CS 4650/7650 ECE 4655/7655 Digital Image Processing 2024
Homework 1A: Point Processes in Python [40 pts]

Out: Tuesday August 27, 2024

Due: Thursday Sep 5 (Midnight)

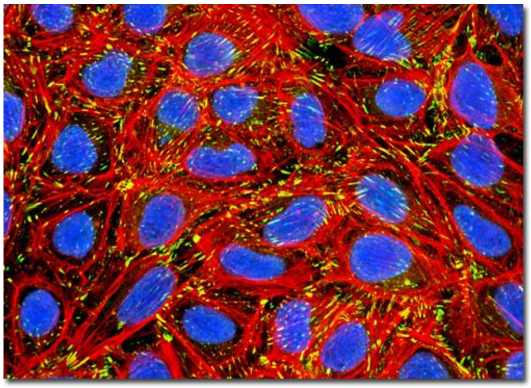

Image credit: https://micro.magnet.fsu.edu/primer/techniques/fluorescence/gallery/cells/u2/u2cellslarge14.html

The goals of this first assignment are:
(1) to become familiar with OpenCV library and Python functions needed in image processing;
(2) trying out basic image read, write, access, display and plotting functions; and
(3) review of point process (intensity transformation) concepts.

[20 pts] Part A1 - Color to Grayscale Conversion (Satisfactory/Unsatisfactory)

Write Python code using the OpenCV library to read a color image (RGB), convert it to a grayscale image and write it out to disk.

Use the following formula to convert from color to grayscale:

```
Gray[x][y]= 0.2989 * Red[x][y] + 0.5871 * Green[x][y] + 0.1140 * Blue[x][y]
```

Note that you will need to use the actual image data structure for accessing the red, green and blue channels.

Program name and arguments:

```
./rgb2gray input_color_image_filename output_grayscale_image_filename
```

[20 pts] Part A2 - Image Binarization (or Thresholding)

Goal: Segmentation of the bone cell nuclei (blue elliptical regions in the image) using image thresholding.

1) Use the grayscale image (computed in Part A1) and convert it to a binary image using a user provided threshold value on the command line and write out the binarized image to disk.

2) Use a color channel of choice and convert it to a binary image using a user provided threshold value on the command line and write out the binarized image to disk.

Program name and arguments:

```
./gray2binary input_grayscale_image_filename output_binary_image_filename threshold_value
```

Implementation issues to consider and be aware of:

- input pixel data type (unsigned char vs float vs double),
- computation pixel data type,
- output pixel data type,
- accessing color channels,
- image dimensions,
- row-column indexing,
- zero vs one convention (array vs matrix representation),
- input and output image file formats

OpenCV (or other libraries) are used read and write a variety of image file formats and include an image display function.

Submission Instructions: Submit electronic version from Canvas (https://courses.missouri.edu/). Submit an archive (tar) or zip file that contains the following directories and files:

1. src - should contain all code files
2. A report in PDF format (not Word or other formats that are difficult to handle with images and equations). Note that you can print to PDF.
3. Images (if any) should be in a separate folder - for example if you tried your own images

Tar file naming convention: DIPFall2024_LastName_FirstName_Assignment#

Windows users can use 7zip to create tar files.

Report Structure: Your Written Report must conform to the following format.

1. Header
   - Course number and name
   - Assignment number and title
   - Your name
   - Date
2. Abstract (2 to 4 sentences describing the problem and the major results obtained)
3. Introduction (A brief statement of the experiments to be performed and the goal of the assignment)
4. Experiments and Results (figures, tables, graphs, …). You need to use given test images.
   a. Input RGB image
   b. Output grayscale image

    c. Input grayscale image
    d. Binarized result 1
    e. Binarized result 2

5. Discussion/Conclusions

   Did the programs work as expected?

   Are the results satisfactory? Why/Why not?

6. References/Appendix (As needed)

References

1. General tools that are useful to look at your input and output images: NIH ImageJ or Fiji which is an extension of ImageJ (opensource Java), imageMagick (opensource), gimp (opensource C++), Paint.Net, irfanview (shareware), Adobe Photoshop, Preview (MacOS), Windows PhotoViewer.
2. Lec2_Introduction_IP2.pdf
3. Lect3_IntensityTransformations.pdf
4. Canvas\Modules\Programming Resources