

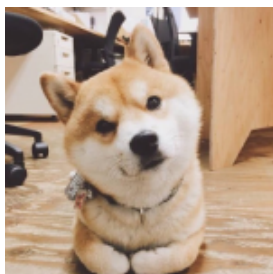
## zoro\_lov3的博客

目录视图

摘要视图

RSS 订阅

## 个人资料



Zoro\_H

关注

发私信

访问：3109次

积分：57

等级：BLOG &gt; 1

排名：千里之外

原创：2篇

转载：0篇

译文：0篇

评论：7条

异步赠书：9月重磅新书升级，本本经典 程序员9月书讯 每周荐书：ES6、虚拟现实、物联网（评论送...）

## FCN制作自己的数据集、训练和测试全流程

标签：ubuntu fcn 图像标注

2017-07-06 15:11

1317人阅读

评论(4)

收藏

举报

分类：

深度学习

版权声明：本文为博主原创文章，未经博主允许不得转载。

\*\*

## FCN制作自己的数据集、训练和测试全流程

\*\*

花了两三周的时间，在导师的催促下，把FCN的全部流程走了一遍，期间走了很多弯路，现在记录一

关闭

## 文章搜索

## 文章分类

[安装心得](#) (1)[深度学习](#) (1)

## 文章存档

[2017年07月](#) (1)[2017年03月](#) (1)

## 阅读排行

[Ubuntu16.04环境下安装caffe及...](#) (1784)[FCN制作自己的数据集、训练...](#) (1310)

## 评论排行

[FCN制作自己的数据集、训练...](#) (4)[Ubuntu16.04环境下安装caffe及...](#) (3)

## 推荐文章

- \* CSDN新版博客feed流内测用户征集令
- \* Android检查更新下载安装
- \* 动手打造史上最简单的 Recycleview 侧滑菜单
- \* TCP网络通讯如何解决分包粘包问题
- \* SDCC 2017之大数据技术实战线上峰会

下。系统环境：ubuntu 16.04LTS

## 一、数据集的制作

注：我的数据集是仿照VOC数据集进行制作的

### 1.resize 数据集

我的GPU显存4G，跑过大的图片带不动，需要resize图片大小，放几个修改图片大小的程序

#### (1) 单张图片resize

```
1 # coding = utf-8
2 import Image
3
4 def convert(width,height):
5     im = Image.open("C:\\xxx\\test.jpg")
6     out = im.resize((width, height),Image.ANTIALIAS)
7     out.save("C:\\xxx\\test.jpg")
8 if __name__ == '__main__':
9     convert(256,256)
```

#### (2) resize整个文件夹里的图片

```
1 # coding = utf-8
2 import Image
3 import os
4
5 def convert(dir,width,height):
6     file_list = os.listdir(dir)
7     print(file_list)
8     for filename in file_list:
```

关闭

\* 快速集成一个视频直播功能

## 最新评论

FCN制作自己的数据集、训练和测试全流程

Zoro\_H : @easonwenwei:那个只是一个展示,就是之后的background 0 0 0 背景 aer...

FCN制作自己的数据集、训练和测试全流程

eason\_cs : 您好,我想问问怎么给图像标注颜色啊,这里说的不是很明白, labelcolormap函数怎么用?

FCN制作自己的数据集、训练和测试全流程

Zoro\_H : @wei19921109:我这里没有重新计算,但不太影响我的结果,您如果有需要可以自行重新计算一下

Ubuntu16.04环境下安装caffe及运行faster-r...

机智的火星人 : 您好,下载的模型的链接有问题是怎么回事

FCN制作自己的数据集、训练和测试全流程

xu-炜 : 你好,想问问,换成新的数据集,训练跟测试时减去的图像均值不需要根据新的数据集重新计算吗?为什么还是 (...

Ubuntu16.04环境下安装caffe及运行faster-r...

HellerF : 非常感谢~你写的是最好最有效的方法~希望以后多交流.....

Ubuntu16.04环境下安装caffe及运行faster-r...

小白莹 : 厉害!

