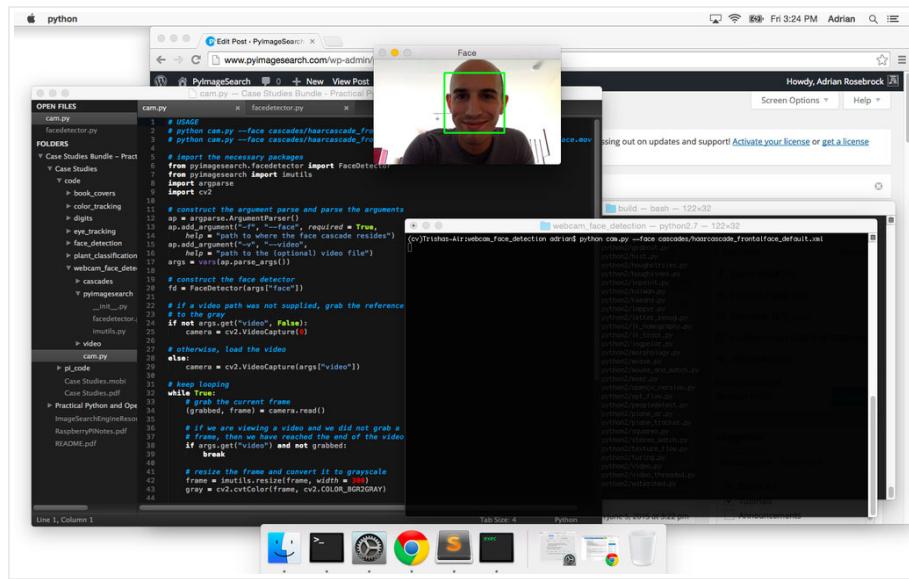




Install OpenCV 3.0 and Python 2.7+ on OSX

by Adrian Rosebrock on June 15, 2015 in [OpenCV 3, Tutorials](#)

19



As I mentioned last week, **OpenCV 3.0 is finally here!**

And if you've been paying attention to my Twitter stream, you may have noticed a bunch of tweets regarding installing OpenCV on OSX and Ubuntu (yep, I've been tweeting a lot lately, but that's just because I'm so excited about the 3.0 release!)

To celebrate OpenCV 3.0, I have decided to perform a series of blog posts that detail how to **install OpenCV 3.0 on both Python 2.7+ and Python 3+**.

We'll also be performing these Python 2.7 and Python 3+ installations on a variety of platforms including **OSX**, **Ubuntu**, and yes, the **Raspberry Pi**.

As I'm sure you already know, OpenCV has never been an effortless library to install. It's not like you can let **pip** or **easy_install** to the heavy-lifting for you. In most cases you'll be pulling down the repo, installing prerequisites, compiling by hand, and hoping that your installation goes smoothly.

With OpenCV 3.0 it doesn't get any easier — *and there are definitely some caveats and gotchas that you need to look out for* (such as the `opencv_contrib` repository — without it, you'll be missing out on some important features, such as SIFT, SURF, etc.)

But don't worry, I've got you covered! Just keep following along with the PyImageSearch blog and I promise these tutorials will get you up and running with OpenCV 3.0 in no time.

We'll go ahead and kick-off our OpenCV 3.0 install fest by installing v3.0 with **Python 2.7+ bindings** on the **OSX** platform.

If you're an Ubuntu or Raspberry Pi user, be sure to keep an eye on PyImageSearch as I'll be posting OpenCV 3.0 install instructions for Ubuntu and the Raspberry Pi as well.

A quick note before we get started: While OpenCV 3.0 is indeed compatible with Python 3+, most computer vision developers are still using Python 2.7 (since OpenCV 2.4.X is only compatible with Python 2.7). If you're a Python 3 user and would like to learn more about how to use OpenCV 3.0 with Python 3, please see my [tutorial on how to use OpenCV 3.0 with Python 3](#). I will be covering OpenCV 3.0 and Python 3+ installation in a future tutorial. But for now, let's stick with what we know and learn how to use OpenCV 3.0 with Python 2.7.

Free 21-day crash course on computer vision & image search engines

UPDATE: The tutorial you are reading now covers how to install OpenCV 3.0 with Python 2.7 bindings on **OSX Yosemite and below**. This tutorial **still works perfectly** if you are using OSX Yosemite or prior, but if you want to install OpenCV on the newer **El Capitan and macOS Sierra** please use [this freshly updated tutorial](#).

How to Install OpenCV 3.0 and Python 2.7+ on OSX

This is our first tutorial in our OpenCV 3.0 install-fest series. In this tutorial I'll be detailing how to install OpenCV 3.0 and Python 2.7+ on the OSX operating system — I'll be covering Python 3+ in a future post.

Let's go ahead and dive into the OpenCV 3.0 and Python 2.7+ install instructions.

Step 1:

The first step we need to do is install Xcode, which is a combination of IDE and software development tools for iOS platforms — most of us already have Xcode installed.

But if you don't, you'll want to open up the App Store application and search for Xcode. From there you'll need to enter your Apple ID username and password:

Figure 1: Installing Xcode on your OSX system.

Step 2:

Now that Xcode is installed, we need to install Homebrew, which is labeled as "*The missing package manager for OSX*" (and they really are not joking about that one). Think of Homebrew as an (almost) equivalent of apt-get for Ubuntu.

To install Homebrew, simply head to the [Homebrew website](#) and simply copy and paste the command underneath the "Install Homebrew" section into your terminal:

Install OpenCV 3.0 and Python 2.7+ on OSX	Shell
1 \$ cd ~	
2 \$ \$ ruby -e "\$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"	

Now that Homebrew is installed, you'll need to update it and grab the latest package (i.e. "formula") definitions. These formulae are simply instructions on how to install a given package or library.

To update Homebrew, simply execute:

Install OpenCV 3.0 and Python 2.7+ on OSX	Shell
1 \$ brew update	

Step 3:

It's bad form to use the system Python as your main interpreter. And this is especially true if you intend on using `virtualenv` and `virtualenvwrapper` (which we will be).

Let's go ahead and use Homebrew to install our user-specific version of Python 2.7:

```
Install OpenCV 3.0 and Python 2.7+ on OSX
1 $ brew install python
```

Note: This tutorial will be covering how to install and setup **OpenCV 3.0 with Python 2.7**. I will be covering how to install **OpenCV 3.0 with Python 3+** later this month.

However, before we proceed, we need to update our `PATH` in our `~/.bash_profile` file to include any system libraries or packages. *This is an absolutely critical step, so be sure not to skip it!*

Open up your `~/.bash_profile` file in your favorite editor (if it does not exist, [create it](#)), and append:

```
Install OpenCV 3.0 and Python 2.7+ on OSX
1 # Homebrew
2 export PATH=/usr/local/bin:$PATH
```

From there, reload your `~/.bash_profile` file to ensure the changes have been made:

```
Install OpenCV 3.0 and Python 2.7+ on OSX
1 $ source ~/.bash_profile
```

As a sanity check, let's confirm that we are using the Homebrew version of Python rather than the system version:

```
Install OpenCV 3.0 and Python 2.7+ on OSX
1 $ which python
2 /usr/local/bin/python
```

If your output of `which python` is `/usr/local/bin/python`, then you are indeed using the Homebrew version of Python. And if your output is `/usr/bin/python`, then you are still using the system version of Python — and you need to go back and ensure that your `~/.bash_profile` file is updated and reloaded correctly.

Again, this is a very important step, so be sure not to skip it!

Step 4:

Alright, time to get `virtualenv` and `virtualenvwrapper` installed and configured correctly. These packages allow us to create separate Python environments for each project we are working on. This is especially useful if you have projects that require different (or conflicting) versions of a given library.

It's **important to note** that `virtualenv` and `virtualenvwrapper` are **by no means required** to install OpenCV 3.0 and Python 2.7+ on OSX. However, **you really should be using these packages** when doing Python development. It's cleaner. Easier to maintain. And **well worth** the upfront effort.

Anyway, to install `virtualenv` and `virtualenvwrapper`, just execute the following command:

```
Install OpenCV 3.0 and Python 2.7+ on OSX
1 $ pip install virtualenv virtualenvwrapper
```

Again, we need to update our `~/.bash_profile` file by appending the following two lines:

```
Install OpenCV 3.0 and Python 2.7+ on OSX
1 # Virtualenv/VirtualenvWrapper
2 source /usr/local/bin/virtualenvwrapper.sh
```

After updating the `~/.bash_profile` file, we need to reload it:

```
Install OpenCV 3.0 and Python 2.7+ on OSX
1 $ source ~/.bash_profile
```

At this point, both `virtualenv` and `virtualenvwrapper` are installed correctly, so we can create our `cv` virtual environment:

```
Install OpenCV 3.0 and Python 2.7+ on OSX
1 $ mkvirtualenv cv
```

Free 21-day crash course on computer vision & image search engines

Interested in computer vision and image search engines, but don't know where to start? Let me help. I've created a free, 21-day crash course that is hand-tailored to give you the best possible introduction to computer vision. Sound good? Enter your email below to start your journey to becoming a computer vision master.

LET'S DO IT!

This command will create a new Python environment that is *entirely sequestered* from our system and Homebrew Python installations. The `cv` virtual environment is where we'll be doing all of our computer vision development (and not to mention, compiling OpenCV 3.0 with Python 2.7+ support).

Step 5:

Now we can start installing some Python packages. We need to install NumPy since the OpenCV Python bindings represent images as multi-dimensional NumPy arrays:

```
Install OpenCV 3.0 and Python 2.7+ on OSX
1 $ pip install numpy
```

Shell

Step 6:

Up until this point we have been mainly focusing on actually *setting up* and *configuring* our developer environment — **here is where the real work starts**.

First, we'll use brew to install the required developers tools, such as the wonderful CMake utility:

```
Install OpenCV 3.0 and Python 2.7+ on OSX
1 $ brew install cmake pkg-config
```

And here we are going to install the necessary image I/O packages. These packages allow you to read and write images in various formats, such as TIFF, etc.

```
Install OpenCV 3.0 and Python 2.7+ on OSX
1 $ brew install jpeg libpng libtiff opencv
```

And finally, let's install libraries that are used to optimize various operations within OpenCV (if we want to go fast).

```
Install OpenCV 3.0 and Python 2.7+ on OSX
1 $ brew install eigen tbb
```

Step 7:

Alright, our system is all setup — **time to compile and install OpenCV 3.0 with Python 2.7+ support**.

The first thing we'll do is change directory to our home directory, followed by pulling down OpenCV from GitHub, and checking out the `3.0.0` version:

```
Install OpenCV 3.0 and Python 2.7+ on OSX
1 $ cd ~
2 $ git clone https://github.com/Itseez/opencv.git
3 $ cd opencv
4 $ git checkout 3.0.0
```

Shell

Update (3 January 2016): You can replace the `3.0.0` version with whatever the current release is (as of right now, it's `3.1.0`). Be sure to check [OpenCV.org](#) for information on the latest release.

Unlike previous versions of OpenCV that were (essentially) self-contained, we need to pull down the extra `opencv_contrib` repo from GitHub as well. The `opencv_contrib` repo which contains extra modules for OpenCV, such as feature detection, local invariant descriptors (SIFT, SURF, etc.), text detection in natural images, line descriptors, and more.

```
Install OpenCV 3.0 and Python 2.7+ on OSX
1 $ cd ~
2 $ git clone https://github.com/Itseez/opencv_contrib
3 $ cd opencv_contrib
4 $ git checkout 3.0.0
```

Shell

Again, make sure that you checkout the **same version** for `opencv_contrib` that you did for `opencv` above, otherwise you could run into compilation errors.

Note: We don't have to pull down the `opencv_contrib` repo if we don't want to. OpenCV will compile and install just fine without it. But if you compile OpenCV **without** `opencv_contrib`, be warned that you'll be missing out on some pretty important features, which will become very obvious, very fast, especially if you're used to working with the 2.4.X version of OpenCV.

Step 8:

Let's setup our OpenCV build by creating the `build` directory:

<https://www.pyimagesearch.com/2015/06/15/install-opencv-3-0-and-python-2-7-on-osx/>

Free 21-day crash course on computer vision & image search engines

Interested in computer vision and image search engines, but don't know where to start? Let me help. I've created a free, 21-day crash course that is hand-tailored to give you the best possible introduction to computer vision. Sound good? Enter your email below to start your journey to becoming a computer vision master.

1
1
1
1

1

1

1

1

Free 21-day crash course on computer
vision & image search engines

```
Install OpenCV 3.0 and Python 2.7+ on OSX
1 $ cd ~/opencv
2 $ mkdir build
3 $ cd build
```

Where we'll use CMake to configure our build:

```
Install OpenCV 3.0 and Python 2.7+ on OSX
1 $ cmake -D CMAKE_BUILD_TYPE=RELEASE -D CMAKE_INSTALL_PREFIX=/usr/local \
2   -D PYTHON2_PACKAGES_PATH=~/virtualenvs/cv/lib/python2.7/site-packages \
3   -D PYTHON2_LIBRARY=/usr/local/Cellar/python/2.7.10/Frameworks/Python.framework/Versions/2.7/bin \
4   -D PYTHON2_INCLUDE_DIR=/usr/local/Frameworks/Python.framework/Headers \
5   -D INSTALL_C_EXAMPLES=ON -D INSTALL_PYTHON_EXAMPLES=ON \
6   -D BUILD_EXAMPLES=ON \
7   -D OPENCV_EXTRA_MODULES_PATH=~/opencv_contrib/modules ..
```

There are some **very important** options we are supplying to CMake here, so let's break them down:

- CMAKE_BUILD_TYPE** : This option indicates that we are building a release binary of OpenCV.
- CMAKE_INSTALL_PREFIX** : The base directory where OpenCV will be installed.
- PYTHON2_PACKAGES_PATH** : The **explicit** path to where our **site-packages** directory lives.
- PYTHON2_LIBRARY** : Path to our Homebrew installation of Python.
- PYTHON2_INCLUDE_DIR** : The path to our Python header files for compilation.
- INSTALL_C_EXAMPLES** : Indicate that we want to install the C/C++ examples after compilation.
- INSTALL_PYTHON_EXAMPLES** : Indicate that we want to install the Python examples after compilation.
- BUILD_EXAMPLES** : A flag that determines whether or not the included OpenCV examples will be built.
- OPENCV_EXTRA_MODULES_PATH** : **This option is extremely important** — here we supply the path to the extra modules we downloaded earlier, indicating that OpenCV should compile the extra modules as well.

Update (3 January 2016): In order to build OpenCV 3.1.0, you need to set **-D INSTALL_C_EXAMPLES=ON**. There is a bug in the OpenCV v3.1.0 CMake build script that can cause errors if you skip this command. There is a bug in the OpenCV v3.1.0 CMake build script that can cause errors if you skip this command. CMake should run without a problem.

Whew, that was a lot of options.

Trust me, it's a lot easier installing OpenCV 3.0 on Ubuntu where these options are *automatically determined* via CMake for us.

But when **using OSX you'll need to explicitly define** the **PYTHON2_PACKAGES_PATH**, **PYTHON2_LIBRARY**, and **PYTHON2_INCLUDE_DIR** yourself. It's a real pain, but if you don't, your compile will fail.

Here's an example of what my CMake output looks like:

```
-- Use Cuda: NO
-- Use OpenCL: YES
-- OpenCL:
--   Version: static
--   Libraries: -framework OpenCL
--   Use AMDFFT: NO
--   Use AMDBLAS: NO
-- Python 2:
--   Interpreter: /Users/adrian/.virtualenvs/cv/bin/python2.7 (ver 2.7.10)
--   Libraries: /usr/local/Cellar/python/2.7.10/Frameworks/Python.framework/Versions/2.7/bin (ver 2.7.10)
--   numpy: /Users/adrian/.virtualenvs/cv/lib/python2.7/site-packages/numpy/core/include (ver 1.9.1)
--   packages path: /Users/adrian/.virtualenvs/cv/lib/python2.7/site-packages
-- Python 3:
--   Interpreter: NO
--   Python (for build): /Users/adrian/.virtualenvs/cv/bin/python2.7
-- Java:
--   ant: NO
--   JNLP: /Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Developer/SDKs/MacOSX10.10.sdk/System/Library/Frameworks/JavaVM.framework/Headers /Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Developer/SDKs/MacOSX10.10.sdk/System/Library/Frameworks/JavaVM.framework/Headers /Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Developer/SDKs/MacOSX10.10.sdk/System/Library/Frameworks/JavaVM.framework/Headers
--   Java wrappers: NO
--   Java tests: NO
-- Matlab:
```

Figure 2: Before compiling OpenCV 3.0 on your OSX system, make sure that cmake has picked up the correct Python interpreter, library, numpy version, and packages path.

Notice how the Python 2 Interpreter, Libraries, numpy version, and packages path have been correctly detected by CMake.

Free 21-day crash course on computer vision & image search engines

Interested in computer vision and image search engines, but don't know where to start? Let me help. I've created a free, 21-day crash course that is hand-tailored to give you the best possible introduction to computer vision. Sound good? Enter your email below to start your journey to becoming a computer vision master.

LET'S DO IT!

You'll also want to make sure that `python2` is in the list of modules **To be built**, like this:

```
-- Extra dependencies: -framework OpenCL -framework Cocoa -framework QTKit -framework QuartzCore -framework A
-- 3rdparty dependencies: libjpeg libwebp libpng libtiff libjasper IlmImf zlib ippicv
-- OpenCV modules:
--   To be built: hal core flann imgproc ml photo reg surface_matching video face imgcodecs shape videoio
--   highgui objdetect optflow superres tracking ts ximgproc xobjdetect xphoto oadss bgsegm biomimicd features2d latentsvm l
--   line_descriptor saliency text calib3d colib datasets rgbd videostab xfeatures2d stitching python2
--   Disabeled: world contrib_world
--   Disabled by dependency: -
--   Unavailable: cudaarithm cudabgsegm cudafeatures2d cudafilters cudaimproc cudalegacy cuda
--   objdetect cudadetectflow cudastereo cudawarping cudev Java python3 viz cvv matlab
-- GUI:
--   QT: NO
--   Cocoa: YES
--   OpenGL support: NO
--   VTK support: NO
-- Media I/O:
--   ZLib: build (ver 1.2.8)
--   JPEG: build (ver 98)
--   WEBP: build (ver 0.3.1)
--   PNG: build (ver 1.5.12)
--   TIFF: build (ver 42 - 4.0.2)
--   JPEG 2000: build (ver 1.900.1)
--   OpenEXR: build (ver 1.7.1)
--   GDAL: NO
-- Video I/O:
--   DC1394 1.x: NO
--   DC1394 2.x: NO
```

Figure 2: Ensuring the "python2" module is in our list of modules.

