

系统：ubuntu16.04 64bit

基本开发环境搭建

1. Linux 发行版

linux有很多发行版，本文强烈建议读者采用新版的 `Ubuntu 16.04 LTS` 的图形界面，与乐观的开源社区；另一方面，Ubuntu是NVIDIA官方支持的个人不建议使用Ubuntu其他版本，由于GCC编译器版本不兼容。
LTS下载地址：<http://www.ubuntu.org.cn/download/desktop>

下载Ubuntu桌面版

Ubuntu 16.04 LTS

Download the latest version of Ubuntu, for desktop PCs and laptops. long-term support – which means five years of free security and main updates, guaranteed.

[Ubuntu 16.04 LTS release notes](#)

Recommended system requirements:


- ✓ 2 GHz dual core processor or better
- ✓ 2 GB system memory
- ✓ 25 GB of free hard drive space
- ✓ Either a DVD drive or a USB port for the installer media
- ✓ Internet access is helpful

通过U盘安装好后，进行初始化环境设置。

2. Ubuntu初始环境设置

- 安装开发包 打开 终端 输入：

```
# 系统升级
>>> sudo apt update
>>> sudo apt upgrade
# 安装python基础开发包
>>> sudo apt install -y python-dev python-pip python-nose gcc g++ git
```

- 安装运算加速库 打开  输入：

```
>>> sudo apt install -y libopenblas-dev liblapack-dev libatlas-base-d
```

3. CUDA开发环境的搭建(CPU加速跳过

如果您的仅仅采用cpu加速，可跳过此步骤 - 下载CUDA8.0

下载地址：<https://developer.nvidia.com/cuda-downloads>

Select Target Platform ⓘ

Click on the green buttons that describe your target platform. Only supported

Operating System

Windows

Linux

Mac OSX

Architecture ⓘ

x86_64

ppc64le

Distribution

Fedora

OpenSUSE

RHEL

CentOS

Version

16.04

14.04

Installer Type ⓘ

runfile (local)

deb (local)

deb (network)

Download Installer for Linux Ubuntu 16.04 x86_64

The base installer is available for download below.

> Base Installer

Installation Instructions:

1. ``sudo dpkg -i cuda-repo-ubuntu1604-8-0-local_8.0.44-1_amd64.deb``
2. ``sudo apt-get update``
3. ``sudo apt-get install cuda``

The CUDA Toolkit contains Open-Source Software. The source code can be found in [Source Code](#). The checksums for the installer and patches can be found in [Installer Checksums](#). For further information, see the [Installation Guide for Linux](#) and the [CUDA Quickstart Guide](#).

之后打开 终端 输入：

```
>>> sudo dpkg -i cuda-repo-ubuntu1604-8-0-local_8.0.44-1_amd64.deb
>>> sudo apt update
>>> sudo apt install cuda
```

自动配置成功就好。

- 将CUDA路径添加至环境变量 在 终端 输入：

```
>>> sudo gedit /etc/bash.bashrc
```

在 `bash.bashrc` 文件中添加：

```
export CUDA_HOME=/usr/local/cuda-8.0
export PATH=/usr/local/cuda-8.0/bin${PATH:+:${PATH}}
export LD_LIBRARY_PATH=/usr/local/cuda-8.0/lib64${LD_LIBRARY_PATH:+:$
```

之后 `source gedit /etc/bash.bashrc` 即可 同样，在 终端 输入：

```
>>> sudo gedit ~/.bashrc
```

在 `.bashrc` 中添加如上相同内容（如果您使用的是 `zsh`，在

- 测试 在 终端 输入：

```
>>> nvcc -V
```

会得到相应的nvcc编译器相应的信息，那么CUDA配置成功

4. 加速库cuDNN（可选）

从官网下载需要注册账号申请，两三天批准。网盘搜索一般x64-v5.1-prod.zip。下载解压出来是名为cuda的文件夹，在CUDA的地方覆盖对应文件夹，在终端中输入：

```
>>> sudo cp include/cudnn.h /usr/local/cuda-8.0/include/  
>>> sudo cp lib64/* /usr/local/cuda-8.0/lib64/
```

Keras框架搭建

相关开发包安装

在 终端 中输入：

```
>>> sudo pip install -U --pre pip setuptools wheel  
>>> sudo pip install -U --pre numpy scipy matplotlib scikit-learn sci  
>>> sudo pip install -U --pre tensorflow-gpu  
# >>> sudo pip install -U --pre tensorflow  
>>> sudo pip install -U --pre keras
```