FCN制作自己的数据集、训练和测试全流程 - zoro\_lov3的博客 - CSDN博客

```

9      path = "
10     path = dir+filename
11     im = Image.open(path)
12     out = im.resize((256,256),Image.ANTIALIAS)
13     print "%s has been resized!"%filename
14     out.save(path)
15
16 if __name__ == '__main__':
17     dir = raw_input('please input the operate dir:')
18     convert(dir,256,256)

```

## (3) 按比例resize

```

1  # coding = utf-8
2  import Image
3  import os
4
5  def convert(dir,width,height):
6      file_list = os.listdir(dir)
7      print(file_list)
8      for filename in file_list:
9          path = "
10         path = dir+filename
11         im = Image.open(path)
12         out = im.resize((256,256),Image.ANTIALIAS)
13         print "%s has been resized!"%filename
14         out.save(path)
15
16 if __name__ == '__main__':
17     dir = raw_input('please input the operate dir:')
18     convert(dir,256,256)

```

关闭

## 2.制作索引图

### (1) 下载labelme

下载地址：<https://github.com/wkentaro/labelme>

下载后按提示打开软件，进行标注，保存会生成后缀为json的文件。

### (2) 生成dataset文件夹

在终端输入指令：

```
1 labelme_json_to_dataset _static/apc2016_obj3.json //这里的文件名根据自己的实际情
```

### (3) 为文件夹下的label.png着色

首先需要对照VOC分割的颜色进行着色，一定要保证颜色的准确性。Matlab代码:

```
1 function cmap = labelcolormap(N)
2
3 if nargin==0
4     N=256
5 end
6 cmap = zeros(N,3);
7 for i=1:N
8     id = i-1; r=0;g=0;b=0;
9     for j=0:7
10         r = bitor(r, bitshift(bitget(id,1),7 - j));
11         g = bitor(g, bitshift(bitget(id,2),7 - j));
12         b = bitor(b, bitshift(bitget(id,3),7 - j));
13         id = bitshift(id,-3);
14     end
15     cmap(i,1)=r; cmap(i,2)=g; cmap(i,3)=b;
```

关闭

```
16 end
17 cmap = cmap / 255;
```

对应的VOC数据集中的颜色类别：

```
1 类别名称 R G B
2 background 0 0 0 背景
3 aeroplane 128 0 0 飞机
4 bicycle 0 128 0
5 bird 128 128 0
6 boat 0 0 128
7 bottle 128 0 128 瓶子
8 bus 0 128 128 大巴
9 car 128 128 128
10 cat 64 0 0 猫
11 chair 192 0 0
12 cow 64 128 0
13 diningtable 192 128 0 餐桌
14 dog 64 0 128
15 horse 192 0 128
16 motorbike 64 128 128
17 person 192 128 128
18 pottedplant 0 64 0 盆栽
19 sheep 128 64 0
20 sofa 0 192 0
21 train 128 192 0
22 tvmonitor 0 64 128 显示器
```

然后使用Python的skimage库进行颜色填充，具体函数是skimage.color.label2rgb(),代码较长，有需要可私信。

关闭

---

-2017年9月26日 新增-

太多人和我私信要代码，前一段时间很忙都没有及时回复大家，所以这里放个链接，需要自取。

<https://github.com/hitzoro/FCN-ColorLabel>

### 3.将填充后的label.png转为灰度图

如果不转，在训练的时候回报错，转换matlab代码如下：

```
1  dirs=dir('F:/xxx/*.png');
2  for n=1:numel(dirs)
3      strname=strcat('F:/xxx/',dirs(n).name);
4      img=imread(strname);
5      [x,map]=rgb2ind(img,256);
6      newname=strcat('F:/xxx/',dirs(n).name);
7      imwrite(x,map,newname,'png');
8  end
```

转化后，在python中检查一下图片的格式：

```
1  In [23]: img = PIL.Image.open('000001_json/label.png')
2  In [24]: np.unique(img)
3  Out[24]: array([0, 1, 2], dtype=uint8)
```

如果输出一致，则索引图制作正确。

## 二、FCN训练自己的数据集

### 1.前期准备

默认已经安装好显卡驱动，cuda，cudnn，opencv。

关闭

最新版caffe下载：<https://github.com/BVLC/caffe>

fcn源代码下载：<https://github.com/shelhamer/fcn.berkeleyvision.org>

caffe的配置安装可以参考我的另一篇博客：[http://blog.csdn.net/zoro\\_lov3/article/details/60581174](http://blog.csdn.net/zoro_lov3/article/details/60581174)

