

Caffe설치(Ubuntu14.04)

Last update : 2016.11.23 15:20

0. 설명

본 자료는 linux에 caffe를 설치하기 위해 작성된 것입니다. 작성자는 리눅스를 처음 깔아본 초보이며, 다시 설치할 때 개인적으로 보려고 작성한 것이므로, 혹시 이 글을 보시고 궁금한 점이 생기셔도 답변을 드릴 능력이 없음을 밝힙니다.

틀린 부분이 있으시면 메일(heejoo.yoon@lge.com)로 보내주시거나, 답글을 주시면 확인이 가능할 경우, 확인 후 수정하겠습니다.

작성자의 설치 환경

ubuntu 14.04 LTS, GeForce GTX 980 Ti, Intel Core i7-6700 CPU @ 3.4GHz x 8

ubuntu 14.04 LTS, GeForce GTX TITAN Black, Intel Core i7-4790 CPU @ 3.6GHz x 8

1. Cuda install

1.1) build-essential install

```
$ sudo apt-get install build-essential
```

아래와 같은 메시지가 나와 실패할 경우, repository를 추가하고 update한 뒤, 다시 설치한다.

```
Reading package lists... Done
Building dependency tree
Reading state information...
Done E: Unable to locate package build-essential
```

```
$ sudo add-apt-repository "deb http://archive.ubuntu.com/ubuntu $(lsb_release -sc) main universe"
$ sudo apt-get update
$ sudo apt-get install build-essential
```

1.2) linux header install

```
$ sudo apt-get install linux-headers-$(uname -r)
```

1.3) CUDA Toolkit download & installation

1.3.1) Download using webpage

Link : <https://developer.nvidia.com/cuda-downloads>

위 경로에서

CUDA8.0 > Linux > x86_64 > Ubuntu > 14.04 > runfile (local)

의 순으로 다운로드 받는다.

1.3.2) Installation

여기서 주의할 점은, gpu를 연산에서 full로 사용하기 위해서는 shell prompt에서 gui를 끄고 설치해야 한다.

<Ctrl> + <Alt> + <F1>을 눌러 shell prompt를 실행한다. (<F1> 에서 <F6>까지 모두 사용가능하며, <Ctrl> + <Alt> + <F7>은 gui환경으로 돌아갈 수 있다.)

```
$ sudo service lightdm stop
```

아래의 방법 등을 이용해 기존에 설치되어 있는 nvidia driver를 지우고 설치하는 것이 좋다. (지우지 않고 설치할 경우, ubuntu의 무한로그인을 경험하게 될 수도 있다.)

```
$ sudo apt-get --purge remove nvidia-*
```

다운로드 경로에 가서 설치한다.

```
$ cd Download
$ ./cuda_8.0.44_Linux.run
```

약관 창이 나타나면 Accept를 입력하고,

Graphic Driver 설치 등 대부분 y(es)라고 대답하고, 반드시 OpenGL libraries 설치를 n(o)라고 해야 한다.

프로그램의 연산에서 GPU를 full로 사용하기 위해서는 반드시 OpenGL libraries를 Off해줘야만 한다. (by 박경업 [kyoungup.park](#))

설치가 끝나면 재부팅한다. (여기서 lightdm을 따로 start 해주지 않아도 재부팅 후, 다시 start된다)

```
$ sudo reboot
```

다음과 같이 driver 설치정보를 확인할 수 있다.

```
$ nvidia-smi
```

```
heejooyoon@HJYoon: ~/opencv-3.1.0
heejooyoon@HJYoon:~/opencv-3.1.0$ nvidia-smi
Wed Nov 23 09:06:34 2016

+-----+
| NVIDIA-SMI 367.48                  Driver Version: 367.48          |
+-----+-----+-----+-----+-----+-----+
| GPU   Name           Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf    Pwr:Usage/Cap|      Memory-Usage | GPU-Util  Compute M. |
|=====+-----+-----+=====+-----+-----+=====+
|    0  GeForce GTX 980 Ti  Off   | 0000:01:00.0    On   |           N/A       |
|  0%   38C    P8     18W / 300W |  108MiB /  6074MiB |      11%    Default  |
+-----+-----+-----+-----+-----+-----+

+-----+
| Processes:                         GPU Memory Usage |
| GPU       PID    Type    Process name                     |              |
|=====+-----+-----+=====+
|    0      1178    G       /usr/bin/X                       |    106MiB    |
+-----+-----+-----+-----+-----+-----+

heejooyoon@HJYoon:~/opencv-3.1.0$
```