If it `python2` is not in this list, and is in the **Unavailable** list, then you need to go back to the CMake configuration screen and add it to the `PYTHON2_PACKAGES_PATH`, `PYTHON2_LIBRARY`, and `PYTHON2_INCLUDE_DIR`.

Now that CMake has properly configured the build, we can compile OpenCV:

Install OpenCV 3.0 and Python 2.7+ on OSX
1 \$ `make -j4`

Where the 4 can be replaced with however many cores you have available on your processor. Here's an example of OpenCV compiling on my system:

```
[ 50%] Building CXX object modules/shape/CMakeFiles/openvc_test_shape.dir/test/test_main.cpp.o
[ 50%] Building CXX object modules/shape/CMakeFiles/openvc_test_shape.dir/test/test_shape.cpp.o
Linking CXX executable ../../bin/openvc_test_shape
[ 50%] Built target openvc_test_shape
Scanning dependencies of target openvc_perf_videoio
[ 50%] Building CXX object modules/videoio/CMakeFiles/openvc_perf_videoio.dir/perf_input.cpp.o
[ 50%] Building CXX object modules/videoio/CMakeFiles/openvc_perf_videoio.dir/perf_min.cpp.o
[ 50%] Building CXX object modules/videoio/CMakeFiles/openvc_perf_videoio.dir/perf_output.cpp.o
Linking CXX executable ../../bin/openvc_perf_videoio
[ 50%] Built target openvc_perf_videoio
Scanning dependencies of target openvc_test_videoio
[ 50%] Building CXX object modules/videoio/CMakeFiles/openvc_test_videoio.dir/test/test_basic_props.cpp.o
[ 50%] Building CXX object modules/videoio/CMakeFiles/openvc_test_videoio.dir/test/test_ffmpeg.cpp.o
[ 50%] Building CXX object modules/videoio/CMakeFiles/openvc_test_videoio.dir/test/test_fourcc.cpp.o
[ 50%] Building CXX object modules/videoio/CMakeFiles/openvc_test_videoio.dir/test/test_framecount.cpp.o
[ 50%] Building CXX object modules/videoio/CMakeFiles/openvc_test_videoio.dir/test/test_min.cpp.o
[ 50%] Building CXX object modules/videoio/CMakeFiles/openvc_test_videoio.dir/test/test_positioning.cpp.o
[ 50%] Building CXX object modules/videoio/CMakeFiles/openvc_test_videoio.dir/test/test_video_to.cpp.o
[ 50%] Building CXX object modules/videoio/CMakeFiles/openvc_test_videoio.dir/test/test_video_pos.cpp.o
Linking CXX executable ../../bin/openvc_test_videoio
[ 50%] Built target openvc_test_videoio
Scanning dependencies of target openvc_test_highgui
[ 50%] Building CXX object modules/highgui/CMakeFiles/openvc_test_highgui.dir/test/test_gui.cpp.o
[ 50%] Building CXX object modules/highgui/CMakeFiles/openvc_test_highgui.dir/test/test_main.cpp.o
Linking CXX executable ../../bin/openvc_test_highgui
[ 50%] Built target openvc_test_highgui
[ 50%] Generating opencv_kernels_objdetect.cpp, opencv_kernels_objdetect.hpp
Scanning dependencies of target openvc_objdetect
[ 50%] Building CXX object modules/objdetect/CMakeFiles/openvc_objdetect.dir/src/cascadedetect.cpp.o
[ 50%] Building CXX object modules/objdetect/CMakeFiles/openvc_objdetect.dir/src/cascadedetect_convert.cpp.o
[ 50%] Building CXX object modules/objdetect/CMakeFiles/openvc_objdetect.dir/src/detection_based_tracker.cpp.o
```

Figure 3: OpenCV 3.0 with Python 2.7+ support compiling on my system.

And assuming that OpenCV compiled without error, you can now install it on your OSX system:

Install OpenCV 3.0 and Python 2.7+ on OSX
1 \$ `make install`

If you get an error message related to permissions (although that really shouldn't happen), you'll need to run the install command as `sudo`:

Install OpenCV 3.0 and Python 2.7+ on OSX
1 \$ `sudo make install`

Free 21-day crash course on computer vision & image search engines

Interested in computer vision and image search engines, but don't know where to start? Let me help. I've created a free, 21-day crash course that is hand-tailored to give you the best possible introduction to computer vision. Sound good? Enter your email below to start your journey to becoming a computer vision master.

Email Address

LET'S DO IT!

1

Step 9:

Assuming you've made it this far, let's perform a sanity check and ensure OpenCV is installed:

```
Install OpenCV 3.0 and Python 2.7+ on OSX
1 $ cd ~/virtualenvs/cv/lib/python2.7/site-packages/
2 $ ls -l cv2.so
3 -rwxr-xr-x 1 adrian staff 2013052 Jun 5 15:20 cv2.so
```

Sure enough, we can see that OpenCV has been installed in our `cv` virtual environment's `site-packages` directory!

As a quick note, you'll be able to find the `cv2.so` file (which is your OpenCV bindings) in your `build/lib` directory as well.

Let's verify our install by firing up a shell and importing OpenCV:

```
Install OpenCV 3.0 and Python 2.7+ on OSX
1 (cv)annalee:~ adrianrosebrock$ python
2 Python 2.7.8 (default, Jul 31 2014, 15:41:09)
3 [GCC 4.2.1 Compatible Apple LLVM 5.1 (clang-503.0.40)] on darwin
4 Type "help", "copyright", "credits" or "license" for more information.
5 >>> import cv2
6 >>> cv2.__version__
7 '3.0.0'
```

Doesn't that `3.0.0` look nice?

Congrats, you have now installed OpenCV 3.0 and Python 2.7+ on your OSX system!

Step 10:

After all this work, let's give our OpenCV 3.0 install a test drive!

Most of my work in computer vision involves image search engines, or more formally, Content-based features to quantify and abstractly represent the contents of an image.

OpenCV 3.0 has numerous updates and changes, but perhaps my personal favorite is an implementation of AKAZE features — *Fast Explicit Diffusion for Accelerated Features in Nonlinear Scale Spaces* by Alcantarilla et al.

Since *Jurassic World* was just released (and *Jurassic Park* is my favorite movie of all time), let's explore how we can compute and extract AKAZE features from the following image:



Figure 4: Our *Jurassic World* test image that we are going to detect keypoints and extract features in using AKAZE.

Open up a new file, name it `test_akaze.py`, and insert the following code:

```
Install OpenCV 3.0 and Python 2.7+ on OSX
1 # import the necessary packages
2 from __future__ import print_function
3 import cv2
4
```

Free 21-day crash course on computer vision & image search engines

Interested in computer vision and image search engines, but don't know where to start? Let me help. I've created a free, 21-day crash course that is hand-tailored to give you the best possible introduction to computer vision. Sound good? Enter your email below to start your journey to becoming a computer vision master.

LET'S DO IT!

```

5 # load the image and convert it to grayscale
6 image = cv2.imread("jurassic_world.jpg")
7 gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
8 cv2.imshow("Original", image)
9
10 # initialize the AKAZE descriptor, then detect keypoints and extract
11 # local invariant descriptors from the image
12 detector = cv2.AKAZE_create()
13 (kps, descs) = detector.detectAndCompute(gray, None)
14 print("keypoints: {}, descriptors: {}".format(len(kps), descs.shape))
15
16 # draw the keypoints and show the output image
17 cv2.drawKeypoints(image, kps, image, (0, 255, 0))
18 cv2.imshow("Output", image)
19 cv2.waitKey(0)

```

And then execute it via:

```

Install OpenCV 3.0 and Python 2.7+ on OSX
1 $ python test_akaze.py
2 keypoints: 762, descriptors: (762, 61)

```

Assuming you have download the [jurassic_world.jpg](#) image and placed it in the same directory, you will see the following output:

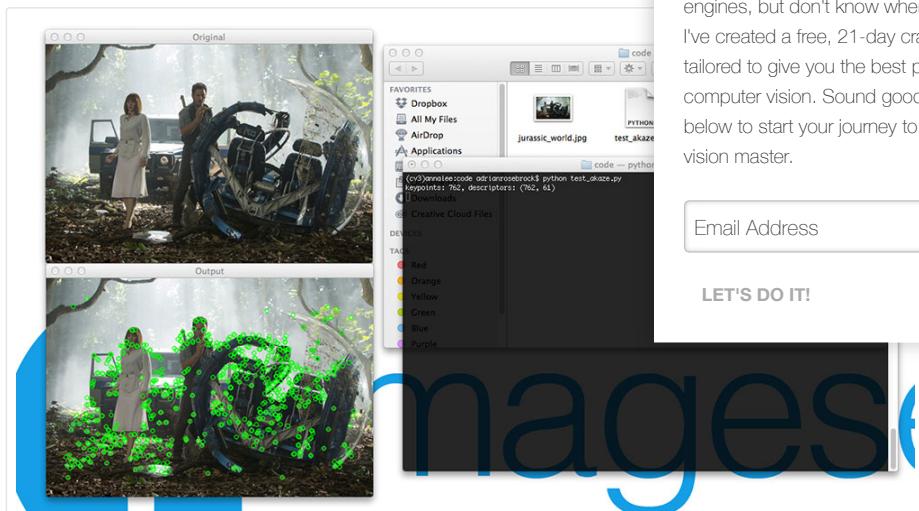


Figure 5: We have successfully been able to detect keypoints, extract AKAZE features, and then draw the keypoints on our image using OpenCV 3.0 and Python 2.7+!

Notice how we have been able to detect keypoints and extract AKAZE features in our image!

Obviously we need to do a lot more work than this to build a useful project using AKAZE features — but this example demonstrates that (1) our OpenCV 3.0 install is working, and (2) we are able to use a unique OpenCV 3.0 feature using Python 2.7.

Summary

OpenCV 3.0 is finally here! And to celebrate the OpenCV 3.0 release, we are going to performing an **OpenCV 3.0 install-fest** for both **Python 2.7+** and **Python 3+** on a variety of operating systems including **OSX**, **Ubuntu**, and the **Raspberry Pi**!

This article kicked-off the install fest by detailing how to setup and install OpenCV 3.0 and Python 2.7+ on the OSX operating system.

Next week, we'll be moving over to Ubuntu and detailing the instructions to get OpenCV 3.0 and Python 2.7 installed on Ubuntu 14.04+ (*hint: it's much easier than OSX*).

Anyway, I hope you enjoyed this post and found it useful!

Please consider subscribing to the PyImageSearch Newsletter by entering your email address in the form below — I'll be sending out updates as new OpenCV 3.0 + Python install instructions are released!

Resource Guide (it's totally free).

Free 21-day crash course on computer vision & image search engines

Interested in computer vision and image search engines, but don't know where to start? Let me help. I've created a free, 21-day crash course that is hand-tailored to give you the best possible introduction to computer vision. Sound good? Enter your email below to start your journey to becoming a computer vision master.

LET'S DO IT!

**Image Search Engine
Resource Guide**
Adrian Rosebrock
pyimagesearch
the awesome of building image search engines

Enter your email address below to get my **free 11-page Image Search Engine Resource Guide PDF**. Uncover **exclusive techniques** that I don't publish on this blog and start building image search engines of your own!

[DOWNLOAD THE GUIDE!](#)

◆ install, opencv 3, python 2.7, tutorials

◀ I was just interviewed on the Talk Python to Me Podcast.

316 Responses to *Install OpenCV 3.0 and Python 2.7+ on OSX*



Jason June 15, 2015 at 12:35 pm #

This is a great tutorial. This helps much more than the homebrew formula built for opencv 3

Only comments are, Line 6 of the cmake command is missing a "\" at the end of it so it executes pre

Also, a question. I am trying to follow the tutorial on this page.

http://docs.opencv.org/master/db/d5c/tutorial_py_bg_subtraction.html

createBackgroundSelectorMOG and createBackgroundSelectorGMG do not seem to be working in OpenCV 3.0 as installed here. I would really like to try GMG on an existing project, and that existing project fails now because MOG seems to be removed. Is this on purpose or am I missing a step to connect these and other OpenCV functions?

Free 21-day crash course on computer vision & image search engines

Interested in computer vision and image search engines, but don't know where to start? Let me help. I've created a free, 21-day crash course that is hand-tailored to give you the best possible introduction to computer vision. Sound good? Enter your email below to start your journey to becoming a computer vision master.

[LET'S DO IT!](#)



Adrian Rosebrock June 15, 2015 at 2:59 pm #

REPLY ↗

Hey Jason, thanks for the comment. I fixed the command and added in the trailing slash, so that shouldn't be a problem anymore.

As for your question, try using `cv2.createBackgroundSubtractorMOG2`. I'm not sure where `cv2.createBackgroundSubtractorMOG` (notice the lack of the trailing "2") went, but it's definitely not in the 3.0 release with Python bindings. I know both are part of the 2.4.11 release though, just with slightly different function names.

The GMG based method is not part of the Python bindings for the 3.0 release (again, I'm not sure why). I think it's available in one of the RC or beta releases (not positive though).

Also, which version of Python are you using? I've noticed that not all of the bindings are compiled and installed when using 2.7 vs. 3.



Jason June 15, 2015 at 5:16 pm #

REPLY ↗

Thank you for getting back to me. I never upgraded to Python 3, Python 2.7.10 is what I am using and has always satisfied my needs. I'm looking forward to more Python bindings from OpenCV in the future.



Patrick June 15, 2015 at 6:40 pm #

REPLY ↗

I've always used brew install opencv to get v2. Is it just a matter of time before v3 is that easy, or has something changed to prevent an easy install? Or have I been doing it wrong.

Free 21-day crash course on computer vision & image search engines



Adrian Rosebrock June 16, 2015 at 6:43 am #

REPLY ↗

There is a brew formula that has been released about 4 days ago for OpenCV 3.0, you can find it [here](#). The biggest issue is the customizability of the opencv_contrib package which now contains important features such as SIFT, SURF, etc. which were part of 2.4.X, but have been marked as optional in 3.0, mainly related to patent reasons.

Furthermore, by compiling from source you can install to a custom location and run multiple versions of Python and OpenCV at the same time! This is especially important since OpenCV 3.0 was just released and we're still finding bugs and backwards compatibility issues. With brew, you won't be able to do this.



Casey June 24, 2015 at 11:13 am #

REPLY ↗

First, thanks for the helpful guide. I was able to get OpenCV 3.0 running in my virtual environment using the Cmake options. Specifically, I had to turn off build_opencv_videoio for an error free compilation. I'm curious though, do you know what other dependancies I need? Perhaps you could post your entire Cmake output.



Adrian Rosebrock June 25, 2015 at 6:34 am #

Hi Casey, OpenCV has a lot of video I/O capabilities, is there a specific one that you are looking for? In my case, I did not have to install any extra dependencies related to video I/O and I can't seem to find any errors. However, that all said, look into the (optional) ffmpeg dependency which should bring in some more video I/O options.



Yong Yuan June 24, 2015 at 11:20 am #

Is it possible to install the OpenCV 3.0 with Homebrew by "brew install opencv3"? I want to use Xcode.

Free 21-day crash course on computer vision & image search engines

Interested in computer vision and image search engines, but don't know where to start? Let me help. I've created a free, 21-day crash course that is hand-tailored to give you the best possible introduction to computer vision. Sound good? Enter your email below to start your journey to becoming a computer vision master.

Email Address

LET'S DO IT!



Adrian Rosebrock June 25, 2015 at 6:27 am #

REPLY ↗

Indeed it is! I'll be doing another set of followup posts regarding installing OpenCV 3.0 with Homebrew within the next few weeks. These posts are simply to help readers who are interested in compiling and installing OpenCV 3.0 from source.



Mark June 29, 2015 at 8:54 am #

REPLY ↗

But will it include the additional plugins like 'amakaze' etc?



Adrian Rosebrock June 29, 2015 at 10:57 am #

REPLY ↗

Hey Mark, all extras from the opencv_contrib repository will be installed. Please see Step 7 where we clone down the git repos for more information.



Joseph July 7, 2015 at 11:49 am #

REPLY ↗

What are your views on using Anaconda? I was able to download the required packages through 'conda install' but the only opencv that's available for download is version 2.4.8. Have you used Anaconda before and if so, is there any way to upgrade it to version 3.0.0?



Adrian Rosebrock July 7, 2015 at 1:27 pm #

REPLY ↗

Free 21-day crash course on computer vision & image search engines

Continuum's Anaconda is a really great tool — but you really won't see me post about it here on PyImageSearch. The Continuum team and myself had a falling out about a year ago, and because of that, I don't use their products.



Joseph July 7, 2015 at 2:36 pm #

REPLY ↗

Thank you for the reply! I also apologize for any negative sentiments I brought with the comment. I was just debating whether I should just go with virtualenv and homebrew python to work with opencv 3.0.0.



Adrian Rosebrock July 7, 2015 at 3:41 pm #

REPLY ↗

No worries Joseph, there were no negative sentiments 😊 I'll be doing a blog post on how to install OpenCV 3.0.0 + Python 2.7 + Homebrew + OpenCV 3 as well.



Daniel Revier January 22, 2016 at 10:53 pm #

Hi Adrian,

Any progress on the Homebrew method of OpenCV3 installation? I can't seem to find anything online.

Thanks for everything!

Free 21-day crash course on computer vision & image search engines

Interested in computer vision and image search engines, but don't know where to start? Let me help. I've created a free, 21-day crash course that is hand-tailored to give you the best possible introduction to computer vision. Sound good? Enter your email below to start your journey to becoming a computer vision master.

