# **linux下安装matlab r2014a**

1 从<http://pan.baidu.com/s/1qYJ9tNm> 下载Matlab2014的Linux版本及破解文件

注:下载的镜像有两部分，解压后合成一个MATHWORKS\_R2014A.iso

2 下载完成后将iso文件挂载到Linux

sudo mkdir /media/matlab

mount -o **loop** path/of/iso/file/filenam.iso /media/matlab

【注意，这里要有loop不然会报错，见文末debug细节】

3.安装

cd /media/matlab

sudo ./install

4.安装过程中使用readme.txt中的序列号

2016a序列号：09806-07443-53955-64350-21751-41297

5. 破解

5.1打开安装目录下的/bin/activate\_matlab.exe

cd /usr/local/MATLAB/R2016a/bin

sudo ./activate\_matlab.exe

5.2选择不使用互联网手动激活：

选择“输入许可证文件的完成路径（包括文件名）”，选择下载的安装包里Crack文件夹中lic文件

5.3将Crack文件夹里的 libmwservices.so拷贝到安装目录下的/bin/glnxa64

sudo cp libmwservices.so /usr/local/MATLAB/R2016a/bin/glnxa64

6.测试

sudo matlab 即可打开matlab

(如果不行，则先source ~/.bashrc)

7.桌面快捷方式

7.1为了避免每次都用root权限打开matlab，通过sudo chown [your ubuntu username] -R ~/.matlab改变权限，直接在终端输入matlab就可以打开MATLAB了。

7.2创建桌面快捷方式：

sudo chmod u+x /usr/share/applications/matlab.desktop

cp /usr/share/applications/matlab.desktop /home/chang/Desktop/

【如果不存在matlab.desktop，可以gedit **/usr/share/applications/matlab.desktop，并输入以下内容保存：**

[Desktop Entry]

Version=2016a

Type=Application

Terminal=false

Exec=matlab -desktop

Name=MATLAB

Icon=/usr/share/icons/hicolor/48x48/apps/matlab.png

Categories=Development;Math;Science

Comment=Scientific computing environment

StartupNotify=true

StartupWMClass=com-mathworks-util-PostVMInit】

debug细节：  
1. 挂载iso文件

命令mount -o **loop** path/of/iso/file/filenam.iso /media/matlab

部分教程用的命令mount -o path/of/iso/file/filenam.iso /media/matlab，可能会报错：mount:can't find /media/matlab in /etc/fstab or /etc/mtab

>>>fstab(/etc/fstab)是Linux下比较重要的配置文件，它包含了系统在启动时挂载文件系统和存储设备的详细信息。fstab就是来专门制定挂载方式和信息的。可以cat /etc/fstab查看

>>>网友给出的方案：往/etc/fstab中增加一条

/data/Matlab\_pkg/R2016a\_glnxa64.iso /home/chang/matlab iso9660 defaults 0 0

我们来举个例子

面是我机子上的fstab文件：  
LABEL=/                 /                       ext3    defaults        1 1  
LABEL=/boot1            /boot                   ext3    defaults        1 2  
tmpfs                   /dev/shm                tmpfs   defaults        0 0  
devpts                  /dev/pts                devpts  gid=5,mode=620  0 0  
sysfs                   /sys                    sysfs   defaults        0 0  
proc                    /proc                   proc    defaults        0 0  
LABEL=SWAP-sda11        swap                    swap    defaults        0 0  
/dev/sda6               /mnt/winE               vfat    defaults        0 0  
/dev/sda8               /mnt/winG               ntfs    defaults        0 0

第一列是需要挂载的文件系统或者是存储设备或者是需要挂载的目录。

第二列是挂载点

第三列是文件系统或者是分区的类型（其实分区类型就是中文件系统）

第四列是以何种形式挂载，比如

rw读写 auto 自动挂载Ro只读等等参数。

不过最常用的是defaults。

defaults是rw，suid，dev，exec，auto，nouser，async等的组合。

  第五列为dump选项，设置是否让备份程序dump备份文件系统，0为忽略，1为备份。  
  第六列为fsck选项，告诉fsck程序以什么顺序检查文件系统，0为忽略。  
‘/etc/fstab’有几个用处：

