Building and installing OpenCV 2.4.8 on Linux

Assuming you already have a working native Linux install (I recommend Ubuntu version 12.04 LTS 64-bit on most newer machines), the first step is to get the required packages to compile OpenCV on Linux. Much of this software may be already on your system, but it can’t hurt to check.

We’ll be using CMake (cross-platform make) and GCC to compile OpenCV and install it on your system. Open the terminal - we’ll get these first, and then the OpenCV dependencies (for example, the ffmpeg/libav library for video compression/encoding). Open the terminal, and type (or copy and paste) the following:

```
sudo apt-get install gcc
```

This will update or install gcc on your system from an internet repository. It will quite likely tell you there is nothing to do (e.g. “gcc is the newest version”). Likewise,

```
sudo apt-get -y install build-essential checkinstall cmake
pkg-config yasm
```

Next, we need to ensure that any older version of the ffmpeg or x264 encoding/compression utilities are removed:

```
sudo apt-get -y remove ffmpeg x264 libx264-dev
```

Now, we may as well install the rest of them in a batch:

```
sudo apt-get install libopencv-dev libtiff4-dev libjpeg-dev
libjasper-dev libavcodec-dev libavformat-dev libswscale-dev
libdcd1394-22-dev libxine-dev libgstreamer0.10-dev
libgstreamer-plugins-base0.10-dev libv4l-dev python-dev
python-numpy libthb-dev libqt4-dev libgtk2.0-dev libfaac-dev
libmp3lame-dev libopencv-amrnb-dev libopencv-amrwb-dev
libtheora-dev libvorbis-dev libxvidcore-dev x264 v4l-utils ffmpeg
libgtk2.0-dev
```

This may take some time depending on your connection speed, so brew yourself a cup of tea or coffee. Next, we’ll retrieve the OpenCV source files themselves.

```
wget -O OpenCV-2.4.8.zip
http://sourceforge.net/projects/opencvlibrary/files/opencv-unix/2.4.8/opencv-2.4.8.zip/download
```

Hmm, perhaps a second cup would be in order- maybe a danish? This is the home stretch—once we have the zip file, we just need to extract it and then build the source with cmake.
Let's create build as a temporary directory within our opencv-2.4.8 folder to build inside of, and again use cd inside it:

```
mkdir build
cd build
```

Finally, we need only make and install it:

```
cmake -D CMAKE_BUILD_TYPE=RELEASE
     -D CMAKE_INSTALL_PREFIX=/usr/local
     -D WITH_TBB=ON
     -D BUILD_NEW_PYTHON_SUPPORT=ON
     -D WITH_V4L=ON
     -D INSTALL_C_EXAMPLES=ON
     -D INSTALL_PYTHON_EXAMPLES=ON
     -D BUILD_EXAMPLES=ON
     -D WITH_QT=ON
     -D WITH_OPENGL=ON
make
sudo make install
```

We now just need to configure OpenCV using ldconfig so that it can be used a shared library in a C/C++ program (for example):

```
sudo sh -c 'echo "/usr/local/lib" > /etc/ld.so.conf.d/opencv.conf'
sudo ldconfig
```

And we're done. But let's test that it works by trying an example. Fortunately, there's a convenient shell script provided that can build all the provided examples for us. Use cd to get to the samples/c directory, then we just need to allow the script as an executable program with chmod and then run it:

```
chmod +x build_all.sh
./build_all
```

Since future tutorials will increasingly deal with object detection and related topics, let's try the object detection example:

```
./find_obj
```
Without any parameters, it will use a default image of a cookie box being detected in a larger image, but you can provide any source image and target you like in this way:

.

./find_obj <source_image> <target_image>