个人资料

thystar

访问： 25454次
积分： 1589
等级： BLOG 4
排名： 第16561名

原创： 129篇 转载： 49篇
译文： 3篇 评论： 14条

文章搜索

文章分类

学习OpenCV--学习笔记 (35)
算法导论 (6)
笔试题目 (4)
PRML (1)
Manifold Learning (6)
读论文整理 (1)
Java Web学习笔记 (23)
机器学习 (4)
我转载的文章 (12)
python机器学习 (3)
pygame (26)
python (0)
随记 (6)
html (15)
Asp.net (2)
php (11)
web前端 (20)
caffe学习 (20)
设计模式 (0)

文章存档

2016年04月 (4)
2016年03月 (2)
2016年02月 (9)
2016年01月 (2)
2015年12月 (10)

caffe学习笔记12 -- R-CNN detection

2016-02-24 16:09 1072人阅读 评论(2) 收藏 举报

分类： caffe学习（19）

这是caffe文档中Notebook Examples的倒数第二个例子，链接地址http://nbviewer.jupyter.org/github/BVLC/caffe/blob/master/examples/detection.ipynb

这个例子用R-CNN做目标检测。

R-CNN是一个先进的目标检测模型，它通过微调caffe模型提供分类区域。对于R-CNN系统和模型的详细介绍，参考

Rich feature hierarchies for accurate object detection and semantic segmentation. Ross Girshick, Jeff Donahue, Trevor Darrell, Jitendra Malik. CVPR 2014.Arxiv 2013.

在本例中，预训练模型基于ImageNet数据集，在并在ILSVRC13上进行微调，输出200个检测分类的得分。需要注意的是：本例中一个原始数据对应所有SVM分类的得分，没有概率校准和类间比较。本例中使用现成的模型只是为了简便，并非完整的R-CNN模型

现在，来检测图caffe-master/examples/images/fish-bike.jpg

首先，需要做一些准备工作：

i. 安装matlab, 具体安装过程参考：http://blog.csdn.net/thystar/article/details/50720691

ii. 添加matlab安装路径，sudo gedit ~/.bashrc, 在文本最后添加：export PATH="/home/sindyz/software/matlab2014/bin":$PATH (我的安装路径)保存后需要重启电脑，要说明的是，这一步是必要的，否则运行时会出现：OSError: [Errno 2] No such file or directory错误。

iii. 下载Selective Search文件，下载地址：https://github.com/sergeyk/selective_search_ijcv_with_python，用于检测候选框，关于Selective Search的算法介绍，参考：http://koen.me/research/selectivesearch/，下载完成后，解压，在matlab下运行demo.m, 无报错信息关闭即可，需要注意的是，如果这个文件不在$CAFFE-ROOT/Python目录下,需要将其添加到PYTHONPATH路径中，我的是：export PYTHONPATH=/home/sindyz/code/matlabCode/selective_search_ijcv_with_python/:$PYTHONPATH。（按自己的情况添加）

完成上述步骤后，还有几处需要注意和修改的地方：

- 在Selective Search文件目录下运行python selective_search.py，看看是否由报错信息，一般来说，如果你添加好了matlab路径，这里不会出什么问题。

展开

阅读排行

caffe学习笔记12 -- R-CN (1070)
OpenCV入门（二十二） (741)
关于windows下caffe配置 (655)
OpenCV入门（十四）-- (341)
caffe学习笔记1--一些学习 (318)
React学习笔记（5）--事 (317)
caffe学习笔记 7 --Image (306)
Manifold Alignment with (291)
OpenCV入门(二十九) -- (291)
Unsupervised Image Ma (286)

评论排行

caffe学习笔记2--caffe的 (3)
Manifold Alignment (2)
caffe学习笔记12 -- R-CN (2)
关于windows下caffe配置 (2)
caffe学习笔记 7 --Image (1)
caffe学习笔记6--训练自己 (1)
caffe学习笔记13 -- Setup (1)
caffe学习笔记14(外篇)--((1)
html5学习笔记（13） (1)
排序算法（四）-- 快速排 (0)

推荐文章

*搭建docker私有仓库
*Android杂谈之RadioGroup+ViewPager制作的底部导航栏
* Android几种常见的多渠道(批量)打包方式介绍
*Oculus Rift, HTC Vive, SONY PSVR的全面对比
*Android View的事件分发机制探索
*Redis源码解析：15Resis主从复制之从节点流程

最新评论

caffe学习笔记12 -- R-CNN detec
thystar: @qq_21402107:可以的，但是你最好用Ipython, 安装时很方便的。
caffe学习笔记12 -- R-CNN detec
qq_21402107: 最初始的部分是在python下运行吗
caffe学习笔记14(外篇)--使用Coc
thystar: 因为我没法截图，所以如果大家在运行时候有出现问题什么的，请在下面评论，共同讨论学习
caffe学习笔记6--训练自己的数据
thystar: 在训练过程中有出现问题随时讨论。共同学习
caffe学习笔记2--caffe的文件结构
thystar: @u010076558:多谢支持，一起学习啦。
caffe学习笔记2--caffe的文件结构
Crossi: 小弟路过，特此一拜（读）
caffe学习笔记2--caffe的文件结构
thystar: caffe初学，希望大家多