* 决定开机时自动挂载哪些介质；
* 指定每个介质挂载时的可选项、载入点；
* 系统用来挂载几个虚拟文件系统。

#### 开机时挂载介质

默认下，开机时‘fstab’中列出的所有介质都将被挂载。如果其中某个介质出了问题，‘mount’会显示出错信息，然后继续下一条目。请注意，这对于网络介质，比如 NFS 或 SMB 共享，也有效。如果想取消开机时的自动挂载，您得在‘/etc/fstab’给相应条目提供‘noauto’可选项。

#### 指定挂载的可选项和载入点

‘mount’命令有两种可选项：常规可选项、文件系统的特殊可选项。

特殊可选项针对特定的文件系统。如果您在‘fstab’中给某个条目提供了特殊可选项（比如用于 Windows FAT 文件系统的 umask=0），只有相应文件系统的介质才可以通过该条目被挂载。如果您想挂载其他文件系统的介质，要么再设置那个可选项，要么创建单独的条目。

指定载入点有个好处，以后可以省略挂载／卸载时的设备文件名。

mount mount\_point

就将在‘/etc/fstab’寻找相应载入点的条目，并按照预先设定完成挂载。

具体ref:blog.csdn.net/zhaole20094463/article/details/7540100

----然而这个解决方案并没有用！！！所以还是加个loop完事。

2.关于MATLAB 中的 opengl错误

com.jogamp.opengl.GLException: X11GLXDrawableFactory - Could not initialize shared resources for X11GraphicsDevice[type .x11, connection :0, unitID 0, handle 0x0, owner false, ResourceToolkitLock[obj 0x535b6672, isOwner false, <15a2f11f, 7c843551>[count 0, qsz 0, owner <NULL>]]]  
at jogamp.opengl.x11.glx.X11GLXDrawableFactory$SharedResourceImplementation.createSharedResource(X11GLXDrawableFactory.java:326)  
at jogamp.opengl.SharedResourceRunner.run(SharedResourceRunner.java:297)  
at java.lang.Thread.run(Unknown Source)  
Caused by: com.jogamp.opengl.GLException: glXGetConfig(0x1) failed: error code Unknown error code 6  
at jogamp.opengl.x11.glx.X11GLXGraphicsConfiguration.glXGetConfig(X11GLXGraphicsConfiguration.java:570)  
at jogamp.opengl.x11.glx.X11GLXGraphicsConfiguration.XVisualInfo2GLCapabilities(X11GLXGraphicsConfiguration.java:500)  
at jogamp.opengl.x11.glx.X11GLXGraphicsConfigurationFactory.chooseGraphicsConfigurationXVisual(X11GLXGraphicsConfigurationFactory.java:434)  
at jogamp.opengl.x11.glx.X11GLXGraphicsConfigurationFactory.chooseGraphicsConfigurationStatic(X11GLXGraphicsConfigurationFactory.java:240)  
at jogamp.opengl.x11.glx.X11GLXDrawableFactory.createMutableSurfaceImpl(X11GLXDrawableFactory.java:524)  
at jogamp.opengl.x11.glx.X11GLXDrawableFactory.createDummySurfaceImpl(X11GLXDrawableFactory.java:535)  
at jogamp.opengl.x11.glx.X11GLXDrawableFactory$SharedResourceImplementation.createSharedResource(X11GLXDrawableFactory.java:283)

... 2 more

这个主要是由于英伟达的独显驱动引起的:

cd /usr/local/MATLAB/R2016b/bin

./matlab -softwareopengl

然后在 MATLAB窗口中运行

opengl('save','software')