## 2.数据集准备

这里我们需要两个数据集包，benchmark和VOC2012，进入fcn/data，新建sbdd文件夹（如果没有），将benchmark解压到sbdd中，将VOC2012解压到data下的pascal文件夹下。

这两个在网上都可以找得到。

这两个数据集有什么用呢？在FCN中VOC数据集的训练需要他俩，benchmark中的dataset用的数据，VOC2012存放测试时的数据。

- 先制作训练时的数据

进入dataset中的img文件夹，这里存放训练用的原图，把原来的原图替换为你自己的原

改train.txt，写入你训练图片的名字，注意不要加后缀。如下即可。

```
1 000001
2 000003
3 000005
4 000007
5 000009
6 000011
7 000013
8 000015
9 000017
10 000019
```

关闭

进入cls文件夹，这里原本需要存放mat格式的文件，但是制作mat文件有点麻烦，参考了网上的资料，修改代码，使得这里也可以直接存放索引图。

方式如下：

- 修改fcn目录下的voc\_layers.py

注释掉原本的load\_label，修改为新的

```
1  # def load_label(self, idx):
2  #     """
3  #     Load label image as 1 x height x width integer array of label indices.
4  #     The leading singleton dimension is required by the loss.
5  #     """
6  #     import scipy.io
7  #     mat = scipy.io.loadmat('{}/cls/{}.mat'.format(self.sbdd_dir, idx))
8  #     label = mat['GTcls'][0]['Segmentation'][0].astype(np.uint8)
9  #     label = label[np.newaxis, ...]
10 #     return label
11
12 def load_label(self, idx):
13     """
14     Load label image as 1 x height x width integer array of label indices.
15     The leading singleton dimension is required by the loss.
16     """
17     im = Image.open('{}/cls/{}.png'.format(self.sbdd_dir, idx))
18     label = np.array(im, dtype=np.uint8)
19     label = label[np.newaxis, ...]
20     return label
```

- 制作测试集数据

关闭



测试集的制作简单一写，进入VOC2012，进入JPEGImages文件夹，里面存放测试用的原图，然后进入SegmentationClass，里面存放测试用的索引图，最后进入ImageSets/Segmentation，有一个名为seg11valid.txt的文件，它和train.txt的性质一样，存放测试用的图片名。

到此，数据集就准备完成了。

### 3.修改网络参数

下载VGG16的预训练模型并放在FCN源码文件夹中的ilsvrc-nets文件夹下

<https://pan.baidu.com/s/1qYJeFfQ>

为了避免运行程序时候出现no module named caffe

在代码中包含import caffe的py文件（solve.py）的第一行加入

```
1 import sys
2 sys.path.append('/home/hitoia/caffe/python')
```

其中，/home/hitoia/caffe/python为你下载的caffe源码中python文件夹的路径

cd进入fcn源码路径

以个人路径为例：/home/hitoia/fcn.berkeleyvision.org/

将其中所有的py文件，例如surgery.py等等，全部复制到voc-fcn32s文件夹中

- solver.prototxt文件修改

进入voc-fcn32s文件夹 打开solver.prototxt

关闭

其中snapshot:10000 表示训练10000次保存一次模型

snapshot\_prefix:"/home/hitoia/fcn.berkeleyvision.org/voc-fcn32s/snapshot/train"

表示训练得到的模型，也就是model存放的路径

在此，我附上个人的solver.prototxt供大家参考

```
1 train_net: "/home/hitoia/fcn.berkeleyvision.org/voc-fcn32s/train.prototxt"
2 test_net: "/home/hitoia/fcn.berkeleyvision.org/voc-fcn32s/val.prototxt"
3 test_iter: 736
4 # make test net, but don't invoke it from the solver itself
5 test_interval: 999999999
6 display: 20
7 average_loss: 20
8 lr_policy: "fixed"
9 # lr for unnormalized softmax
10 base_lr: 1e-10
11 # high momentum
12 momentum: 0.99
13 # no gradient accumulation
14 iter_size: 1
15 max_iter: 100000
16 weight_decay: 0.0005
17 snapshot: 4000
18 snapshot_prefix: "/home/hitoia/fcn.berkeleyvision.org/voc-fcn32s/snapshot/train"
19 test_initialization: false
```

