Windows-Faster-rcnn安装全过程

第0章 软件版本及硬件型号

0.1 软件版本

Win7、CUDA8、CUDNN5.1 for CUDA8、opencv2.4、python版本faster-rcnn、GPU版本、faster训练方式为alt-opt、这里用的faster-rcnn是linux下编译、更改好的（参见1、2教程）

0.2 硬件型号

I7四代CPU、内存4G、固态硬盘250G、影驰GTX1070

第1章 win7系统安装过程

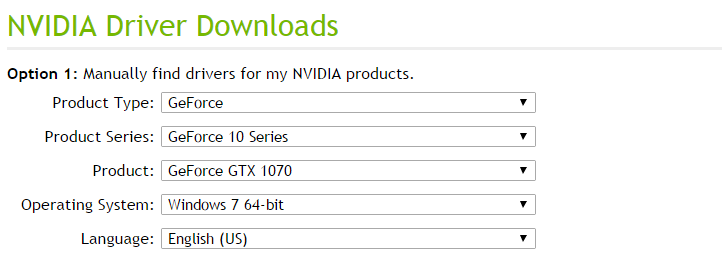
选择光盘安装或U盘安装都可

注：win7系统和win10系统的faster-rcnn安装过程应该是一样的

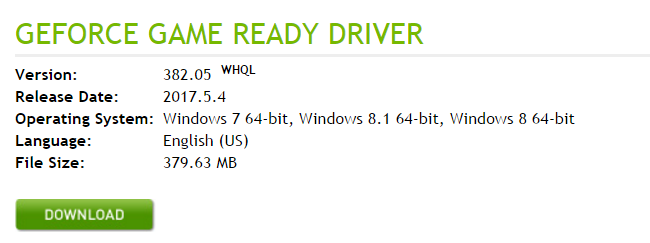
第2章 安装NVIDIA显卡驱动

2.1 查询NVIDIA驱动

首先去官网(<http://www.nvidia.com/Download/index.aspx?lang=en-us>)查看适合自己显卡的驱动：



我下载的驱动版本如下：



2.2 安装NVIDIA驱动

1. 双击.exe文件即可安装

2. 安装完毕后，在桌面空白处单击鼠标右键->个性化->选择Aero主题，若任务栏变为透明，则说明驱动安装成功

第3章 安装VS2013

3.1 下载VS2013

下载地址：<https://www.xp510.com/xiazai/Application/program/24261.html>

下载旗舰版，约2.9G

3.2 安装VS2013

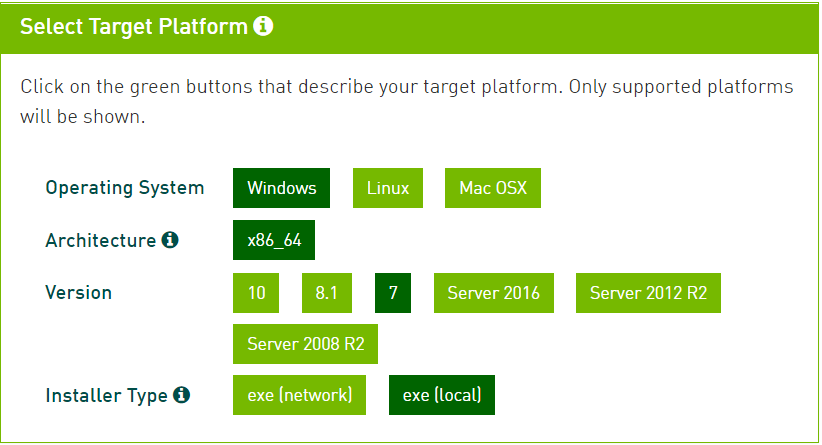
参考网址：<http://jingyan.baidu.com/article/09ea3ede3b2496c0afde3944.html>

第4章 安装CUDA

4.1 下载CUDA

CUDA是NVIDIA的编程语言平台，想使用GPU就必须要使用cuda

首先在官网上(<https://developer.nvidia.com/cuda-downloads>)下载CUDA：



4.2 安装CUDA

1. 双击安装，选择默认路径及组件安装

2. 环境变量配置：

在计算机上点右键，打开属性->高级系统设置->环境变量，可以看到系统中多了CUDA\_PATH和CUDA\_PATH\_V8\_0两个环境变量，接下来，还要在系统中添加以下几个环境变量：