安装完毕后，输入 `python` ，然后输入：

```
>>> import tensorflow  
>>> import keras
```

无错输出即可

```
sudo apt-get install python-pip
```

以下安装，如果不用caffe的话（可选安装opencv

下载opencv，安装依赖项等

- 下载没什么说的，去[官网](#)找想用的版本就行，因为用2
- 安装依赖我是直接用的[高博](#)写的那些，当然也可以参

```
sudo apt-get install build-essential libgtk2.0-dev libjpeg-dev libtiff  
• 1
```

安装opencv

- 安装教程1：http://blog.csdn.net/qq_28115427/article/

安装完成后，python 能够import cv2 就表示安装

- 安装教程2：centos版本：<http://blog.csdn.net/CheNoi>
- 安装教程3：

安装步骤也是照[高博](#)文章做的，把下载的opencv解压到任

```
mkdir build
cd build
cmake ..
make
sudo make install
```

make步骤可能要花挺长时间，可以干点别的事情，当然别

配置opencv

[高博](#)文章里没有这个部分，可以参照[Jacky Liu](#)文章的第11/

- 首先，在 `/etc/ld.so.conf.d/` 路径下新建了 `opencv.conf` 下面的内容：

```
/usr/local/lib
```

然后终端敲命令

```
sudo ldconfig /etc/ld.so.conf
```

- 打开 `/etc/bash.bashrc` 文件，在最后加入下面内容：

```
PKG_CONFIG_PATH=$PKG_CONFIG_PATH:/usr/local/lib/pkgconfig
export PKG_CONFIG_PATH
```


最后进入python环境，import cv2没有报错的话就算安装成

```
cv2.VideoCapture(vid_path)
```

则还需要安装ffmpeg才能读取

安装caffe

安装依赖

```
sudo apt-get install libprotobuf-dev libleveldb-dev libsnappy-dev lib  
sudo apt-get install --no-install-recommends libboost-all-dev
```

然后在caffe的github下载caffe的包下来，解压后，

```
cp Makefile.config.example Makefile.config
```

然后修改里面的

```
USE_CUDNN := 1 是否使用cudnn加速
```

```
CPU_ONLY := 1是不是只使用cpu
```

```
OPENCV_VERSION := 3 opencv是2的还是3的版本的
```

```
WITH_PYTHON_LAYER := 1 使用python layer
```

然后就是编译了，

去到caffe的目录输入

```
make all -j12
```

```
make py
```

然后去到python的目录下，进入python环境输入import caffe，如果没有报错就成功了，否

安装MATLAB

安装资源：MATLAB 镜像 MATHWORKS_R2014A和Crack

注：安装过程需要图形界面，选择ssh -X 远程连接

挂载ISO文件

```
$ sudo mkdir /media/matlab
```

```
$ sudo mount -o loop MATHWORKS_R2014A.iso /media
```

执行安装过程，安装及破解过程参见：~/Crack/readme.txt

```
$ cd /media/matlab
```

```
$ sudo ./install
```

卸载ISO镜像

```
$ sudo umount /media/matlab
```

安装caffe的matlab接口，需要重新编译caffe

- 修改caffe的Makefile.config文件

USE_CUDNN := 1 是否使用cudnn加速

CPU_ONLY := 1是不是只使用cpu

OPENCV_VERSION := 3 opencv是2的还是3的版本的

WITH_PYTHON_LAYER := 1 使用python layer

MATLAB_DIR := /usr/local/MATLAB/R2015b 使用mat

并在大约第410行那一句话CXXFLAGS += -MMD -MP

CXXFLAGS += -MMD -MP

CXXFLAGS += -std=c++11

- 然后在caffe根目录下make clean,重新编译caffe

```
$ make all
```

```
$ make test
```

```
$ make runtest
```

- 最后编译caffe的python wrapper , 与编译Matlab的wrapper

```
$ make matcaffe
```

```
$ cd /caffe-master/python
```

```
$ sudo pip install -r requirements.txt
```

```
$ make pycaffe
```

- export导入编译好的python接口的路径

```
$ export PYTHONPATH=.../caffe-master/python:$PYTHONPATH
```