Install OpenCV 3 with Python 3 on Windows

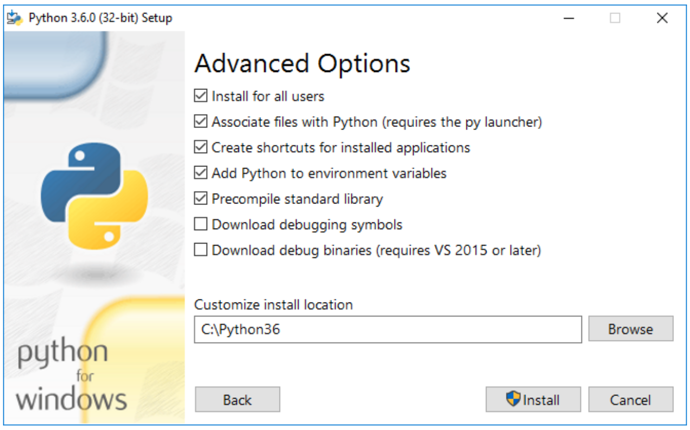
Posted on September 17, 2016 by Sol

Updated 25 February 2017

If you need a short tutorial about how to get started with OpenCV 3.2 programming in Python 3.6 on Windows, you are in the right place. Most articles I found online, including the OpenCV documentation, seem concerned only with Python 2.7.

We'll start by installing the latest stable version of Python 3, which at the time of this writing is 3.6. Head over to https://www.python.org/downloads/ and download the installer. The default Python Windows installer is 32 bits and this is what I will use in this article. If you need the 64 bits version of Python, check the *Looking for a specific release?* section from the above page.

Start the installer and select *Customize installation*. On the next screen leave all the optional features checked. Finally, on the *Advanced Options* screen make sure to check *Install for all users*, *Add Python to environment variables* and *Precompile standard library*. Optionally, you can customize the install location. I've used *C:\Python36*. You should see something like this:



Python 3.6.0 (32-bit) Setup

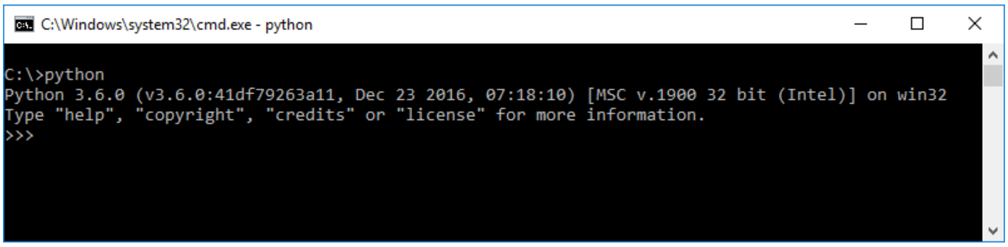
Advanced Options

☑ Install for all users
☑ Associate files with Python (requires the py launcher)
☑ Create shortcuts for installed applications
☑ Add Python to environment variables
☑ Precompile standard library
☐ Download debugging symbols
☐ Download debug binaries (requires VS 2015 or later)

Customize install location: C:\Python36

Press the *Install* button and in a few minutes, depending on the speed of your computer, you should be ready. Now, to check if Python was correctly installed, press and hold the *SHIFT* key and right click with your mouse somewhere on your desktop, select *Open command window here*. Alternatively, on Windows 10, use the bottom left search box to search for *cmd*.

Now, write *python* in the command window and press *Enter*, you should see something like this:



C:\Windows\system32\cmd.exe - python

```
C:\>python
Python 3.6.0 (v3.6.0:41df79263a11, Dec 23 2016, 07:18:10) [MSC v.1900 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Next, download and install the Microsoft Visual C++ 2015 Redistributable. Be sure to select the version corresponding to your Python installer (32 or 64 bits).

At the time of this writing, the *OpenCV for Windows* binary from http://opencv.org/ doesn't have the required *pyd* file for *Python 3*. We'll use a *wheel* package from http://www.lfd.uci.edu/~gohlke/pythonlibs/ in order to install *OpenCV* and *NumPy* which is a prerequisite for *OpenCV*.

Download the *Numpy* version corresponding to your Python installation from here. In my case, I've used *numpy-1.12.0+mkl-cp36-cp36m-win32.whl*

Download the *OpenCV* version corresponding to your Python installation from here. In my case, I've used *opencv_python-3.2.0-cp36-cp36m-win32.whl*.

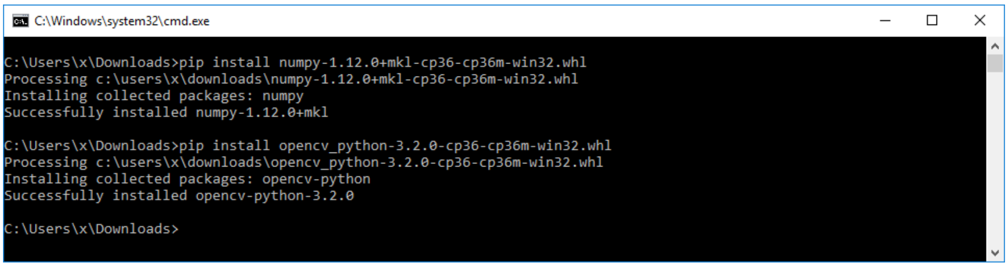
Now, open a *cmd* window like before. You can open this directly in your *Downloads* folder if you *SHIFT* and right click inside it. The idea is to open a *cmd* window where you've downloaded the above two files. Use the next set of commands to install *NumPy* and *OpenCV*:

```
pip install numpy-1.12.0+mkl-cp36-cp36m-win32.whl
```

and

```
pip install opencv_python-3.2.0-cp36-cp36m-win32.whl
```

You should see something like this:



```
C:\Users\x\Downloads>pip install numpy-1.12.0+mkl-cp36-cp36m-win32.whl
Processing c:\users\x\downloads\numpy-1.12.0+mkl-cp36-cp36m-win32.whl
Installing collected packages: numpy
Successfully installed numpy-1.12.0+mkl

C:\Users\x\Downloads>pip install opencv_python-3.2.0-cp36-cp36m-win32.whl
Processing c:\users\x\downloads\opencv_python-3.2.0-cp36-cp36m-win32.whl
Installing collected packages: opencv-python
Successfully installed opencv-python-3.2.0

C:\Users\x\Downloads>
```

Note the *Successfully installed …* message after each command.

At this point, you should be able to play with *OpenCV* and *Python*. Let's try a small test first. Start the *Python* interpreter and write:

```python
import cv2

print(cv2.__version__)
```

If everything was correctly installed, you should see the version number of your *OpenCV* install, in my case this was *3.2.0*.

Let's try a final test, in which I'll show you how to load an image from a file, convert it to gray, and check the results. Start by downloading the next image:

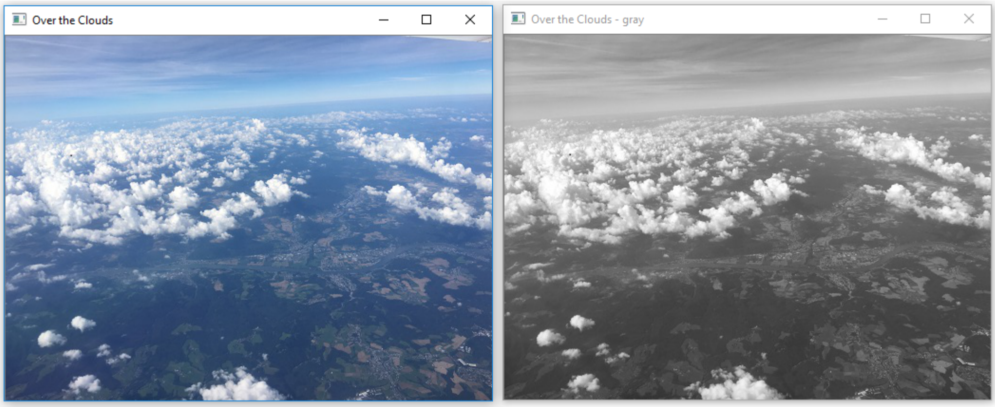
Save it as *clouds.jpg*. In the same folder where you've saved the above image, create a new file *demo.py* and write this small code:

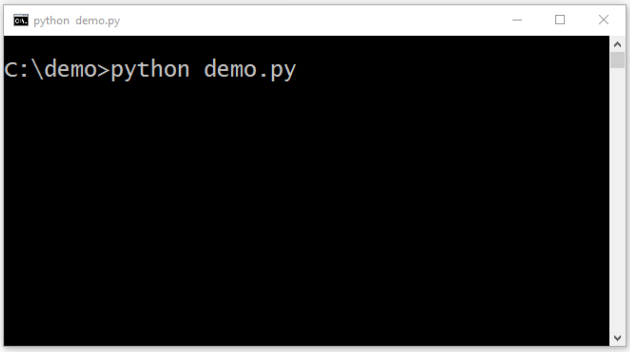
```python
import cv2

image = cv2.imread("clouds.jpg")
gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
cv2.imshow("Over the Clouds", image)
cv2.imshow("Over the Clouds - gray", gray_image)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Open a *cmd* window in this folder and write:

```
python demo.py
```

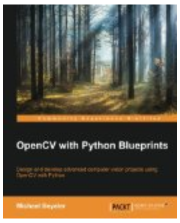
You should see something like this (by default the last image will be over the first one, you need to move the image window in order to see the first image):



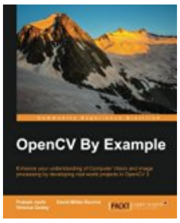


```
C:\demo>python demo.py
```

If you want to learn more about OpenCV and Python I would recommend reading OpenCV with Python Blueprints by M. Beyeler:



or, for OpenCV C++, OpenCV By Example by P. Joshi:



comments powered by Disqus

Categories

Arduino Armadillo Books C C++ C++11 C++14 C++17 C11 C99 Charts Clang Cling C# CUDA Fortran Fractals GCC iOS JavaScript Jekyll Linux macOS Matplotlib Multithreading Node.js NumPy Objective-C OpenCV OpenGL OpenMP OSX Perlin Posix Productivity Python Raspberry Pi Regex Roguelike Scheme SciPy Swift Videos Windows WSL