- 修改$CAFFE-ROOT/python/caffe/detector.py中第86行左右：

  predictions = out[:, :, :, 0].squeeze(axis=(2, 3)) 改为

  predictions = out[self.outputs[0]].squeeze()，否则会报出ValueError: 'axis' entry 2 is out of bounds (-2, 2) 错误

- 修改$CAFFE-ROOT/python/caffe/detector.py中114行左右

  import selective_search_ijcv_with_python as selective_search

  改为

  import selective_search，因为在Selective Search文件目录下，只有selective_search.py模块,否则会出现模块找不到的错误

OK，现在可以开心的运行R-CNN这个例子了。

1. 更改目录，导入相应的包

```python
import os
caffe_root = '/home/sindyz/caffe-master/'
os.chdir(caffe_root)
import sys
sys.path.insert(0,'./python')
```


2. 创建临时目录，导入检测样本

```python
! mkdir -p _temp
! echo examples/images/fish-bike
```

3. 运行selective_search提取候选框，调用C

```python
! python/detect.py --crop_mode=selective_search --
pretrained_model=models/bvlc_reference_rcnn_ilsvrc13/bvlc_reference_rcnn_ilsvrc13.caffemodel --
model_def=models/bvlc_reference_rcnn_ilsvrc13/deploy.prototxt --gpu --
raw_scale=255 _temp/det_input.txt _temp/det_output.h5
```

输出如下内容：

```plain
GPU mode
WARNING: Logging before InitGoogleLogging() is written to STDERR
I0608 10:32:38.067106  6131 net.cpp:42] Initializing net from parameters:
name: "R-CNN-ilsvrc13"
input: "data"
input_dim: 10
input_dim: 3
input_dim: 227
input_dim: 227
state {
  phase: TEST
}
layer {
  name: "conv1"
  type: "Convolution"
  bottom: "data"
  top: "conv1"
  convolution_param {
    num_output: 96
    kernel_size: 11
    stride: 4
  }
}
layer {
  name: "relu1"
  type: "ReLU"
```

