Exercises 1 (Week 1) - Installation

Purpose: The purpose of the this exercise is to install and test the main tools and libraries that you will use them during the course. The instructions are known to be working on Mac OS X and Windows. There is a test subsection after each installation part that helps you to check whether the item has been correctly installed. There is also some examples at the end of the installation guide that use all libraries in an interactive ipython notebook (good for making sure that everything is working) and show you how to load and show images and videos in python. You don't need to read and understand the codes for now.

By the end of the class, you should:

- have all the tools installed properly on your computer.
- be able to import different libraries in the python.
- choose a good IDE for python.
- be able to run an ipython notebook.

Main Tools: Today you will install the following items:

- Programming language: Python 2.7
- Packages: NumPy, SciPy, Matplotlib, OpenCV, and IPython
- IDE: Any IDE you prefer. PyCharm, Spyder, Eclipse+PyDev , and WingIDE have good python support.

Windows

Windows users, Lucky you! you have a quick way to get up and running with python and the needed libraries. Download and install the pythonxy it will install the python and all the other libraries you need.

After installing the pythonxy run the python editor *spyder* (which is already installed) and in the python shell write the the lines below and press enter. This is to make sure that all libraries are installed. You shouldn't get any error after entering each line:

```
import numpy
import matplotlib
import scipy
import cv2
```

If you don't want to use pythonxy or if it has not installed the things properly follow the following instruction for installing the python and the other libraries manually.

Python

Download Python 2.7 from < http://python.org/download/>. Install it at the default address (Remember to download the proper 64bit or 32bit version).

After installation, you have to set the PYTHONPATH permanently in the windows. From the control panel go to "System>Advanced system settings>Environment variables" and:

1. Define a user variable with the name of *PYTHONPATH* and the value of `C:\Python27\;C:\Python27\Scripts\;C:\Python27\Lib\site-packages`
2. Define a system variable with the name of *path* and the value of `C:\Python27\;C:\Python27\Scripts\;C:\Python27\Lib\site-packages`. If you already have a variable called *path*, select edit and write a semicolon *";"* at the end of the existing values and add the value above.

Test

Close the windows command prompt if it is open and open it again. Now you should be able to run the Python by writing ”*python*” in the command prompt. (In case you have any problem see HERE)

Other libraries

All the softwares you need can be downloaded from http://www.lfd.uci.edu/ gohlke/pythonlibs/. Just pay attention to the file names you want to download (*X.win-amd64-py2.7.exe* OR *X.win32-py2.7.exe*).

Go to < http://www.lfd.uci.edu/ gohlke/pythonlibs/ >

Find and download NumPy (e.g., numpy-MKL-V.win-amd64-py2.7.exe) and install it.

Find and download SciPy (e.g, scipy-V.win-amd64-py2.7.exe) and install it.

Find and download matplotlib (e.g, matplotlib-V.win-amd64-py2.7.exe) and install it.

Find and download ipython (e.g, ipython-1.1.0.win-amd64-py2.7.exe) and install it.

Find and download OpenCV (e.g., opencv-python-2.4.3.win-py2.7.exe) and install it.

Test

Run the python and import each library as below. If after each line you don't get any errors everything is ok.

```
import numpy
import matplotlib
import scipy
import cv2
```

Mac OS X

XCode

Mac users should have XCode installed on their machine. Download and install the XCode from App Store if you don't already have it installed.
After installing XCode download XCode command tools (Open XCode -> Preferences -> Downloads -> Components -> Command Line Tools)

Python

You don't need to install the python if you have Mac OS Mavericks because Mavericks comes with python 2.7. You can double-check the version of the python, open the terminal and write python and then you can see the python version.

Download Python 2.7 from < http://python.org/download/>. Install it at the default address (Remember to download the proper 64bit or 32bit version).

Test

Close and open the terminal. Now you should be able to run the Python by writing ”*python*” in the terminal. The version of the running python should be 2.7.

NumPy, Matplotlib and SciPy

You can install these three libraries using the scipy SuperPack. For OSX 10.9 find the download link of the Scipy Superpack Installer in this page.