CUDA\_SDK\_PATH = C:\ProgramData\NVIDIA Corporation\CUDA Samples\v8.0

CUDA\_LIB\_PATH = %CUDA\_PATH%\lib\x64

CUDA\_BIN\_PATH = %CUDA\_PATH%\bin

CUDA\_SDK\_BIN\_PATH = %CUDA\_SDK\_PATH%\bin\win64

CUDA\_SDK\_LIB\_PATH = %CUDA\_SDK\_PATH%\common\lib\x64

然后，在系统变量 PATH 的末尾添加：

;%CUDA\_LIB\_PATH%;%CUDA\_BIN\_PATH%;%CUDA\_SDK\_LIB\_PATH%;%CUDA\_SDK\_BIN\_PATH%;

重新启动计算机。

4.3 配置环境变量

1. 打开~/.bashrc文件：

sudo gedit ~/.bashrc

2. 将以下内容写入到~/.bashrc尾部：

export PATH=/usr/local/cuda-8.0/bin${PATH:+:${PATH}}

export LD\_LIBRARY\_PATH=/usr/local/cuda8.0/lib64${LD\_LIBRARY\_PATH:+:${LD\_LIBRARY\_PATH}}

4.4 验证CUDA8.0已正确安装

1. 打开cmd，输入nvcc –V

结果如下图：



4.5 编译CUDA8.0自带的samples

1. 在C:\ProgramData\NVIDIA Corporation\CUDA Samples\v8.0目录下，用vs2013，打开Samples\_vs2013.sln

如果这时候出现：



此时，是未找到CUDA 8.0.props

解决方案：将C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v8.0\extras\visual\_studio\_integration\MSBuildExtensions下的4个文件复制到C:\Program Files (x86)\MSBuild\Microsoft.Cpp\v4.0\V120\BuildCustomizations目录下，重新启动VS2013

2. 先选择release模式（debug会很慢），选中解决方案Samples\_vs2013，重新生成解决方案

3. 需要说明的是个别例子编译失败，比如自己编译的时候成功135个，失败10个。失败的原因是缺乏相应的头文件；失败也没有什么影响

4. 编译成功后可在C:\ProgramData\NVIDIA Corporation\CUDA Samples\v8.0\bin\win64\Release目录下面找到全部例子的可执行文件

4.6 参考网址

1. <http://www.cnblogs.com/hzm12/p/6422701.html>

2. <http://blog.csdn.net/u011821462/article/details/50145221>

第5章 安装cuDNN（上传版需要去掉账号）

5.1 下载cuDNN

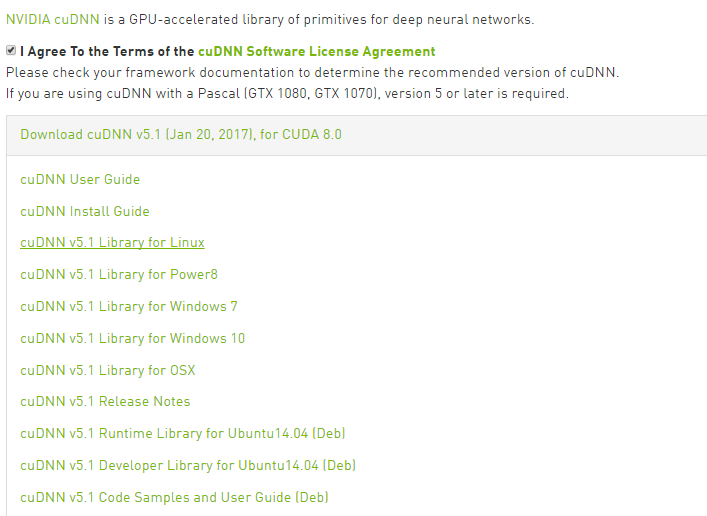
cuDNN是GPU加速计算深层神经网络的库

首先去官网(https://developer.nvidia.com/rdp/cudnn-download)下载cuDNN，需要注册一个账号才能下载：

账号：

密码：

下载for windows7



5.2 安装cuDNN

1. 下载完成后解压，cuda目录下有bin，include，lib三个文件夹

2. 拷贝目录至CUDA8.0的文件夹

分别将cuda\bin、cuda\include、cuda\lib三个目录中的内容拷贝到C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v8.0对应的bin、include、lib目录下即可

