School of Computing
National University of Singapore
Biometrics Course
July 2016

Assignment #2

Part I

This part is to acquaint you with basic Python Image processing.

1. Check that all libraries have been installed on your machine. If not, install them in the following order.

- Numpy, Scipy, Pillow, Matplotlib

2. Open IDLE and change current working directory to your working directory, **e.g.**, *e:/myfolder*.

```
import os
os.chdir("e:/myfolder")
```

3. List the files in the current working directory.

```
os.listdir(".")
```

4. Read an display image with PIL.

- Read image.

  ```
  from PIL import Image
  img = Image.open('lena.jpg')
  ```

- Display image information.

  ```
  print(img.format, img.size, img.mode)
  ```

- Display image.

  ```
  img.show()
  ```

5. Apply simple image processing function.

```
from PIL import ImageEnhance
enhancer = ImageEnhance.Contrast(img)
enhanced_img = enhancer.enhance(2.0)
enhanced_img.show()
```

6. Save image to a file.

```
enhanced_img.save('out.jpg')
```

7. Image and array conversion.

```
import numpy as np
img_array = np.asarray(img)
img = Image.fromarray(img_array)
```

Part II

1. A = [1, 3, 2, 4; 2, 2, 3, 4; 5, 5, 4, 5; 8, 9, 0, 1], B = [1, 2, 3, 4; 2, 1, 3, 0; 4, 1, 3, 4; 2, 4, 3, 4]. Please compute the convolution between A and B by hand. They try to verify your answer by Python code. (Hint: from scipy import signal, use signal.convolve). (4)

2. Study Numpy, Scipy, PIL and Matplotlib libraries. Use lena.png to perform following operations and save the images: (11)

- In the image, rotate a rectangular region by 45 degree counter-clockwise, whose vertices are (100,100), (100,400),(400,100),(400,400). (3)
- Perform histogram equalization on lena.png. Use matplotlib to plot the histogram figure for both original image and processed image. (3)
- Perform Max Filtering, Min Filtering, and Median Filter on lena.png. (3)
- Perform Gaussian Blur with sigma equal to 3 and 5. (2)

3. Color space conversion. Use Python OpenCV functions to perform following operations on 'bee.png' and save the images. (5)

- Read the image.
- Convert the image to HSV color space.
- Perform histogram equalization on V channel by cv2.equalizeHist().
- Convert the result image to BGR color space.
- Show the image by cv2.imshow() and save the image.

4. Edge detection by Sobel filter. Follow the instruction and understand how the edges are detected. (10)

- Read an image as gray scale: img = cv2.imread('test.png',0)
- Construct Sobel's gradient operator Gx, Gy.
- Convolve the image with Gx, Gy and get the gradient maps Mx, My. Hint: use convolve2d() instead of convolve()
- Compute the magnitude the gradient maps: $M = \sqrt{Mx^2 + My^2}$
- Normalize the gradient magnitudes: $M = 255 \times M/max(M)$
- Convert the result to 'uint8' type and save the result: M = np.array(M, dtype='uint8')

Use your own images and run the algorithm.

Submission

Please submit all your code and images for Part II. The code for each question should be in a separate file, named as Qn.py where n is the question number. Zip all your files into a single file and name it as XXX_assignment2.zip, where XXX is your name in English. Upload the zip file to IVLE.

Deadline: 18 July 2:00pm.