Sequence 2 exercises

Exercises marked as homework are supposed to be done at home before the corresponding class session and exercises marked as classroom work are going to be done during the class session.

1. Histogram's Modifications (homework)

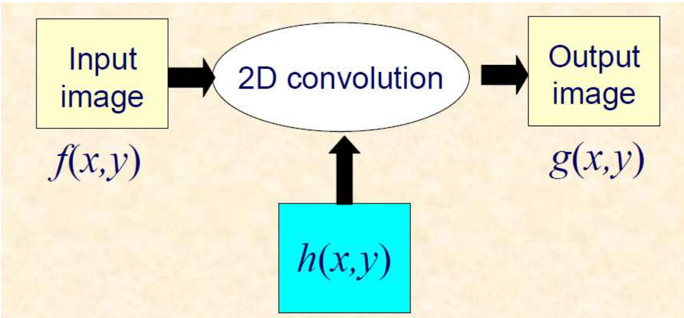
Download the *Sequence2_part1_code.py* file.

- Load and display the *Hawkes_Bay.jpg* image and its histogram before and after applying the *rescale_intensity()* function. Comment the histogram and the visual quality modification. What kind of process is done when using that function?

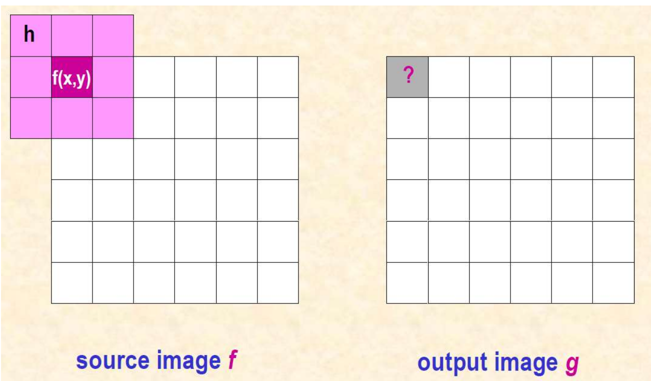
- In order to improve the visual quality of images, histogram's equalization is also possible. Run *Hawkes_Bay.jpg* with the *equalize_hist()* function. Display the modified image and its histogram. Comment the result and compare with the image obtained after applying the recalling function. In your opinion, which image has the best visual quality? Justify.

2. Image noise reduction via filtering operation

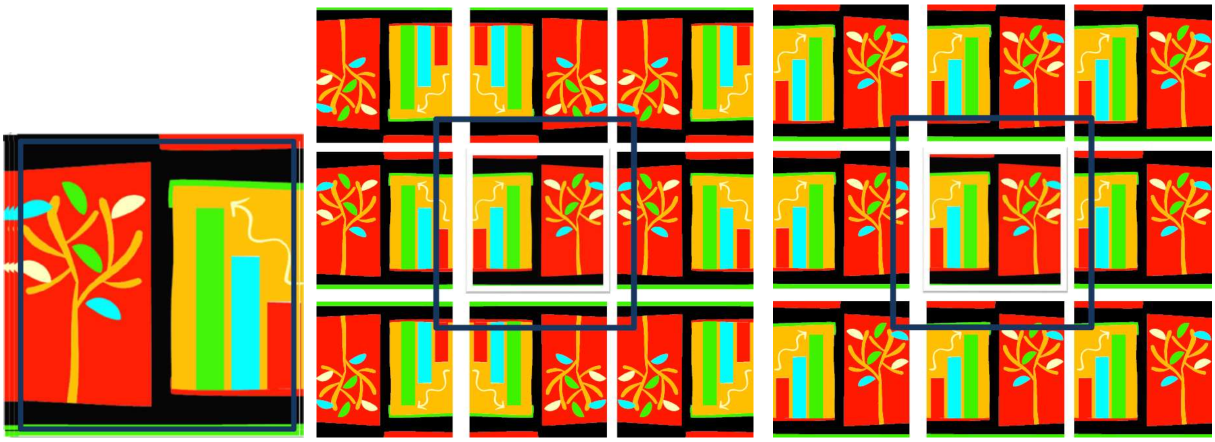
When doing spatial filtering, the global process is the following:



N.B.1: the *skimage.filters* module contains a lot of already implemented 2D filters.
N.B.2: any filtering process involves a convolution mask. Then the question is how to process the pixels that located at the image borders?