Email Address

LET'S DO IT!



Adrian Rosebrock January 23, 2016 at 2:08 pm #

Thanks for reminding me about this. I'll try to get it online within the next month or so.



Hank December 2, 2015 at 6:16 pm #

REPLY ↗

If anyone is interested, you can easily set the location of a conda virtual environment as your python library source, instead of using virtualenv.



Adrian Rosebrock December 3, 2015 at 6:18 am #

Thanks for sharing Hank.



mehul July 7, 2015 at 4:31 pm #

REPLY ↗

hi this is the great guide , i have followed all the steps but finally when i run the code it hits me this error

```
OpenCV Error: Assertion failed (scn == 3 || scn == 4) in cvtColor, file /Users/mehulmistry/opencv/modules/imgproc/src/color.cpp, line 7564
Traceback (most recent call last):
File "test.py", line 6, in
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
cv2.error: /Users/mehulmistry/opencv/modules/imgproc/src/color.cpp:7564: error: (-215) scn == 3 || scn == 4 in function cvtColor
```



Adrian Rosebrock July 7, 2015 at 7:36 pm #

REPLY ↗

Make sure you have downloaded the jurassic_world.jpg image and placed it in the same directory as your `test.py`. You are getting this error because the `jurassic_world.jpg` image is not in the same directory as your Python script.

Free 21-day crash course on computer vision & image search engines

**mehul** July 8, 2015 at 1:07 am #

thank you , for the quick reply

REPLY ↗

**isaac vidas** July 14, 2015 at 11:08 pm #

REPLY ↗

Great tutorial! Thank you very much!

I had a few issues during the cmake step and the "make -j4" step. I don't know if it's because of changes in the installation or a different reason specific to my environment.

When I ran cmake I couldn't get python2 to be in the list of modules to be built.

After taking a look in "opencv\CMakeLists.txt" I found out that some of the parameters were changed:

- 1) PYTHON2_LIBRARY – changed to PYTHON2_LIBRARIES. The value stayed the same.
- 2) PYTHON2_INCLUDE_DIR – was removed and not being used.
- 3) PYTHON2_NUMPY_INCLUDE_DIRS – was added and should be pointed to the include dir of numpy.
- 4) BUILD_opencv_python2 – In the CMakeLists.txt this parameter is used in the condition of building

I ended up with the following command:

```
cmake -D CMAKE_BUILD_TYPE=RELEASE -D CMAKE_INSTALL_PREFIX=/usr/local \
-D PYTHON2_PACKAGES_PATH=~/virtualenvs/cv/lib/python2.7/site-packages \
-D PYTHON2_LIBRARIES=/usr/local/Cellar/python/2.7.10/Frameworks/Python.framework/Versions/ \
-D PYTHON2_NUMPY_INCLUDE_DIRS=~/virtualenvs/cv/lib/python2.7/site-packages/numpy/core \
-D INSTALL_C_EXAMPLES=ON -D INSTALL_PYTHON_EXAMPLES=ON \
-D BUILD_EXAMPLES=ON BUILD_opencv_python2=ON \
-D OPENCV_EXTRA_MODULES_PATH=~/opencv_contrib/modules ..
```

You also need to make sure that PYTHON2_NUMPY_INCLUDE_DIRS is correct. In my case, I didn't first, so it used the default value "/virtualenvs/cv/lib/python2.7/site-packages". While the cmake step following error: "fatal error: 'numpy/ndarrayobject.h' file not found".

I fixed this error by pointing the PYTHON2_NUMPY_INCLUDE_DIRS parameter to the include library "ndarrayobject.h" in this folder.

I've installed this on OS X Yosemite 10.10.4.

Hope this helps.

Free 21-day crash course on computer vision & image search engines

Interested in computer vision and image search engines, but don't know where to start? Let me help. I've created a free, 21-day crash course that is hand-tailored to give you the best possible introduction to computer vision. Sound good? Enter your email below to start your journey to becoming a computer vision master.

LET'S DO IT!

**Adrian Rosebrock** July 15, 2015 at 6:36 am #

REPLY ↗

Thanks so much for the tips Issac! Very interesting that the CMake options have already changed though.

**isaac vidas** July 15, 2015 at 11:42 am #

REPLY ↗

I think so too. It's also weird that there's so little documentation about it.

**Adrian Rosebrock** July 16, 2015 at 6:28 am #

REPLY ↗

Welcome to the world of OpenCV! But that's too be expected. OpenCV 3 is barely a month old, whereas OpenCV 2.4 was the major version for years. It's going to take awhile for good documentation on it to appear. I just hope that PyImageSearch can be one of those "good documentation" websites.

**isaac vidas** July 17, 2015 at 4:02 pm #

REPLY ↗

I think PyImageSearch is already one of those websites.

I've been returning to this site for a while now and I really like what you're doing with it. I have some ideas that I wanted to try with OpenCV and when I wanted to get started, I would always come here. Thank you for all the great tutorials and examples!

Free 21-day crash course on computer vision & image search engines

**Adrian Rosebrock** July 18, 2015 at 7:34 am #

That's really great to hear Isaac! 😊

**Moe** January 11, 2016 at 12:38 am #

REPLY ↗

Thank you so much Isaac, you have saved me from a lot of headaches :)!

**Jay Ludher** April 13, 2016 at 8:35 pm #

Hi! Sorry to bring back to life an old topic. I'm doing this new change as I also looked at. However, when I change to the new format and run cmake, I'm still having python 2 in my list of /usr/local/bin/python2.7 (ver 2.7.11) instead of pointing to Users/Ludhercris/.virtualenvs/cv/bin/

**Chris Alvino** May 6, 2016 at 4:22 pm #

Thanks for including this comment! It saved me a lot of time.

**Trozdzol** May 18, 2016 at 7:47 pm #

OMG! Thank you so much for posting this. I kept running cmake over and over thinking

**Pat1234** June 22, 2016 at 2:50 pm #

REPLY ↗

Thanks so much Isaac!

I have a problem regarding the "make -j4" step. I ran cmake and python2 is in the list of modules to be built.

When I run "make -j4" I get the reply "make: *** No targets specified and no makefile found. Stop."

I am very new to this so it's probably a beginners mistake.

Can anyone help me out?

Thanks!!

**Adrian Rosebrock** June 23, 2016 at 1:13 pm #

REPLY ↗

If you're getting an issue related to "No targets specified", then it's likely that your CMake command exited with errors. Double-check your output from CMake and you'll likely see errors that need to be resolved.

**Nic** August 4, 2016 at 4:44 am #

REPLY ↗

What a lifesaver. Thanks for the clear example.

However, though it almost worked for me, I still had to set INSTALL_C_EXAMPLES=OFF

**bhagyeshm** August 4, 2015 at 1:44 am #

REPLY ↗

Free 21-day crash course on computer vision & image search engines

Oops!! I realised that I am asking for "which python" instead of asking "which python3"..Thanks anyways Adrian for the lovely book 'Practical Python and OpenCV'! Everything is simply there at one place.



Adrian Rosebrock August 4, 2015 at 1:17 pm #

REPLY ↗

Glad it's working for you! And that's awesome that you are enjoying the book! 😊



Alex August 7, 2015 at 1:14 pm #

REPLY ↗

Thank you very much; this was well-needed!

Also thanks to Isaac Vidas, as I was having the same issue.

What would we do without these practical, yet passionate, tech blogs? When will proper documentation be available for OpenCV?

Cheers,
Alex



Adrian Rosebrock August 8, 2015 at 6:33 am #

Thanks for the kind words Alex! 😊



Xi Wang August 9, 2015 at 4:41 am #

Great tutorial! I followed exactly the same steps and got exactly the same results on my MacBook Air. Thank you so much for the accuracy!

Free 21-day crash course on computer vision & image search engines

Interested in computer vision and image search engines, but don't know where to start? Let me help. I've created a free, 21-day crash course that is hand-tailored to give you the best possible introduction to computer vision. Sound good? Enter your email below to start your journey to becoming a computer vision master.

Email Address

LET'S DO IT!



Adrian Rosebrock August 9, 2015 at 7:02 am #

REPLY ↗

Awesome, glad to hear it! 😊



Mike August 12, 2015 at 9:59 pm #

REPLY ↗

I ran into a problem where, after running make -j4, I got an error saying Python.h was not found.

The problem was that when I ran cmake, I used this option:

```
-D PYTHON2_INCLUDE_DIR=/usr/local/Frameworks/Python.framework/Headers
```

That file was an alias to a folder that didn't exist, for some reason. I just changed it to:

```
-D PYTHON2_INCLUDE_DIR=/usr/local/Cellar/python/2.7.10/Frameworks/Python.framework/Headers
```

and it worked.



Adrian Rosebrock August 13, 2015 at 7:04 am #

REPLY ↗

Awesome, thanks for the comment Mike!



Dan October 16, 2016 at 6:41 pm #

REPLY ↗

Hi Mike/Adrian, sorry to bring up an old topic, but I have tried both of those paths that you have suggested and I still keep getting a fatal error around 78% because Python.h cannot be found.

My Python.h file exists in System/Library/Frameworks/Python.frameworks/Headers.

Free 21-day crash course on computer vision & image search engines

Any suggestions on why this might not be working? I've run the Cmake command again a few times trying those paths out but nothing seems to be working. Thanks for your help!



Adrian Rosebrock October 17, 2016 at 4:04 pm #

REPLY ↗

Which version of Python are you using? And which version of the Mac OS?



Kusuma Ramesh August 13, 2015 at 1:40 am #

REPLY ↗

Hi

I am trying to install openCV3.0 and Python2.7 on OSX. After installing python from the brew, making which python on the terminal. It is showing the system version of python and not the python installed.



Adrian Rosebrock August 13, 2015 at 7:06 am #

After updating your `.bash_profile` file did you reload it using:

```
$ source ~/.bash_profile
```

Alternatively, you could launch a new shell and try `which python`

If it is still showing the system version of Python, then it's likely that your `.bash_profile` file w

Free 21-day crash course on computer vision & image search engines

Interested in computer vision and image search engines, but don't know where to start? Let me help. I've created a free, 21-day crash course that is hand-tailored to give you the best possible introduction to computer vision. Sound good? Enter your email below to start your journey to becoming a computer vision master.

LET'S DO IT!



cesar August 17, 2015 at 3:32 pm #

when i use:

```
git clone https://github.com/ltseez/opencv.git
```

i get:

Illegal instruction: 4

im using mac 10.7.5

can u help me, please?



Adrian Rosebrock August 18, 2015 at 6:49 am #

REPLY ↗

That sounds like an issue unrelated to OpenCV, but with git and your OSX install. I'm not quite sure why you would be getting that error (other than OSX 10.7 is pretty dated at this point). You can read more about it [here](#), but again, the issue is most certainly with your git install, not OpenCV.



Chris August 19, 2015 at 9:10 pm #

REPLY ↗

Hi Adrian, I just want to install OpenCV 3.0 on Mac to use with C++. I followed your instruction, but from Step 6 I exited the cv virtual environment and installed the rest globally, because I was afraid that it won't work with XCode (or any other C++ IDE) if I keep installing under the virtual environment. In the end I couldn't find the cv2.so anywhere (either in site-packages or build/lib). I haven't tested with C++ (Xcode) yet but just wanted to ask you if what I did could lead to any potential problem? Overall, how do I get it to work with both Python and C++? Thanks



Adrian Rosebrock August 20, 2015 at 7:01 am #

REPLY ↗

Hey Chris, I'll be honest — I don't do much C++ development, and when I do, it's on Linux. I can't say for sure if this will answer your question. But to answer your question, you should still follow the same install instructions as I did above.

Free 21-day crash course on computer vision & image search engines

command will install OpenCV globally on your machine. The only extra step is to sym-link the `cv2.so` file into your virtual environment. Other than that, you should be able to run Python scripts and compile C++ programs.



Chris August 20, 2015 at 5:33 pm #

REPLY ↗

Cool so this time I followed the exact instruction and I was able to build OpenCV 3.0 for both Python and C++ (Xcode). Thanks a ton. Even though I don't use Python, I'll follow your blog to learn more about features in OpenCV (rather than looking into OpenCV documentation).

Two quick questions though as I'm pretty new to python: (1) How to reactivate the cv virtualenv once I close the current terminal and start a new one? ; (2) Once your test_akaze.py is running from terminal, how do I close it? I tried a couple options (ESC, q, exit(), etc) but the only one that stops it is Ctrl + z, but then the python session is still running and in order to close the image window I have to force close it. Is there a better way? Thanks



Adrian Rosebrock August 21, 2015 at 7:16 am #

1. Use the `workon` command to access the `cv` virtual environment:

\$ `workon cv`

And that will drop you back into the `cv` environment

2. Click on the window first, then press any key — this will close the window.

Free 21-day crash course on computer vision & image search engines

Interested in computer vision and image search engines, but don't know where to start? Let me help. I've created a free, 21-day crash course that is hand-tailored to give you the best possible introduction to computer vision. Sound good? Enter your email below to start your journey to becoming a computer vision master.

Email Address

LET'S DO IT!



Chris August 21, 2015 at 1:11 pm #

Thanks Adrian, all work very well.



Noel García April 21, 2016 at 2:24 pm #

Thanks a lot!



Mark October 7, 2015 at 9:22 pm #

REPLY ↗

Hello Chris. Can you tell me what did you do exactly? I already can use Python+OpenCV to build programs, but it's not very clear how to build with C++ and Xcode. Thanks.



Floren December 11, 2015 at 4:37 am #

I followed these instructions to configure Xcode.

<https://www.youtube.com/watch?v=XJeP1juuHHY>

and my C++ test script worked.



bill September 2, 2015 at 9:55 pm #

REPLY ↗

After wiping an old home-brew install and resetting all the foobared permissions, I am getting the following error wen attempting to make opencv:

```
[ 79%] Linking CXX executable ../../bin/cpp-tutorial-pnp_registration
[ 79%] Built target cpp-tutorial-pnp_registration
[ 79%] Linking CXX executable ../../bin/opencv_perf_stitching
[ 79%] Built target opencv_perf_stitching
make: *** [all] Error 2
```

What is going on and how to fix it?

cmake complains about PythonLibs2.7, but doesn't see the libpython1.7.10 in the Cellar.
make notes that "libopencv_hal.a" has no symbols.

Thanks

oops: that is lib python2.7.10 in the cellar

Thanks Hackeron,

I also had the Python.h problem. Fixed it as suggested above and everything compiled and installed.

never mind. It was the Python.h problem. fixed that and everything worked.

 **Scott Squires** September 3, 2015 at 2:49 pm #

Tried to install twice now.

Same issues each time.

-D PYTHON2_LIBRARY=/usr/local/Cellar/python/2.7.10/Frameworks/Python.framework/Versions/2.

I had to change to python/2.7.10_2 since that was the folder created for some reason.

Any of the make files on cv fail errors in the video codecs. Complains about x86, etc.

In the end import cv2 doesn't work since it can't find cv2

Suggestions?

Free 21-day crash course on computer vision & image search engines

Interested in computer vision and image search engines, but don't know where to start? Let me help. I've created a free, 21-day crash course that is hand-tailored to give you the best possible introduction to computer vision. Sound good? Enter your email below to start your journey to becoming a computer vision master.

LET'S DO IT!

REPLY ↗



Adrian Rosebrock September 4, 2015 at 6:43 am #

If you are getting compile errors, then yes, OpenCV will not be able to finish compiling and install. The library must be compiled correctly and installed correctly before it can be imported.

Try changing your `PYTHON_LIBRARY` to point to the `.dylib` file so it can be linked against. The path on your machine will likely be different, but for me it's:

-D

`PYTHON2_LIBRARY=/usr/local/Cellar/python/2.7.8/Frameworks/Python.framework/Versions/2.7/lib/libpython2.7.dylib`

REPLY ↗



kota November 10, 2015 at 8:37 am #

Thanks Adrian. Previously I used the following but didn't work (everything worked but import cv2)
`PYTHON2_LIBRARY=/usr/local/Cellar/python/2.7.10/Frameworks/Python.framework/Versions/2.7/bin`

Then I changed it to `.dylib` and it worked now.

Since I couldn't find satisfying answer about what `.dylib` does, can you shed some light about why it works now?

REPLY ↗



Adrian Rosebrock November 11, 2015 at 6:38 am #

A dylib file is a dynamic library that's loaded at runtime rather than compile time (hence why your error only happened during the import). If you've ever done any Windows or DOS programming, dylibs are essentially parallel to DLL files.

REPLY ↗



Kuan September 3, 2015 at 3:28 pm #

Just wanted to chime in and say thanks for the very helpful tut.

Free 21-day crash course on computer vision & image search engines



Adrian Rosebrock September 4, 2015 at 6:40 am #

REPLY ↗

Thanks Kuan! 😊



Bernd Meyer September 5, 2015 at 2:10 am #

REPLY ↗

Despite Adrian's generous help I couldn't quite get this to work in my environment, but based on this and another blog I finally found the solution. If you, like me, are tortured by the "unsupported/Eigen/MatrixFunctions" file not found" bug, this solution may also work for you: <http://wp.me/P3ALAE-82>



Adrian Rosebrock September 5, 2015 at 5:24 am #

Thanks for sharing Bernd, I'm glad OpenCV is now installed for you 😊



Nick September 12, 2015 at 1:17 am #

Hi Adrian, I followed your instruction twice and checked every step, but I still could not get it running `sift = cv2.SIFT()` and I got this error: `AttributeError: 'module' object has no attribute 'SIFT'` (`imp /usr/local/include/opencv2` and there's no folder `nonfree` which suppose to include the header for SIFT output in the process that we can check if `opencv_contrib` is added. I did have the line `"-D OPENCV_3=ON` .." in the cmake command. Thanks



Adrian Rosebrock September 12, 2015 at 6:49 am #

REPLY ↗

Hey Nick, you might want to give my post on "[Where did SIFT and SURF go in OpenCV 3?](#)" a read. SIFT and SURF have both moved to the `xfeatures2d` sub-module so you can actually access them via: `cv2.xfeatures2d.SIFT_create()`



Nick September 12, 2015 at 6:09 pm #

REPLY ↗

Oh, thanks, should have asked you earlier. Appreciate it 😊



Adrian Rosebrock September 13, 2015 at 7:22 am #

REPLY ↗

No problem Nick 😊



SPQR September 14, 2015 at 4:33 am #

REPLY ↗

brew doctor now complains with :

Unexpected dylibs:

`/usr/local/lib/libopencv_adas.3.0.0.dylib`
`/usr/local/lib/libopencv_bgsegm.3.0.0.dylib`

...