** GeForce GTX 1080 GPU 의 경우, 추가적으로 GPU driver의 설치가 필요하다. (by 마춘페이 [chunfei.ma](#))

```
$ sudo apt-get install nvidia-457
```

** 기타 nvidia driver가 설치되지 않아, 추가적으로 GPU driver의 설치가 필요할 경우 아래의 방법을 사용해도 된다.

```
$ sudo apt-get install nvidia-current
```

** 위에서도 언급했듯이 nvidia driver를 설치할 때, 인텔CPU의 그래픽카드를 사용하기 위해 OpenGL 옵션을 off해야 한다. (by 박경업 [kyoungup.park](#))

1.4) 환경 설정

.bash의 아래부분에 다음과 같이 환경 설정을 추가한다. (vi 사용법은 vi cheat sheet를 참고)

```
$ vi ~/.bashrc

$ echo 'export PATH=/usr/local/cuda/bin:$PATH' >> ~/.bashrc

$ echo 'export LD_LIBRARY_PATH=$LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/cuda/lib64:/usr/local/lib' >> ~/.bashrc $ source ~/.bashrc
```

2. 의존성 있는 라이브러리 설치[3]

다 설치할 필요는 없을지도 모르지만, 어느것이 필요한지 몰라 검색 후 참고자료에 따라 아래와 같이 다 설치하였다.

```
$ sudo apt-get install -y libprotobuf-dev libleveldb-dev libsnappy-dev libopencv-dev libboost-all-dev libhdf5-serial-dev protobuf-compiler gfortran libjpeg62
libfreeimage-dev libatlas-base-dev git python-dev python-pip libbz2-dev libxml2-dev libxslt1-dev libffi-dev libssl-dev liblmbd-dev

$ sudo apt-get install libgoogle-glog-dev libgflags-dev libsoncpp-dev
```

3. cuDNN install

Link : <https://developer.nvidia.com/rdp/cudnn-download>

위 경로에서 cudnn-8.0-linux-x64-v5.1-ga.tgz 를 다운로드 받는다.

```
$ tar -xv cudnn-8.0-linux-x64-v5.1-ga.tgz  
$ cd cuda $ cp include/* /usr/local/cuda/include  
$ cp lib64/* /usr/local/cuda/lib64
```

4. OpenCv [6]

4.1) Download

```
$ sudo apt-get install cmake git libgtk2.0-dev pkg-config libavcodec-dev libavformat-dev libswscale-dev  
$ git clone https://github.com/opencv/opencv.git  
$ mkdir release  
$ cd release
```

4.2) Building using CMake

```
$ cmake -D CMAKE_BUILD_TYPE=RELEASE -D CMAKE_INSTALL_PREFIX=/usr/local ..
```

Error가 발생 할 경우 다음과 같이 cmake 명령어를 살짝 바꿔 다시 컴파일 할 수 있다. (by [마춘페이 chunfei.ma](#))

```
$ cmake -D CMAKE_BUILD_TYPE=RELEASE -D CMAKE_INSTALL_PREFIX=/usr/local -D CUDA_GENERATION=kepler ..
```

4.3) Compile

```
$ sudo make
```

cuda 8.0에서 지원하지 않는 opencv 버전이 있기 때문에 make할 때, 아래와 같은 Error가 생길 수 있다. (by [마춘페이 chunfei.ma](#))

error가 발생하지 않을 경우 4.4)으로 바로 넘어가면 된다.