First solution is to reduce the size of the filtered image by considering only the pixels that have been fully processed. Others solutions are to replicate the border lines and columns, to increase the image size by mirror symmetry, to increase the image size by circular symmetry… Several solutions are possible but none of them can guarantee very reliable filtered pixels at the image borders. The way of processing the borders is a parameter of each image filtering function.



From left to right, borders replication, mirror symmetry and circular symmetry

2.1 Noise removing (homework)

Download the *Sequence2_part2_code.py* file

Let's consider the two noisy images *House.jpg* and *Train.jpg*.

- Display both images. What kind of noise is it?
- The challenge is to remove the noise as much as possible with edges preservation. What filter to do suggest to do that and why?
- Process the two images with the Gaussian filter and then with the median filter by considering successively a neighborhood of size 3x3. Which is the most appropriate filter and explain why?

2.2 Adaptive mean filter (classwork)

The purpose here is to develop an adaptive version, on a 3x3 neighborhood, of the mean low-pass filter. This filter has to be coded from scratch, it is not available in the *skimage.filters* module. This filter is defined in the following way (s is the current pixel to be processed and p is one of its 3x3 neighbors; τ is a threshold which value is tuned by the user):

$$I(s) = \frac{\sum_{p \in voisinage} c_p I(p)}{\sum_{p \in voisinage} c_p} \quad avec \quad \begin{cases} c_p = 1 \; si \; |I(s) - I(p)| < \tau \\ c_p = 0 \; si \; |I(s) - I(p)| \geq \tau \end{cases}$$

Be careful, the center pixel s itself is always considered in each sum (at least, one value is involved in the averaging process that is the center value).

- Implement the filter.What do you think about this filter and why this adaptive filter might be more efficient than the mean filter?
- In order to study the influence on the image quality of the τ chosen value in adaptive filtering, add a Gaussian noise (0 mean and var=0.01, use the *skimage.util.random_noise()* function) on the *coffee* image (before applying the noise, please convert the coffee image in gray levels). Display the corrupted image and compute the peak signal to noise ratio between the original and the noisy images (cf. see below for the PSNR definition).

Definition of the Mean Square Error MSE between two images I1 and I2 of size MxN:

$$MSE(I1,I2) = \frac{1}{M*N}\sum_{i,j}[I1(i,j) - I2(i,j)]^2$$

Definition of the peak signal to noise ratio PSNR between two 8 bit coded images:

