



### 笔记本以旧换新





**厂家直销 高品质**

**¥0.80/条**

**恒**

访问：440923次

积分：4092

等级：**BLOG > 5**

排名：第8515名

原创：104篇      转载：21篇

译文：0篇      评论：1267条

文章搜索

- 文章分类
- caffe (25)

C++ (46)

leetcode (1)

Deep Learning (8)

Algorithms (10)

python (4)

HDUOJ (1)

数位DP (1)

hihoCoder (3)

POJ (2)

Linux (6)

MFC (1)

计算机网络 (2)

yolo (2)

darknet (2)

目标检测 (3)

windows (1)

C (9)

获取文件夹下所有文件名 (1)

R-FCN (1)

设计模式 (1)

编程珠玑 (3)

☰ 目录视图

☰ 摘要视图

☰ 收藏

图灵赠书——程序员11月书单

【思考】Python这么厉害的原因竟然是！

感恩节赠书：《深度学习》和作译者评选启动！

每周荐书：京东架构、Linux内核、Python全栈

优秀

**[置顶] Faster-RCNN+ZF用自己的数据集训练模型(Python版本)**

2016-05-06 17:33      69587人阅读      评论(156)

☰ 分类：

Deep Learning ( 7 )    caffe ( 24 )

版权声明：本文为博主原创文章，未经博主允许不得转载。

目录(?)    [+]

说明：本博文假设你已经做好了数据集，该数据集格式和VOC2007相同。下面是训练前的一些修改。

( 做数据集的过程可以看[http://blog.csdn.net/sinat\\_30071459/article/details/50723212](http://blog.csdn.net/sinat_30071459/article/details/50723212) )

Faster-RCNN源码下载地址：

Matlab版本：[https://github.com/ShaoqingRen/faster\\_rcnn](https://github.com/ShaoqingRen/faster_rcnn)

**Python**版本:<https://github.com/rbgirshick/py-faster-rcnn>

本文用到的是**Python**版本，在**Linux**下运行。

Matlab版本的训练过程：[http://blog.csdn.net/sinat\\_30071459/article/details/50546891](http://blog.csdn.net/sinat_30071459/article/details/50546891)

准备工作：

1.配置caffe

这个不多说，网上教程很多。

2.其他的注意事项

这里说的挺详细了，认真看看吧。地址：<https://github.com/rbgirshick/py-faster-rcnn>（主要内容如下）

下面大概翻译一下上面网址的内容吧。

( 1 ) 安装cython, python-opencv,easydict

[plain]

01. pip install cython  
02. pip install easydict  
03. apt-get install python-opencv

( 2 ) 下载py-faster-rcnn

[plain]

01. # Make sure to clone with --recursive  
02. git clone --recursive https://github.com/rbgirshick/py-faster-rcnn.git

http://blog.csdn.net/sinat\_30071459/article/details/51332084

1/17



### 笔记本以旧换新







### 厂家直销 高品质

¥0.80/条

YOLO(v1)用自己的数据集训...

(13674)

Windows下用VS2013加载ca...

(11353)

Faster-RCNN+VGG用自己的...

(11116)

基于caffe的图像分类(3)——...

(8953)

基于caffe的图像分类(1)——...

(8607)

#### 评论排行

将数据集做成VOC2007格式...

(421)

Faster-RCNN+ZF用自己的数...

(393)

Faster-RCNN+ZF用自己的数...

(156)

YOLO(v1)用自己的数据集训...

(60)

Windows下用VS2013加载ca...

(57)

封装yolo-windows为动态链...

(43)

R-FCN+ResNet-50用自己的...

(42)

微调CaffeNet用于车颜色识别

(16)

Windows下用Matlab加载caf...

(15)

封装caffe-windows-master...

(12)

#### 推荐文章

\* 【2017年11月27日】CSDN博客更新周报

\* 【CSDN】邀请您来GitChat赚钱啦！

\* 【GitChat】精选——JavaScript进阶指南

\* 改做人工智能之前，90%的人都没能给自己定位

\* TensorFlow 人脸识别网络与对抗网络搭建

\* Vue 移动端项目生产环境优化

\* 面试必考的计算机网络知识点梳理

#### 最新评论

Faster-RCNN+ZF用自己的数据集训练模...

cx2214496 : @mayiyi529:楼主您好，我想问一下我只检测一种 ap怎么算？而且那个loss曲线怎么绘制 ...

