

Module 0.1

# Getting started MacOS

---

**Satya Mallick, Ph.D.**

June 19, 2017

# Table of Content

<b>Installing OpenCV on MacOS</b>	<b>2</b>
Step 1: Install XCode	2
Step 2: Install Homebrew	2
Step 3: Install Python 2 and Python 3	2
Step 4: Install Python libraries	3
Step 5: Install OpenCV 3	3
Step 5.1 : Compile & Install OpenCV	4
Step 5.2 : Add OpenCV's site-packages path to global site-packages	4
Step 6: Make OpenCV 3 python symlink in our virtual environment	4
Step 7: Test OpenCV 3	5
<b>Install Dlib on MacOS</b>	<b>6</b>
<b>For MacOS &gt;= 10.11 &amp; XCode &gt;= 8</b>	<b>6</b>
Step 1: Install OS libraries	6
Step 2: Install Python 2 and/or Python 3	6
Step 3: Install Python libraries	6
Step 4: Install Dlib	6
Step 4.1: Compile C++ library	6
Step 4.2: Compile Python module	6
<b>For MacOS 10.10 &amp; XCode &lt; 8</b>	<b>7</b>
Step 1: Install OS libraries	7
Step 2: Install Python 2 and/or Python 3	7
Step 3: Install Python libraries	7
Step 4: Compile Dlib	7
Step 4.1: Compile C++ library	7
Step 4.2 : Compile Python library	8

# Installing OpenCV on MacOS

## Step 1: Install XCode

Install XCode from the App Store.

If XCode available on the App Store is not compatible with your OS:

1. Find XCode version compatible with your OS from this table.  
[https://en.wikipedia.org/w/index.php?title=Xcode#Version\\_comparison\\_table](https://en.wikipedia.org/w/index.php?title=Xcode#Version_comparison_table)
2. Go to this webpage <https://developer.apple.com/download/more/>
  - a. Login if you have apple developer account. Else, create your account and login
  - b. Search for Xcode and download the version compatible with your OS.
3. Install XCode.
4. After installation open XCode, and accept xcode-build license when prompted.

## Step 2: Install Homebrew

Launch a terminal using Launchpad. From this step onward, all commands will be run in the terminal.

```
ruby -e "$(curl -fsSL
https://raw.githubusercontent.com/Homebrew/install/master/install)"

brew update

# Add Homebrew path to PATH variable
echo "# Homebrew" >> ~/.bash_profile
echo "export PATH=/usr/local/bin:$PATH" >> ~/.bash_profile

source ~/.bash_profile

# Tap the science repo of homebrew
brew tap homebrew/science
```

## Step 3: Install Python 2 and Python 3

```
brew install python python3

brew linkapps python
brew linkapps python3

# check whether Python using homebrew install correctly
```

```
which python # it should output /usr/local/bin/python
which python3 # it should output /usr/local/bin/python3

# check Python versions
python --version
python3 --version
```

Python version (2.6 or 2.7, 3.5 or 3.6) installed on your machine is required to determine path of Python's site-packages. It will be used later.

## Step 4: Install Python libraries

We will use a Virtual Environment to install Python libraries. This allows you to keep your project environment and global environment separate.

```
# Install virtual environment
pip install virtualenv virtualenvwrapper
echo "# Virtual Environment Wrapper"
echo "source /usr/local/bin/virtualenvwrapper.sh" >> ~/.bash_profile
source ~/.bash_profile

##### For Python 2 #####
# Create virtual environment
mkvirtualenv facecourse-py2 -p python2
workon facecourse-py2

# Install python libraries within this virtual environment
pip install numpy scipy matplotlib scikit-image scikit-learn ipython

# Quit virtual environment
deactivate

##### For Python 3 #####
# Create virtual environment
mkvirtualenv facecourse-py3 -p python3
workon facecourse-py3

# Install python libraries within this virtual environment
pip install numpy scipy matplotlib scikit-image scikit-learn ipython

# quit virtual environment
deactivate
```

## Step 5: Install OpenCV 3

Homebrew allows you to install OpenCV with either Python2 or Python3 bindings not both. So follow steps either for Python2 or Python3.