`/usr/local/lib/libopencv_xobjdetect.3.0.0.dylib`
`/usr/local/lib/libopencv_xphoto.3.0.0.dylib`



Adrian Rosebrock September 14, 2015 at 6:10 am #

Free 21-day crash course on computer vision & image search engines

Did you install OpenCV via brew or from the source like detailed in this post? I haven't ran into a problem with brew doctor when compiling from source.



Ashwin September 23, 2015 at 10:06 pm #

REPLY ↗

I wanted know if I will be able to change the number of dedicated cores for OpenCV.

Say, I've already compiled OpenCV and mentioned "make -j4". But now, I want to change it to 2 cores. How do I do it?



Adrian Rosebrock September 24, 2015 at 6:59 am #

REPLY ↗

Hey Ashwin — I think there might be a bit of confusion. The `-j` option controls the number of cores that OpenCV will use once compiled, installed, and performing various operations further improved by compiling with TBB support.



Sarath September 29, 2015 at 12:43 pm #

Hi,

I tried to install opencv for python2.7. i end up with the error Python.h not found.
i checked my `PYTHON_INCLUDE_DIR` path, for me it is `/Library/Frameworks/Python.framework/Headers` present inside it.
i also have Header directory in `/Library/Frameworks/Python.frameworks/Versions/2.7/Headers`. which is there any way to find the correct path?

I did not install python using brew. does it matters? if, How could i remove preexisting python and install?

Please help me..

Free 21-day crash course on computer vision & image search engines

Interested in computer vision and image search engines, but don't know where to start? Let me help. I've created a free, 21-day crash course that is hand-tailored to give you the best possible introduction to computer vision. Sound good? Enter your email below to start your journey to becoming a computer vision master.

Email Address

LET'S DO IT!



Adrian Rosebrock September 30, 2015 at 6:32 am #

REPLY ↗

Yes, it does matter if you did not install Python via brew. If you do not install Python via brew, you will end up compiling against the OSX version of Python. That's not necessarily a bad thing, but you'll need to change your paths accordingly. You do not need to uninstall the system version of Python to install another version — just let brew take care of it for you like I have outlined in this blog post.



Me October 17, 2015 at 3:34 pm #

REPLY ↗

I followed all the steps and Igor it to work. The next time I restarted the computer I used 'workon cv' to switch to the virtual env, but the example code stopped working. A little digging in with 'pip list' showed that I lost the packages...

Any idea how to get the packages to persist between restarts?



Adrian Rosebrock October 18, 2015 at 7:06 am #

REPLY ↗

It sounds like you need to re-access your virtual environment:

```
1 $ workon cv
2 $ python test.py
```

Make sure you are always in the `cv` virtual environment before executing your Python script.



justin ng October 22, 2015 at 5:43 am #

REPLY ↗

Hi Adrian,

Free 21-day crash course on computer vision & image search engines

I have installed OpenCV 3.0.0 on Ubuntu but somehow when using python and check "cv2.__version__", it still returns "2.4.8".

Thus, I can not use some xfeature2d functions.

Could you suggest any ideas to get my python work with OpenCV 3.0.0



Adrian Rosebrock October 22, 2015 at 6:18 am #

REPLY ↗

That's quite strange. Have you previously installed OpenCV 2.4.8 on your system? If so, make sure you are in the **cv** virtual environment so you are seeing the OpenCV 3.0.0 bindings.



Vic October 27, 2015 at 12:48 am #

I attempted to create the virtual environment by:

mkvirtualenv cv,

but i got:

-bash: mkvirtualenv: command not found



Adrian Rosebrock October 27, 2015 at 4:47 am #

I would go back to Step 3 and make sure your `~/.bash_profile` file is properly edited.

Free 21-day crash course on computer vision & image search engines

Interested in computer vision and image search engines, but don't know where to start? Let me help. I've created a free, 21-day crash course that is hand-tailored to give you the best possible introduction to computer vision. Sound good? Enter your email below to start your journey to becoming a computer vision master.

Email Address

LET'S DO IT!



Felicia Amy July 27, 2017 at 6:20 pm #

Hi,

I also have problem with this. Somehow the `virtualenvwrapper.sh` is stored here `"/Library/Frameworks/Python.framework/Versions/2.7/bin/"` instead of `"/usr/local/bin/"`. Do you have any idea what went wrong in my case? Thank you.



hendrick August 2, 2017 at 10:34 am #

REPLY ↗

same problem with me. I already make sure my `~/.bash_profile`. but when I run `which python` the result is `usr/bin/python`



Adrian Rosebrock August 4, 2017 at 7:01 am #

REPLY ↗

Which version of Python and pip did you use to install virtualenv and virtualenvwrapper?



Pawan Ajagond March 28, 2016 at 10:51 am #

REPLY ↗

Use this line of code :

`source /usr/local/bin/virtualenvwrapper.sh`

and then this line of code:`workon cv`



Peter October 28, 2015 at 3:35 am #

REPLY ↗

Hey Adrian,

first I have to thank you for this amazing manual. Everything works fine right now and it was fun to get some background information as well.

After the install process there are two folders in the home directory: `~/opencv` and `~/opencv_contrib`

Free 21-day crash course on computer vision & image search engines

Are they necessary after installing? Can I delete them or what should I do with them?

Thanks and all the best from Germany

Peter



Adrian Rosebrock November 3, 2015 at 10:40 am #

REPLY ↗

Nice, I'm glad the tutorial worked for you Peter! 😊

After you run `make install` you can safely delete `opencv` and `opencv_contrib`.



Peter November 2, 2015 at 3:29 am #

Hey Adrain,

after the install process there are two folders in the home directory. `opencv` and `opencv_contrib`.

Are they necessary or can I delete them?

Best and thanks for the helpful manual.

Peter



Adrian Rosebrock November 3, 2015 at 10:17 am #

Yep! After you've ran `make install` you can safely delete the `opencv` and `opencv_contrib` folders.



Pai November 8, 2015 at 2:26 pm #

Super!

I'm able to install after following instruction given.

Thanks,

Pai

Free 21-day crash course on computer vision & image search engines

Interested in computer vision and image search engines, but don't know where to start? Let me help. I've created a free, 21-day crash course that is hand-tailored to give you the best possible introduction to computer vision. Sound good? Enter your email below to start your journey to becoming a computer vision master.

Email Address

LET'S DO IT!



Adrian Rosebrock November 8, 2015 at 3:18 pm #

REPLY ↗

Great, I'm happy the install instructions worked for you Pai! 😊



Mike Mehr November 17, 2015 at 1:46 am #

REPLY ↗

Hi Adrian,

I got this all working after I realized that my initial error (C compiler cannot create executables) in Cmake had to do with not having the Xcode Command Line tools (I had to look that one up on Google/StackOverflow). Everything else proceeded just fine, including the final test run. Except for not knowing how to quit out from the running Python script (lots to learn about Python!).

Can I now run the install for Python3 as well? Should I? I'll see if there are comments on that page about that issue.

Anyway, thanks for setting all this up and I'm looking forward to learning more with you.

— Mike

Hi Adrian,

I did do this page after doing the 2.7 install page; the brew install python3 got me v3.5.0, so I modified the CMAKE command accordingly, but after the make build, I could not find the cv2.so file anywhere, not in the path specified by Rick, nor the 3.5 path as you suggested. The old 2.7 package in the cv virtualenv is still there and still works under 2.7. I assume if it had put the 3.5-compiled .so file there, it would NOT run under 2.7, so I'm at a loss to explain where it might have gone. Any suggestions? I noted some differences in the output of CMAKE around the Python stuff, but the 3.5 stuff was called out properly.

Best,

Mike M

Free 21-day crash course on computer vision & image search engines

Further update (fixed the problem). I am not sure what step(s) below exactly fixed the problem above, but among the things I did were:

1. Made sure there was only one set of virtualwrapper.sh calls in my .bash_profile
2. Renamed /usr/local/bin/python2.7 temporarily so the old interpreter couldn't be found by CMAKE
3. Reran the CMAKE line, which then only found the 3.5 interpreter.
4. Reran make -j4 so that it was built with Python 3.5
5. Put the /usr/local/bin/python2.7 link back
6. Found the .so file at ~/opencv/build/lib/python3/cv2.cpython-35m-darwin.so and copied it to ~/virtualenvs/cv3/lib/python3.5/site-packages
7. Opened python3 and verified that import cv2 worked.

My clue was noticing that when it wasn't working, the make didn't actually link the .so file in the final step, only building the video-write tutorial. I became suspicious of the line of CMAKE output "- Python (for build): /usr/local/bin/python2.7" and thus my steps #2-4. Only took 2 hours extra! Is this a bug in the build script?

Regards,
Mike



Adrian Rosebrock November 17, 2015 at 6:17 am #

Hey Mike — congrats on getting OpenCV installed! And thanks for following up with all into a single one just for ease of readability). I'm sure other PyImageSearch readers will find your

As you noted, there should only be one set of virtualenv and virtualenvwrapper calls in your .bashrc with Python 2.7 and Python 3. It's also strange that you needed to manually move `python2.7` point to Python 3.5, this should not have happened. But then again, each OSX system is a bit different.

As far as the strange `cv2.so`, there must be a bug in the build script. It only happens for Python 3

Free 21-day crash course on computer vision & image search engines

Interested in computer vision and image search engines, but don't know where to start? Let me help. I've created a free, 21-day crash course that is hand-tailored to give you the best possible introduction to computer vision. Sound good? Enter your email below to start your journey to becoming a computer vision master.

Email Address

LET'S DO IT!



gyurisc November 19, 2015 at 7:25 am #

Thanks for posting this. It is very useful. I could not do the installation without your guide.. 😊



Adrian Rosebrock November 20, 2015 at 6:35 am #

REPLY ↗

No problem!



James December 6, 2015 at 8:11 am #

REPLY ↗

Hi Adrian

Thanks for the great tutorial!

I'm currently doing step 10 and restarted the terminal (for no particular reason), however I get this error:

```
1 James-MBP:~ james$ cd documents
2 James-MBP:documents james$ cd PythonProjects
3 James-MBP:PythonProjects james$ python test_akaze.py
4 Traceback (most recent call last):
5   File "test_akaze.py", line 3, in
6     import cv2
7 ImportError: No module named cv2
```

What's the best way around this?

Best wishes

James



Adrian Rosebrock December 6, 2015 at 11:03 am #

Free 21-day crash course on computer vision & image search engines

Hey James, if you ever open up a new terminal you need to use the `workon` command to re-access your `cv` virtual environment:

```
$ workon cv
```

This will drop you back down into the Python virtual environment and you'll be able to successfully import the OpenCV library.



James December 7, 2015 at 7:19 am #

REPLY ↗

Hey Adrian, thanks for your help and quick reply. Everything is working as it should now!



Adrian Rosebrock December 7, 2015 at 9:31 am #

Fantastic, glad to hear it James!



Daniele December 10, 2015 at 9:59 am #

Hi Adrian, thanks for the great guide! I installed it without virtualenv by skipping the `virtualenv PYTHON2_PACKAGES_PATH=/usr/local/lib/python2.7/site-packages`

It worked without any problem on El Capitan 10.11.1, Really thanks for all your work!



Adrian Rosebrock December 10, 2015 at 2:24 pm #

Nice, I'm glad it worked for you Daniele!

Free 21-day crash course on computer vision & image search engines

Interested in computer vision and image search engines, but don't know where to start? Let me help. I've created a free, 21-day crash course that is hand-tailored to give you the best possible introduction to computer vision. Sound good? Enter your email below to start your journey to becoming a computer vision master.

LET'S DO IT!



David Kadouch December 11, 2015 at 9:58 am #

REPLY ↗

As always this is wonderful tutorial for your winning series of tutorials. One comment is that in the `cmake` command the `PYTHON2_LIBRARY` and `PYTHON2_INCLUDE_DIR` paths for python2 in `cmake` didn't work for me, leading to errors during the build process (`make -j4`). I have python2.11 and not 2.10 as in your example. I found that you can modify the command by pointing everything to the virtual env. This way it's less dependent on how your physical env is configured in reality. Here's the modified command:

```
1 $ cmake -D CMAKE_BUILD_TYPE=RELEASE -D CMAKE_INSTALL_PREFIX=/usr/local \
2   -D PYTHON2_PACKAGES_PATH=~/virtualenvs/cv/lib/python2.7/site-packages \
3   -D PYTHON2_LIBRARY=~/virtualenvs/cv/bin \
4   -D PYTHON2_INCLUDE_DIR=~/virtualenvs/cv/include/python2.7 \
5   -D INSTALL_C_EXAMPLES=ON -D INSTALL_PYTHON_EXAMPLES=ON \
6   -D BUILD_EXAMPLES=ON \
7   -D OPENCV_EXTRA_MODULES_PATH=~/opencv_contrib/modules ..
```

HTH

David



Adrian Rosebrock December 11, 2015 at 11:18 am #

REPLY ↗

Thanks for passing along the updated command David!



Luca March 22, 2016 at 2:52 pm #

REPLY ↗

Thanks so much David! This was killing me for the longest time...

Piyush Santwani March 31, 2016 at 1:42 am #

Free 21-day crash course on computer vision & image search engines



Thank you David Sir.



Tsang-Kai October 31, 2016 at 2:21 pm #

REPLY ↗

This one works for me! Thank you so much, David.



Hilman December 12, 2015 at 7:37 am #

REPLY ↗

Hey, great tutorial. Just want to ask something. Sorry if it sounds fool because I am a beginner (*with spirit to master image processing!*)

1. I am using El Capitan and follow all of your instructions. Is it ok?
2. Why need to install python when in fact the os x come with installed python?



Adrian Rosebrock December 12, 2015 at 10:04 am #

Yes, you can certainly use El Capitan. And the reason you install a separate Python version is just in case anything goes wrong.



Hilman December 12, 2015 at 8:21 am #

I think I've done it. I just copied the cv2.so in the build.lib to the site-packages. Is it ok?



Adrian Rosebrock December 12, 2015 at 10:04 am #

REPLY ↗

Yes, that is perfectly okay.



Hilman December 12, 2015 at 7:24 pm #

REPLY ↗

Hey Adrian, I have another question.

Every time I want to use the opencv 3 with the python for my work, through the terminal, I must run the "mkvirtualenv cv" command first?



Adrian Rosebrock December 13, 2015 at 7:32 am #

REPLY ↗

You only need to run `mkvirtualenv cv` once. After that, you just need to run `workon cv` to access the `cv` virtual environment. You can read more about Python virtual environments [here](#).



Hilman December 14, 2015 at 12:08 am #

REPLY ↗

I've read it. Great article by the way :).

Just to make sure:

1. If I correctly follow the above steps during installing python 2.7+ And opencv 3, I just need to type in "workon cv" and no need to type other command like "source virt1/bin/activate" etc.

2. Before this I've already entered "mkvirtualenv cv" like many times to enter the cv environment as I didn't know the method above. Will it affects anything?

Adrian Rosebrock December 14, 2015 at 6:34 am #

Free 21-day crash course on computer vision & image search engines



1. Correct, once you've already gone through the installation steps you just need to use the `workon cv` command — that's all that is required to drop you down into the `cv` virtual environment.

2. This shouldn't affect anything. The `mkvirtualenv` command is "smart" enough to know that the virtual environment already exists and won't overwrite anything.



Adolfo December 15, 2015 at 8:43 pm #

REPLY ↗

Hi. Thanks very much for the tutorial!

I did everything and it's working but it seems that the SVM module is not included. Anyone else has that problem?



Robert Joseph December 22, 2015 at 3:26 pm #

REPLY ↗

Adrian, thank you so much for your walkthrough! I need `libopencv_contrib.dylib` to be built. In `cmake -DOPENCV_ENABLE_NONFREE=ON ..` (inside `opencv_contrib`) I see cmake output reports that contrib is not being built. gist: <https://goo.gl/ENWUJL>

Does this make any sense to you? Am I missing something obvious? Any help would be MUCH appreciated!



Adrian Rosebrock December 23, 2015 at 6:38 am #

REPLY ↗

Hey Robert, thanks for using a gist, that really helps improve the readability of the output. I'll take a look at that and get back to you.

Your output of CMake is actually correct. Modules such as `xfeatures2d`, `ximgproc`, etc. are not built by default. You can safely go ahead and compile OpenCV and the additional contrib package will be installed.

The `world` and `contrib_world` are the old (unneeded) OpenCV bindings.

Free 21-day crash course on computer vision & image search engines

Interested in computer vision and image search engines, but don't know where to start? Let me help. I've created a free, 21-day crash course that is hand-tailored to give you the best possible introduction to computer vision. Sound good? Enter your email below to start your journey to becoming a computer vision master.

Email Address

LET'S DO IT!



Bangor December 23, 2015 at 5:24 pm #

REPLY ↗

Thanks for the article, I'd have taken years to figure this out on my own!

Random question from a complete amateur who decided to get ambitious: Any idea why CMake would fail with:

CMake Error at samples/gpu/CMakeLists.txt:100 (list):

list sub-command REMOVE_ITEM requires list to be present.

When checking out the just released OpenCV 3.1.0?

Following the instructions and checking out 3.0.0 instead works perfectly, so I'll be using that!



Adrian Rosebrock December 23, 2015 at 6:46 pm #

REPLY ↗

I personally have not tried to compile OpenCV 3.1 yet, so I'm not sure about this error message. I will look into it.



shomz December 24, 2015 at 3:41 pm #

REPLY ↗

Hi, Adrian! I was wondering what steps should be skipped and what should be changed to install without the virtual environment. I tried to install with it, but I am a beginner and probably made a mistake along the way, because it did not work in the end. I plan to use PyCharm, is that ok? Thank you!



