Computing Methods for Experimental Physics and Data Analysis

Data Analysis in Medical Physics

Lecture 4: Basic image processing

Alessandra Retico
alessandra.retico@pi.infn.it
INFN - Pisa

Image enhancement

- Image processing is typically performed to enhance image content
- Two basic instruments used at the low level:
  - optimization of image visualization, accounting for viewer capabilities, e.g.:
    - Histogram modification (histogram stretching, histogram equalization)
  - image filtering aiming to:
    - reduce image noise
    - enhance the visibility of details of interest with respect to the background

- Histogram modification are point operations, i.e. each pixel intensity is modified independently from pixels of its neighborhood
- Image filtering operations are instead based on convolutions, i.e. the pixel values of filtered images are related to those of their neighbor pixels

Image processing in MATLAB

- A large variety of functions for image processing are available in the MATLAB Image Processing toolbox

See demo code:

- Lecture4_demo_morphological_operators.mlx
- Lecture4_demo_image_filtering1.mlx
- Lecture4_demo_image_filtering2.mlx

j

i

$$I(i,j) = \begin{bmatrix} I(1,1) & I(1,2) & I(1,3) & \dots & I(1,N) \\ I(2,1) & I(2,2) & I(2,3) & \dots & I(2,N) \\ I(3,1) & I(3,2) & I(3,3) & \dots & I(3,N) \\ \dots & \dots & \dots & \dots & \dots \\ I(M,1) & I(M,2) & I(M,3) & \dots & I(M,N) \end{bmatrix}$$

y

x

$$\begin{matrix} I(1,1,2) & I(1,2,2) & I(1,3,2) & \dots & I(1,N,2) \end{matrix}$$

$$I(i,j,k) = \begin{bmatrix} I(1,1,1) & I(1,2,1) & I(1,3,1) & \dots & I(1,N,1) \\ I(2,1,1) & I(2,2,1) & I(2,3,1) & \dots & I(2,N,1) \\ I(3,1,1) & I(3,2,1) & I(3,3,1) & \dots & I(3,N,1) \\ \dots & \dots & \dots & \dots & \dots \\ I(M,1,1) & I(M,2,1) & I(M,3,1) & \dots & I(M,N,1) \end{bmatrix} \dots$$

y z

x

The MATLAB Image Processing Toolbox

- Check on the documentation how many functions are available to read, display and process images: https://it.mathworks.com/help/images/index.html

Image Processing Toolbox

Import, Export, and Conversion	101
Display and Exploration	54
Geometric Transformation and Image Registration	43
Image Filtering and Enhancement	145
Image Segmentation and Analysis	102
Deep Learning for Image Processing	16
3-D Volumetric Image Processing	72

Read and Write Image Data from Files

dicominfo	Read metadata from DICOM message
dicomread	Read DICOM image
dicomwrite	Write images as DICOM files
dicomreadVolume	Construct 4-D volume from set of DICOM images
dicomCollection	Gather details about related series of DICOM files
dicomanon	Anonymize DICOM file
dicomdict	Get or set active DICOM data dictionary
dicomdisp	Display DICOM file structure
dicomlookup	Find attribute in DICOM data dictionary
dicomuid	Generate DICOM globally unique identifier
images.dicom.decodeUID	Get information about DICOM unique identifier
images.dicom.parseDICOMDIR	Extract metadata from DICOMDIR file
niftiinfo	Read metadata from NIfTI file
niftiwrite	Write volume to file using NIfTI format
niftiread	Read NIfTI image

Image Filtering

imfilter	N-D filtering of multidimensional images
fspecial	Create predefined 2-D filter
fspecial3	Create predefined 3-D filter
roifilt2	Filter region of interest (ROI) in image
nlfilter	General sliding-neighborhood operations
imgaussfilt	2-D Gaussian filtering of images
imgaussfilt3	3-D Gaussian filtering of 3-D images
wiener2	2-D adaptive noise-removal filtering
medfilt2	2-D median filtering
medfilt3	3-D median filtering
ordfilt2	2-D order-statistic filtering
stdfilt	Local standard deviation of image

Basic Display

imshow	Display image
imfuse	Composite of two images
imshowpair	Compare differences between images
montage	Display multiple image frames as rectangular montage
immovie	Make movie from multiframe image
implay	Play movies, videos, or image sequences
warp	Display image as texture-mapped surface
sliceViewer	Browse image slices
orthosliceViewer	Browse orthogonal slices in grayscale or RGB volume
volshow	Display volume
labelvolshow	Display labeled volume

Morphological Operations

imerode	Erode image
imdilate	Dilate image
imopen	Morphologically open image
imclose	Morphologically close image
imtophat	Top-hat filtering
imbothat	Bottom-hat filtering
imclearborder	Suppress light structures connected to image border
imfill	Fill image regions and holes

Contrast Adjustment

imadjust	Adjust image intensity values or colormap
imadjustn	Adjust intensity values in N-D volumetric image
imcontrast	Adjust Contrast tool
imsharpen	Sharpen image using unsharp masking
imflatfield	2-D image flat-field correction
imlocalbrighten	Brighten low-light image

Histogram transforms

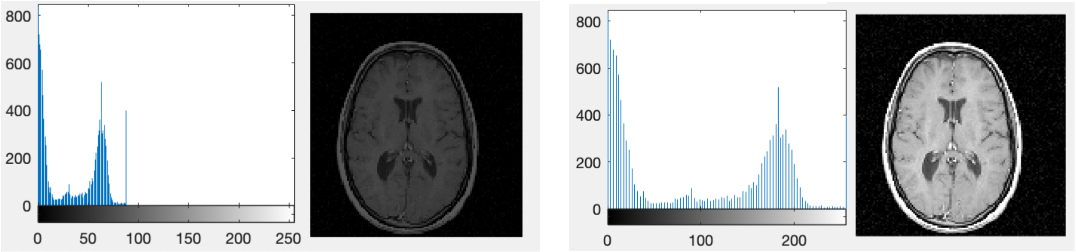
The image histogram is a graphical representation of the intensity distribution of an image (i.e. the number of pixels for each gray level), regardless the position that pixels have in the image.