Find the other versions at the superpack repository HERE

Alternately, if you have curl installed, you can get the script via:

```
$ curl -o install_superpack.sh
https://raw.github.com/fonnesbeck/ScipySuperpack/master/install_superpack.sh
```

After downloading the *install_superpack.sh* you can install it. To install, open a terminal in the directory that the script is located and call:

```
$ sh install_superpack.sh
```

Test

Run the python and import each library as below. If after each line you didn't get any errors these libraries has been installed properly.

```
import numpy
import scipy
import matplotlib
```

Alternative installation through brew

Here is an alternative way of installing the scipy libraries using HomeBrew. Type `which brew` in the terminal. If it shows you the version it means you have brew installed and you just need to update it by running `brew install git` and then: `brew update` inside the terminal. If you get any error it means that you need to install [1] the homebrew first:

Inside a terminal run:

```
ruby -e "$(curl -fsSL https://raw.github.com/Homebrew/homebrew/go/install)"
```

Install the fortran compiler:

```
brew install gfortran
```

In a terminal you can download NumPy and SciPy and Matplotlib by typing

```
git clone https://github.com/numpy/numpy.git
git clone https://github.com/scipy/scipy.git
git clone git://github.com/matplotlib/matplotlib.git
```

Now go to the NumPy folder (`cd NumPy`) and run:

```
python setup.py build
python setup.py install
```

Then go to the SciPy folder (`cd SciPy`) and run:

```
python setup.py build
python setup.py install
```

Then go to Matplotlib folder (`cd Matplotlib`) and run:

[1] https://github.com/mxcl/homebrew/wiki/installation

```
python setup.py build
python setup.py install
```

OpenCV

Download and install CMake
Download OpenCV 2.4.8 Extract OpenCV-2.4.8.tar.bz2 and inside the extracted folder create a folder called *build*. Run cmake. Write the address of the OpenCV folder in the source address, and write the address of the *build* folder as the build address. Press the "Configure" button. Update the python paths as below:

`PYTHON_EXECUTABLE`: change this to your python path (To get the python path write "which python" inside the terminal).
`PYTHON_NUMPY_INCLUDE_DIR`: don't change.
`PYTHON_PACKAGES_PATH`: write the python site-packages path that has been automatically detected as the PYTHON_NUMPY_INCLUDE_DIR. (e.g. /Library/Python/2.7/site-packages). Remember the site-package path, you will need that later.

Press the configure button again. Press the Generate button.
Open the terminal in the *build* folder inside the opencv folder and Run:

```
make -j8
sudo make install
```

Test

Open the python and import opencv by writing :
`import cv2`
If opencv has been installed properly you shouldn't get any error. If you get the segmentation error after importing cv2, it may be because of the python path in your bash profile. Add the path to the python site-package folder into your .bash_profile (open the bash profile using `open .bash_profile`) and add the path (e.g., XXX/Python/2.7/site-packages) to the file:
`export PYTHONPATH=/usr/local/lib/python2.7/site-packages/:$PYTHONPATH".`

Alternative installation through brew

Here is an alternative way of installing the scipy libraries using HomeBrew. Type `which brew` in the terminal. If it shows you the version it means you have brew installed and you just need to update it by running `brew install git` and then: `brew update` inside the terminal. If you get any error it means that you need to install [2] the homebrew first:

Inside a terminal run:
```
ruby -e "$(curl -fsSL https://raw.github.com/Homebrew/homebrew/go/install)"
```

brew will automatically install CMake and checks the OpenCV repository.
First `sudo brew install ffmpeg`
and then
for 64bit: `sudo brew install opencv`
or
for 32bit: `sudo brew install opencv --build32`

IPython

The quickest way to get up and running with IPython is to write the line below in the terminal:
`sudo easy_install ipython[all]`
If the it doesn't work find the alternative ways of installing ipython HERE.

[2] https://github.com/mxcl/homebrew/wiki/installation

Test

Open the terminal and write `ipython`. Inside the ipython shell import the libraries you have already installed:

```
import numpy
import scipy
import matplotlib
import cv2
```

You shouldn't get any error. In case ipython cannot import cv2 it is because ipython is using another version of python. you can fix this by editing the first line of the ipython file. Exit the ipython and write `which ipython` inside the terminal and find the address of the ipython. Go to the ipython folder and open it in a text editor and correct the first line of the file. You need to write the correct path of your python (the one you can see after writing the `which python` in the terminal) in the first line. This will fix the problem of importing the cv2. Another thing you need to test is the ipython notebook. Open the terminal and write `ipython notebook`. It will opens up an interactive ipython shell (ipython notebook) on your browser. If you don't see the ipython notebook and you get the error `infamous ValueError: unknown locale` you can fix that by adding a path to your .bash profile file:

```
export LC_CTYPE="en_US.UTF-8"
```

Examples

Open the folder *exercises1*. Open the terminal inside this folder. Inside the terminal write the following command line that creates an ipython notebook server:

```
ipython notebook
```

This will opens up a page on your browser. Inside the page you will see the list of the notebooks exist in the *exercises1* folder. Select the one that is *Ex1_notebook*. If you don't see the name *Ex1_notebook* in the list find the file *Ex1_notebook.ipynb* in the exercises folder and drag that onto the listing field of the browser page. After opening the notebook you will see an interactive python programming page with some examples in it. You don't need to change the codes or understand them. Just run the code (press run in the Cell menu) and see how the code opens an image and a video file in python.