Faster-RCNN+ZF用自己的数据集训练模...

cx2214496 : 博主很感谢你分享这篇文章，我两个问题（我是入门水平），1：训练出来

如图：

```
~$ git clone --recursive https://github.com/rbgirshick/py-faster-rcnn.git
remote: Counting objects: 1526, done.
remote: Total 1526 (delta 0), reused 0 (delta 0), pack-reused 1526
接收对象中: 100% (1526/1526), 1.34 MiB | 537.00 KiB/s, done.
处理 delta 中: 100% (900/900), done.
检查连接... 完成。
子模组 'caffe-fast-rcnn' (https://github.com/rbgirshick/caffe-fast-rcnn.git) 未
对路径 'caffe-fast-rcnn' 注册
正克隆到 'caffe-fast-rcnn'...
remote: Counting objects: 23976, done.
remote: Compressing objects: 100% (2/2), done.
接收对象中: 15% (3597/23976), 380.01 KiB | 373.00 KiB/s
```

(3) 进入py-faster-rcnn/lib

执行make

如图：

```
~/py-faster-rcnn/lib$ make
python setup.py build_ext --inplace
running build_ext
cythoning utils/bbox.pyx to utils/bbox.c
building 'utils.cython_bbox' extension
creating build
creating build/temp.linux-x86_64-2.7
creating build/temp.linux-x86_64-2.7/utils
x86_64-linux-gnu-gcc -pthread -fno-strict-aliasing -DNDEBUG -g -fwrapv -O2 -Wall
-Wstrict-prototypes -fPIC -I/usr/lib/python2.7/dist-packages/numpy/core/include
-I/usr/include/python2.7 -c utils/bbox.c -o build/temp.linux-x86_64-2.7/utils/b
box.o -Wno-cpp -Wno-unused-function
x86_64-linux-gnu-gcc -pthread -shared -Wl,-O1 -Wl,-Bsymbolic-functions -Wl,-Bsym
bolic-functions -Wl,-z,relro -fno-strict-aliasing -DNDEBUG -g -fwrapv -O2 -Wall
-Wstrict-prototypes -D_FORTIFY_SOURCE=2 -g -fstack-protector --param=ssp-buffe
r-size=4 -Wformat -Werror=format-security build/temp.linux-x86_64-2.7/utils/bbox.o
-o /home/lujie/py-faster-rcnn/lib/utils/cython_bbox.so
cythoning nms/cpu_nms.pyx to nms/cpu_nms.c
building 'nms.cpu_nms' extension
creating build/temp.linux-x86_64-2.7/nms
x86_64-linux-gnu-gcc -pthread -fno-strict-aliasing -DNDEBUG -g -fwrapv -O2 -Wall
-Wstrict-prototypes -fPIC -I/usr/lib/python2.7/dist-packages/numpy/core/include
-I/usr/include/python2.7 -c nms/cpu_nms.c -o build/temp.linux-x86_64-2.7/nms/cp
u_nms.o -Wno-cpp -Wno-unused-function
```

(4) 进入py-faster-rcnn\caffe-fast-rcnn

执行 cp Makefile.config.example Makefile.config

然后，配置Makefile.config文件，可参考我的配置：[Makefile.config文件](#)

配置好Makefile.config文件后，执行：

```
[plain]
01. make -j8 && make pycaffe
```

如图：

```
~/py-faster-rcnn/caffe-fast-rcnn$ make -j8 && make pycaff
NVCC src/caffe/layers/cudnn_tanh_layer.cu
NVCC src/caffe/layers/cudnn_pooling_layer.cu
NVCC src/caffe/layers/relu_layer.cu
NVCC src/caffe/layers/dropout_layer.cu
NVCC src/caffe/layers/absval_layer.cu
NVCC src/caffe/layers/ln2col_layer.cu
NVCC src/caffe/layers/cudnn_lrn_layer.cu
NVCC src/caffe/layers/eltwise_layer.cu
NVCC src/caffe/layers/bnll_layer.cu
NVCC src/caffe/layers/elu_layer.cu
NVCC src/caffe/layers/tile_layer.cu
NVCC src/caffe/layers/cudnn_sigmoid_layer.cu
NVCC src/caffe/layers/scale_layer.cu
NVCC src/caffe/layers/threshold_layer.cu
NVCC src/caffe/layers/euclidean_loss_layer.cu
NVCC src/caffe/layers/smooth_l1_loss_layer.cu
NVCC src/caffe/layers/cudnn_relu_layer.cu
NVCC src/caffe/layers/btas_layer.cu
NVCC src/caffe/layers/embed_layer.cu
NVCC src/caffe/layers/base_data_layer.cu
NVCC src/caffe/layers/slice_layer.cu
NVCC src/caffe/layers/contrastive_loss_layer.cu
```