Using the histogram, pixel transforms can be defines to carry out:

- Histogram stretching – Image gray levels are linearly scaled, stretching out the image range to the maximum range allowed by the display (typically 8-bit levels). It generally improves image contrast.
- Histogram equalization – The most frequent intensity values are spread out, i.e. the intensity range of the image is stretched to cover the maximum allowed range, and so that the histogram is approximately flat. It is used to improve image contrast.

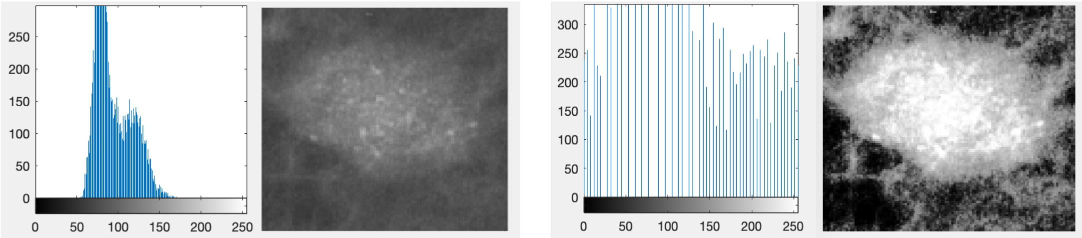
Examples provided in Lecture2_demo2_image_read_display.mlx

Histogram stretching



If the histogram of an image does not contain all gray scales, by stretching the histogram the gray scale distance between neighbor pixel is enlarged, leading to enhanced contrast.

Histogram equalization



Histogram equalization not provided the desired image contrast improvement at that time

Image/object visibility: contrast and signal to noise ratio

- To evaluate image quality and to compare images acquired in different conditions/with different parameters, some standard measures can be computed, e.g.:
  - Contrast between tissues, C_{1-2}
  - Contrast to Noise Ratio, CNR_{1-2}

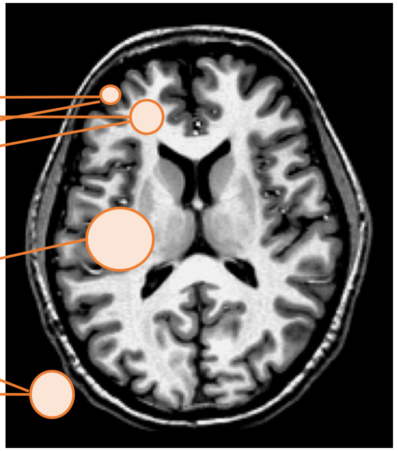
    where C_{1-2} is compared against the standard deviation of the background, $\sigma(BG)$
  - Signal to Noise Ratio, SNR

    where the average signal in a region is compared against $\sigma(BG)$

$$C_{1-2}=(I_1-I_2)/(I_1+I_2)$$

$$CNR_{1-2}=(I_1-I_2)/\sigma(BG_{ROI})$$

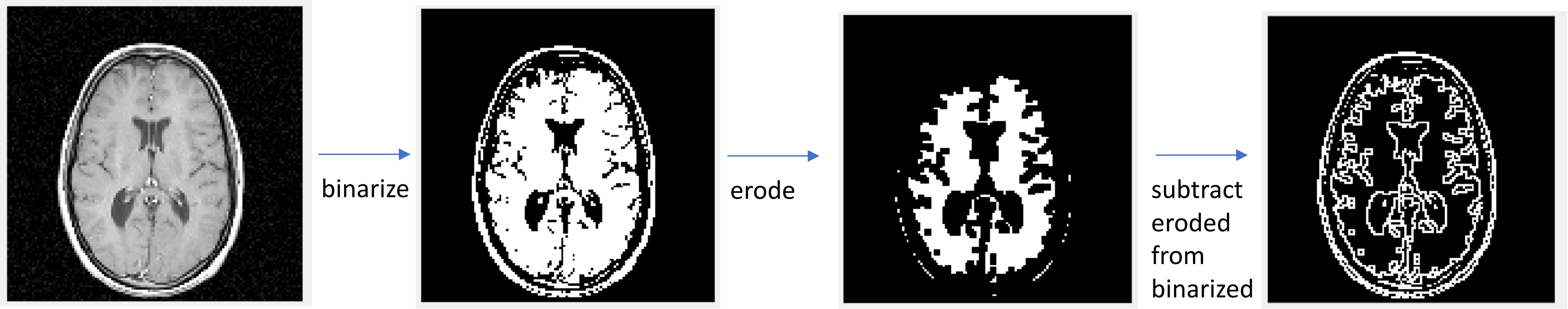
$$SNR=\mu(I_{ROI})/\sigma(BG_{ROI})$$



- To compute these measures Regions of Interests (ROIs) should be defined. They can be:
  - arbitrarily defined geometric portion of the images
  - Anatomically meaningful regions (e.g. segmented gray matter vs. white matter/ brain region vs. outside noise)

Morphological operations

- Mathematical morphology contributes a wide range of operators to image processing
- Morphological operators are useful in image processing to modify the shapes of binary images (they are defined also for gray scale images)
- For a binary image, white pixels are normally taken to represent foreground regions, while black pixels denote background.
- Morphological operators are used in mask modification, edge detection, noise removal, image enhancement and image segmentation.



