

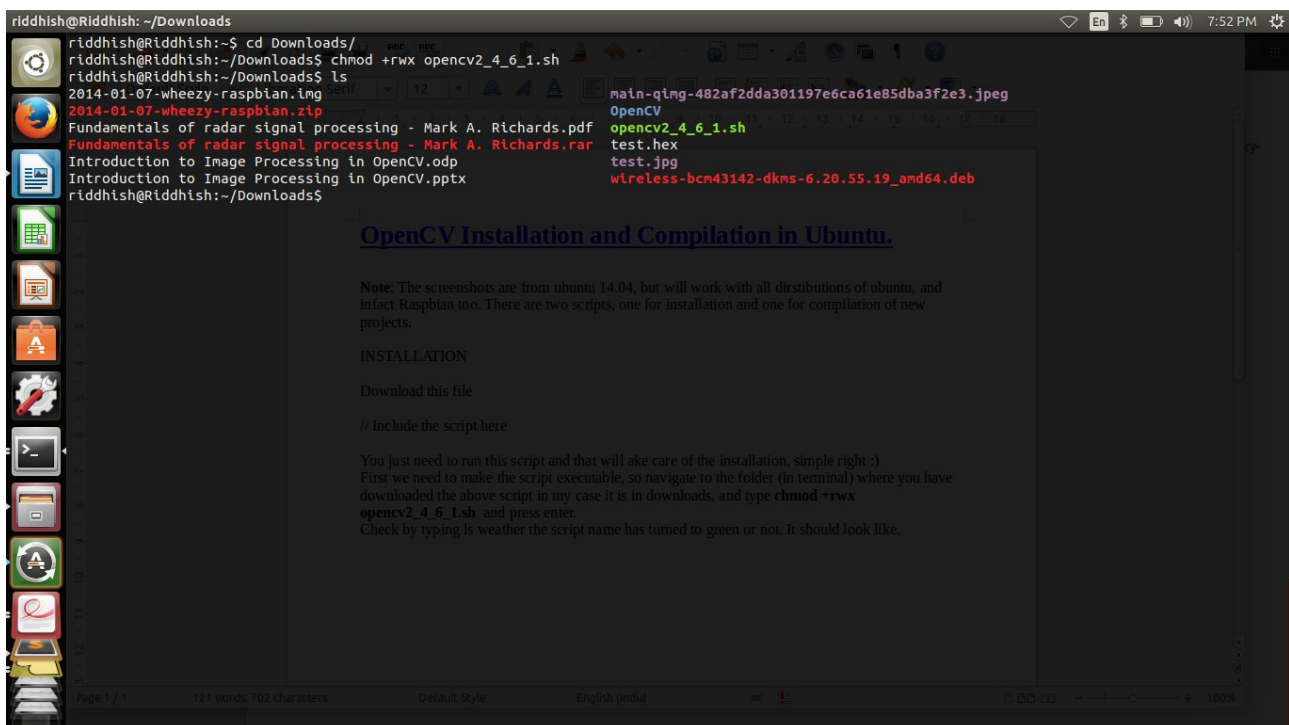
OpenCV Installation and Compilation in Ubuntu.

Note: The screenshots are from ubuntu 14.04, but will work with all distributions of ubuntu, and infact Raspbian too. There are two scripts, one for installation and one for compilation of new projects.

INSTALLATION

Download the opencv2_4_6_1.sh script attached

You just need to run this script and that will take care of the installation, simple right :)
First we need to make the script executable, so navigate to the folder (in terminal) where you have downloaded the above script in my case it is in downloads, and type **chmod +rwx opencv2_4_6_1.sh** and press enter.
Check by typing **ls** whether the script name has turned to green or not. It should look like.

