

Reference Tutorial for: **ML SSD PASCAL-Xilinx Git**

Caffe CPU Only Mode Installation

This is a guide on how to install Caffe for Ubuntu 16.04 and above, without GPU support (No CUDA required).

Prerequisites:

OpenCV:

```
sudo apt-get install libopencv-dev python-opencv
```

OpenBLAS or Atlas

OpenBLAS

```
sudo apt-get install libopenblas-dev
```

Atlas

```
sudo apt-get install libatlas-base-dev
```

Boost

```
sudo apt-get install libboost-all-dev
```

Protobuf (USING PIP)

```
sudo pip install protobuf
```

If you don't have pip yet, install it using the following commands:

```
sudo apt-get install python-pip python-dev build-essential
```

```
sudo pip install --upgrade pip
```

General Dependencies

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

```
sudo apt-get install python-dev
```

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

```
sudo apt-get install python-matplotlib python-numpy python-pil python-scipy
```

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

```
sudo apt-get install python-skimage
```

Getting Caffe

```
git clone https://github.com/BVLC/caffe
```

For weiliu89 fork of caffe:

```
git clone https://github.com/weiliu89/caffe.git
```

You will now find the caffe folder in the directory where you cloned this git repository. (Possibly Home directory)

We have to make a copy of Makefile.config.example, which we generally name as, Makefile.config to which we can make changes based on our system settings.

```
cd caffe
cp Makefile.config.example Makefile.config
```

Making Changes in Makefile.config

Make sure you are still in the caffe directory, then use this command to open Makefile.config

```
gedit Makefile.config
```

Note: The following line numbers may vary.

- On Line 8, uncomment CPU_ONLY :=1

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

# CPU-only switch (uncomment to build without GPU support).
CPU_ONLY := 1
```
- On Line 21, only uncomment OPENCV_VERSION :=3 if you're using OpenCV 3 or above. If you aren't sure, try this in another terminal

```
$python
>>> import cv2
>>> cv2.__version__
'3.0.0'
```

Leave it commented if you get version below 3.0.0 like this

```
Python 2.7.12 (default, Nov 12 2018, 14:36:49)
[GCC 5.4.0 20160609] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import cv2
>>> cv2.__version__
'2.4.9.1'
>>>
```

- On line 25, uncomment CUSTOM_CXX := g++

- On line 50, set BLAS:= open if you've installed OpenBLAS, or let it be the default BLAS := atlas if you've installed Atlas.
- On line 94, change
INCLUDE_DIRS := \$(PYTHON_INCLUDE) /usr/local/include
to
INCLUDE_DIRS := \$(PYTHON_INCLUDE) /usr/local/include /usr/include/hdf5/serial
- On line 95, change
LIBRARY_DIRS := \$(PYTHON_LIB) /usr/local/lib
to
LIBRARY_DIRS:=\$(PYTHON_LIB) /usr/local/lib /usr/lib /usr/lib/x86_64-linux-gnu /usr/lib/x86_64-linux-gnu/hdf5/serial

Whatever **else** you find you need goes here.
INCLUDE_DIRS := \$(PYTHON_INCLUDE) /usr/local/include /usr/include/hdf5/serial
LIBRARY_DIRS := \$(PYTHON_LIB) /usr/local/lib /usr/lib /usr/lib/x86_64-linux-gnu /usr/lib/x86_64-linux-gnu/hdf5/serial

Save and exit.

This should resolve hdf5 errors when running make.

Now Edit the Makefile with command

```
gedit Makefile
```

Add at Line 409 the following

```
-D_FORCE_INLINES
```

```
# Complete build flags.
COMMON_FLAGS += $(foreach includedir,$(INCLUDE_DIRS),-isystem $(includedir))
CXXFLAGS += -pthread -fPIC $(COMMON_FLAGS) $(WARNINGS)
NVCCFLAGS += -D_FORCE_INLINES -ccbin=$(CXX) -Xcompiler -fPIC $(COMMON_FLAGS)
# you may make an older one that is too liberal with the includedir
```

Now within the caffe directory, run the following one after the other

```
make clean
```

```
make -j8
```

```
make test
```

```
make runtest
```

Once all four run without errors, while in the caffe directory, type

```
make py
```

This will build a python wrapper. You will also find a python folder within the caffe folder now.

To use caffe within python, export its path as

```
export PYTHONPATH=~ /home/_username_/caffe/python
```

Replace `_username_` with your username in the system.

Once you've done this, run then python terminal and import caffe