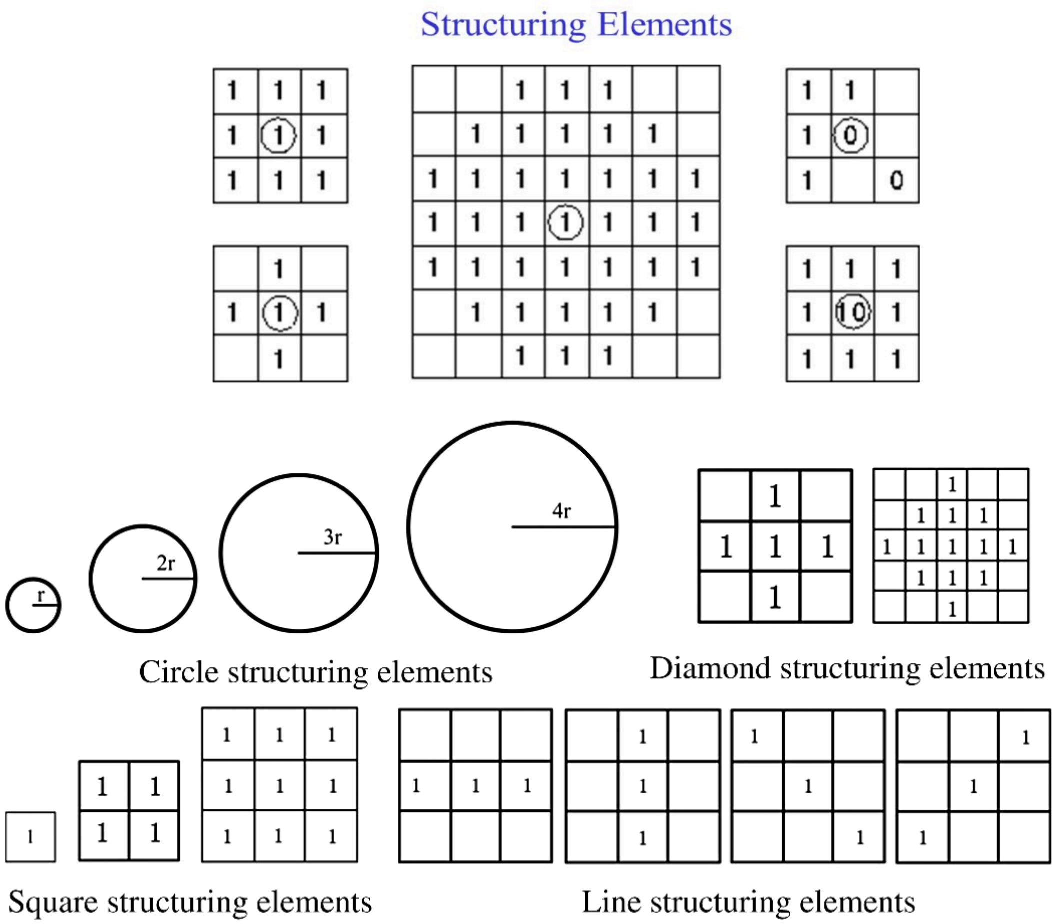
For example, we can use erosion for edge detection by eroding a binarized image and then subtracting it from the binarized image. Only those pixels at the edges of objects that were removed by the erosion are thus highlighted

Morphological operations

- Morphological operations apply a structuring element (SE) to an input image, creating an output image of the same size:
  - The value of each pixel in the output image is obtained by combining the input image and the SE using a set operator (intersection, union, inclusion, complement).
  - The SE defines the neighborhood used to process each pixel.
  - The SE influences the size and shape of objects to process in the image.

- Despite morphological operations can be also be defined for gray-scale images, they are mainly performed on binary images (i.e. masks).

