# [Caffe for windows使用matlab运行RCNN demo](http://blog.csdn.net/jungieve/article/details/45324299)

分类： [计算机视觉](http://blog.csdn.net/Jungieve/article/category/3139107)2015-04-28 00:54 292人阅读 [评论](http://blog.csdn.net/jungieve/article/details/45324299#comments)(0) [收藏](javascript:void(0);) [举报](http://blog.csdn.net/jungieve/article/details/45324299#report)

[Caffe](http://www.csdn.net/tag/Caffe)[RCNN](http://www.csdn.net/tag/RCNN)[MATLAB](http://www.csdn.net/tag/MATLAB)[Windows](http://www.csdn.net/tag/Windows)

在上一篇中配置好Caffe for windows并配置matlab接口之后，我们尝试运行RCNN（区域卷积神经网络）的demo，在给出该方法的思想之前，我们尝试运行demo看看效果如何。此处特别感谢@kai提供指导。   
## 一、下载代码与模型 ##   
首先在<https://github.com/rbgirshick/rcnn.git>下载RCNN的源代码，并在<http://disi.unitn.it/~uijlings/MyHomepage/index.php#page=projects1>下载【Selective Search】(这个东西是什么我们接下来再说，先不要管他。)在ubuntu系统下运行 ./data/fetch\_models.sh下载模型，总共大约1.5G。   
全部下载后，将Selective Search复制到rcnn-master/selective-search，之后下载好的模型解压，得到的caffe\_net和rcnn\_models两个文件夹全部复制到windows系统下rcnn-master/data/。完毕   
## 二、运行demo ##   
打开matlab，将%caffe-root加入matlab的path中。   
cd %rcnn-master 在命令行中输入 rcnn\_build()来生成liblinear。结果如下：

Compiling the anisotropic gauss filtering of:

J. Geusebroek, A. Smeulders, and J. van de Weijer

Fast anisotropic gauss filtering

IEEE Transactions on Image Processing, 2003

Source code/Project page:

http://staff.science.uva.nl/~mark/downloads.html#anigauss

Compiling the segmentation algorithm of:

P. Felzenszwalb and D. Huttenlocher

Efficient Graph-Based Image Segmentation

International Journal of Computer Vision, 2004

Source code/Project page:

http://www.cs.brown.edu/~pff/segment/

Note: A small Matlab wrapper was made.

Compiling liblinear version 1.94

Source code page:

http://www.csie.ntu.edu.tw/~cjlin/liblinear/

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16
* 17

之后在命令行输入key=caffe(‘get\_init\_key’);以验证我们caffe和matlab的wrapper是否正确。   
输入rcnn\_demo查看，运行我们的RCNN\_DEMO，显示如下

Welcome to the PASCAL demo

Running in GPU mode

(To run in CPU mode, call rcnn\_demo(demo\_choice, 0) instead)

Press any key to continue

Initializing R-CNN model (this might take a little while)

done

Computing candidate regions...found 2100 candidates (in 10.069s).

Extracting CNN features from regions...done (in 159.304s).

Scoring regions with detectors...done (in 1.704s)

Applying NMS...done (in 13.593s)

Total 20-class detection time: 192.384s

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11

结果如下：   
