IE 423 Quality Engineering
Project Part 1, due October 30th, 2019

Instructions: Please solve the following exercises using R (http://www.r-project.org/) or Python (https://www.python.org/) as a group of at most 5 members. You are expected to use GitHub Classroom and present your work as an html file (i.e. web page) on your progress journals. There are alternative ways to generate an html page for you work:

- A Jupyter Notebook including your codes and comments. This works for R and Python, to enable using R scripts in notebooks, please check:
    - https://docs.anaconda.com/anaconda/navigator/tutorials/r-lang/
    - https://medium.com/@kyleake/how-to-install-r-in-jupyter-with-irkernel-in-3-steps-917519326e41

  Things are little easier if you install Anaconda (https://www.anaconda.com/). Please export your work to an html file. Please provide your *. ipynb file in your repository and a link to this file in your html report will help us a lot.

- A Markdown html document. This can be created using RMarkdown for R and Python-Markdown for Python

Note that html pages are just to describe how you approach to the exercises in the homework. They should include your codes. You are also required to provide your R/Python codes separately in the repository so that anybody can run it with minimal change in the code. This can be presented as the script file itself or your notebook file (the one with *.ipynb file extension).

The last and the most important thing to mention is that academic integrity is expected! Do not share your code (except the one in your progress journals). You are always free to discuss about tasks but your work must be implemented by yourself. As a fundamental principle for any educational institution, academic integrity is highly valued and seriously regarded at Boğaziçi University.

INTRODUCTION - Quality Control on Images

You are a group of Quality Engineers in a linen manufacturer. “Linen is a textile made from the fibers of the flax plant. Linen is laborious to manufacture, but the fiber is very absorbent and garments made of linen are valued for their exceptional coolness and freshness in hot weather” (Source: Wikipedia).

“Automation of the visual inspection for quality control in production of materials with textures (tiles, textile, leather, etc.) is not widely implemented. A sophisticated system for image acquisition, as well as a fast and efficient procedure for texture analysis is needed for this purpose” (Source: Rimac-Drlje, Snježana, Drago Žagar, and Slavko Rupčić. "Adaptive Image Processing Technique for Quality Control in Ceramic Tile Production." Strojarstvo 52.2 (2010): 205-215.). Suppose you already have the image acquisition system and you are able to obtain pictures of linen (with a resolution of 200x200) as in Figure 1.

As the first part of the project, you are requested to work with any surface (i.e. desk, wall, curtain, sweater and etc.) you would like. In other words, you are not restricted to work with linen images for this part of the project. Linen images to be worked on will be provided for the last part of the project. Please follow the steps below to obtain the image you will work on for this exercise:

- Take a picture of the surface and save it as **.jpg* or **.jpeg* file.
- Using an image editor (i.e. *Paint* in Windows), crop the surface from the image and save it as separate image file (again **.jpg* or **.jpeg*). The resulting image is expected to be a square (i.e. has the same length and width), hence you should crop the image accordingly.
- Resize the cropped image to size 512x512 px (pixel) using an image editor. This image is the one that you will use for this exercise.

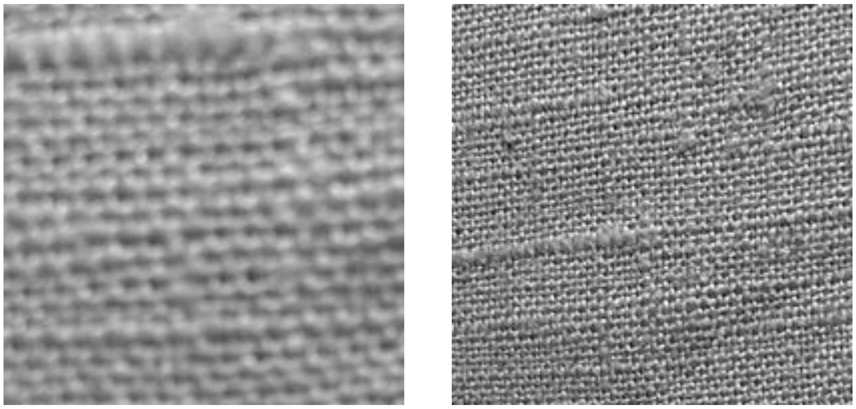


Figure 1: Sample linen images

BACKGROUND - Images

Here is a background information about how a grayscale image is represented on our computers.A grayscale image is basically a matrix where each matrix entry shows the intensity (brightness) level. In other words, when you take a picture with a digital camera, the image is represented by a numerical matrix where the matrix size is defined by the resolution setting of your camera. If your resolution setting is 1280x720, then your image is represented by 1280x720= 921600 pixel values (Actually that is why higher resolution provides better quality pictures).