- solve.py的修改

在这里郑重声明一下：如果训练fcn32s的网络模型，一定要修改solve.py利用transplant的方式获取vgg16的网络权重。

具体操作为：

关闭

```
1 import sys
2 sys.path.append('/home/hitoia/caffe/python')
3 import caffe
4 import surgery, score
5
6 import numpy as np
7 import os
8 import sys
9
10 try:
11     import setproctitle
12     setproctitle.setproctitle(os.path.basename(os.getcwd()))
13 except:
14     pass
15
16 vgg_weights = '../ilsvrc-nets/vgg16-fcn.caffemodel'
17 vgg_proto = '../ilsvrc-nets/VGG_ILSVRC_16_layers_deploy.prototxt'
18 weights = '../ilsvrc-nets/vgg16-fcn.caffemodel'
19 #weights = '../ilsvrc-nets/vgg16-fcn.caffemodel'
20
21 # init
22 #caffe.set_device(int(sys.argv[1]))
23 caffe.set_mode_gpu()
24 caffe.set_device(0)
25
26 #solver = caffe.SGDSolver('solver.prototxt')
27 #solver.net.copy_from(weights)
28 solver = caffe.SGDSolver('solver.prototxt')
29 vgg_net=caffe.Net(vgg_proto,vgg_weights,caffe.TRAIN)
30 surgery.transplant(solver.net,vgg_net)
31 del vgg_net
32
33 # surgeries
34 interp_layers = [k for k in solver.net.params.keys() if 'up' in k]
35 surgery.interp(solver.net, interp_layers)
```

关闭

```
36
37 # scoring
38 val = np.loadtxt('/home/hitoia/fcn.berkeleyvision.org/data/pascal/VOCdevkit/VOC2012/ImageSets/S
39
40 for _ in range(25):
41     solver.step(1000)
42     score.seg_tests(solver, False, val, layer='score')
```

关于VGG\_ILSVRC\_16\_layers\_deploy.prototxt 可以在<http://pan.baidu.com/s/1geLL6Sz>下载。

如果训练fcn16s，则可以直接copy自己的fcn32s的model的权重，不需要transplant，也就是7

solve.py

如果训练fcn8s，则可以直接copy自己的fcn16s的model的权重，不需要transplant,也就是不需

solve.py

只有如此，才能避免loss高居不下的情况

这里的：

```
1 for _ in range(25):
2     solver.step(1000)
3     score.seg_tests(solver, False, val, layer='score')
```

奇怪的现象：修改solver.prototxt中的max\_iter: 100000没有改变最大迭代次数，只有改变这个step里的数字才有用，这里最大迭代次数等于 $25 \times 1000 = 25000$ 次。

- train.prototxt / val.prototxt 修改

所有num\_output 为21 的地方都修改为自己分类数 + 1（加的1是背景），最开始的param\_str需要根

关闭

据自己的情况修改，放一下我自己的

train.prototxt:

```
1 param_str: "{\sbdd_dir\: \"/home/hitoia/fcn.berkeleyvision.org/data/sbdd/benchmark/benchmark_
```

val.prototxt:

```
1 param_str: "{\voc_dir\: \"/home/hitoia/fcn.berkeleyvision.org/data/pascal/VOCdevki
```

准备完成，在voc-fcn32s路径下输入

```
1 python solve.py
```

就可以开始训练

### 三、单张测试

在fcn源码文件夹，找到infer.py。

```
1 import numpy as np
2 from PIL import Image
3 import matplotlib.pyplot as plt
4
5
6 import sys
7 sys.path.append('/home/hitoia/caffe/python')
8 import caffe
9
10 # load image, switch to BGR, subtract mean, and make dims C x H x W for Caffe
```

关闭

```
11 im = Image.open('000030.jpg')
12 in_ = np.array(im, dtype=np.float32)
13 in_ = in_[::-1]
14 in_ -= np.array((104.00698793, 116.66876762, 122.67891434))
15 in_ = in_.transpose((2, 0, 1))
16
17 # load net
18 #net = caffe.Net('voc-fcn8s/deploy.prototxt', 'voc-fcn8s/fcn8s-heavy-pascal.caffemodel', caffe.TEST)
19 net = caffe.Net('voc-fcn32s/deploy.prototxt', 'voc-fcn32s/snapshot/train_iter_24000.caffemodel', caf
20 #net = caffe.Net('voc-fcn8s/deploy.prototxt', 'siftflow-fcn32s/train_iter_100000.caffemodel', caffe.TE
21 # shape for input (data blob is N x C x H x W), set data
22 net.blobs['data'].reshape(1, *in_.shape)
23 net.blobs['data'].data[...] = in_
24 # run net and take argmax for prediction
25 net.forward()
26 out = net.blobs['score'].data[0].argmax(axis=0)
27
28 #plt.imshow(out, cmap='gray');
29 plt.imshow(out);
30 plt.axis('off')
31 plt.savefig('000030_out32.png')
32 #plt.show()
33
```