就可以解决了

3.sudo apt-get install matlab-support安装matlab-support报错：

**Setting up matlab-support (0.0.19) ...**

**cp: cannot create regular file ‘/etc/matlab/debconf’: No such file or directory**

**dpkg: error processing package matlab-support (--configure):**

**subprocess installed post-installation script returned error exit status 1**

**Errors were encountered while processing:**

**matlab-support**

**E: Sub-process /usr/bin/dpkg returned an error code (1)**

----其实就时user没有权限创建文件，所以

sudo mkdir /etc/matlab

再sudo apt-get install matlab-support就没问题了

ref:

<https://www.jianshu.com/p/60038ffa8870>

https://help.ubuntu.com/community/MATLAB

<https://jingyan.baidu.com/article/870c6fc300c2fab03ee4be70.html>

# **编译安装opencv:**

1.安装依赖包

sudo apt-get install cmake git libgtk2.0-dev pkg-config libavcodec-dev libavformat-dev libswscale-dev

其中一些依赖包安装时需要输入matlab安装路径，matlab安装及matlab-support安装可能报错解决方案见 《linux下安装matlab》

2.下载安装包

cd ~/<my\_working \_directory>

git clone

3.编译安装

cmake -D CMAKE\_BUILD\_TYPE=RELEASE -D CMAKE\_INSTALL\_PREFIX=/usr/local -D CUDA\_GENERATION=Auto ..

bebug:

1.make报错

**[ 0%] Building NVCC (Device) object modules/core/CMakeFiles/cuda\_compile.dir/src/cuda/./cuda\_compile\_generated\_matrix\_operations.cu.o**

**nvcc fatal : Unsupported gpu architecture 'compute\_11'**

**CMake Error at cuda\_compile\_generated\_matrix\_operations.cu.o.cmake:208 (message):**

**Error generating**

**/home/chang/OpenCV-2.4.2/release/modules/core/CMakeFiles/cuda\_compile.dir/src/cuda/./cuda\_compile\_generated\_matrix\_operations.cu.o**

**make[2]: \*\*\* [modules/core/CMakeFiles/cuda\_compile.dir/src/cuda/./cuda\_compile\_generated\_matrix\_operations.cu.o] Error 1**

**make[1]: \*\*\* [modules/core/CMakeFiles/opencv\_core.dir/all] Error 2**

**make: \*\*\* [all] Error 2**

----需要查询显卡架构，cmake时将CUDA\_GENERATION的值改成对应的Fermi/Kepler/Maxwell

-----或者CUDA\_GENERATION=Auto

ref:https://www.cnblogs.com/CarryPotMan/p/5377921.html

----然鹅，以上方法都行不通!!!

------算了，还是直接apt-get直接安装吧

sudo apt-get update

sudo apt-get install libcv-dev

ref:blog.csdn.net/tina\_ttl/article/details/52745807

自己编译opencv ref:

blog.csdn.net/xu (opencv安装完全脚本)

<https://github.com/jayrambhia/Install-OpenCV/tree/master/Ubuntu>(opencv安装完全脚本)

blog.csdn.net/honyniu/article/details/46390097

<https://docs.opencv.org/2.4/doc/tutorials/introduction/linux_install/linux_install.html>（官网）

# **安装ffmpeg**

本想按着opencv的套路，直接用apt-get直接安装，安装完之后，cmake程序报错：

g++ -L/usr/lib -L/opt/lib -pipe -Wall -O3 -ggdb -o release/DenseTrackStab .build\_release/DenseTrackStab.o -lopencv\_core -lopencv\_highgui -lopencv\_video -lopencv\_imgproc -lopencv\_calib3d -lopencv\_features2d -lopencv\_nonfree -lopencv\_ocl -lavformat -lavdevice -lavutil -lavcodec -lswscale