When you have a color image, the image stores the information of multiple channels depending on the image type. The most famous one is RGB type where R, G and B stand for “red”, “green” and “blue” respectively. Hence, you have a matrix as in greyscale images representing the intensity for each channel. Combining these matrices generates the color image.

TASKS

Part 1 *(50 points)*

1- Read image as a variable in R/Python. For R, you need to install “jpeg” package to read image into a variable. The variable you obtain is an array with three dimensions (suppose you read image to a variable “img”, try running “str(img)” command to see the structure of the variable. The size of the array should be *512 x 512 x 3* for the three dimensions). If you prefer Python, please find the appropriate module for this task. *(5 points)*

2- What is the structure of the variable that stores the image? What is the dimension? *(10 points)*

a. Display the image. (Hint: google “rasterImage” for R)

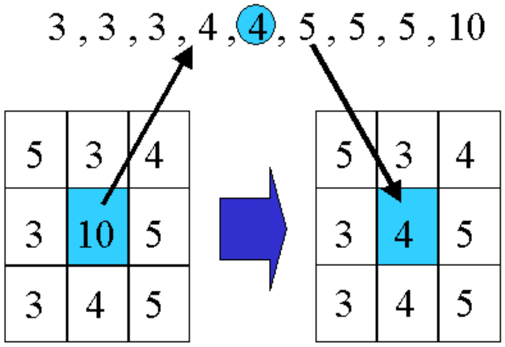
b. Display each channel using “image” function on a single plot. (Hint: google “multiple plots in r” for R)

3- For each channel, take the average of the columns and plot the average as a line plot for each channel on a single plot. *(10 points)*

4- For each channel, subtract one half of the image from the other half (choice of halves is up to you but cropping the image vertically/horizontally into two parts make more sense) and display the new image. (Hint: google “rasterImage” for R) *(10 points)*

5- Median filtering: In signal processing, it is often desirable to be able to perform some kind of noise reduction on an image or signal. The median filter is a nonlinear digital filtering technique, often used to remove noise. Such noise reduction is a typical pre-processing step to improve the results of later processing.

The main idea of the median filter is to run through the image matrix entry by entry, replacing each entry with the median of neighboring entries. The pattern of neighbors is called the "window", which slides, entry by entry, over the entire signal. For 1D signals, the most obvious window is just the first few preceding and following entries, whereas for 2D (or higher-dimensional) signals such as images, more complex window patterns are possible (such as "box" or "cross" patterns). Note that if the window has an odd number of entries, then the median is simple to define: it is just the middle value after all the entries in the window are sorted numerically. For an even number of entries, there is more than one possible median. An example for median filtering is provided in the figure below for 3x3 window:



5	3	4
3	10	5
3	4	5

5	3	4
3	4	5
3	4	5

Apply median filtering to each channel of the image with the following window sizes, display the images (as a color image like in part 2-a) and discuss about the effect of window size based on your observations: *(15 points)*

a) 5x5
b) 11x11
c) 31x31

Part 2 ***(50 points)***

Before working on the tasks, please transform your color image to a greyscale one using an image editor. The other option is to use some package/module to perform the transformation. You will be working on greyscale images for the following tasks:

1. Suppose we are interested in the pixel value distribution of our image. Draw the histogram of the pixel values. Provide an appropriate probability distribution that fits well to the shape you observe (i.e. is it like Normal distribution, uniform or other?). *(5 points)*

2. Assume that pixel values are following the distribution you have proposed in the previous task. Estimate the parameters using the data. (i.e. mean and variance for Normal distribution or min and max for uniform distribution and etc.). *(5 points)*

3. Let's say the pixel values follow the distribution you proposed and its parameters are equal to what you have estimated in part 2. Identify the pixels that are out of the 0.001 probability limits. In other words, find a lower and upper bound that leave 0.001 of the observations on the smaller and larger side of the distribution respectively. Pixels that are out of these bounds should be identified. After finding those pixels, change the value of these pixels to zero (i.e. black color). Display the new image and original image in a plot. What are your observations? Comment on your findings. *(20 points)*

4. Suppose we would like to perform the same operation on the patches of images (i.e. windows of certain size). When local structures are important, performing image operations on the patches might be important. Assume that your window size is 51x51 and you repeated what you have done in the first three tasks. Note that you do not need to draw each patch as requested in the previous task (i.e. task 3) but you are expected to mark the pixels for every patch. After finding those pixels, change the value of these pixels to zero (i.e. black color). Display the new image and original image in a plot. What are your observations? Comment on your findings. *(20 points)*