5.3 参考网址

1. <http://blog.csdn.net/xuzhongxiong/article/details/52717285>

2. <http://blog.csdn.net/baidu_32173921/article/details/53510764>

第6章 安装caffe

6.1 安装python环境

这里选择的是Anaconda2，它是一个一款集成安装环境，包含numpy、scipy等包

下载地址：<https://mirrors.tuna.tsinghua.edu.cn/anaconda/archive/>

1. 打开 Anaconda2 下载地址，选择Anaconda2-4.1.1-Windows-x86\_64.exe

2. 下载完毕后双击运行Anaconda2-4.1.1-Windows-x86\_64.exe

3. 可以选择安装路径

6.2 下载caffe

进入<https://github.com/Microsoft/caffe>，点击 Clone or download-> Download ZIP

6.3 解压caffe

对caffe-master.zip进行解压，生成caffe-master文件夹放到D盘根目录

6.4 打开并修改配置文件

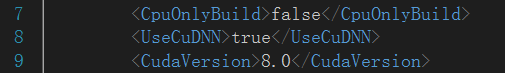
1. 打开到caffe-master/caffe-master/windows 文件夹

复制CommonSettings.props.example文件，并把复制好的文件重命名为CommonSettings.props

2. 用VS2013打开caffe-master/caffe-master/windows/Caffe.sln

3. 打开CommonSettings.props

4. 第7行cpuonlybuild如果true就是cpu版本，false就是gpu版本，gpu版本下面第8行要使用cudnn改为true，cuda的版本改为我们的版本8.0，如图：



5. 第13行的Pythonsupport就是选择支持哪些接口，我们用的是python接口改为true

<PythonSupport>true</PythonSupport>

若此项未改，则运行faster-demo的时候会出现错误：

No module named \_caffe

6. 第21行根据显卡计算能力设置，我的是6.1，设置为

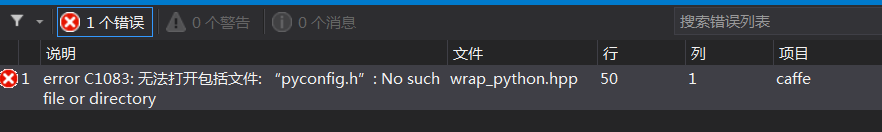
<CudaArchitecture>compute\_61,sm\_61</CudaArchitecture>

7. 第 24 行改为你的cuDNN 的解压文件中cuda文件夹的路径，我的cuda文件夹放在caffe-master下，所以我的路径为：

<CuDnnPath>D:\caffe-master</CuDnnPath>

注：不加这步也编译成功了，可能因为前面安装cudnn时已经操作过了

8. 第 48 行改为Anaconda2的安装文件夹，即python的路径：D:\install\Anaconda2如不更改则会出现如下错误：



6.5 需要安装的相关软件及依赖（这步是多余的，caffe编译的的时候自动安装各种软件及依赖，忽略）

1. protobuf

下载地址：https://github.com/google/protobuf/releases/tag/v3.0.0

下载两个文件：protoc-3.0.0-win32.zip和Source code (zip)