```
abhidan@ubuntu:~/caffe$ export PYTHONPATH=/home/abhidan/caffe/python
abhidan@ubuntu:~/caffe$ $PYTHONPATH
bash: /home/abhidan/caffe/python: Is a directory
abhidan@ubuntu:~/caffe$ python
Python 2.7.12 (default, Nov 12 2018, 14:36:49)
[GCC 5.4.0 20160609] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import caffe
>>> █
```

Now the caffe has been installed for CPU only implementation. This can be done inside a virtual machine.

CAFFE INSTALLATION with Nvidia GPU

This is a guide on how to install Caffe for Ubuntu 16.04 and above, with GPU support (CUDA required)

Prerequisites:

OpenCV:

```
sudo apt-get install libopencv-dev python-opencv
```

OpenBLAS or Atlas

OpenBLAS

```
sudo apt-get install libopenblas-dev
```

Atlas

```
sudo apt-get install libatlas-base-dev
```

Boost

```
sudo apt-get install libboost-all-dev
```

Protobuf (USING PIP)

```
sudo pip install protobuf
```

If you don't have pip yet, install it using the following commands:

```
sudo apt-get install python-pip python-dev build-essential
```

```
sudo pip install --upgrade pip
```

General Dependencies

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

```
sudo apt-get install python-dev
```

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

```
sudo apt-get install python-matplotlib python-numpy python-pil python-scipy
```

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

```
sudo apt-get install python-skimage
```

Install CUDA and CuDNN:

This tutorial was verified with the following configuration:

- Ubuntu 16.04

For any Queries, please visit: www.logictronix.com or mail us at: info@logictronix.com

- NVIDIA GTX 960m
- NVIDIA 390 Graphics Driver

[CUDA v8.0](#) (follow Nvidia instructions and install the runfile along with the patch).

CuDNN v7.0.5 using the “cuDNN v7.0.5 Library for Linux” selection for CUDA 8.0. The following steps were used to install CuDNN:

1. Go to the directory where CuDNN is downloaded.

```
tar -xzvf cudnn-8.0-linux-x64-v7.tgz
```

2. Now you have to copy the two extracted folders to where CUDA was installed, which is most likely at /usr/local/cuda
3. Enter following commands:

```
sudo cp lib64/* /usr/local/cuda/lib64
sudo cp include/* /usr/local/cuda/include
```

4. Now install NCCL v1.2.3 - The following steps were used to install:

- a. Download NCCL 1.2.3: <https://github.com/NVIDIA/nccl/releases>

```
tar -xvf nccl-1.2.3-1-cuda8.0.tar.gz
cd nccl-1.2.3-1-cuda8.0
sudo make install -j
sudo ldconfig /usr/local/cuda/lib64
```

NCCL is not required for DNN v3 and above.

That's it. You have just successfully installed CUDA and CuDNN.

NOTE: If issues are encountered where the graphics driver needs to be installed, please use the following instructions to install:

- a. First remove the other installations via the following:

```
sudo apt-get purge nvidia-cuda*
sudo apt-get purge nvidia-*
```

- b. Enter a terminal session using Ctrl+Alt+F2
- c. Stop lightdm:

```
sudo service lightdm stop
```

- d. Create a file at /etc/modprobe.d/blacklist-nouveau.conf with the following contents:

```
blacklist nouveau
options nouveau modeset=0
```

- e. Then do:

```
sudo update-initramfs -u
```

- f. Add the graphics driver PPA:

```
sudo add-apt-repository ppa:graphics-drivers
```

```
sudo apt-get update
```

- g. Now install and activate the latest drivers:

```
sudo apt-get install nvidia-390
```

Let's move on and install Caffe NVIDIA GPU. This process won't be much difference with the installation in CPU_ONLY mode

Getting Caffe

```
git clone https://github.com/BVLC/caffe
```

For weiliu89 fork of caffe:

```
git clone https://github.com/weiliu89/caffe.git
```

You will now find the caffe folder in the directory where you cloned this git repository. (Possibly Home directory)

We have to make a copy of Makefile.config.example, which we generally name as, Makefile.config to which we can make changes based on our system settings.

```
cd caffe
```

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

Making Changes In Makefile.config

Make sure you are still in the caffe directory, then use this command to open Makefile.config

```
gedit Makefile.config
```

Note: The following line numbers may vary.

- On Line 5, uncomment `USE_CUDNN := 1`

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

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

- On Line 21, only uncomment OPENCV_VERSION :=3 if you're using OpenCV 3 or above. If you aren't sure, try this in another terminal

```
$python
>>> import cv2
>>> cv2.__version__
'3.0.0'
```

Leave it commented if you get version below 3.0.0 like this

```
Python 2.7.12 (default, Nov 12 2018, 14:36:49)
[GCC 5.4.0 20160609] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import cv2
>>> cv2.__version__
'2.4.9.1'
>>>
```

- On line 25, uncomment CUSTOM_CXX := g++
- Ensure line 28 is to set CUDA_DIR := /usr/local/cuda (or wherever your CUDA install is located).