"cudalegacy not compile --nppiGraphcut missing"

```
#include "precomp.hpp"  
  
#if !defined (HAVE_CUDA) || defined  
(CUDA_DISABLE)
```

위 코드를 다음과 같이 수정하여 다시 make하면 된다. [7]

```
#include "precomp.hpp"  
  
#if !defined (HAVE_CUDA) || defined (CUDA_DISABLE) || (CUDART_VERSION >=  
8000)
```

4.4) Installation

```
$ sudo make install
```

5. Caffe Download & Installation [1]

5.1) caffe 소스를 다운, requirements를 통해 python 패키지를 다운

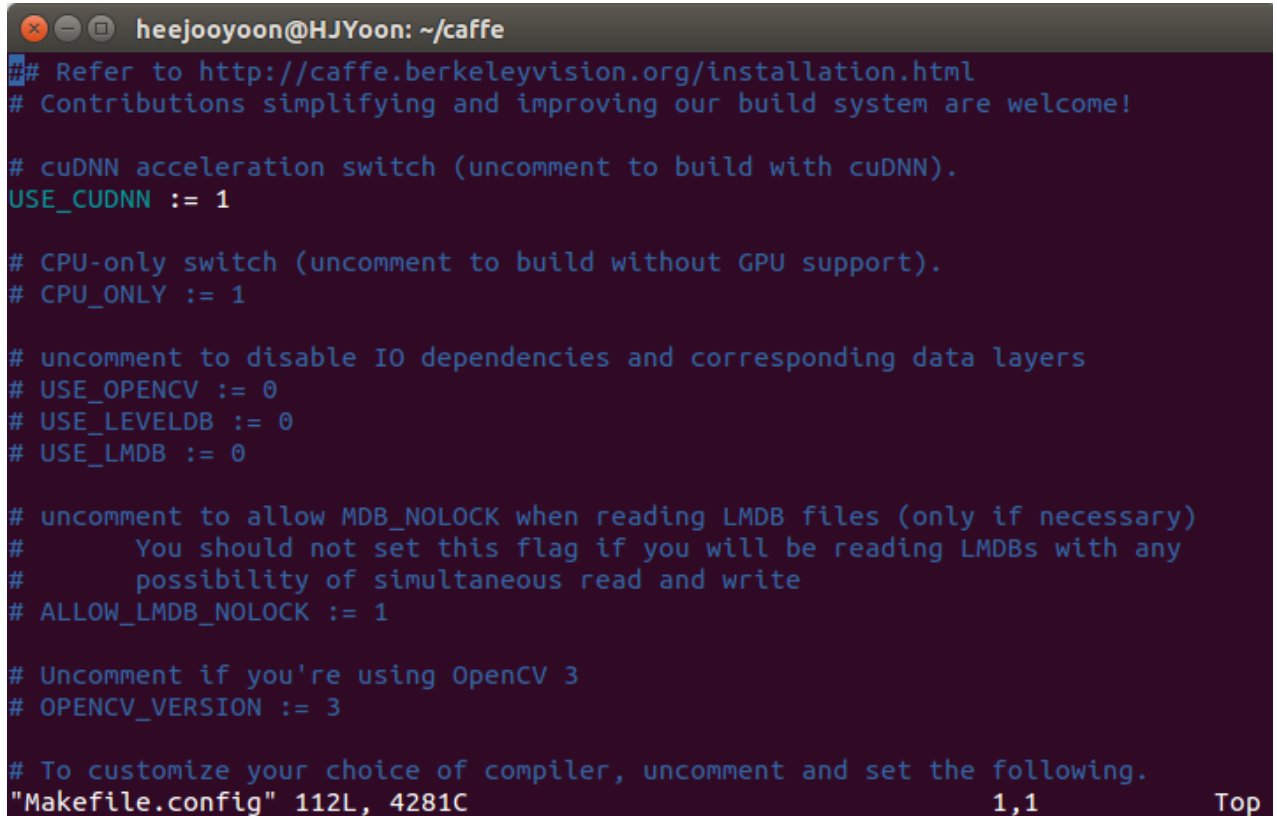
```
$ git clone git://github.com/BVLC/caffe.git  
$ cd caffe  
$ cat python/requirements.txt | xargs -L 1 sudo pip install
```

5.2) Makefile.config 수정[4]