多交流，共同进步啊

caffe学习笔记13 -- Setup
thystar: Caffe初学，希望大家多多指导，关于代码的解释会逐步添加和补充。

caffe学习笔记 7 --Image Classifi
thystar: caffe的学习笔记还没有写完，我先发表一部分，其他的慢慢添加再发表。

关于windows下caffe配置中出现
自来清风: @thystar:遇到了同样的问题，但是不想把cuda改成6.5。要是知道怎么修改7.0就好了

```
  bottom: "conv1"
  top: "conv1"
}
layer {
  name: "pool1"
  type: "Pooling"
  bottom: "conv1"
  top: "pool1"
  pooling_param {
    pool: MAX
    kernel_size: 3
    stride: 2
  }
}
layer {
  name: "norm1"
  type: "LRN"
  bottom: "pool1"
  top: "norm1"
  lrn_param {
    local_size: 5
    alpha: 0.0001
    beta: 0.75
  }
}
layer {
  name: "conv2"
  type: "Convolution"
  bottom: "norm1"
  top: "conv2"
  convolution_param {
    num_output: 256
    pad: 2
    kernel_size: 5
    group: 2
  }
}
layer {
  name: "relu2"
  type: "ReLU"
  bottom: "conv2"
  top: "conv2"
}
layer {
  name: "pool2"
  type: "Pooling"
  bottom: "conv2"
  top: "pool2"
  pooling_param {
    pool: MAX
    kernel_size: 3
    stride: 2
  }
}
layer {
  name: "norm2"
  type: "LRN"
  bottom: "pool2"
  top: "norm2"
  lrn_param {
    local_size: 5
    alpha: 0.0001
    beta: 0.75
  }
}
layer {
  name: "conv3"
  type: "Convolution"
  bottom: "norm2"
  top: "conv3"
  convolution_param {
    num_output: 384
    pad: 1
    kernel_size: 3
  }
}
layer {
  name: "relu3"
  type: "ReLU"
```

```
  bottom: "conv3"
  top: "conv3"
}
layer {
  name: "conv4"
  type: "Convolution"
  bottom: "conv3"
  top: "conv4"
  convolution_param {
    num_output: 384
    pad: 1
    kernel_size: 3
    group: 2
  }
}
layer {
  name: "relu4"
  type: "ReLU"
  bottom: "conv4"
  top: "conv4"
}
layer {
  name: "conv5"
  type: "Convolution"
  bottom: "conv4"
  top: "conv5"
  convolution_param {
    num_output: 256
    pad: 1
    kernel_size: 3
    group: 2
  }
}
layer {
  name: "relu5"
  type: "ReLU"
  bottom: "conv5"
  top: "conv5"
}
layer {
  name: "pool5"
  type: "Pooling"
  bottom: "conv5"
  top: "pool5"
  pooling_param {
    pool: MAX
    kernel_size: 3
    stride: 2
  }
}
layer {
  name: "fc6"
  type: "InnerProduct"
  bottom: "pool5"
  top: "fc6"
  inner_product_param {
    num_output: 4096
  }
}
layer {
  name: "relu6"
  type: "ReLU"
  bottom: "fc6"
  top: "fc6"
}
layer {
  name: "drop6"
  type: "Dropout"
  bottom: "fc6"
  top: "fc6"
  dropout_param {
    dropout_ratio: 0.5
  }
}
layer {
  name: "fc7"
  type: "InnerProduct"
  bottom: "fc6"
  top: "fc7"
```

```
  inner_product_param {
    num_output: 4096
  }
}
layer {
  name: "relu7"
  type: "ReLU"
  bottom: "fc7"
  top: "fc7"
}
layer {
  name: "drop7"
  type: "Dropout"
  bottom: "fc7"
  top: "fc7"
  dropout_param {
    dropout_ratio: 0.5
  }
}
layer {
  name: "fc-rcnn"
  type: "InnerProduct"
  bottom: "fc7"
  top: "fc-rcnn"
  inner_product_param {
    num_output: 200
  }
}
I0608 10:32:38.067556  6131 net.cpp:370] Input 0 -> data
I0608 10:32:38.067576  6131 layer_factory.hpp:74] Creating layer conv1
I0608 10:32:38.067585  6131 net.cpp:90] Creating Layer conv1
I0608 10:32:38.067589  6131 net.cpp:410] conv1 <- data
I0608 10:32:38.067595  6131 net.cpp:368] conv1 -> conv1
I0608 10:32:38.067603  6131 net.cpp:120] Setting up conv1
I0608 10:32:38.108999  6131 net.cpp:127] Top shape: 10 96 55 55 (2904000)
I0608 10:32:38.109035  6131 layer_factory.hpp:74] Creating layer relu1
I0608 10:32:38.109048  6131 net.cpp:90] Creating Layer relu1
I0608 10:32:38.109055  6131 net.cpp:410] relu1 <- conv1
I0608 10:32:38.109063  6131 net.cpp:357] relu1 -> conv1 (in-place)
I0608 10:32:38.109076  6131 net.cpp:120] Setting up relu1
I0608 10:32:38.109233  6131 net.cpp:127] Top shape: 10 96 55 55 (2904000)
I0608 10:32:38.109244  6131 layer_factory.hpp:74] Creating layer pool1
I0608 10:32:38.109257  6131 net.cpp:90] Creating Layer pool1
I0608 10:32:38.109263  6131 net.cpp:410] pool1 <- conv1
I0608 10:32:38.109269  6131 net.cpp:368] pool1 -> pool1
I0608 10:32:38.109277  6131 net.cpp:120] Setting up pool1
I0608 10:32:38.109311  6131 net.cpp:127] Top shape: 10 96 27 27 (699840)
I0608 10:32:38.109318  6131 layer_factory.hpp:74] Creating layer norm1
I0608 10:32:38.109325  6131 net.cpp:90] Creating Layer norm1
I0608 10:32:38.109329  6131 net.cpp:410] norm1 <- pool1
I0608 10:32:38.109335  6131 net.cpp:368] norm1 -> norm1
I0608 10:32:38.109341  6131 net.cpp:120] Setting up norm1
I0608 10:32:38.109349  6131 net.cpp:127] Top shape: 10 96 27 27 (699840)
I0608 10:32:38.109352  6131 layer_factory.hpp:74] Creating layer conv2
I0608 10:32:38.109360  6131 net.cpp:90] Creating Layer conv2
I0608 10:32:38.109364  6131 net.cpp:410] conv2 <- norm1
I0608 10:32:38.109370  6131 net.cpp:368] conv2 -> conv2
I0608 10:32:38.109376  6131 net.cpp:120] Setting up conv2
I0608 10:32:38.109931  6131 net.cpp:127] Top shape: 10 256 27 27 (1866240)
