

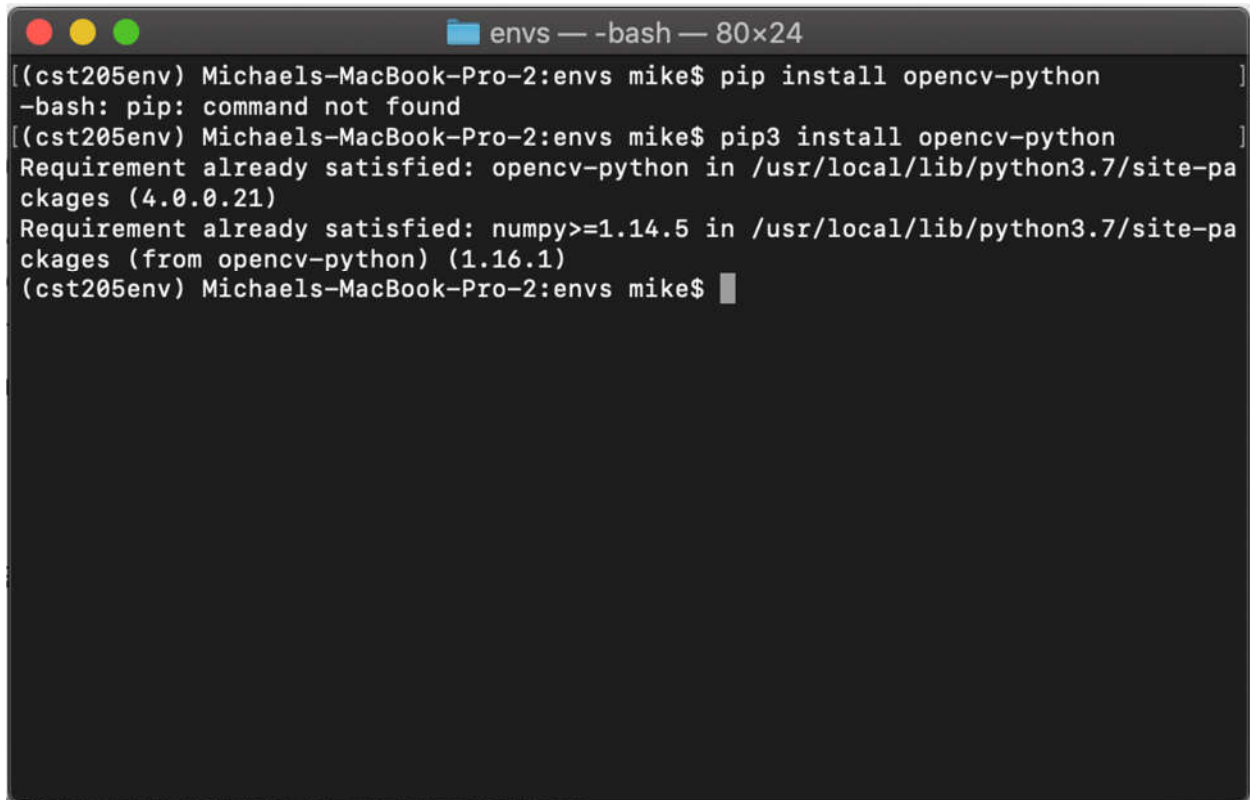
Title: Lab 14: OpenCV

Partners: Michael Avalos-Garcia, Jesus Andres Bernal Lopez, Paul Whipp

Date: 3/25/19

Summary: We all have already installed these packages, as confirmed by the screenshot below.

Task1:

A terminal window titled 'envs — -bash — 80x24' with standard macOS window controls (red, yellow, green buttons). The terminal shows a user 'mike' in an environment 'cst205env' on a 'Michaels-MacBook-Pro-2'. The user runs 'pip install opencv-python', which fails with '-bash: pip: command not found'. Then, the user runs 'pip3 install opencv-python', which succeeds, showing that 'opencv-python' is already installed at version 4.0.0.21 and 'numpy' is also satisfied at version 1.16.1.

```
[(cst205env) Michaels-MacBook-Pro-2:envs mike$ pip install opencv-python ]
-bash: pip: command not found
[(cst205env) Michaels-MacBook-Pro-2:envs mike$ pip3 install opencv-python ]
Requirement already satisfied: opencv-python in /usr/local/lib/python3.7/site-pa
ckages (4.0.0.21)
Requirement already satisfied: numpy>=1.14.5 in /usr/local/lib/python3.7/site-pa
ckages (from opencv-python) (1.16.1)
(cst205env) Michaels-MacBook-Pro-2:envs mike$ █
```

Task 2:

```
1  # 3/27/2019 -- CST 205 -- Lab 14 -- OpenCv
2  # By Michael Avalos-Garcia, Jesus Andres Bernal, and Paul Whipp
3  # Description: This program uses Lab13 code to obtain the url of an image, and
4  #              then loads that image into memory to be used within the program.
5
6  from bs4 import BeautifulSoup
7  from urllib.request import Request, urlopen
8  import numpy as np
9  import cv2
10
11 # My website from CST 336
12 my_site = 'http://pwhipp-cst336.herokuapp.com/labs/lab1/'
13 default_img_url = 'http://pwhipp-cst336.herokuapp.com/labs/lab1/img/poodle_dance.jpg'
14
15 def first_img_found(my_site):
16     # package up request info
17     req = Request(
18         my_site,
19         headers={'User-Agent': 'Mozilla/5.0'}
20     )
21     # make request, convert to BeautifulSoup object
22     resp = urlopen(req)
23     bs_obj = BeautifulSoup(resp.read(), 'lxml')
24     for link in bs_obj.findAll("img"):
25         if 'src' in link.attrs:
26             return my_site + str(link.attrs['src'])
27     return default_img_url
28
29 resp = urlopen(first_img_found(my_site))
30 image = np.asarray(bytearray(resp.read()), dtype="uint8")
31 image = cv2.imdecode(image, cv2.IMREAD_COLOR)
32
33 cv2.imshow("Result", image)
34 cv2.waitKey()
```

The code given:

```
# make sure you have this import --> from urllib.request import urlopen
r = urlopen(" ... your image URL here ... ")
image = np.asarray(bytearray(r.read()), dtype="uint8")
image = cv2.imdecode(image, cv2.IMREAD_COLOR)
```

Stores an image url in `r` (resp in our code) and then takes the image url and converts the image into an array of raw, mutable data bytes. It then returns an image from a buffer in memory. The last line, reads the image from the buffer in memory and returns a matrix.

Task 3:

For this task we amended our Task 2 code with the following:

```
image2 = cv2.applyColorMap(image, cv2.COLORMAP_RAINBOW )  
cv2.imshow("Result of Rainbow Map", image2)  
cv2.waitKey()
```

This applies a rainbow effect, then displays the new image, waiting for a keypress before closing.