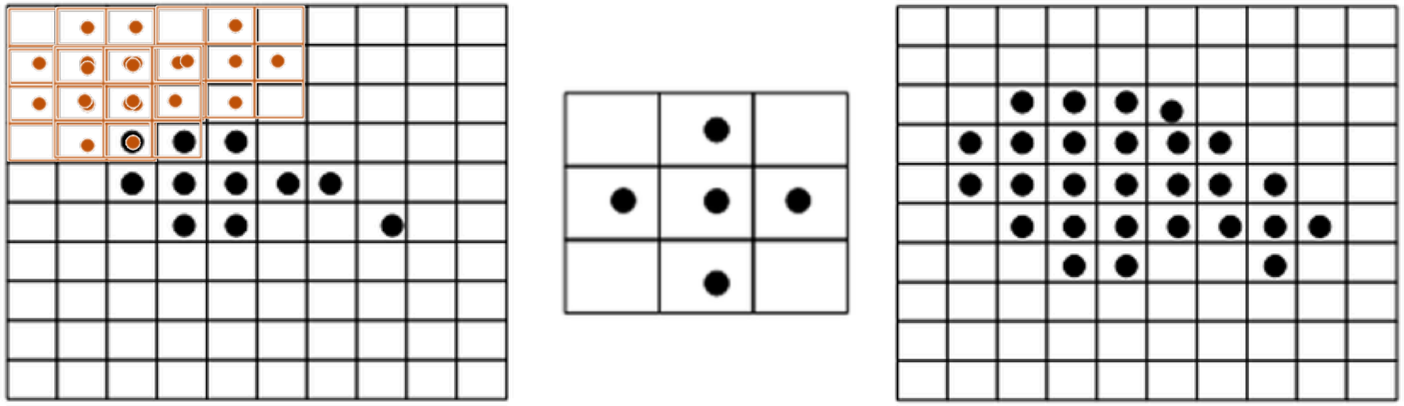
See demo code:
- Lecture4_demo_morphological_operators.mlx



Dilation

- Dilation (enlarges the boundary of regions of foreground pixels of a binary image):
  - X is the set of coordinates corresponding to the input binary image
  - K is the set of coordinates of the structuring element (SE)
  - K_x is the translation of K so that its origin is at x
  - The dilation of X by K is based on logical OR of SE and binary image, i.e. it is the set of all points x such that the intersection of K_x with X is not empty

$$X \oplus K = \{x \mid K_x \cap X \neq \varnothing\}$$



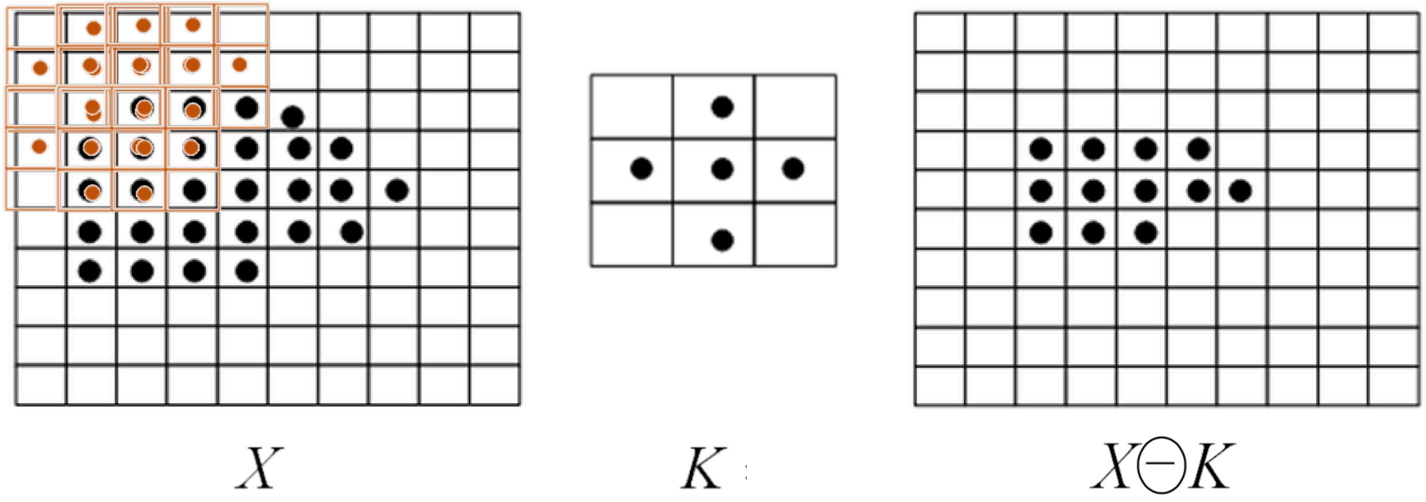
X $\qquad$ K $\qquad$ $X \oplus K$

Tip: Dilation and subtraction of the original image can be used to find the border of an object

Erosion

- Erosion (erodes the boundary of regions of foreground pixels of a binary image):
  - *X* is the set of coordinates corresponding to the input binary image
  - *K* is the set of coordinates of the structuring element (SE)
  - K_x is the translation of *K* so that its origin is at *x*
  - The erosion of *X* by *K* is based on logical AND of SE and binary image, the set of all points *x* such that K_x is a subset of *X*

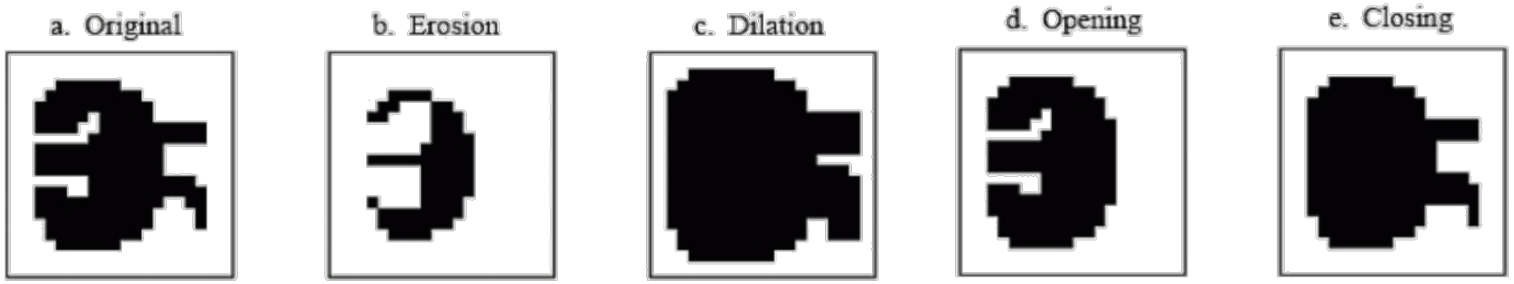
$$X \ominus K = \{x \mid K_x \subseteq X\}$$