I0608 10:32:38.109947  6131 layer_factory.hpp:74] Creating layer relu2
I0608 10:32:38.109954  6131 net.cpp:90] Creating Layer relu2
I0608 10:32:38.109959  6131 net.cpp:410] relu2 <- conv2
I0608 10:32:38.109966  6131 net.cpp:357] relu2 -> conv2 (in-place)
I0608 10:32:38.109972  6131 net.cpp:120] Setting up relu2
I0608 10:32:38.110002  6131 net.cpp:127] Top shape: 10 256 27 27 (1866240)
I0608 10:32:38.110008  6131 layer_factory.hpp:74] Creating layer pool2
I0608 10:32:38.110014  6131 net.cpp:90] Creating Layer pool2
I0608 10:32:38.110018  6131 net.cpp:410] pool2 <- conv2
I0608 10:32:38.110024  6131 net.cpp:368] pool2 -> pool2
I0608 10:32:38.110030  6131 net.cpp:120] Setting up pool2
I0608 10:32:38.110136  6131 net.cpp:127] Top shape: 10 256 13 13 (432640)
I0608 10:32:38.110144  6131 layer_factory.hpp:74] Creating layer norm2
I0608 10:32:38.110152  6131 net.cpp:90] Creating Layer norm2
I0608 10:32:38.110157  6131 net.cpp:410] norm2 <- pool2
I0608 10:32:38.110162  6131 net.cpp:368] norm2 -> norm2
I0608 10:32:38.110168  6131 net.cpp:120] Setting up norm2
I0608 10:32:38.110175  6131 net.cpp:127] Top shape: 10 256 13 13 (432640)
I0608 10:32:38.110179  6131 layer_factory.hpp:74] Creating layer conv3
I0608 10:32:38.110187  6131 net.cpp:90] Creating Layer conv3
```

```
I0608 10:32:38.110191  6131 net.cpp:410] conv3 <- norm2
I0608 10:32:38.110198  6131 net.cpp:368] conv3 -> conv3
I0608 10:32:38.110203  6131 net.cpp:120] Setting up conv3
I0608 10:32:38.111160  6131 net.cpp:127] Top shape: 10 384 13 13 (648960)
I0608 10:32:38.111176  6131 layer_factory.hpp:74] Creating layer relu3
I0608 10:32:38.111183  6131 net.cpp:90] Creating Layer relu3
I0608 10:32:38.111189  6131 net.cpp:410] relu3 <- conv3
I0608 10:32:38.111194  6131 net.cpp:357] relu3 -> conv3 (in-place)
I0608 10:32:38.111202  6131 net.cpp:120] Setting up relu3
I0608 10:32:38.111232  6131 net.cpp:127] Top shape: 10 384 13 13 (648960)
I0608 10:32:38.111238  6131 layer_factory.hpp:74] Creating layer conv4
I0608 10:32:38.111243  6131 net.cpp:90] Creating Layer conv4
I0608 10:32:38.111248  6131 net.cpp:410] conv4 <- conv3
I0608 10:32:38.111253  6131 net.cpp:368] conv4 -> conv4
I0608 10:32:38.111260  6131 net.cpp:120] Setting up conv4
I0608 10:32:38.112344  6131 net.cpp:127] Top shape: 10 384 13 13 (648960)
I0608 10:32:38.112357  6131 layer_factory.hpp:74] Creating layer relu4
I0608 10:32:38.112365  6131 net.cpp:90] Creating Layer relu4
I0608 10:32:38.112370  6131 net.cpp:410] relu4 <- conv4
I0608 10:32:38.112375  6131 net.cpp:357] relu4 -> conv4 (in-place)
I0608 10:32:38.112381  6131 net.cpp:120] Setting up relu4
I0608 10:32:38.112411  6131 net.cpp:127] Top shape: 10 384 13 13 (648960)
I0608 10:32:38.112416  6131 layer_factory.hpp:74] Creating layer conv5
I0608 10:32:38.112422  6131 net.cpp:90] Creating Layer conv5
I0608 10:32:38.112427  6131 net.cpp:410] conv5 <- conv4
I0608 10:32:38.112432  6131 net.cpp:368] conv5 -> conv5
I0608 10:32:38.112439  6131 net.cpp:120] Setting up conv5
I0608 10:32:38.113263  6131 net.cpp:127] Top shape: 10 256 13 13 (432640)
I0608 10:32:38.113279  6131 layer_factory.hpp:74] Creating layer relu5
I0608 10:32:38.113286  6131 net.cpp:90] Creating Layer relu5
I0608 10:32:38.113291  6131 net.cpp:410] relu5 <- conv5
I0608 10:32:38.113297  6131 net.cpp:357] relu5 -> conv5 (in-place)
I0608 10:32:38.113303  6131 net.cpp:120] Setting up relu5
I0608 10:32:38.113333  6131 net.cpp:127] Top shape: 10 256 13 13 (432640)
I0608 10:32:38.113339  6131 layer_factory.hpp:74] Creating layer pool5
I0608 10:32:38.113347  6131 net.cpp:90] Creating Layer pool5
I0608 10:32:38.113350  6131 net.cpp:410] pool5 <- conv5
I0608 10:32:38.113356  6131 net.cpp:368] pool5 -> pool5
I0608 10:32:38.113363  6131 net.cpp:120] Setting up pool5
I0608 10:32:38.113502  6131 net.cpp:127] Top shape: 10 256 6 6 (92160)
I0608 10:32:38.113520  6131 layer_factory.hpp:74] Creating layer fc6
I0608 10:32:38.113528  6131 net.cpp:90] Creating Layer fc6
I0608 10:32:38.113533  6131 net.cpp:410] fc6 <- pool5
I0608 10:32:38.113538  6131 net.cpp:368] fc6 -> fc6
I0608 10:32:38.113545  6131 net.cpp:120] Setting up fc6
I0608 10:32:38.140440  6131 net.cpp:127] Top shape: 10 4096 (40960)
I0608 10:32:38.140478  6131 layer_factory.hpp:74] Creating layer relu6
I0608 10:32:38.140492  6131 net.cpp:90] Creating Layer relu6
I0608 10:32:38.140498  6131 net.cpp:410] relu6 <- fc6
I0608 10:32:38.140506  6131 net.cpp:357] relu6 -> fc6 (in-place)
I0608 10:32:38.140516  6131 net.cpp:120] Setting up relu6
I0608 10:32:38.140576  6131 net.cpp:127] Top shape: 10 4096 (40960)
I0608 10:32:38.140583  6131 layer_factory.hpp:74] Creating layer drop6