**/usr/bin/ld: cannot find -lavdevice**

找了一些解决方案，然而并解决不了

貌似在系统lib找不到avdevice.so动态链接库，apt-get安装的并找不到ffmpeg的安装文件夹，所以索性手动编译0.11.1版本的ffmpeg

ref:

blog.csdn.net/u012814360/article/details/39345431

blog.csdn.net/u012545304/article/details/23041799

1.安装yasm

sudo apt-get install yasm

2.安装各种依赖包（包可能不全，以后用到可以再装）：

sudo apt-get install build-essential libgtk2.0-dev libjpeg-dev libtiff4-dev libjasper-dev libopenexr-dev cmake python-dev python-numpy python-tk libtbb-dev libeigen2-dev yasm libfaac-dev libopencore-amrnb-dev libopencore-amrwb-dev libtheora-dev libvorbis-dev libxvidcore-dev libx264-dev libqt4-dev libqt4-opengl-dev sphinx-common texlive-latex-extra libv4l-dev libdc1394-22-dev libavcodec-dev libavformat-dev libswscale-dev

3.下载安装包、解压

<http://lear.inrialpes.fr/people/wang/download/ffmpeg-0.11.1.tar.bz2>

4.配置、安装ffmpeg

cd ffmpeg-0.11-1

4.1配置  
./configure --enable-gpl --enable-version3 --enable-nonfree --enable-postproc --enable-libfaac --enable-libopencore-amrnb --enable-libopencore-amrwb --enable-libtheora --enable-libvorbis --enable-libxvid --enable-x11grab --enable-swscale –enable-shared

安装路径设置为/usr/local/ffmpeg

./configure --enable-shared --prefix=/usr/local/ffmpeg

4.2编译安装

make && sudo make install

4.3添加环境变量

sudo gedit ~/.bashrc

文件中添加一行：export PATH=$PATH:/usr/local/ffmpeg/bin

然后

source ~/.bashrc

4.4测试

ffmpeg –help

报错：

ffmpeg: error while loading shared libraries: libavdevice.so.54: cannot open shared object file: No such file or directory

---查找库的路径：

sudo find / -name "libavdevice.so.54"

---在/etc/ld.so.conf添加添加一行：/usr/local/ffmpeg/lib

sudo gedit /etc/ld.so.conf

添加后文件内容为：

*include /etc/ld.so.conf.d/\*.conf*

*/usr/local/ffmpeg/lib*

然后：

sudo ldconfig

再次测试，成功～！

REF：blog.csdn.net/sahusoft/article/details/7388617

for TDD

# **安装caffe(matlab interface)**

caffe.berkeleyvision.org/install\_apt.html

>>>

sudo apt-get install libprotobuf-dev libleveldb-dev libsnappy-dev libopencv-dev libhdf5-serial-dev protobuf-compiler

sudo apt-get install --no-install-recommends libboost-all-dev

sudo apt-get install libgflags-dev libgoogle-glog-dev liblmdb-dev

sudo apt-get install libatlas-base-dev

cp Makefile.config.examples Makefile.config

gedit Makefile.config

-------------------------------

USE\_CUDNN := 1 //如果使用GPU

MATLAB\_DIR := /usr/local/MATLAB/R2016a //确保目录下 bin文件夹中有mex二进制文件

ANACONDA\_HOME := /home/chang/anaconda2/envs/caffe

PYTHON\_INCLUDE := $(ANACONDA\_HOME)/include $(ANACONDA\_HOME)/include/python2.7 //安装在anaconda环境下，添加路径3

