Primrose
Machine Learning Experts

Today we will practice the OpenCV library binding to the python langauage. OpenCV is written in C++ and has a huge set of functionalities.

1. Install OpenCV using the command: pip install opencv-python
2. If you get an error that "Visual Studio Redistributabel package is missing - install it. Google it and you will get to the correct page.
3. The import command for OpenCV in python is: import cv2
4. Load a color image using imread http://docs.opencv.org/3.0-beta/doc/py_tutorials/py_gui/py_image_display/py_image_display.html
5. Show the image and then keep the window open for 10 seconds (imshow, waitkey)
6. Show the image and then keep the window open till a command is entered in the window
7. Print the value of the pixel (RGB) at location (100,100) http://docs.opencv.org/3.0-beta/doc/py_tutorials/py_core/py_basic_ops/py_basic_ops.html#basic-ops
8. Print only the blue value of this pixel
9. Access this pixel value and then modify it's value using the "item" and "itemset" functions
10. Print the image shape, size and dtype values to the screen
11. Choose a rectangluar area in the image, copy it's values to a variable and then paste it in another area in the pixel
12. Split the RGB image into three different images: R,G,B and show them. Use the "split" function
13. Merge these values back to a RGB image using the "merge" function
14. Do the same operation now with NumPy indexing of the image
15. Add borders around the image: http://docs.opencv.org/2.4/doc/tutorials/imgproc/imgtrans/copyMakeBorder/copyMakeBorder.html
    Use two differnet flags: `BORDER_REPLICATE` 2. `BORDER_CONSTANT`
16. Convert the RGB image into a Gray level image using "cvtColor" and "show"
17. Create a binary image from the Gray image using "threshold" and show
18. Blur the image using "GaussianBlur" and show
19. Create a Gaussian kernel and filter using "`filter2D`"
20. Create a three level pyramid using "pyrDown" and "pyrUp"
21. Resize the image using "resize" and show
22. Perform an Affine transformation using "warpAffine"

23. Create a rotation matrix from rotation angle using: getRotationMatrix2D and thenperform the affine transofrm using warpAffine