a. 解压两个文件

b. 将protoc-3.0.0-win32\bin\protoc.exe 拷贝进入源代码文件夹protobuf-3.0.0/src中

c. 将protobuf-3.0.0拷贝到C盘根目录（否则无法cd）

d. 已管理员身份运行cmd

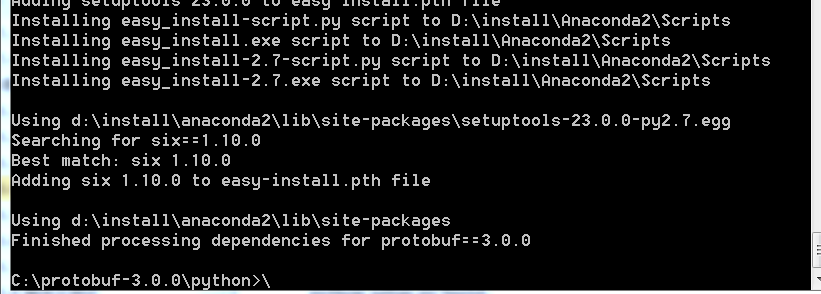
e. 依次执行：

cd C:\protobuf-3.0.0\python

python setup.py build

python setup.py install

最后显示下图表明安装成功



注：如果不安装protobuf直接编译caffe，则会出现如下错误：

C:\Users\JC\Desktop\proto.PNG

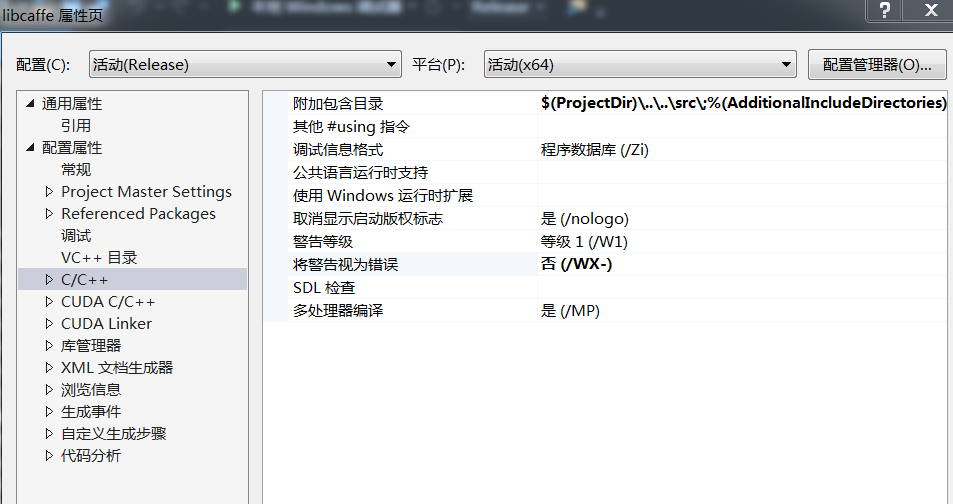
6.6 编译libcaffe（必须先编译）

1. 右键libcaffe->设置为启动项

2. 选择编译类型为release，x64

3. 关闭视警告为错误(即设置为No)如果不设置的话在编译boost库的时候会由于文字编码的警告而报错

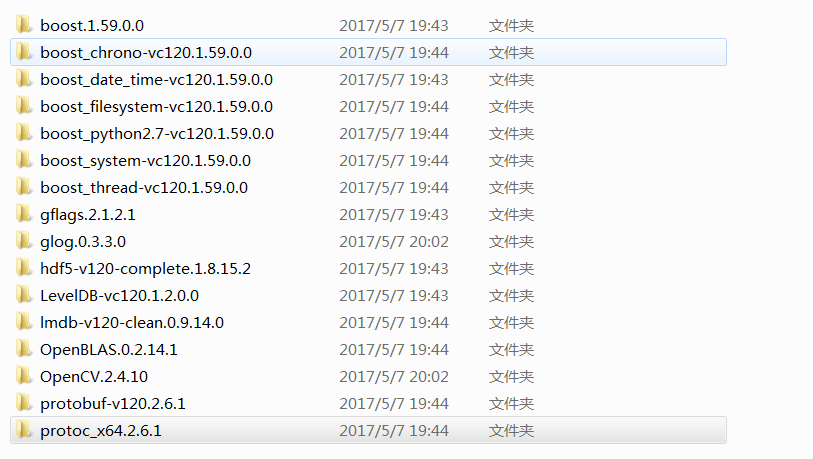
2、3设置如下图所示：



4. 然后就在开始编译，先编译的是libcaffe：右键libcaffe->重新生成

这个时候,在你刚刚解压出来的caffe-master文件夹下面就应该出现一个名叫Build的文件夹，以后编译的成功和运行需要的文件都会存放在Build\X64\Relase下面

5. 注意：之所以选择微软官方的版本，是因为在编译过程中会自动下载一些依赖库（包括opencv2.4），但在这部操作时必须连网，(见下图)不用手动选择，所以非常简单方便，路径是：D:\NugetPackages



6. 如果要重新编译caffe，需要先清理解决方案

否则会出现57个错误

6.7 编译解决方案下剩余项目

在libcaffe单独编译成功之后，右键caffe解决方法->点击生成解决方案（不是重新生成解决方案，会把编译好的libcaffe删除）

编译成功后显示：

13> test\_all.vcxproj -> D:\caffe-master\windows\..\Build\x64\Release\test\_all.exe

========== 生成: 成功 15 个，失败 0 个，最新 1 个，跳过 0 个 ==========

6.8 MNIST数据集测试

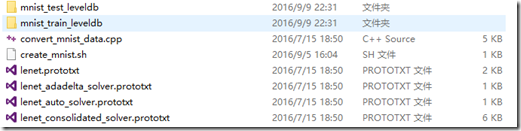
配置caffe完成后，我们可以利用MNIST数据集对caffe进行测试，过程如下：

