Phenomena such as inefficient or excessive lighting, motion blur, and sudden scene changes can degrade the quality of recorded digital images. These phenomena often result in obtaining "*noisy*" images. In order to enhance images and make them more suitable for further processings, some preprocessing operations are usually needed. In this lab, you will implement some of these operations in grayscale images.

Important Note: You should complete the lab until the end of the lab hours and submit all your codes to SUCourse as a single zip file. Deadline for in-lab code submission to SUCourse is **17:00**.

Things to do:

Your functions must be as generic as possible, i.e., don't make any assumptions about the size, the type and the colors of the images. Your functions must convert the image to grayscale if it is colored and you must employ the row and coloumn numbers of the images as variables.

- Histogram equalization is an example of a point operator in which the contrast of an image is adjusted using the image's histogram.

Apply histogram equalization to the given image below using MATLAB's built-in "**histeq**" function, and obtain histograms of both original and resulting images using "**imhist**" function.

Figure 1: Original Image

- Linear scaling is an example of a point operator in which the pixel values of images are linearly scaled between u_{min} and u_{max} using a gradation function as follows:

$$g(u) = b(u + a) \tag{1}$$

where, $a = -u_{min}$, $b = \frac{G_{max}}{(u_{max} - u_{min})}$ and $G_{max} = 255$.

Now write a function which takes an image as input and returns the "**linearly scaled version**" of it. u_{min} and u_{max} of the returned image should be 0 and 255, respectively. Your function's name should be "**lab1linscale.m**". Compare histograms of the resulting linearly scaled and the histogram equalized images.

Your linear scaling results should look as follows:

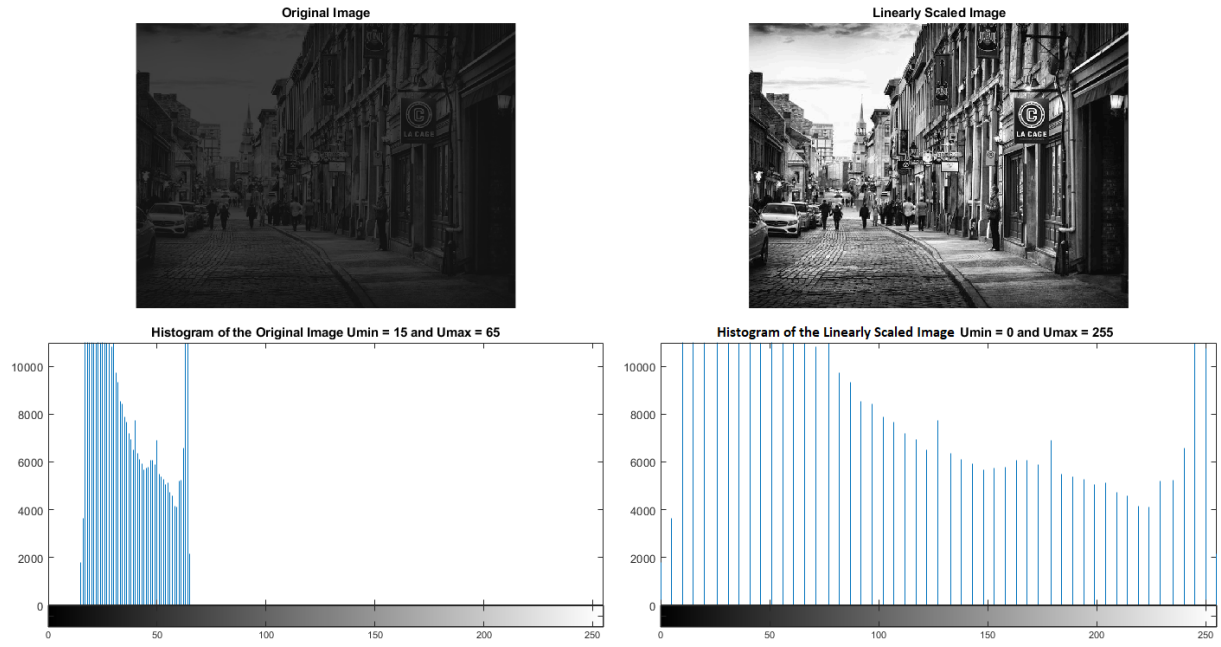

Figure 2: Linear Scaling of Images

- Conditional scaling is another example of a point operator in which an image J is mapped into image J_{new} such that J_{new} has the same mean and variance as a reference image I using a gradation function as follows:

$$g(u) = b(u + a) \tag{2}$$

where, $a = \mu_I \frac{\sigma_J}{\sigma_I} - \mu_J$ and $b = \frac{\sigma_I}{\sigma_J}$.