Basic morphological operations

- **Dilation** adds pixels to the boundaries of objects in an image. It is based on the logical OR operation.
  - The value of the output pixel is the *maximum* value of all pixels in the neighborhood. In a binary image, a pixel is set to 1 if any of the neighboring pixels have the value 1.
  - Morphological dilation makes objects more visible and fills in small holes in objects.
- **Erosion** removes pixels on object boundaries. It is based on the logical AND operation.
  - The value of the output pixel is the *minimum* value of all pixels in the neighborhood. In a binary image, a pixel is set to 0 if any of the neighboring pixels have the value 0.
  - Morphological erosion removes islands and small objects so that only substantive objects remain.
- **Opening** removes small objects from an image while preserving the shape and size of larger objects.
  - It **erodes** an image and then **dilates** the eroded image, using the same structuring element for both operations.
- **Closing** fills small holes from an image while preserving the shape and size of the objects in the image.
  - It **dilates** an image and then **erodes** the dilated image, using the same structuring element for both operations.

The number of pixels added or removed from the objects in an image depends on the size and shape of the **structuring element** used to process the image.



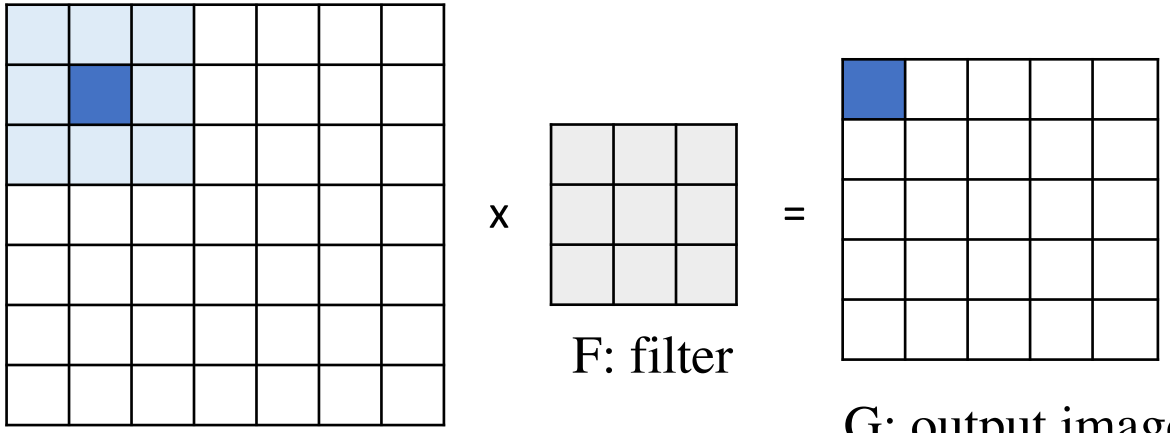
Spatial filters

Spatial filters are implemented to improve image appearance → better contrast and visibility of features of interest.

Image enhancement in the spatial domain is achieved by applying convolutional operations, i.e. the pixel value is combined with those of its neighbors using filter *kernels*.

Specific masks (*kernels)* are used to obtain:

- Image denoising (e.g. mean, median, Gaussian filters)
- Edge detection (e.g. standard deviation, Sobel, Canny filters)



H: input image

F: filter

G: output image

Each pixel of the output image is the weighted sum of the input pixels within a region defined by the mask, with the elements of the mask defining the weights.

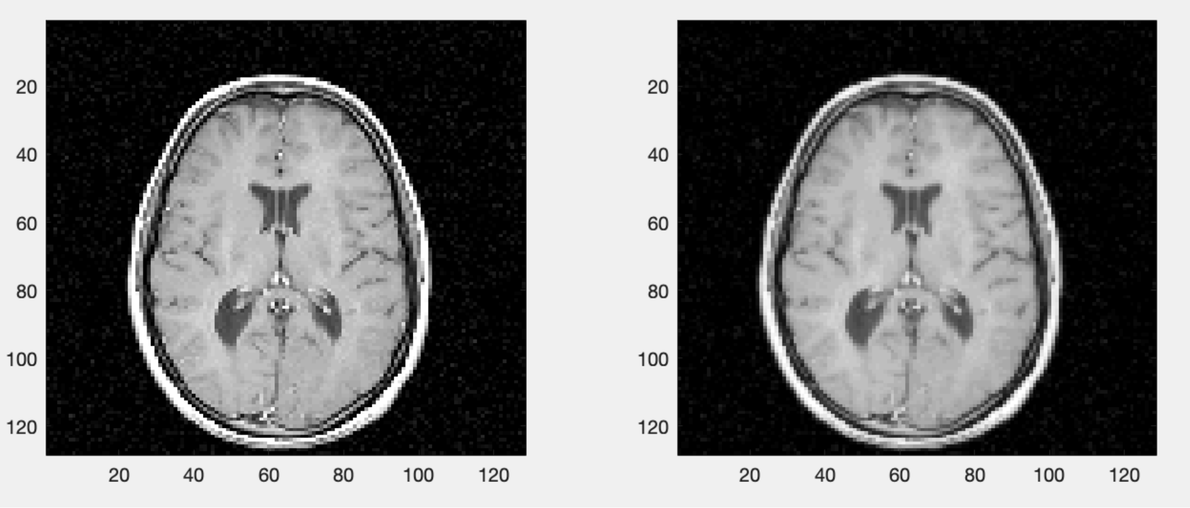
$$\mathbf{G} = \mathbf{F} * \mathbf{H} \qquad a = (k-1)/2$$