1. 先下载mnist数据集：

这里需要注意的是caffe并不直接通过下载得到的四个文件进行训练，而是会把它转化为lmdb或leveldb格式进行读取，实现这个转化的代码是\examples\mnist的convert\_mnist\_data.cpp，但也没必要看，因为它的运行脚本create\_mnist.sh显然是linux下的东西

目前更很好的办法就是直接网盘下载，这里分享一个<http://pan.baidu.com/s/1c2G9qyk> ，提取码：xama

里面是已经经过转换的leveldb格式的训练集和测试集，把这两个文件夹直接放到\examples\mnist目录下，如下图所示：



2. 接着再用VS打开\examples\mnist目录下的lenet\_train\_test.prototxt，做如下修改以正确指定训练集和测试集

name: "LeNet"

layer {

name: "mnist"

type: "Data"

top: "data"

top: "label"

include {

phase: TRAIN

}

transform\_param {

scale: 0.00390625

}

data\_param {

source: "examples/mnist/mnist\_train\_leveldb"

batch\_size: 64

backend: LEVELDB

}

}

layer {

name: "mnist"

type: "Data"

top: "data"

top: "label"

include {

phase: TEST

}

transform\_param {

scale: 0.00390625

}

data\_param {

source: "examples/mnist/mnist\_test\_leveldb"

batch\_size: 100

backend: LEVELDB

}

}

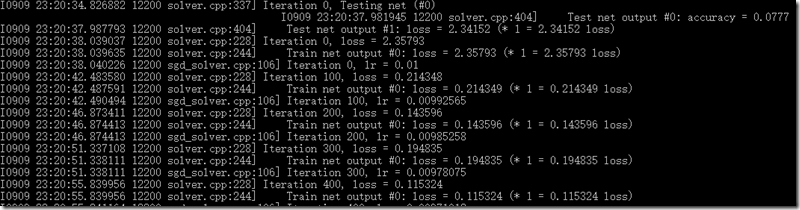
3. 完成上述工作后就可以编写bat脚本进行正式训练了

在D:\caffe-master文件夹下新建一个run.txt并写入以下内容：

Build\x64\Release\caffe.exe train --solver=examples\mnist\lenet\_solver.prototxt

pause

将后缀名改成bat后双击运行，不出意料，应该会出现类似如下的训练过程：



4. 当训练结束后，会在目录\examples\mnist下产生训练出的模型(文件后缀名为caffemodel和solverstate)

备注（这步不需要操作）：cd到D:\caffe-master\Build\x64\Release

1. 先进入D盘根目录（需要中转一下），在cmd中输入

d:

2. cd到release

cd D:\caffe-master\Build\x64\Release

参考网址：<http://www.cnblogs.com/yixuan-xu/p/5858595.html>

6.9 参考网址

1. <http://blog.csdn.net/jasonzhangoo/article/details/54866120>

2. <http://blog.csdn.net/xierhacker/article/details/51834563>

3. <http://www.cnblogs.com/yixuan-xu/p/5858595.html>

第7章 安装opencv2.4

7.1 下载opencv2.4.13.2

Linux下我安装的是3.1版本，这里装的是2.4版本，两个版本都可以

1. 下载地址：<https://github.com/opencv/opencv/releases/tag/2.4.13.2>

2. 选择opencv-2.4.13.2-vc14.exe进行下载

7.2 安装

1. 双击.exe安装，选择解压路径（其实就是一个解压的过程）

2. 选择opencv-2.4.13.2-vc14.exe进行下载

3. 将解压后opencv文件夹放大任意全英文目录下

7.3 建立与python2.7的连接

1. 安装完成后，将D:\install\opencv\build\python\2.7\x64下的cv2.pyd拷贝到D:\install\Anaconda2\Lib\site-packages下面

2. 在cmd下面输入python进入python命令行，然后输入import cv2，如没有错误信息证明安装成功

7.4 参考网址

1. <http://blog.csdn.net/qq_14845119/article/details/52354394>

第8章 faster rcnn安装

8.1 获得faster rcnn

直接把ubuntu下编译好faster-rcnn复制过来

8.2 复制caffe文件建立连接

1. 将D:\faster-rcnn\caffe-fast-rcnn\python下的caffe文件夹删除

2. 把D:\caffe-master\Build\x64\Release\pycaffe文件夹下的caffe文件夹复制到1的目录中

caffe编译下Build/x64/Release/pycaffe/caffe目录复制到py-faster-rcnn/caffe-faster-rcnn/python目录下，替换原有文件

