

# Caffe

---

Deep learning framework by the [BVL](#)

Created by

[Yangqing Jia](#)

Lead Developer

[Evan Shelhamer](#)



## Installation

---

Prior to installing, have a glance through this guide and take note of the details for your platform. We install and run Caffe on Ubuntu 14.04 and 12.04, OS X 10.10 / 10.9 / 10.8, and AWS. The official Makefile and `Makefile.config` build are complemented by an automatic CMake build from the community.

- [Prerequisites](#)
- [Compilation](#)
- [Hardware](#)
- Platforms: [Ubuntu guide](#), [OS X guide](#), and [RHEL / CentOS / Fedora guide](#)

When updating Caffe, it's best to `make clean` before re-compiling.

## Prerequisites

---

Caffe has several dependencies:

- [CUDA](#) is required for GPU mode.
  - library version 7.0 and the latest driver version are recommended, but 6.\* is fine too
  - 5.5, and 5.0 are compatible but considered legacy
- [BLAS](#) via ATLAS, MKL, or OpenBLAS.

- **Boost**  $\geq 1.55$
- `protobuf`, `glog`, `gflags`, `hdf5`

Optional dependencies:

- **OpenCV**  $\geq 2.4$  including 3.0
- IO libraries: `lmdb`, `leveldb` (note: `leveldb` requires `snappy`)
- `cuDNN` for GPU acceleration (v3)

Pycaffe and Matcaffe interfaces have their own natural needs.

- For Python Caffe: Python 2.7 or Python 3.3+, `numpy` ( $\geq 1.7$ ), `boost-python`
- For MATLAB Caffe: MATLAB with the `mex` compiler.

**cuDNN Caffe:** for fastest operation Caffe is accelerated by drop-in integration of **NVIDIA cuDNN**. To speed up your Caffe models, install cuDNN then uncomment the `USE_CUDNN := 1` flag in `Makefile.config` when installing Caffe. Acceleration is automatic. The current version is cuDNN v3; older versions are supported in older Caffe.

**CPU-only Caffe:** for cold-brewed CPU-only Caffe uncomment the `CPU_ONLY := 1` flag in `Makefile.config` to configure and build Caffe without CUDA. This is helpful for cloud or cluster deployment.

## CUDA and BLAS

---

Caffe requires the CUDA `nvcc` compiler to compile its GPU code and CUDA driver for GPU operation. To install CUDA, go to the **NVIDIA CUDA website** and follow installation instructions there. Install the library and the latest standalone driver separately; the driver bundled with the library is usually out-of-date. **Warning!** The 331.\* CUDA driver series has a critical performance issue: do not use it.

For best performance, Caffe can be accelerated by **NVIDIA cuDNN**. Register for free at the cuDNN site, install it, then continue with these installation instructions. To compile with cuDNN set the `USE_CUDNN := 1` flag set in your `Makefile.config`.

Caffe requires BLAS as the backend of its matrix and vector computations. There are several implementations of this library. The choice is yours:

- **ATLAS**: free, open source, and so the default for Caffe.
- **Intel MKL**: commercial and optimized for Intel CPUs, with a free trial and **student** licenses.
  1. Install MKL.
  2. Set up MKL environment (Details: **Linux**, **OS X**). Example: source `/opt/intel/mkl/bin/mklvars.sh intel64`
  3. Set `BLAS := mkl` in `Makefile.config`
- **OpenBLAS**: free and open source; this optimized and parallel BLAS could require more effort to install, although it might offer a speedup.
  1. Install OpenBLAS
  2. Set `BLAS := open` in `Makefile.config`

## Python and/or MATLAB Caffe (optional)

---

### Python

The main requirements are `numpy` and `boost.python` (provided by `boost`). `pandas` is useful too and needed for some examples.

You can install the dependencies with

```
for req in $(cat requirements.txt); do pip install $req; done
```

but we suggest first installing the **Anaconda** Python distribution, which provides most of the necessary packages, as well as the `hdf5` library dependency.

To import the `caffe` Python module after completing the installation, add the module directory to your `$PYTHONPATH` by `export`

`PYTHONPATH=/path/to/caffe/python:$PYTHONPATH` or the like. You should not import the module in the `caffe/python/caffe` directory!

Caffe's Python interface works with Python 2.7. Python 3.3+ should work out of the box without protobuf support. For protobuf support please install protobuf 3.0 alpha (<https://developers.google.com/protocol-buffers/>). Earlier Pythons are your own adventure.

### MATLAB

Install MATLAB, and make sure that its `mex` is in your `$PATH`.

Caffe's MATLAB interface works with versions 2015a, 2014a/b, 2013a/b, and 2012b.

## Windows

There is an unofficial Windows port of Caffe at [niuzhiheng/caffe:windows](https://github.com/niuzhiheng/caffe-windows). Thanks [@niuzhiheng](https://twitter.com/niuzhiheng)!

## Compilation

---

Caffe can be compiled with either Make or CMake. Make is officially supported while CMake is supported by the community.

### Compilation with Make

---

Configure the build by copying and modifying the example `Makefile.config` for your setup. The defaults should work, but uncomment the relevant lines if using Anaconda Python.

```
cp Makefile.config.example Makefile.config
# Adjust Makefile.config (for example, if using Anaconda Python, or if cuDNN is desired)
make all
make test
make runtest
```

- For CPU & GPU accelerated Caffe, no changes are needed.
- For cuDNN acceleration using NVIDIA's proprietary cuDNN software, uncomment the `USE_CUDNN := 1` switch in `Makefile.config`. cuDNN is sometimes but not always faster than Caffe's GPU acceleration.
- For CPU-only Caffe, uncomment `CPU_ONLY := 1` in `Makefile.config`.

To compile the Python and MATLAB wrappers do `make pycaffe` and `make matcaffe` respectively. Be sure to set your MATLAB and Python paths in `Makefile.config` first!

**Distribution:** run `make distribute` to create a `distribute` directory with all the Caffe headers, compiled libraries, binaries, etc. needed for distribution to other machines.

**Speed:** for a faster build, compile in parallel by doing `make all -j8` where 8 is

the number of parallel threads for compilation (a good choice for the number of threads is the number of cores in your machine).

Now that you have installed Caffe, check out the [MNIST tutorial](#) and the [reference ImageNet model tutorial](#).

## Compilation with CMake

---

In lieu of manually editing `Makefile.config` to configure the build, Caffe offers an unofficial CMake build thanks to @Nerei, @akosiorek, and other members of the community. It requires CMake version  $\geq 2.8.7$ . The basic steps are as follows:

```
mkdir build
cd build
cmake ..
make all
make install
make runtest
```

See [PR #1667](#) for options and details.

## Hardware

---

**Laboratory Tested Hardware:** Berkeley Vision runs Caffe with K40s, K20s, and Titans including models at ImageNet/ILSVRC scale. We also run on GTX series cards (980s and 770s) and GPU-equipped MacBook Pros. We have not encountered any trouble in-house with devices with CUDA capability  $\geq 3.0$ . All reported hardware issues thus-far have been due to GPU configuration, overheating, and the like.

**CUDA compute capability:** devices with compute capability  $\leq 2.0$  may have to reduce CUDA thread numbers and batch sizes due to hardware constraints. Your mileage may vary.

Once installed, check your times against our [reference performance numbers](#) to make sure everything is configured properly.

Ask hardware questions on the [caffe-users group](#).