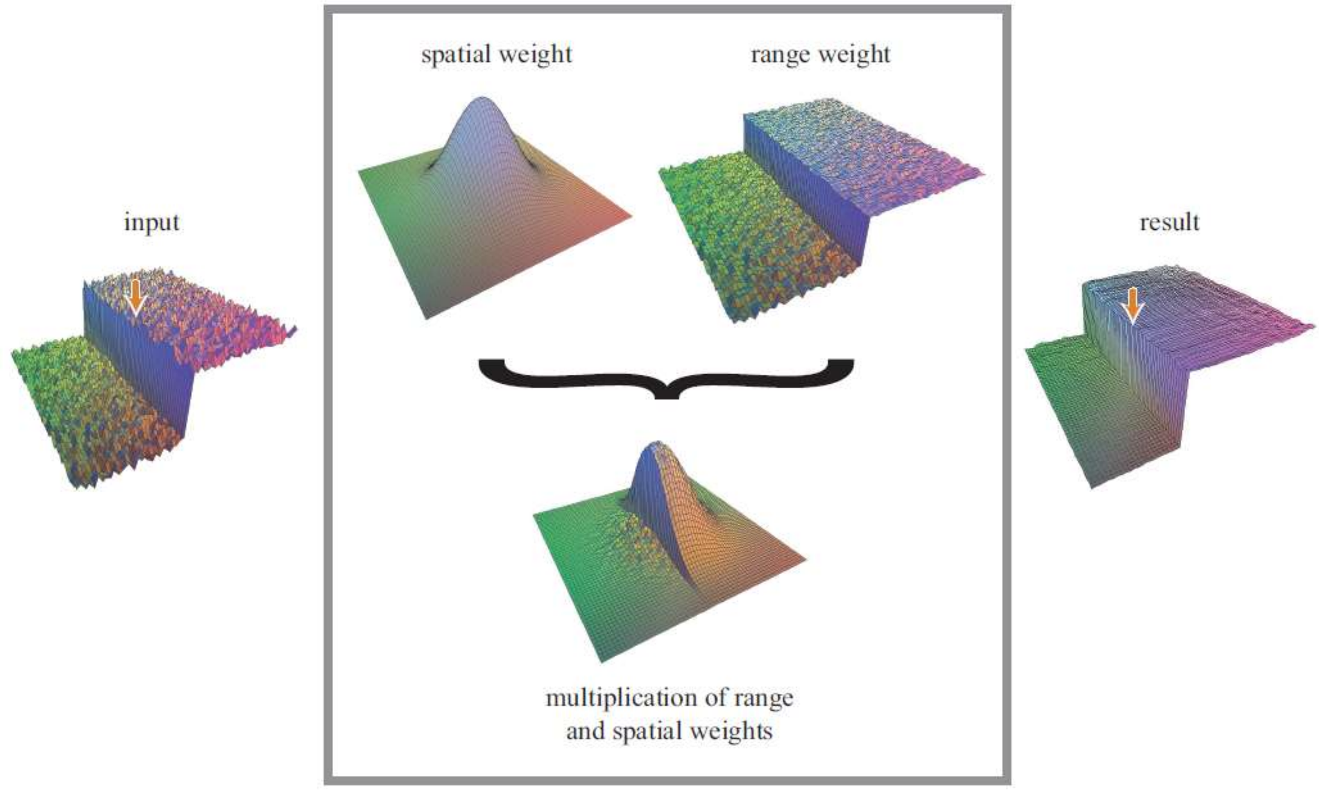
$$PSNR = 10log_{10}(\frac{255^2}{MSE})$$

N.B. to compute the PSNR, use the *skimage.metrics.peak-signal-noise-ratio()* function

- Process the image with the adaptive average filter and a τ value running from 10 to 250 (increment of 10 between 2 consecutive values). Draw the evolution of the PSNR computed between the original image and the filtered image with respect to τ.
- Is it possible to select an optimal τ value on that curve? Do you visually see differences on the filtered images obtained with τ=20, τ=optimal value and τ=220? Display the corresponding images.

2.3 Bilateral filtering (classwork)

The bilateral filter averages pixels based on their spatial closeness AND their luminance similarity. As a consequence, only pixels belonging to a specific neighbor and with a grey level belonging to a predefined range [g-s_0 ; g+s_1] where g is the central pixel grey level are averaged.



Each pixel is replaced by a weighted average of its neighbors. Each neighbor is weighted by a spatial component that penalizes distant pixels (according to a Gaussian law in the example) and a range component that penalizes pixels with a different intensity. The combination of both components ensures that only nearby similar pixels contribute to the final result. The weights are represented for the central pixel (under the arrow).
The figure is reproduced from: Fast bilateral filtering for the display of high-dynamic-range images *Durand and Dorsey* ACM SIGGRAPH conference (c) 2002

- Apply the bilateral filter on the noisy coffee image (cf. the *mean_bilateral* function) and compute the psnr. The range of grey levels to be taken into account has to be tuned. Try different values and keep the best ones. Compare the new psnr value with the previous psnr value obtained with the adaptive filter and comment the visual quality of the result. What is the main advantage of the bilateral filter?