Appendix: Python Tutorial

http://cs231n.github.io/python-numpy-tutorial/

Table of contents:

1. Python
   - Basic data types
   - Containers
     - Lists
     - Dictionaries
     - Sets
     - Tuples
   - Functions
   - Classes
2. Numpy
   - Arrays
   - Array indexing
   - Datatypes
   - Array math
   - Broadcasting
3. SciPy
   - Image operations
   - MATLAB files
   - Distance between points
4. Matplotlib
   - Plotting
   - Subplots
   - Images

Lists: Python Containers

Python includes several built-in container types: lists, dictionaries, sets, and tuples.

A list is the Python equivalent of an array, but is resizeable and can contain elements of different types:

Example: Python code of lists

```
xs = [3, 1, 2]          # Create a list
print(xs, xs[2])        # Prints "[3, 1, 2] 2"
print(xs[-1])           # Negative indices count from the end of the list; prints "2"
xs[2] = 'foo'           # Lists can contain elements of different types
print(xs)               # Prints "[3, 1, 'foo']"
xs.append('bar')        # Add a new element to the end of the list
print(xs)               # Prints "[3, 1, 'foo', 'bar']"
x = xs.pop()            # Remove and return the last element of the list
print(x, xs)            # Prints "bar [3, 1, 'foo']"
```

numpy 2D Array

Python includes several built-in container types: lists, dictionaries, sets, and tuples.
Example: Python 2D array:

```
test = [1, 2, 3]
test[0] = [1, 2, 3]
test[1] = [1, 2, 3]
test[2] = [1, 2, 3]
```

Defined array dimension

```
ubuntu@ubuntu-ThinkPad-Yoga-14:~/OpenCV/samples/cpp/harry-test/lec3-videoCapture$ python
Python 2.7.6 (default, Nov 23 2017, 15:49:48)
[GCC 4.8.4] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> test = [1, 2, 3]
>>> test[0] = [1, 2, 3]
>>> test[1] = [1, 2, 3]
>>> test[2] = [1, 2, 3]
>>> print(test)
[[1, 2, 3], [1, 2, 3], [1, 2, 3]]
>>>
```

Example: use numpy for openCV program

```
import numpy as np
a=np.array([[1,1,1],[2,2,2],[3,3,3]])
```

```
>>> a=np.array([[1,1,1],[2,2,2],[3,3,3]])
>>> a
array([[1, 1, 1],
       [2, 2, 2],
       [3, 3, 3]])
>>>
```

numpy 2D Convolution in OpenCV

```
# HL: convolution.py
# 1. define user kernel
import numpy as np
import cv2

cap = cv2.VideoCapture(0)
print ('type q to quit')
while(True):
    # Capture frame-by-frame
    gray, frame = cap.read()
    # Display the resulting frame
    cv2.imshow('Harry: colour video', frame)
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    cv2.imshow('Harry: gray video',gray)
    kernel = np.array([[1, 0, -1],
                       [1, 0, -1],
                       [1, 0, -1]])
    dst = cv2.filter2D(gray,-1,kernel)
    cv2.imshow('Harry: convolution on gray', dst)
    if cv2.waitKey(1) & 0xFF == ord('q'):
       cap.release()
       cv2.destroyAllWindows()
        break
```

```
cv2.VideoCapture(0)
cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
cv2.filter2D(gray,-1,kernel)
```