I0608 10:32:38.140589  6131 net.cpp:90] Creating Layer drop6
I0608 10:32:38.140594  6131 net.cpp:410] drop6 <- fc6
I0608 10:32:38.140599  6131 net.cpp:357] drop6 -> fc6 (in-place)
I0608 10:32:38.140605  6131 net.cpp:120] Setting up drop6
I0608 10:32:38.140611  6131 net.cpp:127] Top shape: 10 4096 (40960)
I0608 10:32:38.140616  6131 layer_factory.hpp:74] Creating layer fc7
I0608 10:32:38.140622  6131 net.cpp:90] Creating Layer fc7
I0608 10:32:38.140630  6131 net.cpp:410] fc7 <- fc6
I0608 10:32:38.140636  6131 net.cpp:368] fc7 -> fc7
I0608 10:32:38.140643  6131 net.cpp:120] Setting up fc7
I0608 10:32:38.153045  6131 net.cpp:127] Top shape: 10 4096 (40960)
I0608 10:32:38.153095  6131 layer_factory.hpp:74] Creating layer relu7
I0608 10:32:38.153105  6131 net.cpp:90] Creating Layer relu7
I0608 10:32:38.153112  6131 net.cpp:410] relu7 <- fc7
I0608 10:32:38.153120  6131 net.cpp:357] relu7 -> fc7 (in-place)
I0608 10:32:38.153129  6131 net.cpp:120] Setting up relu7
I0608 10:32:38.153200  6131 net.cpp:127] Top shape: 10 4096 (40960)
I0608 10:32:38.153206  6131 layer_factory.hpp:74] Creating layer drop7
I0608 10:32:38.153214  6131 net.cpp:90] Creating Layer drop7
I0608 10:32:38.153219  6131 net.cpp:410] drop7 <- fc7
I0608 10:32:38.153224  6131 net.cpp:357] drop7 -> fc7 (in-place)
I0608 10:32:38.153231  6131 net.cpp:120] Setting up drop7
I0608 10:32:38.153237  6131 net.cpp:127] Top shape: 10 4096 (40960)
I0608 10:32:38.153242  6131 layer_factory.hpp:74] Creating layer fc-rcnn
I0608 10:32:38.153249  6131 net.cpp:90] Creating Layer fc-rcnn
I0608 10:32:38.153254  6131 net.cpp:410] fc-rcnn <- fc7
```

```
I0608 10:32:38.153259  6131 net.cpp:368] fc-rcnn -> fc-rcnn
I0608 10:32:38.153267  6131 net.cpp:120] Setting up fc-rcnn
I0608 10:32:38.154058  6131 net.cpp:127] Top shape: 10 200 (2000)
I0608 10:32:38.154080  6131 net.cpp:194] fc-rcnn does not need backward computation.
I0608 10:32:38.154085  6131 net.cpp:194] drop7 does not need backward computation.
I0608 10:32:38.154090  6131 net.cpp:194] relu7 does not need backward computation.
I0608 10:32:38.154095  6131 net.cpp:194] fc7 does not need backward computation.
I0608 10:32:38.154100  6131 net.cpp:194] drop6 does not need backward computation.
I0608 10:32:38.154105  6131 net.cpp:194] relu6 does not need backward computation.
I0608 10:32:38.154110  6131 net.cpp:194] fc6 does not need backward computation.
I0608 10:32:38.154115  6131 net.cpp:194] pool5 does not need backward computation.
I0608 10:32:38.154129  6131 net.cpp:194] relu5 does not need backward computation.
I0608 10:32:38.154134  6131 net.cpp:194] conv5 does not need backward computation.
I0608 10:32:38.154139  6131 net.cpp:194] relu4 does not need backward computation.
I0608 10:32:38.154145  6131 net.cpp:194] conv4 does not need backward computation.
I0608 10:32:38.154150  6131 net.cpp:194] relu3 does not need backward computation.
I0608 10:32:38.154155  6131 net.cpp:194] conv3 does not need backward computation.
I0608 10:32:38.154160  6131 net.cpp:194] norm2 does not need backward computation.
I0608 10:32:38.154165  6131 net.cpp:194] pool2 does not need backward computation.
I0608 10:32:38.154170  6131 net.cpp:194] relu2 does not need backward computation.
I0608 10:32:38.154175  6131 net.cpp:194] conv2 does not need backward computation.
I0608 10:32:38.154180  6131 net.cpp:194] norm1 does not need backward computation.
I0608 10:32:38.154193  6131 net.cpp:194] pool1 does not need backward computation.
I0608 10:32:38.154198  6131 net.cpp:194] relu1 does not need backward computation.
I0608 10:32:38.154203  6131 net.cpp:194] conv1 does not need backward computation.
I0608 10:32:38.154208  6131 net.cpp:235] This network produces output fc-rcnn
I0608 10:32:38.154220  6131 net.cpp:482] Collecting Learning Rate and Weight Decay.
I0608 10:32:38.154227  6131 net.cpp:247] Network initialization done.
I0608 10:32:38.154232  6131 net.cpp:248] Memory required for data: 62425920
E0608 10:32:38.221285  6131 upgrade_proto.cpp:618] Attempting to upgrade input file specified usir
I0608 10:32:38.324671  6131 upgrade_proto.cpp:626] Successfully upgraded file specified using depr
Loading input...
selective_search_rcnn({'/home/ouxinyu/caffe-master/examples/images/fish-
bike.jpg'}, '/tmp/tmpu85WGa.mat')
Processed 1570 windows in 17.131 s.
/usr/lib/python2.7/dist-packages/pandas/io/pytables.py:2487: PerformanceWarning:
your performance may suffer as PyTables will pickle object types that it cannot
map directly to c-types [inferred_type->mixed,key->block1_values] [items->['prediction']]

  warnings.warn(ws, PerformanceWarning)
Saved to _temp/det_output.h5 in 0.025 s.
```