Adrian Rosebrock December 25, 2015 at 12:32 pm #

Free 21-day crash course on computer vision & image search engines

You can certainly use PyCharm with OpenCV. Please see [this post](#) for more information, although it does assume you are using a Python virtual environment.

As for the steps required to *not* use the virtual environment, simply skip the installation of `virtualenv` and `virtualenvwrapper` along with the `mkvirtualenv` and `workon` commands.



Sanjay December 30, 2015 at 1:17 pm #

REPLY ↗

After struggling through old documentation on OpenCV this is a breather with new and updated versions. Thank you!!



Adrian Rosebrock December 30, 2015 at 2:47 pm #

Awesome, I'm glad the instructions worked for you Sanjay! 😊



Jackson Isaac January 1, 2016 at 2:43 am #

Hi,

Nice tutorial. With 3.1.0 version there is an issue in `samples/gpu/CMakeLists.txt` line 10.

Hence while building 3.1.0 version I had to drop `-D INSTALL_C_EXAMPLES=ON`. Didn't try any o

Free 21-day crash course on computer vision & image search engines

Interested in computer vision and image search engines, but don't know where to start? Let me help. I've created a free, 21-day crash course that is hand-tailored to give you the best possible introduction to computer vision. Sound good? Enter your email below to start your journey to becoming a computer vision master.

Email Address

LET'S DO IT!



Adrian Rosebrock January 1, 2016 at 7:23 am #

Thanks for the tip Jackson! I have confirmed this issue as well.



Rohan January 6, 2016 at 10:56 pm #

REPLY ↗

I couldn't install it the first time around.

After removing all of the brew and pip installations from their respective directories, I removed python 2.7.

I re-installed with brew and made sure 'brew doctor' didn't show any errors.

Then, the installation worked properly.



Adrian Rosebrock January 7, 2016 at 6:37 am #

REPLY ↗

Thanks for sharing your experience Rohan!



Moe January 11, 2016 at 12:37 am #

REPLY ↗

Hey Adrian, thank you so much for this great tutorial.

Two questions:

First, in step 9, how do I get the (cv) before your computer name in the terminal? Is it necessary to always run "mkvirtualenv cv"? Also, am I understanding correctly that the (cv) is called the virtual environment?

Second, in the very last step while testing. How do I terminate/exit the process after running "test_akaze.py" properly? Pressing ctrl-C does not work as it simply prints "^C" in the terminal, also the Python output has no exit button. Should I simply quit the Python window with cmd+Q?

Thank you once again for what you do!

Free 21-day crash course on computer vision & image search engines



Adrian Rosebrock January 11, 2016 at 6:35 am #

REPLY ↗

Hey Moe, to answer your questions:

1. The `mkvirtualenv` command only needs to be run once. After that, you just need to run `workon cv` to drop into the `cv` virtual environment.
2. Click on the window and press any key. This will exit the Python script.



Moe January 11, 2016 at 3:44 pm #

REPLY ↗

Thanks for the reply, Adrian.

If I ran “mkvirtualenv cv” multiple times, what are the consequences? Does it just rebuild or something because I ran it a second time before your reply.

Thanks :)



Adrian Rosebrock January 12, 2016 at 6:33 am #

Running `mkvirtualenv cv` multiple times won't hurt anything — it's smart enough to see if it exists and will not overwrite it. It just drops you down into the environment.



Vihbor January 11, 2016 at 5:49 am #

file `cv2.so` doesn't exit..what to do?

Free 21-day crash course on computer vision & image search engines

Interested in computer vision and image search engines, but don't know where to start? Let me help. I've created a free, 21-day crash course that is hand-tailored to give you the best possible introduction to computer vision. Sound good? Enter your email below to start your journey to becoming a computer vision master.

Email Address

LET'S DO IT!



Adrian Rosebrock January 11, 2016 at 6:36 am #

REPLY ↗

There are many reasons why the `cv2.so` file might not show up. I would suggest taking a look at the Troubleshooting section of this post.



mandysmoak January 25, 2016 at 2:35 pm #

REPLY ↗

I am having the same issue. Unfortunately, I found no solutions on the suggested Troubleshooting section.

Any other suggestions??



Adrian Rosebrock January 25, 2016 at 4:00 pm #

REPLY ↗

As mentioned in the “Troubleshooting” section, it's really hard to diagnose why OpenCV may fail to import. The best suggestions are to (1) ensure you are in the `cv` virtual environment prior to trying to import OpenCV and (2) ensure that OpenCV compiled without any errors.



mandysmoak January 26, 2016 at 11:20 am #

REPLY ↗

Found my problem! I didn't notice that `python2` was in the Unavailable section, rather than the `OpenCV` after the `cmake` command. I used the one provided by isaac above and everything works!

Thank you so much for this tutorial!



Adrian Rosebrock January 26, 2016 at 5:52 pm #

Free 21-day crash course on computer vision & image search engines

Congrats on resolving the issue!

REPLY ↗

**aaz15** January 18, 2016 at 5:12 am #

Hello Adrian,

I just crossed your blog and its packed with excellent resources. Thank you for your guide.
While testing, I came accross this error

```
AttributeError: 'module' object has no attribute 'AKAZE_create'  
any ideas why?
```

**Adrian Rosebrock** January 18, 2016 at 3:24 pm #

Which version of OpenCV are you using? It works under v3.0.0.

**isunchy** March 13, 2016 at 4:54 pm #

I came across the same error.

What should I test opencv under OpenCV v3.1.0?

**versionHell** March 25, 2016 at 11:12 am #

Hi. I also had this error. I was able to successfully run the sample after doing the follow

1. I had originally built 3.1.0, so I then built 3.0.0
- 1a. I have python 2.7.11, so I had to change this line:

```
PYTHON2_LIBRARY=/usr/local/Cellar/python/2.7.11/Frameworks/Python.framework/Versions/2.7/bin
```

2. It still wasn't working, but I determined it was because I was running python outside of the virtual environment (my desktop), which is linked to openCV 2.4 instead of 3

3. copying the sample code and image to the .virtualenvs/cv/lib/python2.7 folder, then running it from there worked as expected

Free 21-day crash course on computer vision & image search engines

Interested in computer vision and image search engines, but don't know where to start? Let me help. I've created a free, 21-day crash course that is hand-tailored to give you the best possible introduction to computer vision. Sound good? Enter your email below to start your journey to becoming a computer vision master.

**Adrian Rosebrock** March 27, 2016 at 9:17 am #

REPLY ↗

Once you have successfully created the virtual environment, you *do not* need to put code directly in the `.virtualenvs/cv/lib/python2.7` folder. Instead, just use the `workon` command to access the virtual environment and then execute your Python script:

```
1 $ workon cv
2 $ python your_script.py
```

**Nate** January 24, 2016 at 2:45 pm #

REPLY ↗

Thank you for this.

**Adrian Rosebrock** January 25, 2016 at 4:07 pm #

REPLY ↗

I'm glad it worked for you Nate! 😊

Jiezhi January 31, 2016 at 1:37 am #

Free 21-day crash course on computer vision & image search engines



Great!

It helps me a lot!

**Adrian Rosebrock** January 31, 2016 at 8:56 am #

REPLY ↗

I'm glad to hear it Jiezhil! 😊

**Rahul** February 1, 2016 at 3:20 pm #

REPLY ↗

thank you for the wonderful setup instructions..

**Adrian Rosebrock** February 2, 2016 at 10:31 am #

REPLY ↗

No problem, I'm happy the install instructions worked for you Rahul!

**Gani Siva Kumar** February 4, 2016 at 9:24 am #

REPLY ↗

Thank you, I had successfully installed opencv. How can I add scipy now to virtualenv to work?

I installed scipy through "pip install scipy". But it showing import error while working in virtualenv.

**Adrian Rosebrock** February 4, 2016 at 9:30 am #

REPLY ↗

You need to access the `cv` virtual environment and then use pip to install it:

```
1 $ workon cv
2 $ pip install scipy
```

**Gani Siva Kumar** February 4, 2016 at 12:43 pm #

REPLY ↗

Thanks Adrian! It worked =D

**Niko** February 10, 2016 at 5:22 am #

REPLY ↗

I just installed a fresh copy of the newest el capitano version.

Im using python version 2.7.11 and OpenCV Version 3.1.0. And yes i set "INSTALL_C_EXAMPLES=OFF -D" so CMake was able to compile without any errors.

I followed the tutorial till step 9 i had no issues. But i can't find the "cv2.so" file, the openCV bindings in my "~/.virtualenvs/cv/lib/python2.7/site-packages/" directory. I've been using "workon cv" . Of course i cant sym-link cv2.so and python error : "ImportError: No module named cv2". The troubleshooting post from <https://www.pyimagesearch.com/2015/10/26/how-to-install-opencv-3-on-raspbian-jessie/> doesn't solve the problem.

Which is the best workaround for now ?

I don't know why cv2.so was not created.

Greeds Niko

**Adrian Rosebrock** February 10, 2016 at 4:35 pm #

REPLY ↗

Can you confirm that the `make` command successfully compiled OpenCV? If so, check if `cv2.so` file should be in there. Once you find it, you can move it/sym-link it into the virtual environment.**Free 21-day crash course on computer vision & image search engines**

REPLY ↗

**at0mb0y** February 19, 2016 at 2:14 am #

Hi, as some other I have trouble to install openCV. I'll not bug the comment with my bash output but can you be clearer at the step 6. Should we brew install while we are in the virtualenv cv ? or outside ? it's not clearly explain we we are in or out.
Thanks for your post

**Adrian Rosebrock** February 19, 2016 at 6:48 am #

REPLY ↗

The `brew` package should have been installed way back in Step 2, long before you install `virtualenv`. Once you've installed `brew`, you then proceed with the tutorial.

**at0mb0y** February 29, 2016 at 10:44 pm #

Sorry that's not what I mean.
At the step 6 we are in the virtual env or out ?
should we type at sometimes "deactivate" to get out of virtualenv before step 6 ?

**Adrian Rosebrock** March 1, 2016 at 3:44 pm #

No, do not leave the virtual environment. It is important that you stay inside the virtual environment so that Python picks up the correct Python version.

Free 21-day crash course on computer vision & image search engines

Interested in computer vision and image search engines, but don't know where to start? Let me help. I've created a free, 21-day crash course that is hand-tailored to give you the best possible introduction to computer vision. Sound good? Enter your email below to start your journey to becoming a computer vision master.

LET'S DO IT!

REPLY ↗

**Lirone** February 22, 2016 at 8:33 am #

Thank you !
So just to be sure, If I already have in my laptop OpenCV 2.4, It won't be overwritten right ? I can use both on my computer without problem (meaning without create any conflict ?)

**Adrian Rosebrock** February 22, 2016 at 4:21 pm #

REPLY ↗

Technically yes, but sometimes there are gremlins in the system. I normally only run `sudo make install` for one of the installations, from which I then run a compile for each OpenCV + Python version and then delete everything but the `code/build` directory. I then create a virtual environment that points to each respective version.

**Lirone** February 23, 2016 at 10:24 am #

REPLY ↗

Thank you Adrian for your answer.
I made a silly mistake. In the step 8 `PYTHON2_LIBRARY=/usr/local/Cellar/python/2.7.10/Frameworks/Python.framework`
I didn't notice that I have python 2.7.11 installed, therefore it didn't work when I wanted to import cv2.
I replaced 2.7.10 with 2.7.11, re make, and it worked. 😊

**Adrian Rosebrock** February 23, 2016 at 3:19 pm #

REPLY ↗

Congrats on resolving the issue Lirone 😊

Free 21-day crash course on computer vision & image search engines



Zakai February 26, 2016 at 9:53 pm #

REPLY ↗

I'm not sure if my first question is going to post or not, but I have resolved it. Now I am having another problem where after the build has compiled and I go to install it the cv2.so file isn't located in my site-packages directory. It shows up in my opencv/build directory, so I believe the install goes correctly but is not linking with the cv virtualenv and python.



Adrian Rosebrock February 27, 2016 at 10:16 am #

REPLY ↗

Hey Zakai — I'll look into the spam filter to see if your first comment got caught there. In any case, you should check the `dist-packages` directory as well. Finally, if it's still not there, then manually copy the `cv2.so` file from `build` into the `site-packages` directory of your virtual environment.



Akhila February 28, 2016 at 3:32 pm #

Hello!

Thank you for the great tutorial! I am interested in using openCV 3.1.0, specifically the DNN module. I am able to install 3.1.0. I know the contributions were installed because "cv2.xfeatures2d.SIFT_create()"

Do you have any thoughts on how I can use it?

Thanks!

Akhila

Free 21-day crash course on computer vision & image search engines

Interested in computer vision and image search engines, but don't know where to start? Let me help. I've created a free, 21-day crash course that is hand-tailored to give you the best possible introduction to computer vision. Sound good? Enter your email below to start your journey to becoming a computer vision master.

LET'S DO IT!



Adrian Rosebrock February 29, 2016 at 3:30 pm #

REPLY ↗

I personally haven't used the DNN sub-module of OpenCV yet. My understanding is that the DNN module is a placeholder to load and run pre-trained Caffe or Torch networks. It's not actually used to currently *train* a network. I also don't think this module is exposed to the Python bindings (yet).



Vijay Kamarashi February 29, 2016 at 2:39 pm #

REPLY ↗

Hi Adrian,

Not sure what I did, but I deleted the terminal instance, launched a new instance, did "workon cv" and reran cmake (with python includes and libraries pointing to the virtualenvs environment as suggested by some helpful person above). And cmake was able to configure correctly this time. Not sure why it failed first time around. By the way, for people not so familiar with cmake, I found it best to delete the build directory and create a fresh one everytime you make a new cmake attempt. Just my 2c to add to the "folklore".

Regards,

Vijay



Adrian Rosebrock February 29, 2016 at 3:25 pm #

REPLY ↗

Thanks for the input Vijay, it's much appreciated! 😊



Jose Sotelo March 10, 2016 at 2:14 am #

REPLY ↗

Hello. I'm following your tutorial and everything was perfect until Step 10. I'm pretty new into python, OpenCv and programming at all, so I can't create a new file.... It must be inside Xcode, the terminal or in a text file? I'm lost :(((

Thank you for any help

Free 21-day crash course on computer vision & image search engines



Adrian Rosebrock March 10, 2016 at 12:01 pm #

REPLY ↗

You can create the sample Python script in whatever text editor you would like. That could be XCode, Sublime Text, nano, vim, emacs, TextEdit, etc. Once you have created the Python script, open up your Terminal and execute the script.



Rob March 15, 2016 at 10:26 am #

REPLY ↗

Thank you for this tutorial!

In case it helps anyone:

In my case "make -j2" (I have a dual core MBP) failed with "fatal error: 'hdf5.h' file not found". In cmake I noticed that it was searching /Users/rob/anaconda/... (my previous Python installation used Anaconda) for HDF5.

Even though my .bash_profile had the right path and I ran source, I noticed that cmake was still finding

I did two things at once so I'm not sure which fixed it:

First, I updated Anaconda by running "conda update conda" and it turns out my Anaconda installation even though "which python" was returning the Homebrew installation, I completely removed the anaconda and created a new virtual environment.

I recompiled and there were no errors.

Free 21-day crash course on computer vision & image search engines

Interested in computer vision and image search engines, but don't know where to start? Let me help. I've created a free, 21-day crash course that is hand-tailored to give you the best possible introduction to computer vision. Sound good? Enter your email below to start your journey to becoming a computer vision master.

Email Address

LET'S DO IT!

REPLY ↗



Adrian Rosebrock March 15, 2016 at 4:26 pm #

Thanks for sharing your experience Rob, that's super helpful!



Ernie April 29, 2016 at 9:05 pm #

REPLY ↗

Interesting — I'm having the same issue, yet I don't have Anaconda installed 😞

No idea how to fix it.



Sujit Pal October 31, 2016 at 8:23 pm #

REPLY ↗

Six months too late for Ernie, but maybe someone else has this problem and stumbles upon this comment...

I was trying to install OpenCV 3.1.0 with Anaconda Python 2.7.12 using Adrian's guide. For the cmake, I skipped setting the PYTHO2_* variables hoping they will get set to the Anaconda defaults. During "make -j4", I had the same problem with the missing hdf5.h that is mentioned in the thread above.

I was ultimately able to fix it with the hack proposed by @avtomatons in OpenCV Issues #6016 (and replicated in #6050), ie, adding the following lines to the OPENCV_HOME/modules/python/common.cmake.

```
find_package(HDF5)
include_directories(${HDF5_INCLUDE_DIRS})
```

Also, many thanks to Adrian for writing this awesome how-to for installing OpenCV 3.x on Python (with SURF and SIFT). Just verified that I can create a SURF object in my REPL like so:

```
>>> import cv2
>>> surf = cv2.xfeatures2d.SIFT_create()
>>> surf
```



Adrian Rosebrock November 1, 2016 at 8:53 am #

REPLY ↗

Thanks for sharing Sujit!

Free 21-day crash course on computer vision & image search engines

REPLY ↗

**Anton** March 16, 2016 at 5:38 pm #

Hi Adrian,

Thanks so much for this – this was tremendously helpful. I was able to successfully carry out the OpenCV 3.1.0 installation on my Mac, but I'm having a problem as I now need to find cv2 equivalents to certain (older) cv functions and I'm struggling mightily to figure out what they might be. In particular, I'm looking for cv2 equivalents for:

```
cv.FindStereoCorrespondenceGC()
cv.GetReal2D()
cv.CreateStereoGCState()
```

I'm looking for these too, but they're less important:

```
cv.Set2D()
cv.LoadImage() (here I can probably use imread())
cv.CreateMat()
cv.ConvertScale()
```

More concretely, what I'm trying to do is create a disparity map using the Graph Cuts (GC) implementation in OpenCV 3? If so, is there any way to access the older functions?

I'm willing to try the C++ route (as opposed to Python), if need be.

Thanks very much for your help!

Free 21-day crash course on computer vision & image search engines

Interested in computer vision and image search engines, but don't know where to start? Let me help. I've created a free, 21-day crash course that is hand-tailored to give you the best possible introduction to computer vision. Sound good? Enter your email below to start your journey to becoming a computer vision master.