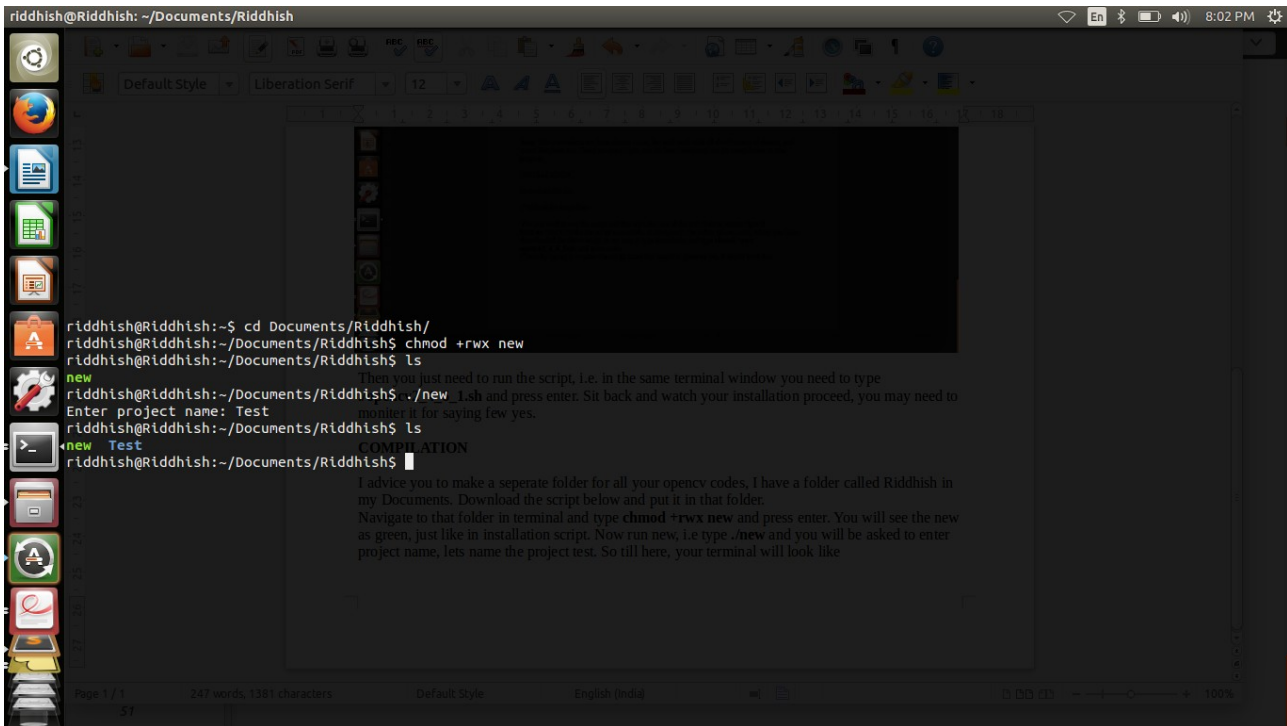


```
riddhish@Riddhish: ~/Downloads
riddhish@Riddhish:~$ cd Downloads/
riddhish@Riddhish:~/Downloads$ chmod +rwx opencv2_4_6_1.sh
riddhish@Riddhish:~/Downloads$ ls
2014-01-07-wheezy-raspbian.img
2014-01-07-wheezy-raspbian.zip
Fundamentals of radar signal processing - Mark A. Richards.pdf
Fundamentals of radar signal processing - Mark A. Richards.rar
Introduction to Image Processing in OpenCV.odp
Introduction to Image Processing in OpenCV.pptx
riddhish@Riddhish:~/Downloads$
```

Then you just need to run the script, i.e. in the same terminal window you need to type **./opencv2_4_6_1.sh** and press enter. Sit back and watch your installation proceed, you may need to monitor it for saying few yes.

COMPILATION

I advice you to make a separate folder for all your opencv codes, I have a folder called Riddhish in my Documents. Download the new script and put it in that folder.
Navigate to that folder in terminal and type **chmod +rwx new** and press enter. You will see the new as green, just like in installation script. Now run new, i.e type **./new** and you will be asked to enter project name, lets name the project test. So till here, your terminal will look like



Now each time you need a new code you will execute new file and the separate folder will be created for that code.