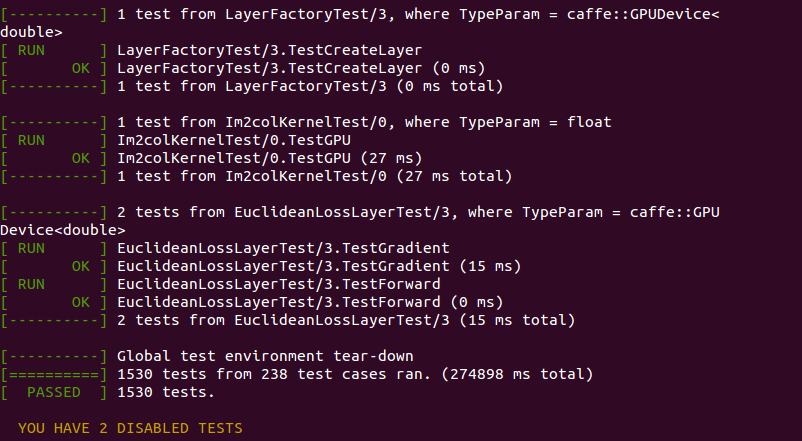
PYTHON\_LIB := $(ANACONDA\_HOME)/lib $(ANACONDA\_HOME)/lib/python2.7/site-packages/

USE\_OPENCV:=1

-------------------------------

make all

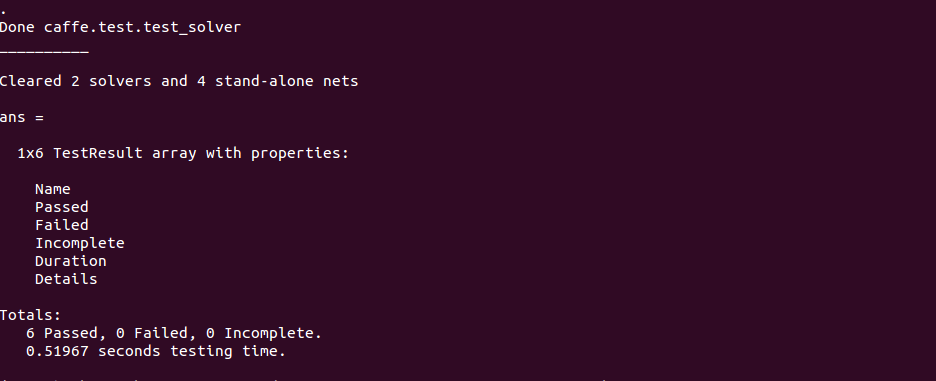
make runtest



编译matlab 接口：

make matcaffe

make mattest



跑一个demo:

运行caffe目录下代码：/caffe/matlab/demo/classification\_demo.m

运行之前先下载网络模型：

bvlc\_reference\_caffenet.caffemodel（url: http://dl.caffe.berkeleyvision.org/bvlc\_reference\_caffenet.caffemodel）

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

DEBUG:

1.make all时报错：ld：cannot find atlas

sudo apt-get install libatlas-base-dev

2.make runtest时报错：

.build\_release/tools/caffe: error while loading shared libraries: libprotobuf.so.15: cannot open shared object file: No such file or directory

make: \*\*\* [runtest] Error 127

参考blog.csdn.net/sahusoft/article/details/7388617中的两种方法：

1.查找libprotobuf.so.15 的路径，在/etc/ld.so.conf添加，ldconfig

2.在～/.bashrc 中添加LD\_LIBRARY\_PATH，source ~/.bashrc

for fisher vector

# **vlFeat**

官网instruction:[www.vlfeat.org/install-matlab.html](http://www.vlfeat.org/install-matlab.html)

中文版：blog.csdn.net/breeze5428/article/details/41982633

1.下载www.vlfeat.org/download.html，解压，

2.在matlab中：（VLFEATROOT为解压所在路径，VLFEATROOT/toolbox中有mex文件夹）

run('VLFEATROOT/toolbox/vl\_setup')

3.检测安装结果：

vl\_version verbose

4.永久配置：

在你matlab的任一个搜索路径里创建一个startup.m文件，添加一行：

run('VLFEATROOT/toolbox/vl\_setup')

>>>关于Matlab启动时自动运行某个脚本 (即某个.m文件)：

matlab启动时自动运行脚本需满足的条件：

* 脚本的名称必须为startup.m
* 脚本文件必须在matlab的可搜索到的目录下，亦即如下图（matlab菜单->set path）：