LET'S DO IT!

**Adrian Rosebrock** March 17, 2016 at 10:38 am #

Hey Anton, thank sfor the comment. To be honest with you, I do very little work with stereo vision functions, but I'm pretty sure [this is a resource that would help out](#).

As for `cv.Set2D`, images are represented as NumPy arrays in OpenCV 2.4 and OpenCV 3. To set a pixel value, simply access and set a pixel value: `image[y, x] = (255, 255, 255)`

The `cv.CreateMat` function is also not needed since we can use NumPy to allocate memory for an image: `image = np.zeros((height, width), dtype="uint8")`

Finally, the `cv.ConvertScale` function is not known as `cv2.convertScaleAbs`.

I hope that helps! And if you decide you want to learn more about the basics of OpenCV 2.4/3+ along with Python, be sure to take a look at [Practical Python and OpenCV](#), which will help get you up to speed quickly.

REPLY ↗

**Anton** March 28, 2016 at 3:42 pm #

Hi Adrian,

Thanks a bunch for your reply and tips. I was actually able to find and compile a pure C++ implementation of Graph Cuts which approximates the functionality of `cv.FindStereoCorrespondenceGC()`, so my problem is solved! The implementation is here, thanks to a French CV scholar called Pascal Monasse:

<http://www.ipol.im/pub/art/2014/97/>

The results are frankly tremendous (and a demo is provided.) Again, the power of Open Source shines through 😊

Again, thanks for your great site!

REPLY ↗

**Adrian Rosebrock** March 28, 2016 at 6:22 pm #

Very nice, thanks for sharing Anton!

REPLY ↗

**Cehasli** March 18, 2016 at 8:56 pm #

Thank you for sharing, Adrian. This article help me a lot 😊

Free 21-day crash course on computer vision & image search engines



Adrian Rosebrock March 19, 2016 at 9:12 am #

REPLY ↗

Happy I could help! 😊



mrfksiv March 21, 2016 at 1:06 pm #

REPLY ↗

Thanks for this extremely concise tutorial! The AKAZE example works perfectly? But what is this anyway?:p



Adrian Rosebrock March 21, 2016 at 6:32 pm #

AKAZE is a type of keypoint and local invariant descriptor. We often use it to describe recognizing the covers of book, which I cover inside [Practical Python and OpenCV](#).



Ade March 24, 2016 at 5:40 pm #

Thanks Adrian for the excellent writeup.

For those struggling with getting Python 3.5 to work with OpenCV 3, I was able to make it work by cloning

<http://peekay.org/2016/03/24/opencv-osx-python-3-bindings/>



Adrian Rosebrock March 27, 2016 at 9:37 am #

Thanks for sharing Ade! 😊



David Sterling April 3, 2016 at 7:33 pm #

REPLY ↗

Thanks for the fantastic write-up Adrian — I followed your instruction but built OpenCV 3.1.0 on a fresh install of El Capitan 10.11. Everything went smoothly and Python bindings seem to work so far.

For anyone interested in the Matlab bindings the openCV 3.1 "contrib" installation doesn't seem to be quite ready for prime time. The matlab compiler/code generator is only compiling *some* of the cpp files needed to implement the matlab bindings. Specifically line #30 in compile.cmake located in opencv_contrib/modules/matlab

```
file(GLOB SOURCE_FILES "${CMAKE_CURRENT_BINARY_DIR}/src/*.cpp")
```

should be something like:

```
file(GLOB SOURCE_FILES "${CMAKE_CURRENT_BINARY_DIR}/src/*.cpp" "${CMAKE_CURRENT_BINARY_DIR}/src/private/*.cpp")
```

so that the cpp files in build/modules/matlab/src/private are also compiled. Without these "Bridge" files some of the matlab bindings still work, but others (e.g. VideoCapture) produce errors about undefined "Bridge" functions) like this:

```
>> camera = cv.VideoCapture();
```

Undefined function or variable 'VideoCaptureBridge'.

Error in cv/VideoCapture (line 15)

```
this.ptr_ = VideoCaptureBridge('new', varargin{:});
```

A second issues with opencv_contrib/modules/matlab is that the compiler settings for Xcode 7.2.1

Apple LLVM version 7.0.2 (clang-700.1.81)

Target: x86_64-apple-darwin14.5.0

Thread model: posix

aren't quite right as there is an issue with c++11 standards and resulting collision between cmath and math.h that causes some of the cpp files in the source/private to throw compiler errors. Maybe this is why they weren't included in the first place??

Free 21-day crash course on computer vision & image search engines

Interested in computer vision and image search engines, but don't know where to start? Let me help. I've created a free, 21-day crash course that is hand-tailored to give you the best possible introduction to computer vision. Sound good? Enter your email below to start your journey to becoming a computer vision master.

LET'S DO IT!



Adrian Rosebrock April 4, 2016 at 9:29 am #

REPLY ↗

Thanks for the detailed comment David! I'm not much of a MATLAB users, but I know other PyImageSearch readers are and this comment is super helpful.



Matt Lashinsky April 9, 2016 at 7:56 pm #

REPLY ↗

Awesome, got it to work!! Dude, you're such a great teacher.

I had some issues setting up the build and ran into issues during compilation, but that's because I installed openCV in a different directory rather than the home directory. So it's important to be really diligent about specifying those paths to make sure they'

Thanks Adrian.



Adrian Rosebrock April 13, 2016 at 7:10 pm #

Thanks for the kind words Matt, I'm happy the tutorial was able to help!



João Crav April 13, 2016 at 3:40 pm #

How can I open the `~/.bash_profile`? very noob here



Adrian Rosebrock April 13, 2016 at 6:47 pm #

You can open the `~/.bash_profile` file using your favorite text editor. If you're just getting used to using the command line, then I highly recommend using nano:

```
$ nano ~/.bash_profile
```

Free 21-day crash course on computer vision & image search engines

Interested in computer vision and image search engines, but don't know where to start? Let me help. I've created a free, 21-day crash course that is hand-tailored to give you the best possible introduction to computer vision. Sound good? Enter your email below to start your journey to becoming a computer vision master.

LET'S DO IT!



Michiel April 19, 2017 at 4:35 pm #

REPLY ↗

I have the same problem as above, I use Sublime text 2, after which i type the '\$ nano ~/.bash_profile' tag at the line which i find by unhiding the console. I'm very new to this, any idea where I am going wrong?



Adrian Rosebrock April 21, 2017 at 11:01 am #

REPLY ↗

Hey Michiel — you should be typing `nano ~/.bash_profile` into your terminal, not Sublime Text.

If you are very new to computer vision, OpenCV, and the terminal, I would suggest working through my book, [Practical Python and OpenCV](#) which will teach you the basics.

Furthermore, the Quickstart Bundle and Hardcopy Bundle of Practical Python and OpenCV include a pre-configured Ubuntu VirtualBox virtual machine with OpenCV pre-installed. This will allow you to skip the installation step and start learning immediately. Be sure to take a look!



CC April 19, 2016 at 4:16 am #

REPLY ↗

Thank you for this nice tutorial. I use OSX 10.10. A little bit of feedback:

1. ffmpeg is also needed. (easy installation by brew)

2. I got some errors and linking errors when I didn't set

```
-D CMAKE_OSX_SYSROOT=/Applications/Xcode.app/Contents/Developer/Platforms/MacOSX.platform/Developer/SDKs/MacOSX.sdk
WITH_QUICKTIME=OFF -D BUILD_opencv_hdf=OFF
```

Free 21-day crash course on computer vision & image search engines

, as a reference for people who can't successfully install opencv by following this tutorial.



Adrian Rosebrock April 19, 2016 at 6:50 am #

REPLY ↗

Interesting, I personally haven't ran into an issue where FFMPEG was required. Were you getting an error message at install related to FFMPEG?



Hilman April 26, 2016 at 6:30 am #

REPLY ↗

Hey Adrian, I need to ask something.

I already installed Python 2.7 with OpenCV 3.0 using this tutorial, but now I will need to upgrade to CUDA. I have the opencv_contrib. How can I do that? Do I need to delete anything first?

And because of problems with Matplotlib before this, I also think I need to download back the Python 2.7 version.

Thanks.



Adrian Rosebrock April 26, 2016 at 5:12 pm #

REPLY ↗

You don't need to revert your Python version, just use an old version of matplotlib. I recommend 1.5.1.

As for uninstalling OpenCV, you *should* be able to delete all `libopencv*` files from `/usr/local/lib` (the OpenCV bindings (i.e., the `cv2.so` file)).



Hilman April 26, 2016 at 6:11 pm #

REPLY ↗

Thanks! As for the matplotlib issue, it is solved! Thank you so much! This makes me loves this blog even more! Keep up the good work of educating us! 😊



Adrian Rosebrock April 28, 2016 at 3:26 pm #

REPLY ↗

Awesome, I'm glad to hear it's resolved! 😊



Hilman April 26, 2016 at 6:25 pm #

REPLY ↗

About uninstalling the OpenCV, can i just `sudo make uninstall` it? I've read that this is possible.



Adrian Rosebrock April 28, 2016 at 3:27 pm #

REPLY ↗

I personally have never done this so I can't say whether or not it would work. However, this would imply that you kept your original Makefile generated from CMake. Most people delete these after OpenCV has been compiled and installed.



Samvaran April 26, 2016 at 6:26 pm #

REPLY ↗

Thank you for the great tutorial! It was very clear and incredibly helpful – there's nothing else like this that I could find on the web.

I was able to build everything successfully and install everything just fine. My question is – if I want to redo the build and installation (e.g. if I want to install CUDA files and compile it with those added), and I repeat these steps (i.e. cmake and make), will anything bad happen given that all the files have already been installed? For instance, is there any kind of "uninstall" necessary, or can I just repeat these steps?

Free 21-day crash course on computer vision & image search engines



Adrian Rosebrock April 28, 2016 at 3:25 pm #

REPLY ↗

If you decide to recompile and re-install OpenCV, then nothing bad *should* happen. You'll simply overwrite the existing files with the new ones.



ryan April 28, 2016 at 4:09 pm #

REPLY ↗

thank you for the tutorial.

i'm getting this error when running the script though:

```
File "test_akaze.py", line 15, in
detector = cv2.AKAZE_create()
AttributeError: 'module' object has no attribute 'AKAZE_create'
```

any help would be appreciated.



Adrian Rosebrock April 30, 2016 at 4:08 pm #

REPLY ↗

Hey Ryan — can you confirm which version of OpenCV you installed? Also, it would be helpful if you could post your code above. Both of them mentioned similar issues.



Lucky Dube April 29, 2016 at 5:21 am #

REPLY ↗

Hello Adrian, I'm a beginner and your tutorial was very useful. I managed to get all the way through it without any issues. However, I'm having trouble running the code. I keep getting an error message that says "aaz15" above. Both of them mentioned similar issues.

Thanks for your help

Free 21-day crash course on computer vision & image search engines

Interested in computer vision and image search engines, but don't know where to start? Let me help. I've created a free, 21-day crash course that is hand-tailored to give you the best possible introduction to computer vision. Sound good? Enter your email below to start your journey to becoming a computer vision master.

LET'S DO IT!



Adrian Rosebrock April 30, 2016 at 4:00 pm #

REPLY ↗

All you need to do is download the image and the code. You can copy and paste the code into your favorite text editor and save the file. Both the code and the image should be saved to the same folder on your computer.



stefan April 29, 2016 at 11:40 pm #

REPLY ↗

Thank you Adrian for your post on this issue which is LACKING from the opencv.org site. They cover installation to most ALL other OSes except OS X. I successfully followed your clear recipe and also especially appreciate all the thoughtful comments of your other readers. I am however getting an error with cmake -ffmpeg/avformat.h file not found. I wrote up all the details here:

<https://stackoverflow.com/questions/36949053/opencv3-1-compilation-error-ffmpeg-avformat-h-file-not-found>

How can I troubleshoot this?

Regarding the PYTHON2_NUMPY_INCLUDE_DIRS

the actual .h files are here:

~/virtualenvs/cv/lib/python2.7/site-packages/numpy/core/include/numpy

not here

~/virtualenvs/cv/lib/python2.7/site-packages/numpy/core/include

Well, I tracked down another error. Tried compiling with CUDA on but apparently that doesn't work so easily on a mac. It has to do with compiler incompatibility with Clang and CUDA. Apparently I have to set a different compiler with CUDA_HOST_COMPILER. Do you have any insight about this?

I solved the ffmpeg problem. It seems there are certain source files that brew install ffmpeg doesn't get

brew uninstall ffmpeg

brew install ffmpeg --build-from-source

Free 21-day crash course on computer vision & image search engines

Known issue.

<https://github.com/Itseez/opencv/issues/6435>



Adrian Rosebrock April 30, 2016 at 3:51 pm #

REPLY ↗

Thanks for the comments Stefan! I've consolidated them into a single thread to make them a bit easier to read. Congrats on resolving the FFmpeg issue. As for CUDA, I try to avoid enabling that on OSX if at all possible. It's always given me problems when compiling as well.



Greg n May 5, 2016 at 4:59 pm #

REPLY ↗

These are super clear and useful instructions. Thanks for taking the time and for all you did for us. I appreciate anything that hadn't already been covered.



Adrian Rosebrock May 6, 2016 at 4:35 pm #

REPLY ↗

Thanks for the kind words Greg, I'm happy the tutorial helped!



Mathilde May 7, 2016 at 8:33 am #

REPLY ↗

Hi,

Thank you for this tutorial, this really help me out! I did everything and it's working great! But now I have a question. What should I do? Do I have to restart the c-make part?

Free 21-day crash course on computer vision & image search engines

Interested in computer vision and image search engines, but don't know where to start? Let me help. I've created a free, 21-day crash course that is hand-tailored to give you the best possible introduction to computer vision. Sound good? Enter your email below to start your journey to becoming a computer vision master.

Email Address

LET'S DO IT!



Adrian Rosebrock May 7, 2016 at 12:34 pm #

REPLY ↗

The matplotlib library is completely independent from OpenCV, so there is absolutely no need to worry about CMake. What is the error you are getting with matplotlib?



Damon May 7, 2016 at 8:18 pm #

REPLY ↗

Thanks for the great tutorial, Adrian! I really appreciate the effort! I've managed to get core OpenCV working, but I can't seem to get any of the useful stuff working, like the SURF classifier. I'm told that modules such as xfeatures2d aren't available. These are the modules found in opencv_contrib. I've followed your guide to the tee, but for some reason the extra modules simply are not linked to the core module. Then dylib files even exist at my /usr/local/lib folder. Any ideas?



Adrian Rosebrock May 8, 2016 at 8:08 am #

REPLY ↗

My favorite way to debug if the `opencv_contrib` modules are installed is to investigate the output of CMake. Under the list of components/packages to be built, you should see a list of OpenCV modules. Ideally, you'll also see the `xfeatures2d` module in their, indicating that your compile will include the `opencv_contrib`. If you do not see these extra modules, then it's likely that your path to the `opencv_contrib` directory is incorrect. I would suggest deleting your current `build` directory, re-creating it, and then re-running CMake, supplying the correct path to `opencv_contrib`.



Sumanth Mallya May 11, 2016 at 11:05 am #

REPLY ↗

Brilliant tutorial ! Everything worked like clockwork, however when i was trying to use openCV in IDLE i was getting an cv2 not found error, after a little research and trying things out i found the solution, i had to edit my .bash_profile file to add a path variable added to the file after which it worked perfectly. Hope this helps someone with the same issue.

Free 21-day crash course on computer vision & image search engines

PS – I'm guessing you can also amend the below code for a different location of the cv2.so file

```
export PYTHONPATH=/Users/Sumanth/opencv/build/lib:$PYTHONPATH
```



Adrian Rosebrock May 12, 2016 at 3:39 pm #

REPLY ↗

Are you referring to the GUI version of IDLE? If so, the GUI IDLE doesn't place nice with Python virtual environments. Otherwise, if you want to access IDLE via command like, just be sure to execute the `worlon` command first.



Olivera May 19, 2016 at 12:36 pm #

Hi Adrian. I need your help, please. I try to follow your directions to install openCV on El Capitan and virtualenvwrapper, but when I update the bash_profile with the lines you have it shows me "no such cv virtual environment. When I add the command "mkvirtualenv cv" it shows me "the command not found" but I'm not succeeding. Please, help me. Thank you in advance.



Adrian Rosebrock May 19, 2016 at 5:58 pm #

In order to run `mkvirtualenv`, you'll need to make sure you have your `~/.bash_profile`

```
$ nano ~/.bash_profile
```



shahram September 3, 2016 at 7:55 pm #

I have exactly the same problem. Any help?



Vidhi Jain June 5, 2016 at 7:12 am #

REPLY ↗

For my system, even the sudo make install command seems to be failing. Any suggestions?



Adrian Rosebrock June 5, 2016 at 11:20 am #

REPLY ↗

If `sudo make install` isn't working, then the `make` command likely failed as well. Check the output of `make` to ensure that OpenCV compiled correctly.



Mehdi June 14, 2016 at 11:55 am #

REPLY ↗

Awesome tutorial!



Adrian Rosebrock June 15, 2016 at 12:35 pm #

REPLY ↗

Thanks Mehdi!



Geert June 23, 2016 at 2:29 pm #

REPLY ↗

Thank you for this awesome and easy to follow tutorial. Now I want to use matplotlib but I do not get it to work. I tried to install it with pip within the virtual environment. There were no problems there but when I try to import matplotlib in my code which is outside the virtual environment, I get the error `ModuleNotFoundError: No module named 'matplotlib'`.

Free 21-day crash course on computer vision & image search engines

`RuntimeError: Python is not installed as a framework. The Mac OS X backend will not be able to function correctly if Python is not installed as a framework. See the Python documentation for more information on installing Python as a framework on Mac OS X. Please either reinstall Python as a framework, or try one of the other backends. If you are Working with Matplotlib in a virtual enviroment see 'Working with Matplotlib in Virtual environments' in the Matplotlib FAQ`

Please help me and tell me what i'm doing wrong. Thank you in advance.



