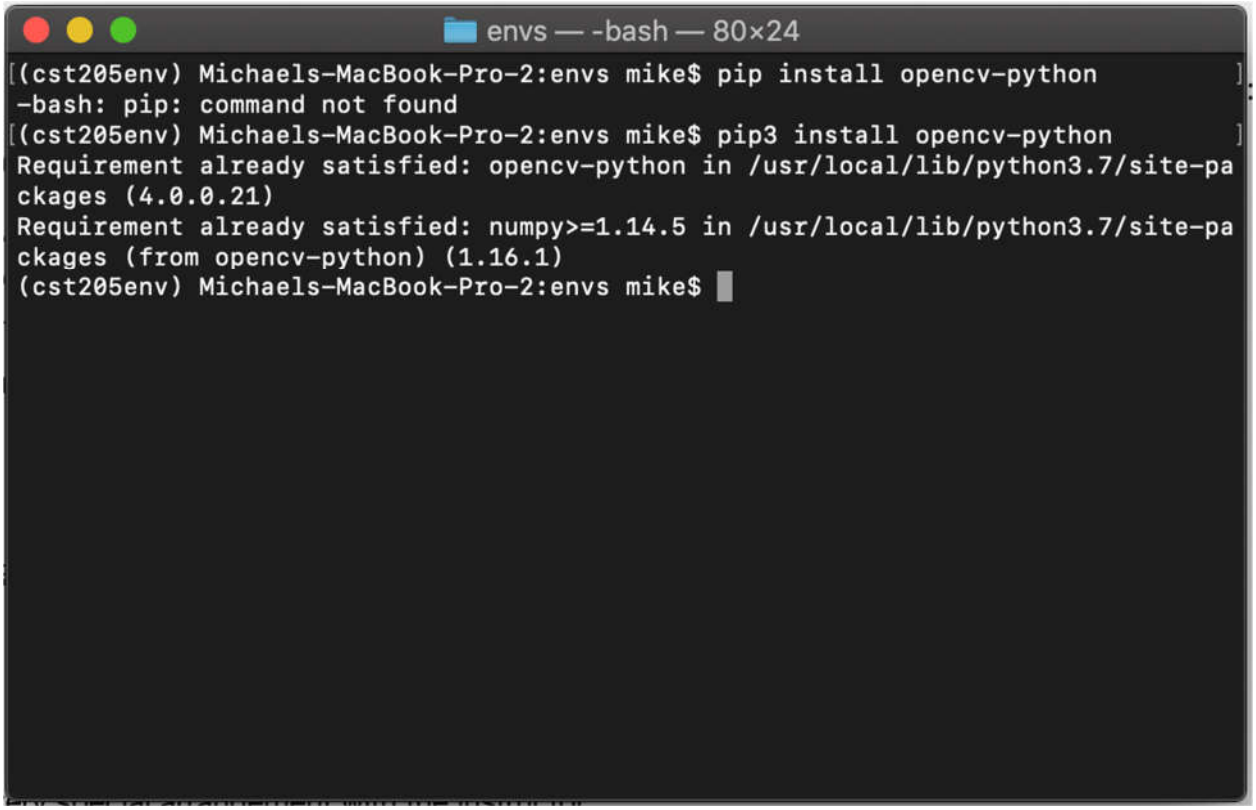
Title: Lab 14: OpenCV

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:



Task 2:

```python
# 3/27/2019 -- CST 205 -- Lab 14 -- OpenCv
# By Michael Avalos-Garcia, Jesus Andres Bernal, and Paul Whipp
# Description:  This program uses lab13 code to obtain ther url of an image, and
#               then loads that image into memory to be used within the program.

from bs4 import BeautifulSoup
from urllib.request import Request, urlopen
import numpy as np
import cv2

# My website from CST 336
my_site = 'http://pwhipp-cst336.herokuapp.com/labs/lab1/'
default_img_url = 'http://pwhipp-cst336.herokuapp.com/labs/lab1/img/poodle_dance.jpg'

def first_img_found(my_site):
    # package up request info
    req = Request(
        my_site,
        headers={'User-Agent': 'Mozilla/5.0'}
        )
    # make request, convert to BeautifulSoup object
    resp = urlopen(req)
    bs_obj = BeautifulSoup(resp.read(), 'lxml')
    for link in bs_obj.findAll("img"):
        if 'src' in link.attrs:
            return my_site + str(link.attrs['src'])
    return default_img_url

resp = urlopen(first_img_found(my_site))
image = np.asarray(bytearray(resp.read()), dtype="uint8")
image = cv2.imdecode(image, cv2.IMREAD_COLOR)

cv2.imshow("Result", image)
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.