(5) 下载VOC2007数据集

提供一个百度云地址：<http://pan.baidu.com/s/1mhMKKw4>

解压，然后，将该数据集放在py-faster-rcnn\data下，用你的数据集替换VOC2007数据集。（替换Annotations，ImageSets和JPEGImages）



(用你的Annotations, ImagesSets和JPEGImages替换py-faster-rcnn\data\VOCdevkit2007\VOC2007中对应文件夹)

(6) 下载ImageNet数据集下预训练得到的模型参数(用来初始化)

提供一个百度云地址: <http://pan.baidu.com/s/1hsxx8OW>

解压, 然后将该文件放在py-faster-rcnn\data下

下面是训练前的一些修改。

## 1.py-faster-

rcnn/models/pascal\_voc/ZF/faster\_rcnn\_alt\_opt/stage1\_fast\_rcnn\_caffe.pb

### 修改

```
[plain]
01. layer {
02.   name: 'data'
03.   type: 'Python'
04.   top: 'data'
05.   top: 'rois'
06.   top: 'labels'
07.   top: 'bbox_targets'
08.   top: 'bbox_inside_weights'
09.   top: 'bbox_outside_weights'
10.   python_param {
11.     module: 'roi_data_layer.layer'
12.     layer: 'RoIDataLayer'
13.     param_str: "'num_classes': 16" #按训练集类别改, 该值为类别数+1
14.   }
15. }
```

```
[plain]
01. layer {
02.   name: "cls_score"
03.   type: "InnerProduct"
04.   bottom: "fc7"
05.   top: "cls_score"
06.   param { lr_mult: 1.0 }
07.   param { lr_mult: 2.0 }
08.   inner_product_param {
09.     num_output: 16 #按训练集类别改, 该值为类别数+1
10.     weight_filler {
11.       type: "gaussian"
12.       std: 0.01
13.     }
14.     bias_filler {
15.       type: "constant"
16.       value: 0
17.     }
18.   }
19. }
```

```
[plain]
01. layer {
02.   name: "bbox_pred"
03.   type: "InnerProduct"
04.   bottom: "fc7"
05.   top: "bbox_pred"
06.   param { lr_mult: 1.0 }
07.   param { lr_mult: 2.0 }
08.   inner_product_param {
09.     num_output: 64 #按训练集类别改, 该值为(类别数+1)*4
10.     weight_filler {
11.       type: "gaussian"
12.       std: 0.001
13.     }
14.   }
15. }
```



```

14.         bias_filler {
15.             type: "constant"
16.             value: 0
17.         }
18.     }
19. }

```

## 2.py-faster-

rcnn/models/pascal\_voc/ZF/faster\_rcnn\_alt\_opt/stage1\_rpn\_t

```

[plain]
01. layer {
02.     name: 'input-data'
03.     type: 'Python'
04.     top: 'data'
05.     top: 'im_info'
06.     top: 'gt_boxes'
07.     python_param {
08.         module: 'roi_data_layer.layer'
09.         layer: 'RoIDataLayer'
10.         param_str: "'num_classes': 16" #按训练集类别改, 该值为类别数+1
11.     }
12. }

```

## 3.py-faster-

rcnn/models/pascal\_voc/ZF/faster\_rcnn\_alt\_opt/stage2\_fast\_rcnn\_train.pt

修改

```

[plain]
01. layer {
02.     name: 'data'
03.     type: 'Python'
04.     top: 'data'
05.     top: 'rois'
06.     top: 'labels'
07.     top: 'bbox_targets'
08.     top: 'bbox_inside_weights'
09.     top: 'bbox_outside_weights'
10.     python_param {
11.         module: 'roi_data_layer.layer'
12.         layer: 'RoIDataLayer'
13.         param_str: "'num_classes': 16" #按训练集类别改, 该值为类别数+1
14.     }
15. }