Now write a function which takes "**two images**" as inputs and returns the "**conditionally scaled version**" of the first image. Map the first image into the returned image such that the resultant image has the same mean and variance as the second image. Your function's name should be "**lab1condscale.m**".

Your results should look as follows:

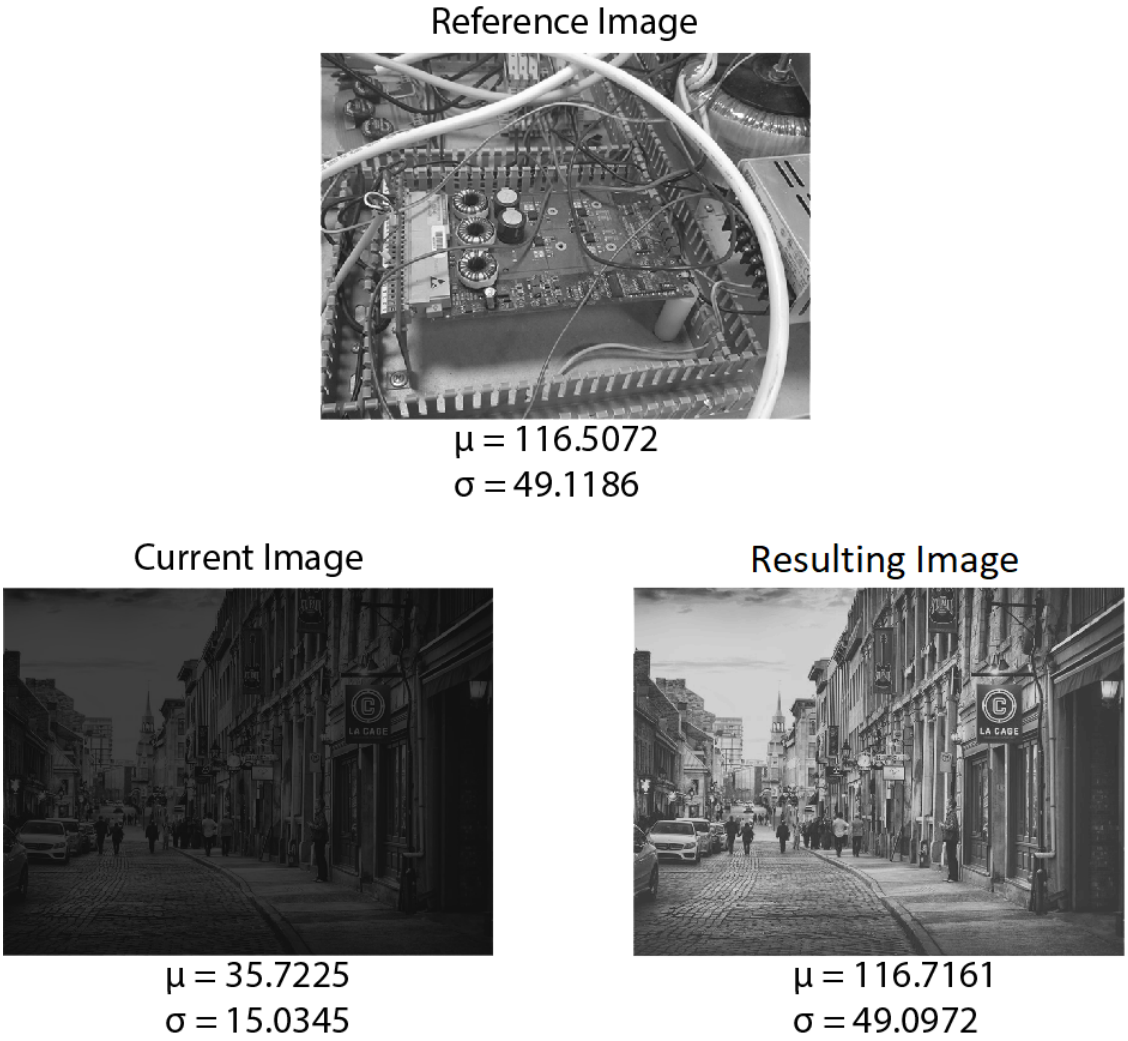

Figure 3: Conditional Scaling of Images

- Local mean filter (Box filter) is an example of local operator which is used to attenuate noise in the acquired images by convolving it with a sliding window W_P of size $(2k+1)\times(2k+1)$. Box filter is realized by replacing each pixel of an image with the average of its neighborhood as follows:

$$\mu_{W_p(I)} = \frac{1}{(2k+1)^2} \sum_{i=-k}^{+k} \sum_{j=-k}^{+k} I(x+i, y+j) \tag{3}$$

$$J(p) = \mu_{W_p(I)} \tag{4}$$

Now Write a function which takes an image and a number (for window size) as inputs and returns the "**local mean (box filter) filtered version**" of the image. Your function's name should be "lab1locbox.m".