4. 运行后输出的文件名，选择的窗口，检测得分存放在~/_temp/det_outpu.h5文件中，查看结果：

```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline

df = pd.read_hdf('_temp/det_output.h5', 'df')
print(df.shape)
print(df.iloc[0])
```

输出：

```
(1570, 5)
prediction    [-2.62247, -2.84579, -2.85122, -3.20838, -1.94...
ymin                                                     79.846
xmin                                                       9.62
ymax                                                     246.31
xmax                                                    339.624
Name: /home/sindyz/caffe-master/examples/images/fish-bike.jpg, dtype: object
```

Selective Search选出了1570个区域，作为R-CNN的输入，图与图的候选框的数量随图像的内容和大小不同而改变，也就是说：selective search不是尺度不变的。

通常，detect.py在运行大量图片时是非常高效的：首先，对所有图片提取候选框，用GPU批处理这些窗口，输出结

果。只要在images_file中列出图像名，就可以批处理了。

仅管本例中只给出了Imagenet的R-CNN检测，但是detect.py可以适应不同caffe模型的输入维度，批处理规模及输出类别。你可以根据需要选择模型定义和预处理模型，参考detect.py --help根据数据选择参数。

5. 加载ILSVRC13的检测类别名称，做预测的DataFrame, 注意，通过./data/ilsvrc12/get_ilsvrc12_aux.sh获取数据

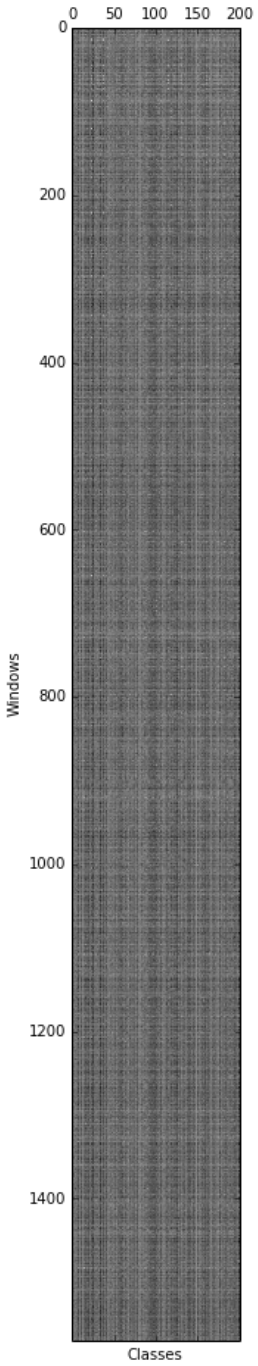
```python
with open('../data/ilsvrc12/det_synset_words.txt') as f:
    labels_df = pd.DataFrame([
        {
            'synset_id': l.strip().split(' ')[0],
            'name': ' '.join(l.strip().split(' ')[1:]).split(',')[0]
        }
        for l in f.readlines()
    ])
labels_df.sort('synset_id')
predictions_df = pd.DataFrame(np.vstack(df.prediction.values), columns=labels_df['name'])
print(predictions_df.iloc[0])
```