$$g(i,j) = \sum_{m=-a}^{a} \sum_{n=-a}^{a} f(m,n)h(i-m,\ j-n)$$

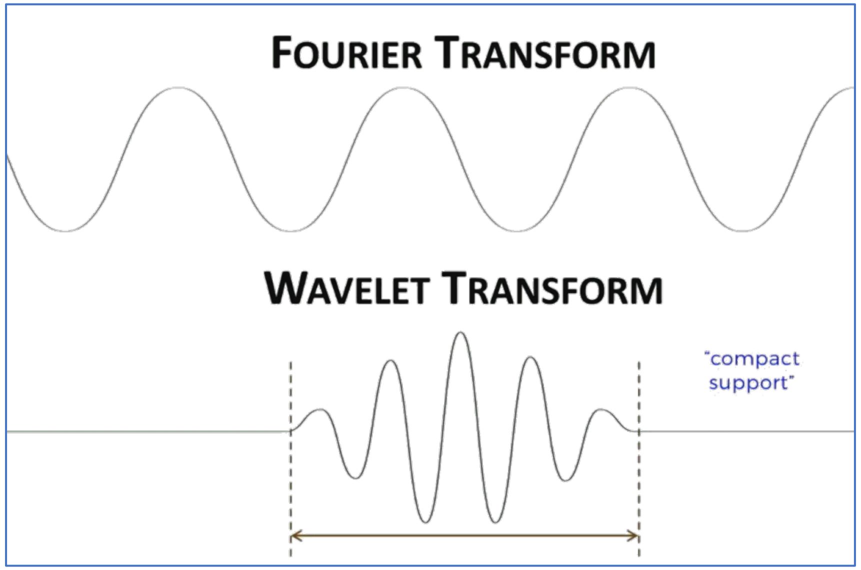
Spatial filters

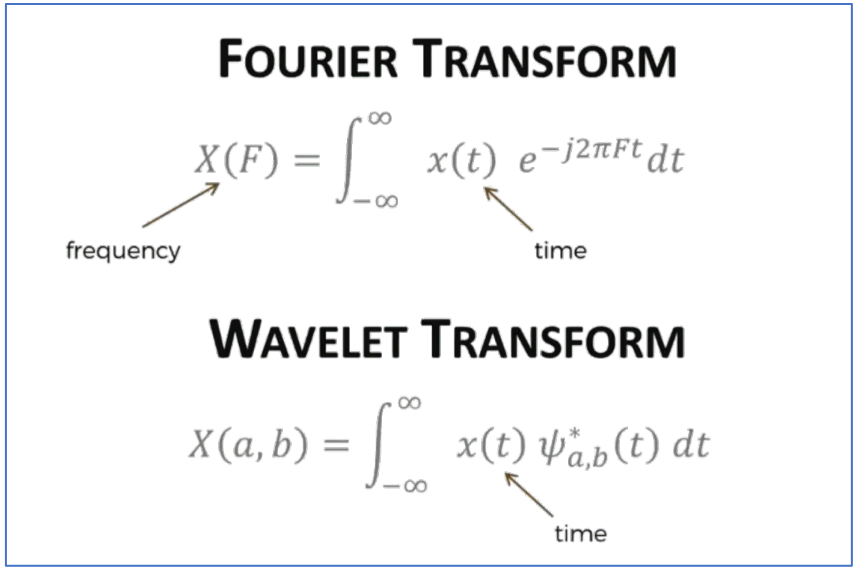
- Pixel values are combined with the values of their neighbors through convolution operations
- The resulting output image pixels often have non-integral values, thus it is necessary to work with floating-point numbers to avoid "round-off" errors
- The computational complexity for convolution of an image of M x M pixels with a mask of size k x k is of the order of k^2 per pixel, based on the number of multiply-and-adds
- When we apply denoising filters, the resulting image is smoothed and appears with reduced noise. However, the sharpness of edges is also reduced.