```
# CUDA directory contains bin/ and lib/ directories that we need.
CUDA_DIR := /usr/local/cuda
# On Ubuntu 14.04, if cuda tools are installed via
# "sudo apt-get install nvidia-cuda-toolkit" then use this instead
# CUDA_DIR := /usr
```

- On line 35 there is a definition for CUDA_ARCH. Delete the lines that end in sm_20 and sm_21 (these are incompatible with the version of cuda being used).

```
# CUDA architecture setting: going with all of them.
# For CUDA < 6.0, comment the lines after *_35 for compatibility.
CUDA_ARCH := -gencode arch=compute_20,code=sm_20 \
             -gencode arch=compute_20,code=sm_21 \
             -gencode arch=compute_30,code=sm_30 \
             -gencode arch=compute_35,code=sm_35 \
             -gencode arch=compute_50,code=sm_50 \
             -gencode arch=compute_52,code=sm_52 \
             -gencode arch=compute_61,code=sm_61
```

- On line 50, set BLAS:= open if you've installed OpenBLAS, or let it be the default BLAS := atlas if you've installed Atlas.

- On line 94, change

```
INCLUDE_DIRS := $(PYTHON_INCLUDE) /usr/local/include
to
INCLUDE_DIRS := $(PYTHON_INCLUDE) /usr/local/include /usr/include/hdf5/serial
```

- On line 95, change

```
LIBRARY_DIRS := $(PYTHON_LIB) /usr/local/lib
to
```

```
LIBRARY_DIRS:=$(PYTHON_LIB) /usr/local/lib /usr/lib /usr/lib/x86_64-linux-gnu /usr/lib/x86_64-  
linux-gnu/hdf5/serial
```

```
# Whatever else you find you need goes here.  
INCLUDE_DIRS := $(PYTHON_INCLUDE) /usr/local/include /usr/include/hdf5/serial  
LIBRARY_DIRS := $(PYTHON_LIB) /usr/local/lib /usr/lib /usr/lib/x86_64-linux-gnu /usr/lib/x86_64-linux-gnu/hdf5/serial
```

Save and exit.

This should resolve hdf5 errors when running make.

Now Edit the Makefile with command

```
gedit Makefile
```

Add at Line 409 the following

```
-D_FORCE_INLINES  
# Complete build flags.  
COMMON_FLAGS += $(foreach includedir,$(INCLUDE_DIRS),-isystem $(includedir))  
CXXFLAGS += -pthread -fPIC $(COMMON_FLAGS) $(WARNINGS)  
NVCCFLAGS += -D_FORCE_INLINES -ccbin=$(CXX) -Xcompiler -fPIC $(COMMON_FLAGS)  
# may may make an older one that is too liberal with uninitialized
```

Now within the caffe directory, run the following one after the other

```
make clean
```

```
make -j8
```

```
make test
```

```
make runtest
```

Once all four run without errors, while in the caffe directory, type

```
make py
```

This will build a python wrapper. You will also find a python folder within the caffe folder now.

To use caffe within python, export its path as

```
export PYTHONPATH=~/.home/_username_/caffe/python
```

Replace `_username_` with your username in the system.

Once you've done this, run then python terminal and import caffe

```
abhidan@ubuntu:~/caffe$ export PYTHONPATH=/home/abhidan/caffe/python
abhidan@ubuntu:~/caffe$ $PYTHONPATH
bash: /home/abhidan/caffe/python: Is a directory
abhidan@ubuntu:~/caffe$ python
Python 2.7.12 (default, Nov 12 2018, 14:36:49)
[GCC 5.4.0 20160609] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import caffe
>>>
```

Now the caffe has been installed for CUDA. This cannot be done inside a virtual machine.

References:

- i. <https://gist.github.com/nikitametha/c54e1abecff7ab53896270509da80215>
- ii. https://github.com/yahoo/open_nsfw/issues/13
- iii. <https://chunml.github.io/ChunML.github.io/project/Installing-Caffe-Ubuntu/>
- iv. https://blog.csdn.net/weixin_41399111/article/details/83025208
- v. https://github.com/yahoo/open_nsfw/issues/13

Thank You

-----www.LogicTronix.com-----

For any queries or issues mail us at ***info@logictronix.com***