输出：

```plain
name
accordion      -2.622471
airplane       -2.845789
ant            -2.851220
antelope       -3.208377
apple          -1.949950
armadillo      -2.472936
artichoke      -2.201685
axe            -2.327404
baby bed       -2.737924
backpack       -2.176764
bagel          -2.681061
balance beam   -2.722538
banana         -2.390628
band aid       -1.598909
banjo          -2.298197
...
trombone        -2.582361
trumpet         -2.352853
turtle          -2.360860
tv or monitor   -2.761042
unicycle        -2.218468
vacuum          -1.907718
violin          -2.757080
volleyball      -2.723690
waffle iron     -2.418540
washer          -2.408994
water bottle    -2.174899
watercraft      -2.837426
whale           -3.120339
wine bottle     -2.772961
zebra           -2.742914
Name: 0, Length: 200, dtype: float32
```

6. 查看激活值并可视化

```python
plt.gray()
plt.matshow(predictions_df.values)
plt.xlabel('Classes')
plt.ylabel('Windows')
```



7. 取得分最大值，并输出

```python
max_s = predictions_df.max(0)
max_s.sort(ascending=False)
print(max_s[:10])
```

输出

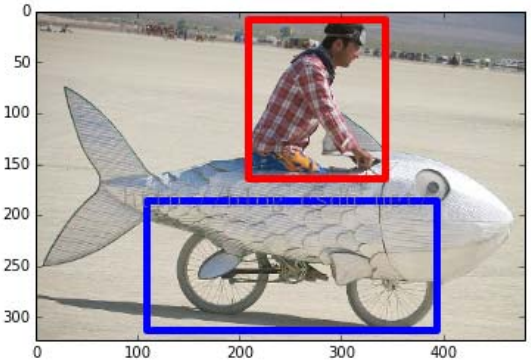
```
name
person          1.835771
bicycle         0.866109
unicycle        0.057079
motorcycle     -0.006122
banjo          -0.028208
turtle         -0.189833
electric fan   -0.206787
cart           -0.214237
lizard         -0.393519
helmet         -0.477942
dtype: float32
```

8. 检测结果最高的是人和自行车，检测还需要定位，于是，选择得分最高的人和自行车来定位

```python
# Find, print, and display the top detections: person and bicycle.
i = predictions_df['person'].argmax()
j = predictions_df['bicycle'].argmax()

# Show top predictions for top detection.
f = pd.Series(df['prediction'].iloc[i], index=labels_df['name'])
print('Top detection:')
print(f.order(ascending=False)[:5])
print('')

# Show top predictions for second-best detection.
f = pd.Series(df['prediction'].iloc[j], index=labels_df['name'])
print('Second-best detection:')
print(f.order(ascending=False)[:5])

# Show top detection in red, second-best top detection in blue.
im = plt.imread('examples/images/fish-bike.jpg')
plt.imshow(im)
currentAxis = plt.gca()

det = df.iloc[i]
coords = (det['xmin'], det['ymin']), det['xmax'] - det['xmin'], det['ymax'] - det['ymin']
currentAxis.add_patch(plt.Rectangle(*coords, fill=False, edgecolor='r', linewidth=5))

det = df.iloc[j]
coords = (det['xmin'], det['ymin']), det['xmax'] - det['xmin'], det['ymax'] - det['ymin']
currentAxis.add_patch(plt.Rectangle(*coords, fill=False, edgecolor='b', linewidth=5))
```

输出

```plain
Top detection:
name
person              1.835771
swimming trunks    -1.150371
rubber eraser      -1.231106
turtle             -1.266037
plastic bag        -1.303266
dtype: float32

Second-best detection:
name
bicycle     0.866109
unicycle   -0.359140
scorpion   -0.811621
lobster    -0.982891
lamp       -1.096809
dtype: float32
```