Your results should look as follows:

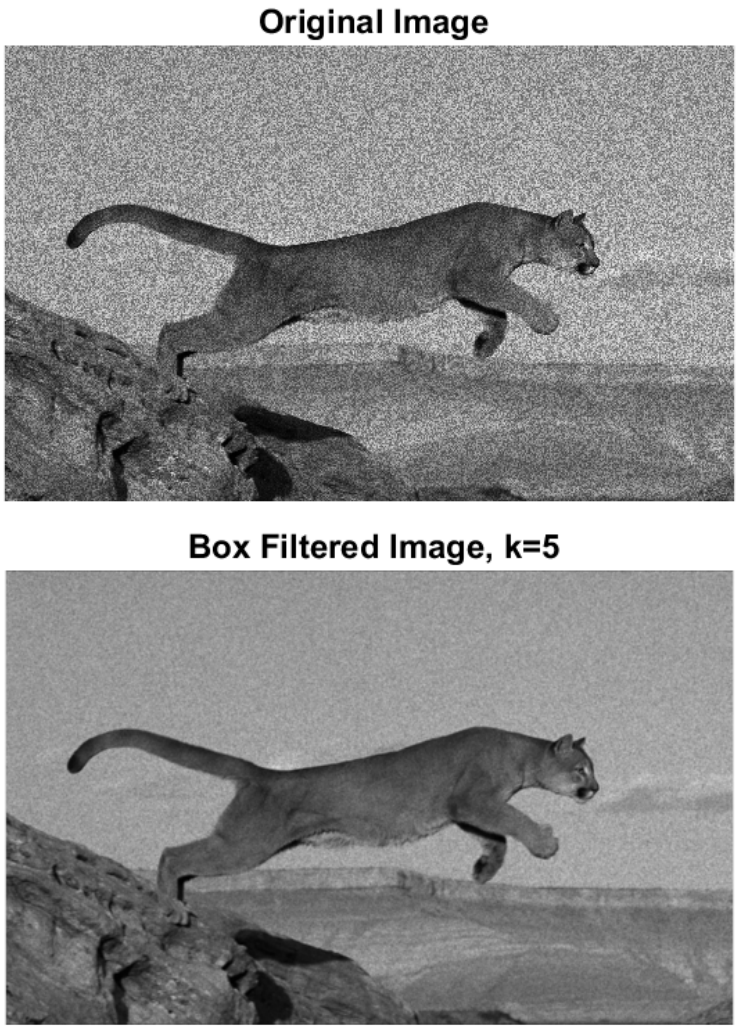

Figure 4: Box Filtering of Images

- Local max and local min filters are examples of local operators which are used for dilation and erosion of pixels in images by convolving the image with a sliding window W_P of size $(2k+1)\times(2k+1)$. Local max and min filters are realized by replacing each pixel of an image with the maximum and minimum of its neighborhood as follows:

$$J(p)_{max} = max\{I(x+i, y+j) : -k \leq i \leq k \ \wedge \ -k \leq j \leq k\} \tag{5}$$

$$J(p)_{min} = min\{I(x+i, y+j) : -k \leq i \leq k \ \wedge \ -k \leq j \leq k\} \tag{6}$$

Now write a function which takes an image and a number (for window size) as inputs and returns the "**local maximum filtered version**" and "**local minimum filtered version**" of the image. Your function's name should be "lab1locmaxmin.m".

Your results should look as follows:

Figure 5: Local Maximum and Minimum Filtering of Images

Post Lab

Post lab reports must include brief explanations of the functions that you implemented in this lab.

- Provide resulting images by utilizing all these functions with different parameters such as window size. Discuss your results.
- Write a program that calculates the histogram of a given image. Make your own implementation of histogram equalization.

Deadline for post-lab report submission to SUCourse: **19 October 2021, 23:55**.