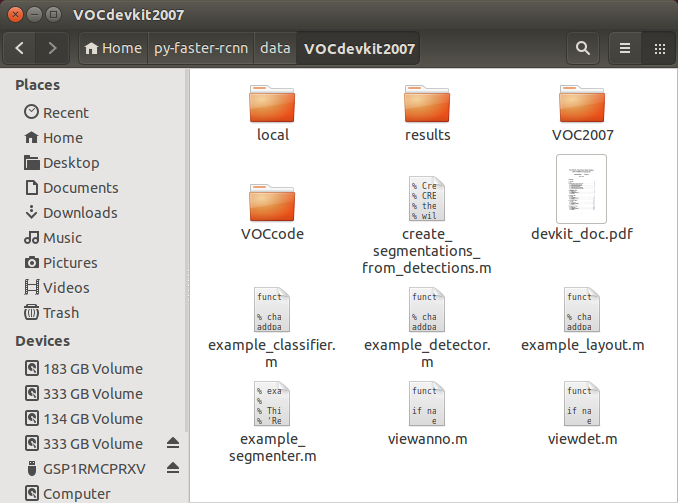
faster\_rcnn\_models目录如下：



Imagenet\_models目录如下：

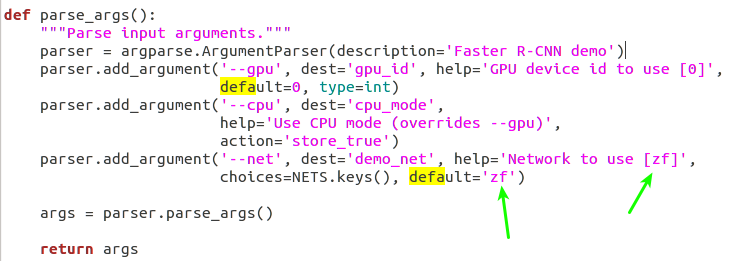


VOCdevkit2007目录如下：

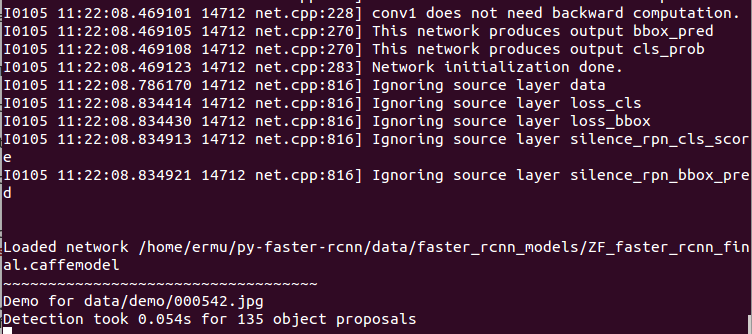


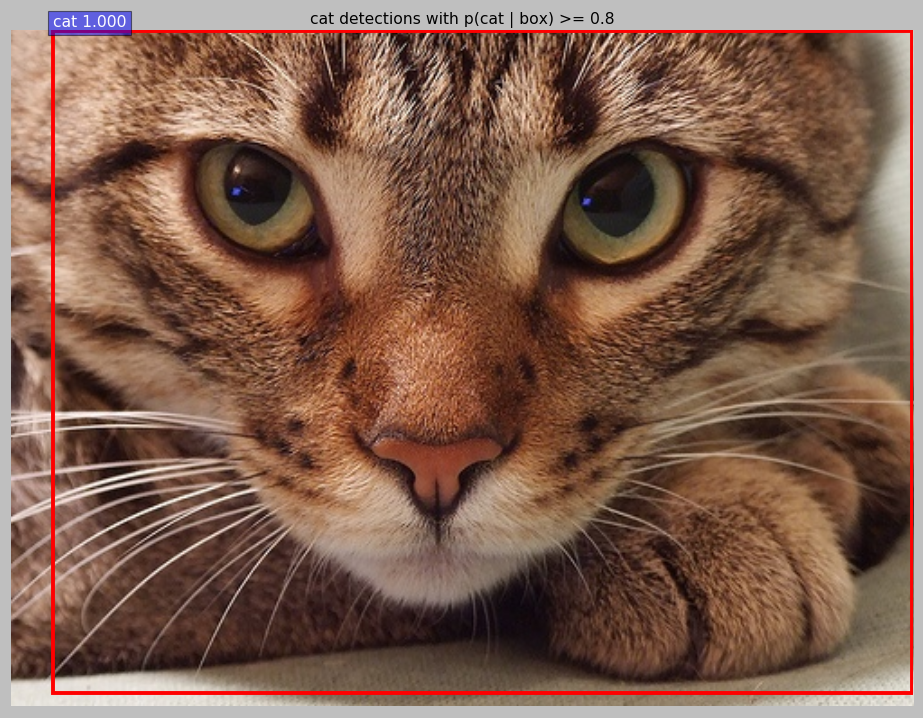
## 测试

这里选择zf网络进行测试，为了方便，我对tools/demo.py文件中将默认网络从vgg16改成zf，且只对一张图像进行测试



在tools目录下执行python demo.py