9. 拿所有的自行车检测，并用NMS避免窗口重叠。

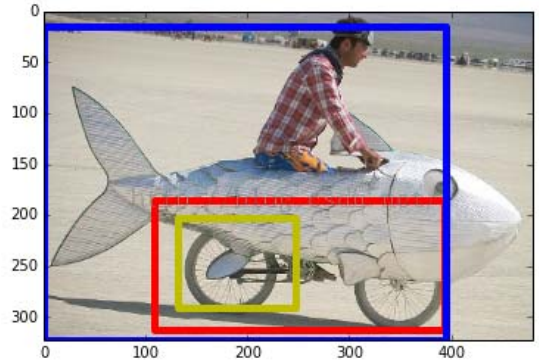
```python
def nms_detections(dets, overlap=0.3):
    """
```

```
    Non-maximum suppression: Greedily select high-scoring detections and
    skip detections that are significantly covered by a previously
    selected detection.

    This version is translated from Matlab code by Tomasz Malisiewicz,
    who sped up Pedro Felzenszwalb's code.

    Parameters
    ----------
    dets: ndarray
        each row is ['xmin', 'ymin', 'xmax', 'ymax', 'score']
    overlap: float
        minimum overlap ratio (0.3 default)

    Output
    ------
    dets: ndarray
        remaining after suppression.
    """
    x1 = dets[:, 0]
    y1 = dets[:, 1]
    x2 = dets[:, 2]
    y2 = dets[:, 3]
    ind = np.argsort(dets[:, 4])

    w = x2 - x1
    h = y2 - y1
    area = (w * h).astype(float)

    pick = []
    while len(ind) > 0:
        i = ind[-1]
        pick.append(i)
        ind = ind[:-1]

        xx1 = np.maximum(x1[i], x1[ind])
        yy1 = np.maximum(y1[i], y1[ind])
        xx2 = np.minimum(x2[i], x2[ind])
        yy2 = np.minimum(y2[i], y2[ind])

        w = np.maximum(0., xx2 - xx1)
        h = np.maximum(0., yy2 - yy1)

        wh = w * h
        o = wh / (area[i] + area[ind] - wh)

        ind = ind[np.nonzero(o <= overlap)[0]]

    return dets[pick, :]
```

```
[python]
scores = predictions_df['bicycle']
windows = df[['xmin', 'ymin', 'xmax', 'ymax']].values
dets = np.hstack((windows, scores[:, np.newaxis]))
nms_dets = nms_detections(dets)
```

10. 显示排名前3的NMS处理过的自行车，注意得分最高的红色的框与其他框之间的差异

```
[python]
plt.imshow(im)
currentAxis = plt.gca()
colors = ['r', 'b', 'y']
for c, det in zip(colors, nms_dets[:3]):
    currentAxis.add_patch(
        plt.Rectangle((det[0], det[1]), det[2]-det[0], det[3]-det[1],
        fill=False, edgecolor=c, linewidth=5)
    )
print 'scores:', nms_dets[:3, 4]
```



自行车的检测是个简单的实例，因为在训练数据中由这个类别的数据，但是人的结果是一个真正的检测因为训练数据中没有这个类别的数据。

下面，你也可以用自己的图像做检测。

11. 删除_temp目录

```
[python]
01. !rm -rf _temp
```

参考资料：

http://nbviewer.jupyter.org/github/BVLC/caffe/blob/master/examples/detection.ipynb

http://nbviewer.jupyter.org/github/ouxinyu/ouxinyu.github.io/blob/master/MyCodes/caffe-master/detection.ipynb

上一篇 caffe学习笔记10 -- Fine-tuning a Pretrained Network for Style Recognitio
下一篇 caffe学习笔记13 -- Setup

我的同类文章

caffe学习（19）

- caffe学习笔记14(外篇)--使… 2016-04-18 阅读 15
- caffe学习笔记1.1-- caffe的M… 2016-03-10 阅读 116
- caffe中matlab接口配置 2016-02-24 阅读 129
- caffe学习笔记11 -- Net Surg… 2016-02-29 阅读 110
- caffe学习笔记3.1 -- caffe的… 2016-04-20 阅读 8
- caffe学习笔记6--训练自己的… 2016-04-18 阅读 20
- caffe学习笔记5 -- Alex's CIF… 2016-04-12 阅读 31
- caffe学习笔记13 -- Setup 2016-02-24 阅读 130
- caffe学习笔记10 -- Fine-tuni… 2016-02-24 阅读 149

更多文章

参考知识库

猜你在找

查看评论

1楼 qq_21402107 前天 18:39发表

最初始的部分是在python下运行吗

Re: thystar 前天 22:09发表

回复qq_21402107：可以的，但是你最好用Ipython, 安装时很方便的。

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

核心技术类目

全部主题 Hadoop AWS 移动游戏 Java Android iOS Swift 智能硬件 Docker OpenStack
VPN Spark ERP IE10 Eclipse CRM JavaScript 数据库 Ubuntu NFC WAP jQuery
BI HTML5 Spring Apache .NET API HTML SDK IIS Fedora XML LBS Unity
Splashtop UML components Windows Mobile Rails QEMU KDE Cassandra CloudStack
FTC coremail OPhone CouchBase 云计算 iOS6 Rackspace Web App SpringSide Maemo
Compuware 大数据 aptech Perl Tornado Ruby Hibernate ThinkPHP HBase Pure Solr
Angular Cloud Foundry Redis Scala Django Bootstrap