```

```

[plain]
01. layer {
02.     name: "cls_score"
03.     type: "InnerProduct"
04.     bottom: "fc7"
05.     top: "cls_score"
06.     param { lr_mult: 1.0 }
07.     param { lr_mult: 2.0 }
08.     inner_product_param {
09.         num_output: 16 #按训练集类别改, 该值为类别数+1
10.         weight_filler {
11.             type: "gaussian"
12.             std: 0.01
13.         }
14.         bias_filler {
15.             type: "constant"
16.             value: 0
17.         }
18.     }
19. }

```



[plain]

```

01. layer {
02.   name: "bbox_pred"
03.   type: "InnerProduct"
04.   bottom: "fc7"
05.   top: "bbox_pred"
06.   param { lr_mult: 1.0 }
07.   param { lr_mult: 2.0 }
08.   inner_product_param {
09.     num_output: 64 #按训练集类别改, 该值为(类别数+1)*4
10.     weight_filler {
11.       type: "gaussian"
12.       std: 0.001
13.     }
14.     bias_filler {
15.       type: "constant"
16.       value: 0
17.     }
18.   }
19. }

```

#### 4.py-faster-

rcnn/models/pascal\_voc/ZF/faster\_rcnn\_alt\_opt/stage2\_rpn\_train.pt修改

[plain]

```

01. layer {
02.   name: 'input-data'
03.   type: 'Python'
04.   top: 'data'
05.   top: 'im_info'
06.   top: 'gt_boxes'
07.   python_param {
08.     module: 'roi_data_layer.layer'
09.     layer: 'RoIDataLayer'
10.     param_str: "'num_classes': 16" #按训练集类别改, 该值为类别数+1
11.   }
12. }

```

#### 5.py-faster-

rcnn/models/pascal\_voc/ZF/faster\_rcnn\_alt\_opt/faster\_rcnn\_test.pt修改

[plain]

```

01. layer {
02.   name: "cls_score"
03.   type: "InnerProduct"
04.   bottom: "fc7"
05.   top: "cls_score"
06.   inner_product_param {
07.     num_output: 16 #按训练集类别改, 该值为类别数+1
08.   }
09. }

```

[plain]

```

01. layer {
02.   name: "bbox_pred"
03.   type: "InnerProduct"
04.   bottom: "fc7"
05.   top: "bbox_pred"
06.   inner_product_param {
07.     num_output: 64 #按训练集类别改, 该值为(类别数+1)*4
08.   }
09. }

```

#### 6.py-faster-rcnn/lib/datasets/pascal\_voc.py修改

(1)



[plain]

```

01. class pascal_voc(imdb):
02.     def __init__(self, image_set, year, devkit_path=None):
03.         imdb.__init__(self, 'voc_' + year + '_' + image_set)
04.         self._year = year
05.         self._image_set = image_set
06.         self._devkit_path = self._get_default_path() if devkit_path is None \
07.             else devkit_path
08.         self._data_path = os.path.join(self._devkit_path, 'VOC' + self._year)
09.         self._classes = ('__background__', # always index 0
10.             '你的标签1', '你的标签2', '你的标签3', '你的标签4'
11.         )

```

上面要改的地方是

修改训练集文件夹：

[plain]

```

01. self._data_path = os.path.join(self._devkit_path, 'VOC'+self._year)

```

用你的数据集直接替换原来VOC2007内的Annotations，ImageSets和JPEGImages即可,以免出现各种错误。

修改标签：

[plain]

```

01. self._classes = ('__background__', # always index 0
02.     '你的标签1', '你的标签2', '你的标签3', '你的标签4'
03. )

```

修改成你的数据集的标签就行。

( 2 )

[html]

```

01. cls = self._class_to_ind[obj.find('name').text.lower().strip()]

```

这里把标签转成小写，如果你的标签含有大写字母，可能会出现KeyError的错误，所以建议标签用小写字母。

( 去掉lower应该也行 )

