

Assignment 0: Introduction to Python for Computer Vision

This assignment is a self-directed study to introduce you to the basics of Python for computer vision. There is nothing to hand in and no marks to be given. You should aim to complete this assignment by the date specified on the course web page.

For this assignment and most of the others you will have a choice of the environment to use.

1. You can use undergraduate unix lab machines remotely, which come pre-loaded with Enthought Python Distribution (EPD) with **Python 3**. The default is Python version 3.6.10 on undergraduate unix lab machines. The following libraries that we will use in the assignments are also installed **OpenCV** (cv2) version 3.3.1, **PIL** version 2.0, **numpy** version 1.17.3 for python 3, and **scipy** version 1.3.3 for python 3.
2. You can install and use python and corresponding libraries on your personal machine. To do so please install Enthought Python Distribution available at <https://www.enthought.com/enthought-deployment-manager/>.
3. You can use the CoLab environment made available by Google: <https://colab.research.google.com>. Note that unlike the other two environments, this one uses Jupyter Notebooks. This has a couple of benefits. The main ones are (a) you can produce your write-ups directly from the notebook by exporting PDFs, and (b) it makes it easier for TAs and myself to potentially help you with code issues. There are many instruction guides on the web and YouTube that show you how to use CoLab. For example this one <https://www.geeksforgeeks.org/how-to-use-google-colab/> is pretty good; I will also ask TAs to conduct a training session. Please note that Assignment 6 will require use of CoLab, so in the interest of getting familiar with the environment you are encouraged to start earlier.

Note to export a PDF from your Jupyter Notebook in CoLab you will need to make sure the copy of the notebook is available in the working directory and add, and run, the following commands from a cell.

```
!apt-get install texlive texlive-xetex texlive-latex-extra pandoc
!pip install py pandoc
!jupyter nbconvert --to PDF "Assignment0.ipynb"
```

We will assume that by default you are using versions of python and corresponding libraries that are available on the undergraduate unix lab machines. If you use different versions of the tools please specify this on top of the submitted file in the comments. If TAs are unable to reproduce your results with a reasonable amount of effort, points will be deducted. Of course this does not apply to this assignment as it is ungraded, but will apply to all the future ones.

The Assignment

1) Python Distribution

On an undergraduate lab machine: To start the Python distribution from a unix shell, do the following:

```
# start Python 3
python3
```

Note: “>>>” is the Python shell prompt (and you're ready to go).

On your personal machine: Enthought installs both Python and a collection of programming libraries for scientific and mathematical work. The key libraries we are interested in are PIL, numpy, matplotlib, and scipy; they will be shipped with Enthought. Enthought can be installed on Windows, OSX, or Unix.

Note: Please check the Python and Canopy version if you are using a department machine. In general, we are not very strict about versioning; however, some starter code might break if you are using a different version. Most of them should be an easy fix, e.g., `print a` in Python 2 and `print(a)` in python 3.

2) Working with Python

Python v3.6.10 documentation (<https://docs.python.org/release/3.6.10/>) is the official documentation page for the Python 3. It includes the tutorial (<http://docs.python.org/release/3.6.10/tutorial/index.html>) for getting started. This Python Quick Guide is suggested as appropriate for someone in a 400-level Computer Science course who is not otherwise familiar with Python. For CPSC 425, you will need to know how to perform flow control and mathematical operations, how to create functions, how Python handles scoping through indentation, and file reading/writing. Follow the Python Quick Guide (or equivalent) and make sure you are comfortable with everything in the Basic Tutorial up to (but not including) "Exceptions." The material in the Advanced Tutorial, "Classes/Objects" (and beyond) won't be needed in this course.

Additionally, you will be using numpy extensively and many lines of code in your assignments can be replaced with a few lines of numpy code. Please study a introduction tutorial on numpy ([a href="https://docs.scipy.org/doc/numpy/user/quickstart.html"](https://docs.scipy.org/doc/numpy/user/quickstart.html)). For more numpy materials, please visit this page (<https://docs.scipy.org/doc/numpy/user/basics.html>).

Some students might prefer watching tutorials as opposed to learning the documentation. You may find the Python and numpy tutorials on YouTube helpful.

3) Specific Examples

First download this test image, [peacock.png](#) (Image credit: Tristram Southey).

Since the version of Python we are using is an interpreted language, you can either just start Python at the command line and work from there or work from your favourite text editor and run the Python code. In future you will need to submit your code but not for this assignment.

```
# import the packages we need for this assignment
from PIL import Image
import numpy as np

# open the test image
# Note: If you didn't launch Python from the same directory where you saved
#       the file, peacock.png, you'll need to provide the full path name as
#       the argument to Image.open
im = Image.open('peacock.png')

# display relevant Image class attributes: dimensions (width, height),
# pixel format and file format
print im.size, im.mode, im.format
```

```
# Note: PIL does not have a built-in image display tool. Instead, principally
# for debugging, there's a show method which saves an image to a temporary file
# on disk and calls a platform dependent external display utility
# (the default being "xv" on unix, and the "Paint" program on Windows).

# display the image
im.show()

# if this does not work on your system, try the imshow function from
# matplotlib's pyplot module by uncommenting the next three lines
# import matplotlib.pyplot as plt
# plt.imshow(im)
# plt.show()

# convert the image to a black and white "luminance" greyscale image
im = im.convert('L')

# select a 100x100 sub region (containing the peacock's head)
im2 = im.crop((475,130,575,230))

# save the selected region
im2.save('peacock_head.png', 'PNG')

# PIL and numpy use different internal representations
# convert the image to a numpy array (for subsequent processing)
im2_array = np.asarray(im2)

# compute the average intensity
average = np.mean(im2_array)

# Note: we need to make a copy to change the values of an array created using
# np.asarray
im3_array = im2_array.copy()

# add 50 to each pixel value (clipping above at 255, the maximum uint8 value)
# Note: indentation matters
for x in range(0,100):
    for y in range(0,100):
        im3_array[y,x] = min(im3_array[y,x] + 50, 255)

# convert the result back to a PIL image and save
im3 = Image.fromarray(im3_array)
im3.save('peacock_head_bright.png', 'PNG')

# again make a copy of the (original) 100x100 sub-region
im4_array = im2_array.copy()

# this time, reduce the intensity of each pixel by half
# Note: this converts the array to a float array
im4_array = im4_array * 0.5

# convert the array back to a unit8 array so we can write to a file
im4_array = im4_array.astype('uint8')

# convert the numpy array back to a PIL image and save
im4 = Image.fromarray(im4_array)
im4.save('peacock_head_dark.png', 'PNG')

# let's generate our own image, a simple gradient test pattern
# make a 1-D array of length 256 with the values 0 - 255
grad = np.arange(0,256)

# repeat this 1-D array 256 times to create a 256x256 2-D array
grad = np.tile(grad,[256,1])
```

```
# convert to uint8 and then to a PIL image and save
im5 = Image.fromarray(grad.astype('uint8'))
im5.save('gradient.png', 'PNG')
```

Downloads

- Jupyter Notebook for Python 3 (to be used with CoLab): [Assignment0.ipynb](#).
- Python 3 file: [Assignment0_py3.py](#).

Deliverables

There is nothing to hand in for this assignment and no marks will be given. The objective is for you to get the Enthought Python Distribution (EPD) up and running on the platform of your choice, sufficient to run the above examples. The specific examples chosen demonstrate basic functionality needed in future assignments. Thus, it is to your benefit also to try to understand what is happening at each step in each example.

© 2020 Leonid Sigal | Design by [Andreas Viklund](#)