## Step 5.1 : Compile & Install OpenCV

```
##### For Python 2 #####
# Compile OpenCV 3 with Python2 bindings
brew install opencv3 \
  --with-contrib --with-examples --with-nonfree \
  --with-qt --with-tbb --with-ffmpeg --with-gstreamer
```

```
##### For Python 3 #####
# Compile OpenCV3 with Python3 bindings
brew install opencv3 \
  --with-contrib --with-examples --with-nonfree \
  --with-qt --with-tbb --with-ffmpeg --with-gstreamer \
  --with-python3 --without-python
```

## Step 5.2 : Add OpenCV's site-packages path to global site-packages

When brew is done compiling and installing OpenCV3, we will update path of site-packages directory which contains cv2.so file to Homebrew Python's site-packages directory. Depending upon the Python version you have (2.6/2.7 or 3.5/3.6) these paths would be different.

```
echo /usr/local/opt/opencv3/lib/python2.7/site-packages >>
/usr/local/lib/python2.7/site-packages/opencv3.pth
```

## Step 6: Make OpenCV 3 python symlink in our virtual environment

Path to OpenCV's Python library will be different depending upon which Python version you have. Double check the exact path and filename on your machine. Use this command to find out the path on your machine.

```
find /usr/local/opt/opencv3/lib/ -name cv2*.so
```

```
##### For Python 2 #####
cd ~/.virtualenvs/facecourse-py2/lib/python2.7/site-packages/
ln -s /usr/local/opt/opencv3/lib/python2.7/site-packages/cv2.so cv2.so
```

```
##### For Python 3 #####
cd ~/.virtualenvs/facecourse-py3/lib/python3.6/site-packages/
ln -s /usr/local/opt/opencv3/lib/python3.6/site-packages/cv2.cpython-36m-darwin.so
cv2.so
```

## Step 7: Test OpenCV 3

```
# activate virtual environment
##### For Python 2 #####
workon facecourse-py2
##### For Python 3 #####
workon facecourse-py3
#####

# open ipython
ipython
# import cv2 and print version
import cv2
print(cv2.__version__)
# If OpenCV3 is installed correctly, the above command should give output 3.2.0
# Press CTRL+D to exit ipython
```

## Install Dlib on MacOS

The installation instructions are slightly different for different versions of the operating system and XCode. Please choose the right one for you.

### For MacOS >= 10.11 & XCode >= 8

#### Step 1: Install OS libraries

```
brew cask install xquartz  
brew install gtk+3 boost  
brew install boost-python --with-python3
```

#### Step 2: Install Python 2 and/or Python 3

We have already installed Python while installing OpenCV.

#### Step 3: Install Python libraries

We have completed this step too.

#### Step 4: Install Dlib

The steps for installing Dlib are listed below

##### Step 4.1: Compile C++ library

```
brew install dlib
```

##### Step 4.2: Compile Python module

Now activate “facecourse-py2” or “facecourse-py3” virtual environment based on whether you were following the process for Python2 or Python3.

```
##### For Python 2 #####  
workon facecourse-py2  
##### For Python 3 #####  
workon facecourse-py3
```

We will now install Dlib using pip.

```
pip install dlib
```

Installation is complete. You can now exit from virtualenv using deactivate.

```
deactivate
```

## For MacOS 10.10 & XCode < 8

### Step 1: Install OS libraries

```
brew cask install xquartz  
brew install gtk+3 boost  
brew install boost-python --with-python3
```

### Step 2: Install Python 2 and/or Python 3

We have already installed Python while installing OpenCV.

### Step 3: Install Python libraries

We have completed this step too.

### Step 4: Compile Dlib


The steps for installing Dlib are listed below

#### Step 4.1: Compile C++ library

Dlib uses few C++11 features which XCode 7 does not support. We have patched Dlib v19.4 by replacing this feature with Dlib's internal function. This patched version of Dlib compiles with XCode 7.

```
git clone https://github.com/vaibhawchandel/dlib.git  
cd dlib  
git checkout v19.4-thread-local-patch  
mkdir build  
cd build  
cmake ..
```





```
cmake --build . --config Release
make install
cd ..
```

## Step 4.2 : Compile Python library

Now activate “facecourse-py2” or “facecourse-py3” virtual environment based on whether you were following the process for Python2 or Python3.

You can install Python bindings for dlib either by compiling yourself or using pip.

```
##### For Python 2 #####
workon facecourse-py2

##### For Python 3 #####
workon facecourse-py3

#####

python setup.py install

# Installation is complete. Now exit from virtual env
deactivate
```

Since we installed dlib in facecourse-py3/facecourse-py2 virtualenv, dlib was installed only in this virtualenv and not globally.

Whenever you want to run a Python script which uses Dlib and/or OpenCV, you have to first activate the virtual environment using command workon.