Adrian Rosebrock June 25, 2016 at 1:40 pm #

REPLY ↗

There are some known issues with Python virtual environments and the latest release of matplotlib that cause this error. As far as I know, there is no fix yet, but for the time being, you can use a previous version of matplotlib and it will work on.

\$ pip install matplotlib==1.4.3

If you already have matplotlib installed, make sure you uninstall it before you execute the above command.



Chris August 6, 2016 at 8:06 pm #

Was wondering if anyone was having an issue when importing cv2 in python. The install failed with the following: ImportError: numpy.core.multiarray failed to import

Anyone have a fix?



Adrian Rosebrock August 7, 2016 at 8:10 am #

REPLY ↗

Hey Chris — it sounds like you do not have NumPy installed on your system or you do not have it in your Python virtual environment. I would suggest going back to Step #5 and ensuring you have NumPy installed.



Chris August 7, 2016 at 12:59 pm #

REPLY ↗

Thanks for the advice. I tried that to no avail. I removed all traces of NumPy and reinstalled it ensuring the paths are correct in the cmake but still get the error "ImportError: numpy.core.multiarray failed to import" I am a bit of a noob when it comes to working in the terminal so probably just missing something small.



Adrian Rosebrock August 8, 2016 at 6:42 pm #

REPLY ↗

This definitely seems to be an issue either related to your `PYTHONPATH` or your Python virtual environment. Unfortunately, without physical access to your machine, I can't diagnose what the exact issue is. Make sure you are in the `cv` virtual environment before installing NumPy and running CMake.



Tanya August 9, 2016 at 2:07 pm #

REPLY ↗

Trying to do "pip install virtualenv virtualenvwrapper". However, when I edit my `.bash_profile` as the instructions say and run "source `~/.bash_profile`" I get the error "/usr/local/bin/virtualenvwrapper.sh: No such file or directory". Then when I run "which virtualenv" to see where the virtualenv is being installed, I get "/Library/Frameworks/Python.framework/Versions/2.7/bin/virtualenv". I've tried uninstalling and then installing again, but that has not worked. How can I fix this problem?



Adrian Rosebrock August 10, 2016 at 9:28 am #

REPLY ↗

Hey Tanya — I suspect this is a Python versioning issue. Did you install Python via Homebrew? Did you edit your `.bash_profile` file? I would run `which python` and see if it's the same Python that virtualenv is using.

Free 21-day crash course on computer vision & image search engines

Interested in computer vision and image search engines, but don't know where to start? Let me help. I've created a free, 21-day crash course that is hand-tailored to give you the best possible introduction to computer vision. Sound good? Enter your email below to start your journey to becoming a computer vision master.

Email Address

LET'S DO IT!



Apple August 15, 2016 at 9:45 pm #

REPLY ↗

Thanks for your tutorial, I have the similar problem. Trying to do "pip install virtualenv virtualenvwrapper", but when I run "source ~/.bash_profile", I was told "No module named virtualenvwrapper. virtualenvwrapper.sh: There was a problem running the initialization hooks." I am sure the output of 'which python' is /usr/local/bin/python, and I have run "pip install virtualenv virtualenvwrapper"



Adrian Rosebrock August 16, 2016 at 1:00 pm #

REPLY ↗

Try running `pip freeze` and ensuring that both `virtualenv` and `virtualenvwrapper` are present, then the issue is likely due to the path updates to your `.bash_profile` file.



E July 21, 2017 at 5:56 pm #

I have this same problem. When I run `which python`, I get "/usr/local/bin/python" but "/Library/Frameworks/Python.framework/Versions/2.7/bin/virtualenv". How do I make it so that the virtualenv is installed on the correct python?



Adrian Rosebrock July 24, 2017 at 3:49 pm #

It's hard to say what the exact issue is without physical access to your machine. My impression that pip, for whatever reason, is using the original version, not the Homebrew-installed one.



MTS August 30, 2016 at 1:38 pm #

REPLY ↗

Hello, i just using python 2.7 and will instal opencv, so i must using other hardware (rashberry) or can just using computer ?



Adrian Rosebrock August 31, 2016 at 1:44 pm #

REPLY ↗

Hm, I'm not sure what you are asking in this comment. If you are using Python 2.7 on your OSX machine, follow this tutorial. If you are using a Raspberry Pi, follow one of these tutorials.



Mike Sharps September 15, 2016 at 5:25 am #

REPLY ↗

Brilliant tutorial. Followed instructions step by step and Installed Opencv 3.1.0 on El Capitan no problems. Ubuntu is next. What IDE / Dev environment do you recommend I've just been using either textmate or emacs for some basic python projects but not an IDE as such. Also I'm not too sure where to go next to explore opencv, any suggestions ?



Adrian Rosebrock September 15, 2016 at 9:30 am #

REPLY ↗

If you're just getting started learning OpenCV, then you should absolutely go through Practical Python and OpenCV. This book will help you get up to speed super quickly. It also covers a number of interesting computer vision projects including face detection, handwriting recognition, and identifying the covers of books.

As for an IDE, take a look at PyCharm. I detail how to setup PyCharm for computer vision development in this blog post.



joas September 16, 2016 at 1:47 pm #

Free 21-day crash course on computer vision & image search engines

Hello Adrian, I can't find cv2.so in the / site-packages when I exactly install everything . Please help



Adrian Rosebrock September 19, 2016 at 1:23 pm #

REPLY ↗

Hey Joas — after `make` has finished executing, make sure you take a look in the `build/lib` directory and see if the `cv2.so` file is in there.



minho September 22, 2016 at 5:00 pm #

REPLY ↗

Thanks for your awesome tutorial!

I'm first time to python opencv and OSX.

I got error in 'make -j4' step like follows

```
1 [ 26%] Built target opencv_face
2 [ 26%] Built target opencv_shape
3 /Users/minhominho/opencv/modules/videoio/src/cap_qtkit.mm:46:9: fatal error:
4 #import
5   ^
6 1 error generated.
```

how can I solve this problem ? T_T

Free 21-day crash course on computer vision & image search engines

Interested in computer vision and image search engines, but don't know where to start? Let me help. I've created a free, 21-day crash course that is hand-tailored to give you the best possible introduction to computer vision. Sound good? Enter your email below to start your journey to becoming a computer vision master.

LET'S DO IT!



Adrian Rosebrock September 23, 2016 at 6:51 am #

REPLY ↗

I assume based on this error message that you are using either OSX – El Capitan or mac OS X. I've been breaking OpenCV builds because QTKit is no longer bundled with the OS and the OpenCV build system is a mess right now and I'm actually investigating it myself. In the meantime, I would suggest downgrading to OSX Yosemite (10.10) where QTKit is still found.



Alexander September 28, 2016 at 11:48 pm #

REPLY ↗

One solution for this for now is to simply disable the videoio module while compiling. You can do that with this flag in cmake: `-D BUILD_opencv_videoio=OFF` (from <https://github.com/opencv/opencv/issues/6913>)

After that I also had another issue, "fatal error: 'opencv2/highgui.hpp' file not found" around 86% of the way through. I'm not sure if this is caused because I'm running Sierra, but I couldn't find anything online about it. Disabling the xphoto module let it compile though. (the flag is `"BUILD_opencv_xphoto=OFF"`)

Hopefully they'll fix these issues soon though so we can recompile with these two modules.



Adrian Rosebrock September 30, 2016 at 6:46 am #

REPLY ↗

Thanks for the tip Alexander. I got OpenCV 3.1 to compile on El Capitan yesterday. I'm planning on trying Sierra later this weekend. I'll be sure to turn my findings into a blog post 😊



Peter October 4, 2016 at 2:56 am #

REPLY ↗

Hi Alexander,

I have the same error too, could you help me please?

Thanks

Free 21-day crash course on computer vision & image search engines



As I mentioned in the previous comment, I'll be finalizing my steps and turning the result into a new blog post within the next couple of weeks.



Roger October 9, 2016 at 3:26 pm #

Hi Adrian,

Your post has been really helpful, thank you very much. I have enrolled to your course and followed the tutorial so far, but hit the brick when tried to install the opencv 3.1.0 on macOS Sierra 10.12.

Since you have mentioned that you will post new updated steps, I was wondering if you have any update on this matter?

Many thanks



Adrian Rosebrock October 11, 2016 at 1:03 pm #

Hey Roger — I'm planning on creating a brand new blog post for macOS monitoring the OpenCV commits to determine the right time to create this blog pos



Madeline January 27, 2017 at 2:20 pm #

I am encountering this same problem now. Is there an updated solution? I would like to compile OpenCV with the videoio module.



Adrian Rosebrock January 28, 2017 at 6:47 am #

Please follow my updated tutorials for macOS.



Lance Legel September 27, 2016 at 11:03 pm #

REPLY ↗

You're the boss, thanks so much.

Just a note that as of today, I wasn't able to compile with 3.1.0 hardcoded into the git checkout; I had to simply work with the master branches for both opencv and opencv_contrib, following this error: <http://answers.opencv.org/question/94067/help-analyze-make-error-in-opencv-contrib3/>

Anyways, keep killing it, and best regards.



Adrian Rosebrock September 28, 2016 at 10:39 am #

REPLY ↗

Thanks for sharing Lance — very strange indeed!



Margareta October 9, 2016 at 6:27 am #

REPLY ↗

Great guide!

However I couldn't install opencv without doing some modification on El captain. I did the following modifications:

1. Never checkout out a certain version from the gitrepo, I only cloned down the repo and skipped the checkout part for both opencv and opencv_contrib
2. Had -D INSTALL_C_EXAMPLES=ON instead of off as suggested.



Adrian Rosebrock October 11, 2016 at 1:04 pm #

Free 21-day crash course on computer vision & image search engines

Great job getting OpenCV installed on your system Margareta! I'm in the process of creating updated OpenCV install tutorials for both El Capitan and macOS Sierra to avoid confusions like these in the future.



Roger October 12, 2016 at 5:56 am #

REPLY ↗

Hi Margareta,

Thanks you very much!

I have followed your steps and successfully installed opencv 3.1.0 on macOS Sierra 10.12; executed the test drive suggested in this tutorial, so far so good :)

To Adrian, I hope this information is useful for your new tutorials.

Kind regards

Free 21-day crash course on computer vision & image search engines

Interested in computer vision and image search engines, but don't know where to start? Let me help. I've created a free, 21-day crash course that is hand-tailored to give you the best possible introduction to computer vision. Sound good? Enter your email below to start your journey to becoming a computer vision master.

Email Address

LET'S DO IT!



Belal C October 13, 2016 at 9:32 pm #

REPLY ↗

I got around the issue with compiling on macOS Sierra by switching to the 'pr7266' branch (<https://github.com/opencv/opencv/pull/7266>).

I'm pretty sure the AVFoundation functionality developed to replace QTkit is merged with the master branch now.

Anyway, thanks so much for this thorough guide Adrian – really helped me as a beginner to compiling OpenCV from source (I was getting errors with QTkit which were broken down!)



Adrian Rosebrock October 15, 2016 at 9:57 am #

REPLY ↗

Thanks for sharing Belal! Once we get another update to OpenCV (ideally v3.2) I'll make another blog post on installing on OSX. You also like to see macOS Sierra mature a bit.



Alex Ogilvie October 18, 2016 at 7:26 am #

REPLY ↗

Hi Adrian,

Thanks for the tutorial. I've managed to get OpenCV installed within the virtual environment I set up, but need to have it installed on my main environment. When I am trying to run python scripts that use OpenCV from MATLAB, MATLAB cannot find the cv2 module. Do you have any idea how to get OpenCV onto the main machine?

Thanks,



Adrian Rosebrock October 20, 2016 at 8:57 am #

REPLY ↗

Hey Alex — I honestly haven't used MATLAB in many years so I'm not sure regarding this.



Alex Ogilvie October 21, 2016 at 7:28 am #

REPLY ↗

I guess what I'm trying to ask is how do you access python from outside the virtual environment? I've been into the site-packages folder of the virtual environment and the main environment, and a lot of the files – particularly the .so and .py files – aren't in the main environment. Could this be why it's not working?



Adrian Rosebrock October 23, 2016 at 10:20 am #

REPLY ↗

Typically I would suggest not creating Python projects *without* a virtual environment. If you want to use Python in your main environment or of your virtual environment(s) simply copy the `cv2.so` files into your system `site-packages` folder.

Free 21-day crash course on computer vision & image search engines

REPLY ↗

**Christian Egglin** October 21, 2016 at 11:26 pm #

I figured it out using Belal's solution, I guess I should have checked before I commented. But now I am having the problem around the 85% mark:
`/Users/Christian/opencv/modules/python/src2/cv2.cpp:6:10: fatal error: 'Python.h' file not found`
`#include`

Most people commenting seem to have suggested that it come from the CMake step at the -D
`PYTHON2_INCLUDE_DIR=/usr/local/Frameworks/Python.framework/Headers.`

But Python.h is located there. I am not sure how to fix this. I am using Python 2.7 and have Sierra

Thanks again,
Christian

**Adrian Rosebrock** October 23, 2016 at 10:15 am #

Installing OpenCV on macOS Sierra is a major pain right now. I'm currently working on
`PYTHON2_INCLUDE_DIR` should look something like this:

`PYTHON2_INCLUDE_DIR=/usr/local/Cellar/python/2.7.12/Frameworks/Py`

Keep in mind that you should be compiling against the *Homebrew* version of Python, not your system's Python.

**Walid Ahmed** October 25, 2016 at 10:21 am #

Hi Adrian

I have both OpenCV2.4 and OpenCV3 installed on my Mac with python.

My code now only sees how OpenCV2.4, can I direct my system to use OpenCV3?

Thanks

Free 21-day crash course on computer vision & image search engines

Interested in computer vision and image search engines, but don't know where to start? Let me help. I've created a free, 21-day crash course that is hand-tailored to give you the best possible introduction to computer vision. Sound good? Enter your email below to start your journey to becoming a computer vision master.

LET'S DO IT!

**Adrian Rosebrock** November 1, 2016 at 9:40 am #

REPLY ↗

It really depends on how you actually installed both OpenCV 2.4 and OpenCV 3. Without knowing that I can't direct you.

**Benabbou** October 30, 2016 at 5:02 am #

REPLY ↗

Hi Adrian,

Thanks for this tutorial i find it very useful and explicit however I got stuck at Compiling error

```
1 [code] /Users/anasnadir/opencv/modules/videoio/src/cap_qtkit.mm:46:9: fatal error: 'QTKit/QTKit.h'
2   file not found
3 #import ^
4 ^
5 1 error generated.
```

I don't really get what is the problem .

**Benabbou** October 30, 2016 at 6:54 am #

REPLY ↗

I solved the Problem by deleting OpenCv and OpenCv_rep .. directories and Cloned new ones but this time I used the Master Version I didn't check out on both of them, then I did The other steps .

Free 21-day crash course on computer vision & image search engines



Adrian Rosebrock November 1, 2016 at 9:13 am #

REPLY ↗

Please see my reply to Rakshith above where I have addressed this question.



Rakshith October 30, 2016 at 8:05 am #

REPLY ↗

Hey Adrian, I'm getting a fatal error at "make -j4" step.

```
1 [ 21%] Building CXX object modules/videoio/CMakeFiles/opencv_videoio.dir/src/cap_qtkit.mm.o
2 [ 21%] Built target opencv_dnn
3 /Users/rakshithgb/opencv/modules/videoio/src/cap_qtkit.mm:46:9: fatal error:
4   'QTKit/QTKit.h' file not found
5 #import
6 ^
7 1 error generated.
8 make[2]: *** [modules/videoio/CMakeFiles/opencv_videoio.dir/src/cap_qtkit.mm]
9 make[1]: *** [modules/videoio/CMakeFiles/opencv_videoio.dir/all] Error 2
10 make: *** [all] Error 2
```

Is there any fix to this?



Adrian Rosebrock November 1, 2016 at 9:13 am #

REPLY ↗

Hey Rakshith — this error is because you are using either El Capitan or macOS Sierra this issue for good, but in the meantime, make sure you download the latest code from the GitHub will ensure you are compiling against the bleeding edge version of OpenCV where the issue has

Free 21-day crash course on computer vision & image search engines

Interested in computer vision and image search engines, but don't know where to start? Let me help. I've created a free, 21-day crash course that is hand-tailored to give you the best possible introduction to computer vision. Sound good? Enter your email below to start your journey to becoming a computer vision master.

LET'S DO IT!



Rakshith November 2, 2016 at 6:06 am #

REPLY ↗

Yeah I am on Sierra. However Alexander had posted a work around, will that work on Sierra? Or can you please post steps to compile with the latest version?



Adrian Rosebrock November 3, 2016 at 9:45 am #

REPLY ↗

The solution is to simply checkout the latest version of OpenCV from GitHub instead of downloading a tagged .zip archive. This will indeed work on Sierra. I'll be posting an updated tutorial covering macOS Sierra (ideally) within the next month.



Ajay November 13, 2016 at 1:04 am #

REPLY ↗

Adrian: you Rock. As of Nov 2016 this is still the best resource on how to install opencv from source in virtualenv on mac



Adrian Rosebrock November 14, 2016 at 12:07 pm #

REPLY ↗

Thanks so much Ajay, I really appreciate that! 😊Congrats on getting OpenCV installed on your Mac.



Jacob December 6, 2016 at 2:19 pm #

REPLY ↗

Thanks for this tutorial!

At "make -j4" I'm getting at fatal error at 31% that says "QTKit/QTKit.h' file not found"

Any ideas how to solve this?

Free 21-day crash course on computer vision & image search engines



Adrian Rosebrock December 7, 2016 at 9:41 am #

REPLY ↗

Hey Jacob — it sounds like you're using macOS Sierra. Please follow [this updated tutorial](#) as I state at the top of the blog post.



J December 15, 2016 at 4:15 pm #

REPLY ↗

Is this <http://www.mobileway.net/2015/02/14/install-opencv-for-python-on-mac-os-x/> → a same setup for installing and running OpenCV and Python too?



Adrian Rosebrock December 18, 2016 at 8:49 am #

Install OpenCV via Homebrew is a different process. The post you linked is for OpenCV 3 + Homebrew online this Monday, December 19th 2016.