See demo code:
- Lecture4_demo_image_filtering1.mlx



Wavelet Transform





Wavelets allow a general way to represent and analyze images at multiresolution

Continuous wavelet transform

- In a continuous space wavelet are scaled and translated version of a mother wavelet $\Psi(x)$

$$\psi_{s,\tau}(x) = \frac{1}{\sqrt{s}}\psi\left(\frac{x-\tau}{s}\right)$$

- The continuous wavelet transform of a signal transforms a continuous function of one variable into a continuous function of two variables: translation and scale

Wavelet Transform

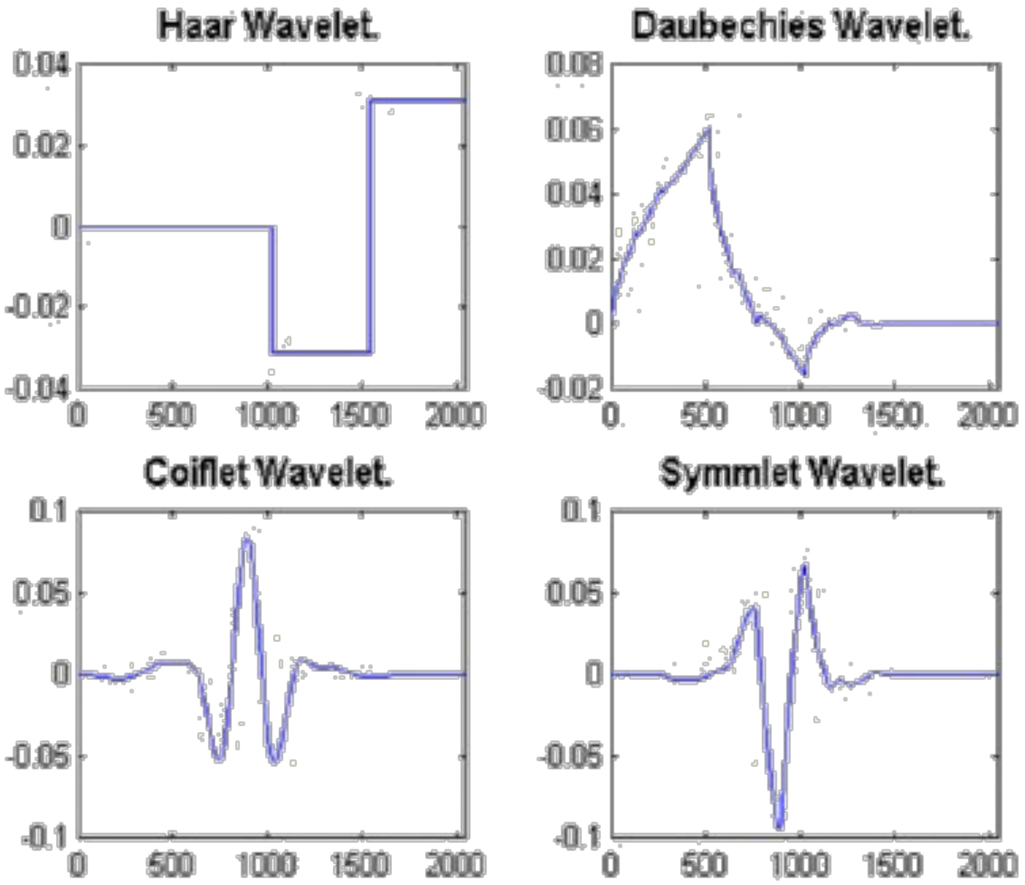
Discrete Wavelet Transform (DWT)

- We do not need to calculate wavelet coefficients at every possible scale
- We can choose scales based on powers of two, and find the signal components at different scales

$$\psi_{j,k}(x) = 2^{j/2}\,\psi(2^j x - k)$$

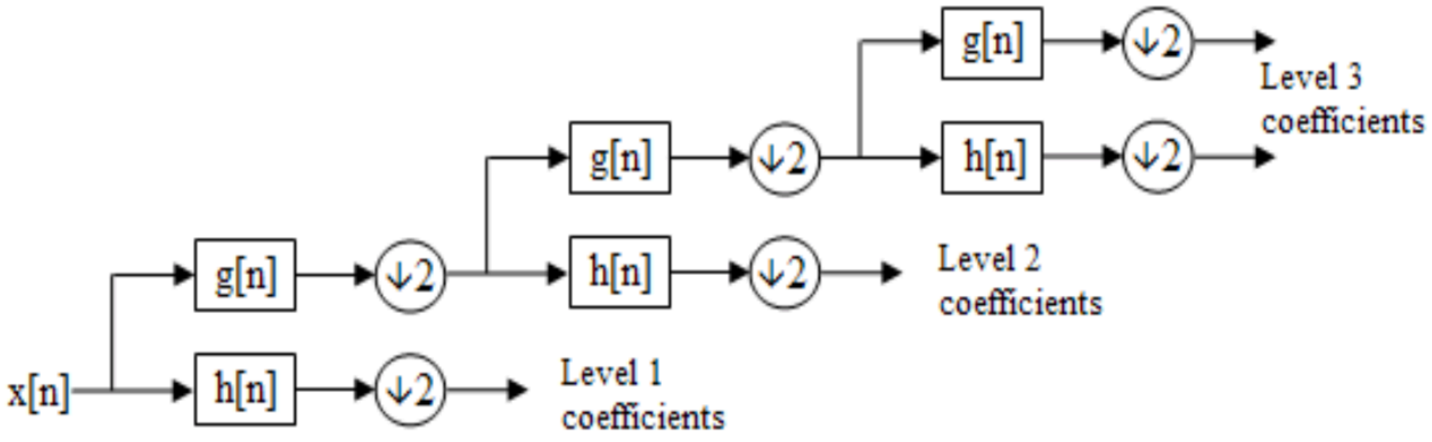
- The wavelet coefficients measure how closely correlated the wavelet is with each section of the signal

