Installation in Linux

These steps have been tested for Ubuntu 10.04 but should work with other distros as well.

Required Packages

- GCC 4.4.x or later
- CMake 2.8.7 or higher
- Git
- GTK+2.x or higher, including headers (libgtk2.0-dev)
- pkg-config
- Python 2.6 or later and Numpy 1.5 or later with developer packages (python-dev, python-numpy)
- ffmpeg or libav development packages: libavcodec-dev, libavformat-dev, libswscale-dev
- [optional] libtbb2 libtbb-dev
- [optional] libdc1394 2.x
- [optional] libjpeg-dev, libpng-dev, libtiff-dev, libjasper-dev, libdc1394-22-dev

The packages can be installed using a terminal and the following commands or by using Synaptic Manager:

```
[compiler] sudo apt-get install build-essential
[required] sudo apt-get install cmake git libgtk2.0-dev pkg-config liba
[optional] sudo apt-get install python-dev python-numpy libtbb2 libtbb-
```

Getting OpenCV Source Code

You can use the latest stable OpenCV version available in *sourceforge* or you can grab the latest snapshot from our Git repository.

Getting the Latest Stable OpenCV Version

- Go to our page on Sourceforge;
- Download the source tarball and unpack it.

Getting the Cutting-edge OpenCV from the Git Repository

Launch Git client and clone OpenCV repository

In Linux it can be achieved with the following command in Terminal:

```
cd ~/<my_working _directory>
git clone https://github.com/Itseez/opencv.git
```

Building OpenCV from Source Using CMake, Using the Command Line

1. Create a temporary directory, which we denote as <cmake_binary_dir>, where you want to put the generated Makefiles, project files as well the object files and output binaries.

2. Enter the <cmake_binary_dir> and type

   ```
   cmake [<some optional parameters>] <path to the OpenCV source directory>
   ```

   For example

   ```
   cd ~/opencv
   mkdir release
   cd release
   cmake -D CMAKE_BUILD_TYPE=RELEASE -D CMAKE_INSTALL_PREFIX=/usr/local ..
   ```

3. Enter the created temporary directory (<cmake_binary_dir>) and proceed with:

   ```
   make -j8 # -j8 runs 8 jobs in parallel.
            # Change 8 to number of hardware threads available.
   sudo make install
   ```

Note: If the size of the created library is a critical issue (like in case of an Android build) you can use the `install/strip` command to get the smallest size as possible. The *stripped* version appears to be twice as small. However, we do not recommend using this unless those extra megabytes do really matter.

Help and Feedback

You did not find what you were looking for?

Ask a question on the **Q&A forum**.
If you think something is missing or wrong in the documentation, please file a **bug report**.