You will see that in Test folder we just created there is a blank Test.cpp inside it, here you will write your code, so here we will write a simple test code to load an image and display it on a window.

```

#include "opencv2/imgproc/imgproc.hpp"
#include "opencv2/highgui/highgui.hpp"
#include <stdlib.h>
#include <stdio.h>

using namespace std;
using namespace cv;

int main(){
    Mat img;
    img = imread("lena.jpg",1);
    namedWindow("Input", CV_WINDOW_AUTOSIZE);
    imshow("Input",img);
    waitKey(0);
    return 0;
}

```

Now inside each project you need to build another folder called 'build' (I prefer to name it this :P) by typing **mkdir build** , press enter and navigate to build. Then type **cmake ..** this you will just need to do it once per project. Till here you will have in your terminal something like.

```

riddhish@Riddhish: ~/Documents/Riddhish/Test/build
riddhish@Riddhish:~$ cd Documents/Riddhish/Test/
riddhish@Riddhish:~/Documents/Riddhish/Test$ nano Test.cpp
riddhish@Riddhish:~/Documents/Riddhish/Test$ ls
CMakeLists.txt Test.cpp
riddhish@Riddhish:~/Documents/Riddhish/Test$ mkdir build
riddhish@Riddhish:~/Documents/Riddhish/Test$ ls
build
riddhish@Riddhish:~/Documents/Riddhish/Test$ cd build/
riddhish@Riddhish:~/Documents/Riddhish/Test/build$ ls
riddhish@Riddhish:~/Documents/Riddhish/Test/build$ cmake ..
-- The C compiler identification is GNU 4.8.2
-- The CXX compiler identification is GNU 4.8.2
-- Check for working C compiler: /usr/bin/cc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Check for working CXX compiler: /usr/bin/c++
-- Check for working CXX compiler: /usr/bin/c++ -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Configuring done
-- Generating done
-- Build files have been written to: /home/riddhish/Documents/Riddhish/Test/build
riddhish@Riddhish:~/Documents/Riddhish/Test/build$ ls
lena.jpg;1)
namedWindow("Image", CV_WINDOW_AUTOSIZE);
imshow("Image",img);
waitKey(0);
return 0;
}

Now inside each project you need to build another folder called 'build' (I prefer to name it this :P)
by typing mkdir build , press enter and navigate to build. Then type cmake .. this you will just
need to do it once per project. Till here you will have in your terminal something like.

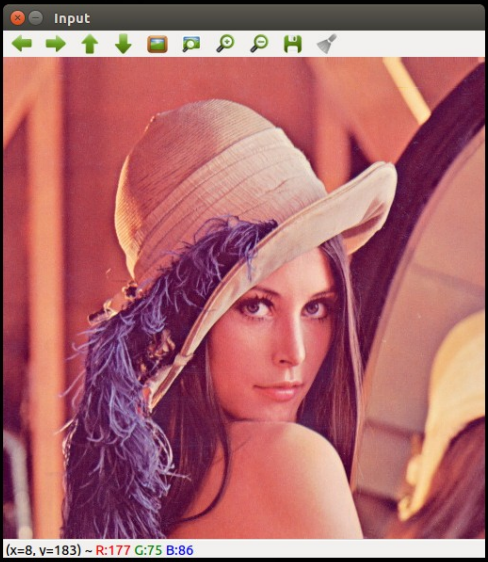
```

Then to compile your code each time you edit your Test.cpp (or whatever the name might be). Just type **make** inside build, and if the compilation is successful then on typing **ls** we will see a green **Test** inside build, that is our executable code, and execute it by typing **./Test** and press enter. Voila! Ok, one more thing you need to have the image we are using i.e. lena.jpg inside the build folder. Hence till the execution of code we have.

```

Input
riddhish@Riddhish:~/Documents/Riddhish/Test/build$ make Test.cpp
Scanning dependencies of target Test
[100%] Building CXX object CMakeFiles/Test.dir/Test.cpp.o
Linking CXX executable Test
[100%] Built target Test
riddhish@Riddhish:~/Documents/Riddhish/Test/build$ ls
CMakeCache.txt CMakeFiles cmake_install.cmake lena.jpg Makefile Test
riddhish@Riddhish:~/Documents/Riddhish/Test/build$ ./Test
init done
opengl support available
main()
{
    cv::Mat img;
    img = imread("lena.jpg",1);
    namedWindow("Input",CV_WINDOW_AUTOSIZE);
    imshow("Input",img);
    waitKey(0);
    return 0;
}

```



- by
Riddhish Bhalodia
(riddhishb@gmail.com)

// ping me if you face any problems