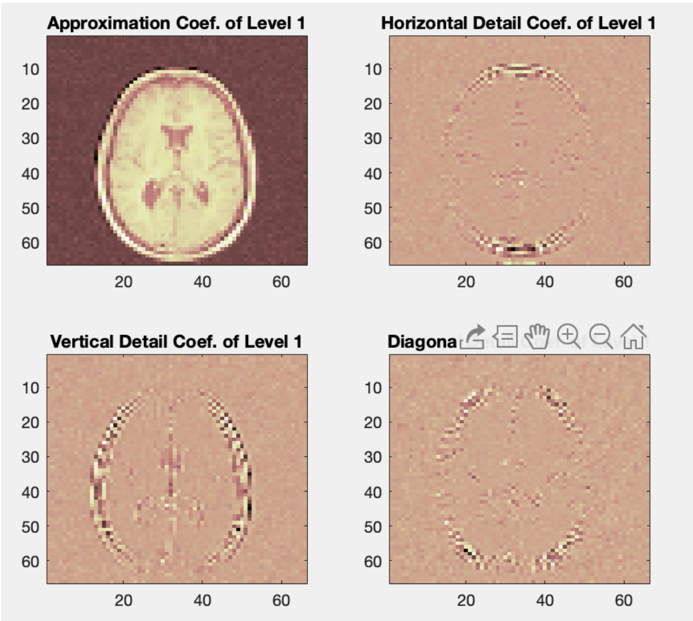
- Tip: choose a wavelet that matches the shape of the image components



DWT decomposition

- The DWT of a signal is calculated by passing it through a series of filters: a low-pass filter *g* and simultaneously a high-pass filter *h*:
  - the detail coefficients are obtained from the high-pass filter *h*
  - approximation coefficients from the low-pass filter *g*
- The filter output of *g* is then subsampled by 2 and undergoes again *h* and *g* filters





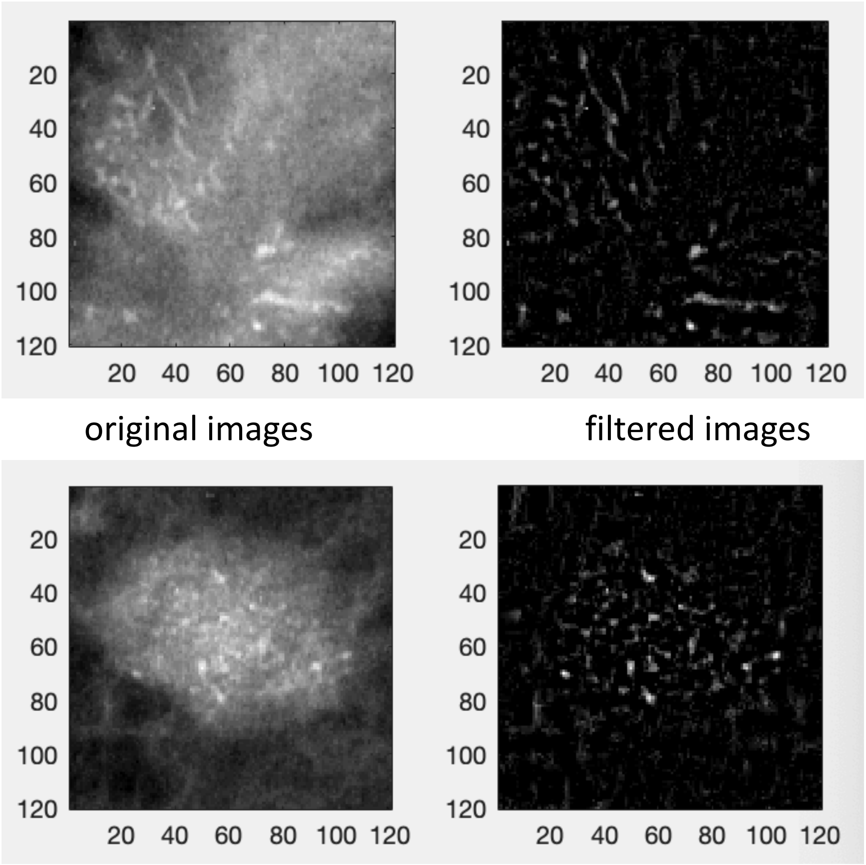
Wavelet-based filters

- DWT are very useful for:
  - image compression (e.g., in the JPG-2000 standard)
  - removing noise

- To build a wavelet-based image processing filter:
  - Compute the 2D wavelet transform
  - Alter the wavelet coefficient
  - Compute the inverse transform

See demo code:
- Lecture4_demo_image_filtering2.mlx

Wavelet-based filter to highlight microcalcifications in mammograms and to discard heterogeneous tissue background



References and sources

- Books
  - Digital Image Processing for Medical Applications, Geoff Dougherty
  - Handbook of Medical Image Processing and Analysis, Isaac N. Bankman
  - Image Processing and Acquisition using Python, Ravishankar Chityala & Sridevi Pudipeddi

- Sources
  - http://doc.sid.unipi.it/Campus_Matlab
  - https://it.mathworks.com/help/matlab/getting-started-with-matlab.html
  - https://it.mathworks.com/videos/
  - https://it.mathworks.com/help/images/index.html
  - https://it.mathworks.com/help/matlab/external-language-interfaces.html
  - https://www.youtube.com/watch?v=ZnmvUCtUAEE

- (See also)
  - https://scikit-image.org Image processing in Python