Gunnar Dittmar January 2, 2017 at 5:14 am #

Thanks for the tutorial. It works with OpenCV 3.2 as well.



Adrian Rosebrock January 4, 2017 at 11:01 am #

Awesome, glad to hear it Gunnar! 😊



Syeda Hasan January 11, 2017 at 3:09 pm #

REPLY ↗

Hey,

Thank so much for the installation guide. I am trying to detect face in a video and store it as a series of images at different time intervals (for instance: the video is about 1 minute long and I want to capture the face every 10 seconds -that is 6 times) and store it in different files, is there an easy way to do that?



Adrian Rosebrock January 12, 2017 at 7:57 am #

REPLY ↗

To start, you'll want to determine the frame rate of your video. Most videos are in the range 24-32 frames per second. Then, you just set a counter. You detect a face. Then start your counter and wait (for example) 24 * 10 frames until you detect your next face.



Danny January 29, 2017 at 9:55 pm #

REPLY ↗

Hi, I am a beginner. I followed every step above and have successfully installed the OpenCV. I have a question now. Right now I can only run the OpenCV-installed python in the terminal where I can access the virtual environment and it is not convenient to edit at all. When I try to write a program in the python IDLE without the terminal, I cannot import OpenCV anymore and I have to pretend it is imported. So is there any way to import OpenCV library while I am editing it, instead of only using OpenCV when running the program?

I don't know if I clearly explained it, but hope you can get my point.

Thanks for your awesome tutorial again!



Adrian Rosebrock January 30, 2017 at 4:23 pm #

REPLY ↗

Python IDLE does not respect virtual environments. You should use either (1) the terminal version of IDLE or (2) use Jupyter Notebooks. The GUI version of IDLE is not recommended.

Free 21-day crash course on computer vision & image search engines

**sponge** February 24, 2017 at 9:50 pm #

REPLY ↗

hello, first thanks so much for ur installation guide, I just follow ur steps, but in the step 4, when I pip install virtualenv virtualenvwrapper, I will always have a problem that
No module named virtualenvwrapper
virtualenvwrapper.sh: There was a problem running the initialization hooks.

If Python could not import the module virtualenvwrapper.hook_loader,
check that virtualenvwrapper has been installed for
VIRTUALENVWRAPPER_PYTHON=/usr/local/bin/python and that PATH is
set properly.
I do not know how to solve it ,really hope ur advice
thanks so much

**jazz hands** June 16, 2017 at 11:01 am #

I have the same problem! any ideas?

**Mirna** March 5, 2017 at 9:59 am #

Thank you for the great tutorial, Adrian!

**Adrian Rosebrock** March 6, 2017 at 3:43 pm #

No problem Mirna, I'm happy I could help 😊

**Pat** March 8, 2017 at 5:19 pm #

REPLY ↗

Adrian,

Thanks for posting this. I'm new to OpenCV and this has really helped me get up and running without any grief.

When I make my first million from OpenCV I'll buy you a beer.

**Adrian Rosebrock** March 10, 2017 at 3:55 pm #

REPLY ↗

Congrats on getting OpenCV installed Pat! I look forward to that beer 😊

**Ed** March 14, 2017 at 6:25 am #

REPLY ↗

Awesome tutorial, thank you!

I came across a build error on Mac 10.12.3:
changing the makelist file of freetype module made the trick 😊
https://github.com/opencv/opencv_contrib/issues/919

**Adrian Rosebrock** March 15, 2017 at 8:56 am #

REPLY ↗

Thank you for sharing Ed!

Free 21-day crash course on computer vision & image search engines

Interested in computer vision and image search engines, but don't know where to start? Let me help. I've created a free, 21-day crash course that is hand-tailored to give you the best possible introduction to computer vision. Sound good? Enter your email below to start your journey to becoming a computer vision master.

LET'S DO IT!



zrb March 31, 2017 at 4:11 pm #

REPLY ↗

I had the same problem. This works great, thanks!



Jaypee March 19, 2017 at 6:16 pm #

REPLY ↗

I am having problem in the step 8 when I run the \$ make -j4. Can anyone help me fix it. It would be of great help. Thanks.

The error is as follows:

```
[ 30%] Building CXX object modules/videoio/CMakeFiles/opencv_videoio.dir/src/cap_qtkit.mm.o
/Users/akshalsharma/opencv/modules/videoio/src/cap_qtkit.mm:46:9: fatal error: 'QTKit/QTKit.h'
file not found
```



Adrian Rosebrock March 21, 2017 at 7:30 am #

REPLY ↗

Please read the other comments before posting. This question has been addressed. You can search for "QTKit/QTKit.h" file not found" in the search bar.



crazyfang March 21, 2017 at 1:57 am #

REPLY ↗

I met a question that fatal error: 'Python.h' file not found #include when i make.What should i do?



Adrian Rosebrock March 21, 2017 at 7:08 am #

REPLY ↗

Your PYTHON_LIBRARY configuration is incorrect. Double-check the path to your Python include directory.



Paul April 17, 2017 at 8:30 am #

REPLY ↗

How to clean everything and start over???

please help



Adrian Rosebrock April 17, 2017 at 10:51 am #

REPLY ↗

The easiest way? Re-install macOS/OSX. Otherwise without knowing what step you are having trouble with it's impossible to provide any suggestion on how to clean and restart the process.



Guy Hoffman May 23, 2017 at 10:39 pm #

REPLY ↗

Great tutorial!

Small comment – in OpenCV 3.2.0 you need to go to

```
~/opencv_contrib/modules/freetype/CMakeLists.txt
```

and replace

```
 ${freetype2_LIBRARIES} ${harfbuzz_LIBRARIES}
```

with

```
 ${FREETYPE_LIBRARIES} ${HARFBUZZ_LIBRARIES}
```

Free 21-day crash course on computer vision & image search engines

Interested in computer vision and image search engines, but don't know where to start? Let me help. I've created a free, 21-day crash course that is hand-tailored to give you the best possible introduction to computer vision. Sound good? Enter your email below to start your journey to becoming a computer vision master.

LET'S DO IT!

REPLY ↗

**ARUN UPADHYAYA** May 27, 2017 at 7:25 am #

Hi,Adrian,

I successfully executed step3 but while i am trying to execute step 4 its telling me that pip command not found then i tried to install pip using sudo easy_install pip but then its telling command not found,I am screwed please guide me further.
Thanks.

REPLY ↗

**Edgar Acosta** June 18, 2017 at 3:57 pm #

USING OPEN CV IN OTHER PYTHON VIRTUAL ENVIRONMENT:

I follow this Procedure to Install Open CV... and It Works... I install it in the CV Virtual Environment..
But i need to use it in Tensor Flow, Tensor Flow is in another Virtual Environment. How can i use OpenCV
Please help..

Free 21-day crash course on computer vision & image search engines

Interested in computer vision and image search engines, but don't know where to start? Let me help. I've created a free, 21-day crash course that is hand-tailored to give you the best possible introduction to computer vision. Sound good? Enter your email below to start your journey to becoming a computer vision master.

[LET'S DO IT!](#)
**Adrian Rosebrock** June 20, 2017 at 11:05 am #

Hey Edgar — simply access the Python virtual environment that has OpenCV installed

```
1 $ workon your_env_name
2 $ pip install tensorflow
```

REPLY ↗

**nck** July 12, 2017 at 12:40 pm #

Thanks a lot. Everything was almost smooth.

- 1º I had to change my version for 2.7.13.
- 2º I had a problem with cmake which was solved in the comments.
- 3º Adding some code in step 4 which I found in stackexchange, because some code was missing.

**Michelle** July 24, 2017 at 4:28 pm #

Hi Adrian! I've followed this tutorial in the past to download OpenCV in one of my computers and worked like a charm. However, now that I am installing it in another one I had the issue that when I type "which python" it still gives me the system version.

Further, when I type "which python2" it correctly gives me the /usr/local/bin location. Seems like the home-brew installation is installing python as such. At the end I followed an instruction that appeared in the terminal saying that if I wanted the /usr/bin/local version then I should add this path to the bash_profile: "/usr/local/opt/openssl/bin:\$PATH"

The installation worked when I did that but I still get /usr/bin after writing "which python", and the new path above when I try "which python2". Any thoughts on this?

Thanks a million!

REPLY ↗

**Adrian Rosebrock** July 28, 2017 at 10:18 am #

It's hard to say without physical access to your machine, but it seems like your \$PATH has been updated to point to your system install of Python before Homebrew. I would double-check your `.bash_profile` and try to debug this. \$PATH issues can be a real pain to debug.

REPLY ↗

**Daniel** August 11, 2017 at 4:25 pm #

This had stumped me too for a while, but it seems since the original publishing of Homebrew version of Python have changed slightly (possibly could update article?).

Free 21-day crash course on computer vision & image search engines

As per the caveats of `brew info python`:

> This formula installs a python2 executable to /usr/local/bin.
 If you wish to have this formula's python executable in your PATH then add
 the following to ~/.bash_profile:
`export PATH="/usr/local/opt/python/libexec/bin:$PATH"`

So the initial `usr/local/bin` for Homebrew, and the above path for Homebrew's Python. Works after that!



Adrian Rosebrock August 14, 2017 at 1:16 pm #

REPLY ↗

Thanks for sharing Daniel!



Daniel YEH October 1, 2017 at 10:46 pm #

I added `export PATH="/usr/local/opt/python/libexec/bin:$PATH"` to the bash profile
 command can't work!
 It showed command not found.
 I need to type the following code to let terminal work properly:
`export PATH="/usr/bin:/bin:/usr/sbin:/sbin"
 export PATH="/usr/local/bin:/usr/local/sbin:$PATH"`
 but my problem of python"2" didn't solve...



David August 14, 2017 at 6:09 pm #

I struggled with this too because no matter what, "which python" still pointed to my system Python rather than brew's.

However, "which python2" points to the brew Python:

```
1 $ which python
2 /usr/bin/python
3 $ which python2
4 /usr/local/bin/python2
5 $ which python3
6 /usr/local/bin/python3
```

It is also good to check `pip`:

```
1 $ which pip
2 /usr/local/bin/pip
3 $ which pip3
4 /usr/local/bin/pip3
```

The next step and trouble is when running the `virtualenvwrapper.sh` script from your `.bash_profile`.

It will complain when you source your `.bash_profile` after adding the line: `source /usr/local/bin/virtualenvwrapper.sh`:

```
1 $ source ~/.bash_profile
2 -bash: /usr/local/bin: is a directory
3 virtualenvwrapper.sh: There was a problem running the initialization hooks.
4
5 If Python could not import the module virtualenvwrapper.hook_loader,
6 check that virtualenvwrapper has been installed for
7 VIRTUALENVWRAPPER_PYTHON=/usr/local/bin and that PATH is
8 set properly."
```

That prompted me to read and modify that script.

```
$ sudo vim /usr/local/bin/virtualenvwrapper.sh
```

on Line 50 change

```
VIRTUALENVWRAPPER_PYTHON="$(command \which python)"  

_to_  

VIRTUALENVWRAPPER_PYTHON="$(command \which python2)"
```

Free 21-day crash course on computer vision & image search engines

Interested in computer vision and image search engines, but don't know where to start? Let me help. I've created a free, 21-day crash course that is hand-tailored to give you the best possible introduction to computer vision. Sound good? Enter your email below to start your journey to becoming a computer vision master.

LET'S DO IT!

That will get the right path to Python2 for you (brew's Python).

Then:

```
$ source ~/.bash_profile
```

And now you're ready to:

```
$ mkvirtualenv cv -p python2
```

and if you want python3

```
$ mkvirtualenv cvpy3 -p python3
```

Hopefully this helps someone and isn't just a one off issue that I've had to figure out.



kram September 6, 2017 at 1:23 am #

I'm still getting the same complaint even after editing the virtualenvwrapper.sh, I changed it to python3. Any idea what else can be wrong?



Adrian Rosebrock September 7, 2017 at 7:12 am #

I actually was working on this problem yesterday. Update your `.bash_profile`:

```
1 # virtualenv and virtualenvwrapper
2 export VIRTUALENVWRAPPER_PYTHON=/usr/local/bin/python3
3 source /usr/local/bin/virtualenvwrapper.sh
```

Then `source` it again. From there it should work. You might also have to edit the `virtualenvwrapper.sh` contents (I didn't have to edit the file).

Free 21-day crash course on computer vision & image search engines

Interested in computer vision and image search engines, but don't know where to start? Let me help. I've created a free, 21-day crash course that is hand-tailored to give you the best possible introduction to computer vision. Sound good? Enter your email below to start your journey to becoming a computer vision master.

LET'S DO IT!

Trackbacks/Pingbacks

[Install OpenCV 3.0 and Python 2.7+ on Ubuntu - PyImageSearch](#) - June 22, 2015

[...] Last week we kicked-off the OpenCV 3.0 install fest by detailing how to install OpenCV 3.0 and Python 2.7+ on the OSX platform. [...]

[Install OpenCV 3.0 and Python 3.4+ on OSX - PyImageSearch](#) - June 29, 2015

[...] Two weeks ago we kicked off the OpenCV 3.0 install-fest with a tutorial on how to install OpenCV 3.0 and Python 2.7 on OSX. [...]

Leave a Reply

 Name (required)

 Email (will not be published) (required)

 Website

SUBMIT COMMENT

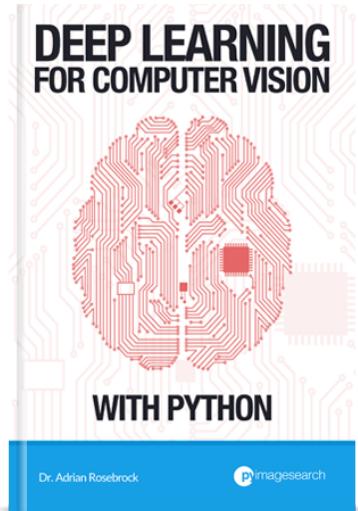
Resource Guide (it's totally free).

Click the button below to get my [free 11-page Image Search Engine Resource Guide PDF](#). Uncover exciting new ways to build image search engines of your own.

Free 21-day crash course on computer vision & image search engines


[Download for Free!](#)

Deep Learning for Computer Vision with Python Book



You're interested in deep learning and computer vision, *but you don't know how to get started*. Let me help. [My deep learning](#).

[CLICK HERE TO PRE-ORDER MY NEW BOOK](#)

You can detect faces in images & video.



Are you interested in **detecting faces in images & video?** But **tired of Googling for tutorials** that never work? Then let me help! I guarantee that my new book will turn you into a **face detection ninja** by the end of this weekend. [Click here](#) to give it a shot yourself.

[CLICK HERE TO MASTER FACE DETECTION](#)

PyImageSearch Gurus: NOW ENROLLING!

Free 21-day crash course on computer vision & image search engines

Interested in computer vision and image search engines, but don't know where to start? Let me help. I've created a free, 21-day crash course that is hand-tailored to give you the best possible introduction to computer vision. Sound good? Enter your email below to start your journey to becoming a computer vision master.

[LET'S DO IT!](#)

Free 21-day crash course on computer vision & image search engines

The PyImageSearch Gurus course is now enrolling! Inside the course you'll learn how to perform:

- Automatic License Plate Recognition (ANPR)
- Deep Learning
- Face Recognition
- and much more!

Click the button below to learn more about the course, take a tour, and get 10 (FREE) sample lessons.

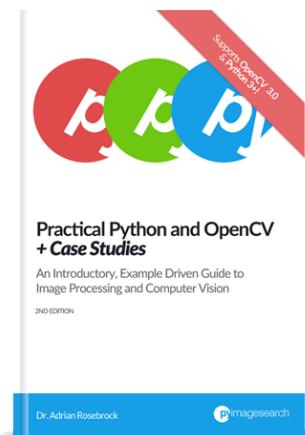
TAKE A TOUR & GET 10 (FREE) LESSONS

Hello! I'm Adrian Rosebrock.



I'm an entrepreneur and Ph.D who has launched two successful image search engines, ID hacks I've learned along the way.

Learn computer vision in a single weekend.



Want to learn computer vision & OpenCV? I can teach you in a **single weekend**. I know. It sounds crazy, but it's no joke. My new book is your **guaranteed, quick-start guide** to becoming an OpenCV Ninja. So why not give it a try? [Click here](#) to become a computer vision ninja.

[CLICK HERE TO BECOME AN OPENCV NINJA](#)

Subscribe via RSS



Never miss a post! Subscribe to the PyImageSearch RSS Feed and keep up to date with my image search engine tutorials, tips, and tricks

POPULAR

Install OpenCV and Python on your Raspberry Pi 2 and B+

FEBRUARY 23, 2015

Home surveillance and motion detection with the Raspberry Pi, Python, OpenCV, and Dropbox

JUNE 1, 2015

Install guide: Raspberry Pi 3 + Raspbian Jessie + OpenCV 3

APRIL 18, 2016

How to install OpenCV 3 on Raspbian Jessie

Free 21-day crash course on computer vision & image search engines

Interested in computer vision and image search engines, but don't know where to start? Let me help. I've created a free, 21-day crash course that is hand-tailored to give you the best possible introduction to computer vision. Sound good? Enter your email below to start your journey to becoming a computer vision master.

Email Address

LET'S DO IT!

OCTOBER 26, 2015

Basic motion detection and tracking with Python and OpenCV

MAY 25, 2015

Accessing the Raspberry Pi Camera with OpenCV and Python

MARCH 30, 2015

Ubuntu 16.04: How to install OpenCV

OCTOBER 24, 2016

Search

Search...

Find me on [Twitter](#), [Facebook](#), [Google+](#), and [LinkedIn](#).
© 2017 PylimageSearch. All Rights Reserved.

**Free 21-day crash course
on computer vision &
image search engines**

Interested in computer vision and image search engines, but don't know where to start? Let me help. I've created a free, 21-day crash course that is hand-tailored to give you the best possible introduction to computer vision. Sound good? Enter your email below to start your journey to becoming a computer vision master.

 Email Address

LET'S DO IT!

**Free 21-day crash course on computer
vision & image search engines**