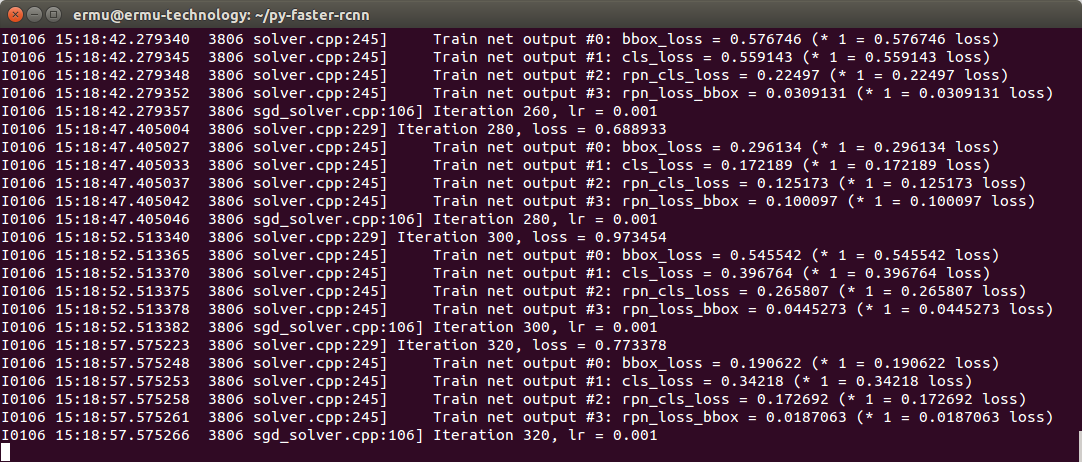




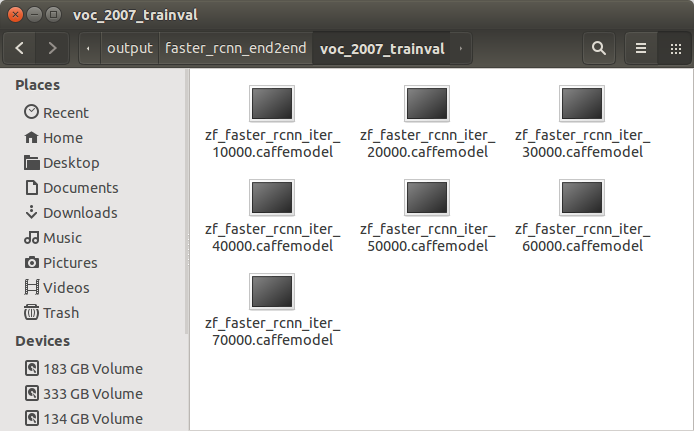
## 训练

在py-faster-rcnn目录下执行：

python ./tools/train\_net.py --gpu 0 --solver models/pascal\_voc/ZF/faster\_rcnn\_end2end/solver.prototxt --weights data/imagenet\_models/ZF.v2.caffemodel --imdb voc\_2007\_trainval --iters 70000 --cfg experiments/cfgs/faster\_rcnn\_end2end.yml