建议训练的标签还是用小写的字母，如果最终需要用大写字母或中文显示标签，可参考：

[http://blog.csdn.net/sinat\\_30071459/article/details/51694037](http://blog.csdn.net/sinat_30071459/article/details/51694037)

## 7.py-faster-rcnn/lib/datasets/imdb.py修改

该文件的append\_flipped\_images(self)函数修改为：

[plain]

```

01. def append_flipped_images(self):
02.     num_images = self.num_images
03.     widths = [PIL.Image.open(self.image_path_at(i)).size[0]
04.         for i in xrange(num_images)]
05.     for i in xrange(num_images):
06.         boxes = self.roidb[i]['boxes'].copy()
07.         oldx1 = boxes[:, 0].copy()
08.         oldx2 = boxes[:, 2].copy()
09.         boxes[:, 0] = widths[i] - oldx2 - 1
10.         print boxes[:, 0]
11.         boxes[:, 2] = widths[i] - oldx1 - 1
12.         print boxes[:, 2]
13.         assert (boxes[:, 2] >= boxes[:, 0]).all()

```





```

14.         entry = {'boxes' : boxes,
15.                   'gt_overlaps' : self.roidb[i]['gt_overlaps'],
16.                   'gt_classes' : self.roidb[i]['gt_classes'],
17.                   'flipped' : True}
18.         self.roidb.append(entry)
19.         self._image_index = self._image_index * 2

```

这里assert (boxes[:, 2] >= boxes[:, 0]).all()可能出现AssertionError, 具体参

<http://blog.csdn.net/xzzppp/article/details/52036794>

!!! 为防止与之前的模型搞混,训练前把output文件夹删除(或改个其他名), 还

rcnn/data/cache中的文件和  
py-faster-rcnn/data/VOCdevkit2007/annotations\_cache中的文件删除(如果  
至于学习率等之类的设置,可在py-faster-rcnn/models/pascal\_voc/ZF/faster\_rcnn\_alt\_opt中  
的solve文件设置,迭代次数可在py-faster-rcnn\tools\train\_faster\_rcnn\_alt\_opt.py中修改:

```

[plain]
01. max_iters = [80000, 40000, 80000, 40000]

```

分别为4个阶段(rpn第1阶段, fast rcnn第1阶段, rpn第2阶段, fast rcnn第2阶段)的迭代次数。  
可改成你希望的迭代次数。

如果改了这些数值,最好把py-faster-rcnn/models/pascal\_voc/ZF/faster\_rcnn\_alt\_opt里对应的  
solver文件(有4个)也修改, stepsize小于上面修改的数值。

## 8.开始训练

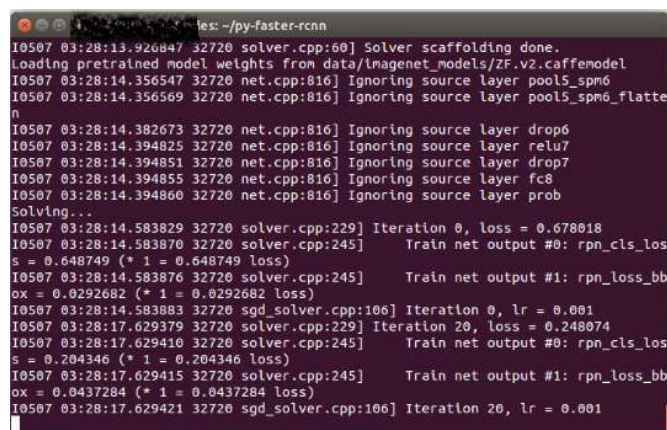
进入py-faster-rcnn, 执行:

```

[plain]
01. ./experiments/scripts/faster_rcnn_alt_opt.sh 0 ZF pascal_voc

```

这样, 就开始训练了。



## 9.测试

将训练得到的py-faster-rcnn\output\faster\_rcnn\_alt\_opt\\*\*\*\_trainval中ZF的caffemodel拷贝至  
py-faster-rcnn\data\faster\_rcnn\_models(如果没有这个文件夹,就新建一个), 然后, 修改:  
py-faster-rcnn\tools\demo.py, 主要修改:

```

[plain]
01. CLASSES = ('__background__',
02.              '你的标签1', '你的标签2', '你的标签3', '你的标签4')