关闭

其中，`net = caffe.Net('voc-fcn32s/deploy.prototxt', 'voc-fcn32s/snapshot/train_iter_24000.caffemodel', caffe.TEST)`，其中`train_iter_24000.caffemodel`是我训练后得到的模型。

如果没有deploy文件，可以参考如下方法：

首先，根据你利用的模型，例如模型是voc-fcn32s的，那么你就去voc-fcn32s的文件夹，里面有train.prototxt文件，将文件打开，全选，复制，新建一个名为deploy.prototxt文件，粘贴进去，

然后ctrl+F 寻找所有名为loss的layer 只要有loss 无论是loss还是geo\_loss 将这个layer统统删除,这就是此次的deploy.prototxt。

大功告成，至此整个流程全部完成。整个过程心酸不断，fcn的资料不多，求助了很多，在此感谢

**无奈的小心酸，深度学习思考者**

对我的帮助。

参考博客：

<http://blog.csdn.net/wangkun1340378/article/details/70238290>

<http://blog.csdn.net/u010402786/article/details/72883421>

[http://blog.csdn.net/supe\\_king/article/details/55657136](http://blog.csdn.net/supe_king/article/details/55657136)

顶 踩  
0 0

关闭

- [上一篇](#) Ubuntu16.04环境下安装caffe及运行faster-rcnn

#### 相关文章推荐

- FCN训练自己的数据集及测试
- caffe fcn数据集制作 -标签

- Presto的服务治理与架构在京东的实践与应用--王哲...
- FCN网络训练 终极版
- 深入掌握Kubernetes应用实践--王渊命
- 使用caffe-future完成FCN网络的训练
- Python基础知识汇总
- Eclipse快捷键大全(转载)
- Android核心技术详解
- Retrofit 从入门封装到源码解析
- 深度学习研究理解6:OverFeat:Integrated Recognition...
- 自然语言处理工具Word2Vec
- 使用FCN做图像语义分割(实践篇)
- caffe下fcn数据集制作的简化
- py-rfcn算法caffe配置, 训练及应用到自己的数据集
- R-FCN、SSD、YOLO2、faster-rcnn和labelImg实验...

## 查看评论



eason\_cs

2楼 2017-09-26 15:...

您好,我想问问怎么给图像标注颜色啊,这里说的不是很明白, labelcolormap函数怎么用?



Zoro\_H

Re: 2017-09-26 20:2

回复eason\_cs: 那个只是一个展示,就是之后的background 0 0 0 背景 aeroplane 128 0 0 飞机 这些具体数值都是怎么来的,真正制作索引图需要参考我放在文章里的新增的代码。



xu-炜

1楼 2017-09-12 12:24发表

你好,想问问,换成新的数据集,训练跟测试时减去的图像均值不需要根据新的数据集重新计算吗?为什么还是 ( 104.00698793,116.6687676 2,122.67891434 ) ?

Zoro\_H

Re: 2017-09-24 21:48发表

关闭






回复xu-炜：我这里没有重新计算，但不太影响我的结果，  
您如果有需要可以自行重新计算一下

您还没有登录,请[\[登录\]](#)或[\[注册\]](#)

\* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

[公司简介](#) | [招贤纳士](#) | [广告服务](#) | [联系方式](#) | [版权声明](#) | [法律顾问](#) | [问题报告](#) | [合作伙伴](#) | [论坛反馈](#)

[网站客服](#)   [杂志客服](#)   [微博客服](#)   [webmaster@csdn.net](mailto:webmaster@csdn.net)   400-660-0108 | 北京创新乐知信息技术有限公司 版权所有 | 江苏知之为计算机有限公司 | 江苏乐知信息技术有限公司  
京 ICP 证 09002463 号 | Copyright © 1999-2017, CSDN.NET, All Rights Reserved 

关闭