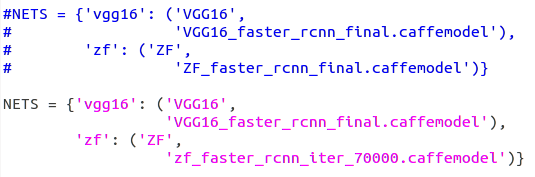


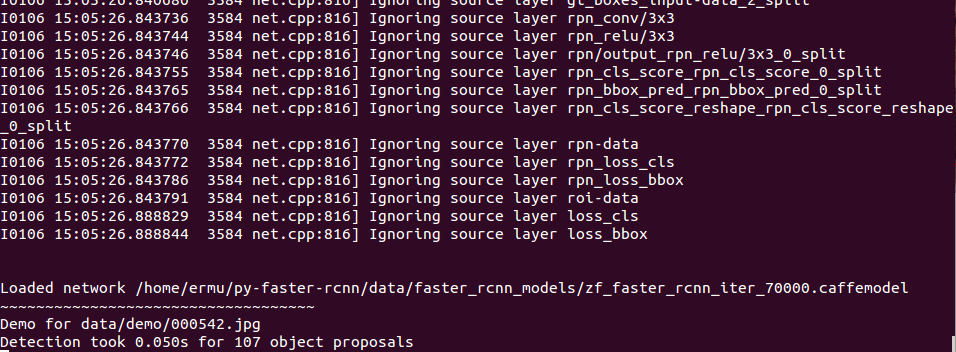
训练结果：

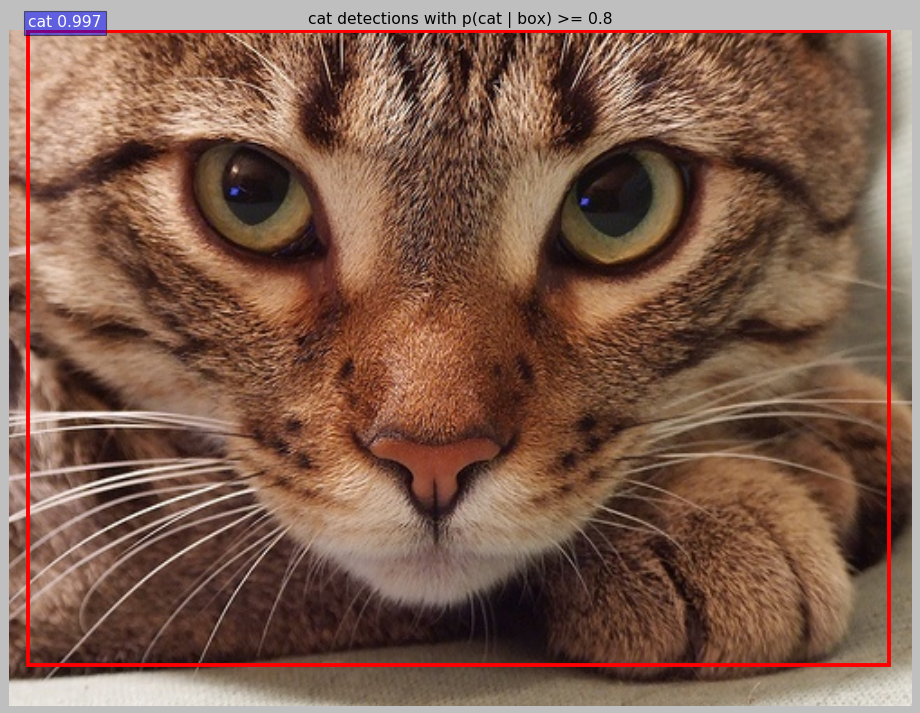


## 再测试

将训练结果zf\_faster\_rcnn\_iter\_70000.caffemodel拷贝到data/faster\_rcnn\_models目录下，并在demo.py文件中修改相应的模型名称，在tools目录下再次执行python demo.py：







训练成功！