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

Makefile.config를 예제파일에서 복사 한 뒤, 아래 옵션 부분을 찾아 주석처리된 부분의 주석을 제거한다.(vi 등을 이용해서 해당 줄의 #을 지운다.)

```
## USE_CUDNN := 1
```



```
heejooyoon@HJYoon: ~/caffe
# Refer to http://caffe.berkeleyvision.org/installation.html
# Contributions simplifying and improving our build system are welcome!

# cuDNN acceleration switch (uncomment to build with cuDNN).
USE_CUDNN := 1

# CPU-only switch (uncomment to build without GPU support).
# CPU_ONLY := 1

# uncomment to disable IO dependencies and corresponding data layers
# USE_OPENCV := 0
# USE_LEVELDB := 0
# USE_LMDB := 0

# uncomment to allow MDB_NOLOCK when reading LMDB files (only if necessary)
#     You should not set this flag if you will be reading LMDBs with any
#     possibility of simultaneous read and write
# ALLOW_LMDB_NOLOCK := 1

# Uncomment if you're using OpenCV 3
# OPENCV_VERSION := 3

# To customize your choice of compiler, uncomment and set the following.
"Makefile.config" 112L, 4281C 1,1 Top
```

Reference : <https://github.com/BVLC/caffe/pull/1667>

5.3) Compile [4,5]

```
$ make clean
$ make all
$ make test
$ make runtest
```

OK가 나오면 설치 끝.

5.4) 환경 설정 (.bashrc)

caffe의 설치 경로에 주의해서 아래와 같이 입력한다. 아래와 같이 쓰면 vi 사용없이 .bashrc에 추가된다.

```
$ echo 'PATH=/usr/local/cuda/bin:$PATH' >> ~/.bashrc
$ echo 'LD_LIBRARY_PATH=/usr/local/lib:$LD_LIBRARY_PATH' >> ~/.bashrc
$ echo 'LD_LIBRARY_PATH=/home/heejooyoon/caffe/build/lib:$LD_LIBRARY_PATH' >> ~/.bashrc
$ echo 'LD_LIBRARY_PATH=/usr/local/cuda/lib64:$LD_LIBRARY_PATH' >> ~/.bashrc
$ echo 'export PATH' >> ~/.bashrc $ echo 'export LD_LIBRARY_PATH' >> ~/.bashrc $ source ~/.bashrc
```

vi로 직접입력해도 된다. 아래 6줄 처럼 경로 추가가 되지만 하면 된다.

```
heejooyoon@HJYoon: ~/Downloads

if [ -f ~/.bash_aliases ]; then
    . ~/.bash_aliases
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
    if [ -f /usr/share/bash-completion/bash_completion ]; then
        . /usr/share/bash-completion/bash_completion
    elif [ -f /etc/bash_completion ]; then
        . /etc/bash_completion
    fi
fi

PATH=/usr/local/cuda/bin:$PATH
LD_LIBRARY_PATH=/usr/local/lib:$LD_LIBRARY_PATH
LD_LIBRARY_PATH=/home/heejooyoon/caffe/build/lib:$LD_LIBRARY_PATH
LD_LIBRARY_PATH=/usr/local/cuda/lib64:$LD_LIBRARY_PATH

export PATH
export LD_LIBRARY_PATH
```

122,1 Bot

Reference

- [1] <http://yujuwon.tistory.com/entry/DeepLearning-%EC%9A%B0%EB%B6%84%ED%88%AC-1404%EC%97%90-Caffe-%EC%84%A4%EC%B9%98-%ED%95%98%EA%B8%B0>
- [2] <http://haanjack.github.io/cuda/2016-02-29-cuda-linux/>
- [3] <http://bigblue.tistory.com/entry/%EC%9A%B0%EB%B6%84%ED%88%AC-%EB%A6%AC%EB%88%85%EC%8A%A4-%EC%84%A4%EC%B9%98-%EB%B0%8F-Caffe-%EC%84%A4%EC%B9%98>
- [4] <http://www.whydsp.org/316>
- [5] <http://jangji.tistory.com/197>
- [6] http://docs.opencv.org/2.4/doc/tutorials/introduction/linux_install/linux_install.html#linux-installation
- [7] <http://answers.opencv.org/question/95148/cudalegacy-not-compile-nppigraphcut-missing/>

Special Thanks to

박경업 kyoungup.park

마춘페이 chunfei.ma