8.3 替换windows下的lib包

1. 下载MrGF修改的lib

下载地址：<https://github.com/MrGF/py-faster-rcnn-windows>

2. 删除D:\faster-rcnn下的原始lib，将1中下载的lib文件拷贝进来

注：若不替换lib包，则会出现：

no module named gpu\_nms错误

8.4 param\_str\_问题

运行demo时遇到AttributeError:‘ProposalLayer’ object has no attribute ‘param\_str\_’问题：

打开D:\faster-rcnn\lib\rpn下proposal\_layer.py文件，修改param\_str\_为param\_str

8.5 没有roi\_pooling层问题

运行demo时遇到Check failed: registry.count(type) == 1 (0 vs. 1) Unknown layer问题，这个是因为caffe工程中没有roi\_pooling层，在2016年9月21日在github上下载的caffe中已经有了这个layer，但windows branch的工程文件中没有添加进去

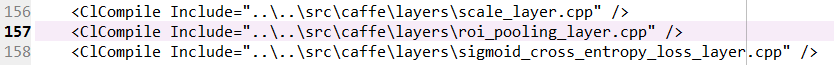
1. 找到D:\caffe-master\windows\libcaffe下的文件

2. 用spyder打开libcaffe.vcxproj文件

a. 在157行加入

<ClCompile Include="..\..\src\caffe\layers\roi\_pooling\_layer.cpp" />

如图：



b. 在265行加入

<ClInclude Include="..\..\include\caffe\layers\roi\_pooling\_layer.hpp" />

如图：

C:\Users\JC\Desktop\2.PNG

c. 在344行加入

<CudaCompile Include="..\..\src\caffe\layers\roi\_pooling\_layer.cu" />

如图：

C:\Users\JC\Desktop\3.PNG

2. 用spyder打开libcaffe.vcxproj.filters

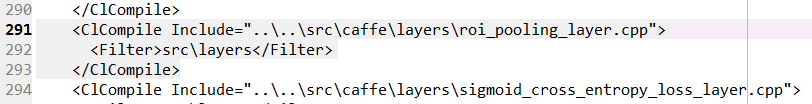
a. 在291行加入

<ClCompile Include="..\..\src\caffe\layers\roi\_pooling\_layer.cpp">

<Filter>src\layers</Filter>

</ClCompile>

如图：



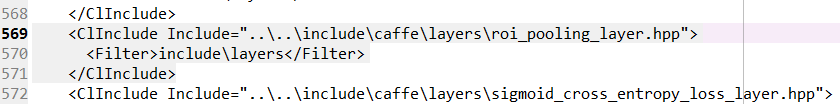
b. 在569行加入

<ClInclude Include="..\..\include\caffe\layers\roi\_pooling\_layer.hpp">

<Filter>include\layers</Filter>

</ClInclude>

如图：



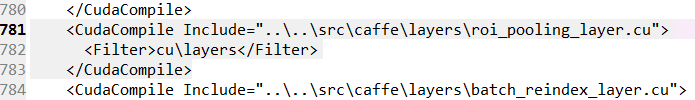
c. 在781行加入

<CudaCompile Include="..\..\src\caffe\layers\roi\_pooling\_layer.cu">

<Filter>cu\layers</Filter>

</CudaCompile>

如图：



8.6 解决以上问题后重新编译caffe

1. 解决上面的问题后，需要重新编译caffe，参见6.6

a. 先删除caffe-master下Build文件夹

b. 清理解决方案

c. 按照6.6节进行重新编译

2. 重新复制caffe文件建立连接，参见8.2节

8.7 keyerror问题

运行demo.py是出现proposal\_layer.py的keyerror:’1’错误

解决方案：

1. 用spyder打开D:\faster-rcnn\lib\rpn下proposal\_layer.py文件

2. 将64行

C:\Users\JC\Desktop\7.PNG

改为：cfg\_key = 'TEST'#str(self.phase)

8.8 运行demo.py

解决上面的问题后，即可运行demo

1. 打开dos界面

2. 输入：

d:

3. 输入：

cd D:\faster-rcnn\tools

4. 输入：

python demo.py

5. 等待运行完，显示结果图

8.9 参考网址

1. <http://www.th7.cn/system/win/201609/181466.shtml>

2. <http://www.voidcn.com/blog/yueqiang525/article/p-6390560.html>

3. <http://blog.csdn.net/sunmelon/article/details/54600178>