```



改成你的数据集标签；

```
[plain]
01. NETS = {'vgg16': ('VGG16',
02.         'VGG16_faster_rcnn_final.caffemodel'),
03.         'zf': ('ZF',
04.               'ZF_faster_rcnn_final.caffemodel')}
```

上面ZF的caffemodel改成你的caffemodel。

```
[plain]
01. im_names = ['1559.jpg', '1564.jpg']
```

改成你的测试图片。(测试图片放在py-faster-rcnn\data\demo中)

## 10.结果

在py-faster-rcnn下，

执行：

```
[plain]
01. ./tools/demo.py --net zf
```

或者将默认模型改为zf：

```
[html]
01. parser.add_argument('--net', dest='demo_net', help='Network to use [vgg16]',
02.                     choices=NETS.keys(), default='vgg16')
```

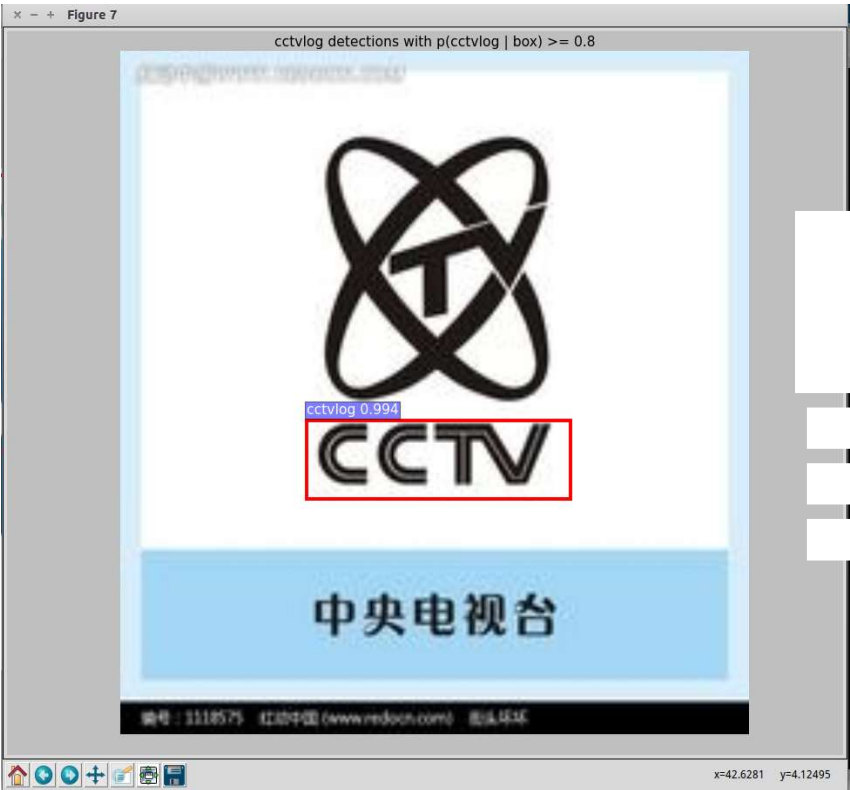
修改：

```
[html]
01. default='zf'
```

执行：

```
[plain]
01. ./tools/demo.py
```





顶 28 踩 0

- 上一篇 MFC：用另存为对话框保存文件
- 下一篇 Matlab截图小工具

相关文章推荐

- 用自己的数据训练Faster-RCNN
- MySQL在微信支付下的高可用运营--莫晓东
- Faster-RCNN+ZF用自己的数据集训练模型(Matlab版本)
- 容器技术在58同城的实践--姚远
- py-faster-rcnn用自己的数据训练模型
- SDCC 2017之容器技术实战线上峰会
- 使用faster rcnn训练自己的数据 (py-faster-rcnn)
- SDCC 2017之数据库技术实战线上峰会
- Faster\_rcnn训练自己的数据集
- 腾讯云容器服务架构实现介绍--董晓杰
- faster rcnn训练自己的数据集demo和训练过程er...
- 微博热点事件背后的数据库运维心得--张冬洪
- Faster-RCNN训练自己的数据集
- pva-faster-rcnn配置安装及训练自己的数据集
- py-faster-rcnn + cpu安装及训练自己的数据集
- Faster-rcnn训练自己的数据库