

Computing Methods for Experimental Physics and Data Analysis

Data Analysis in Medical Physics

Lecture 6: Image segmentation

Alessandra Retico

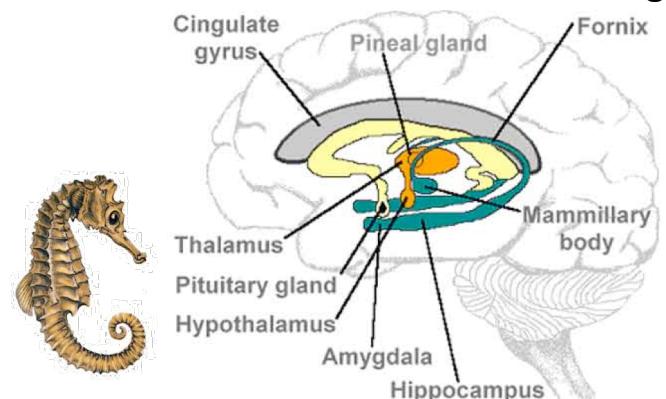
alessandra.retico@pi.infn.it

INFN - Pisa

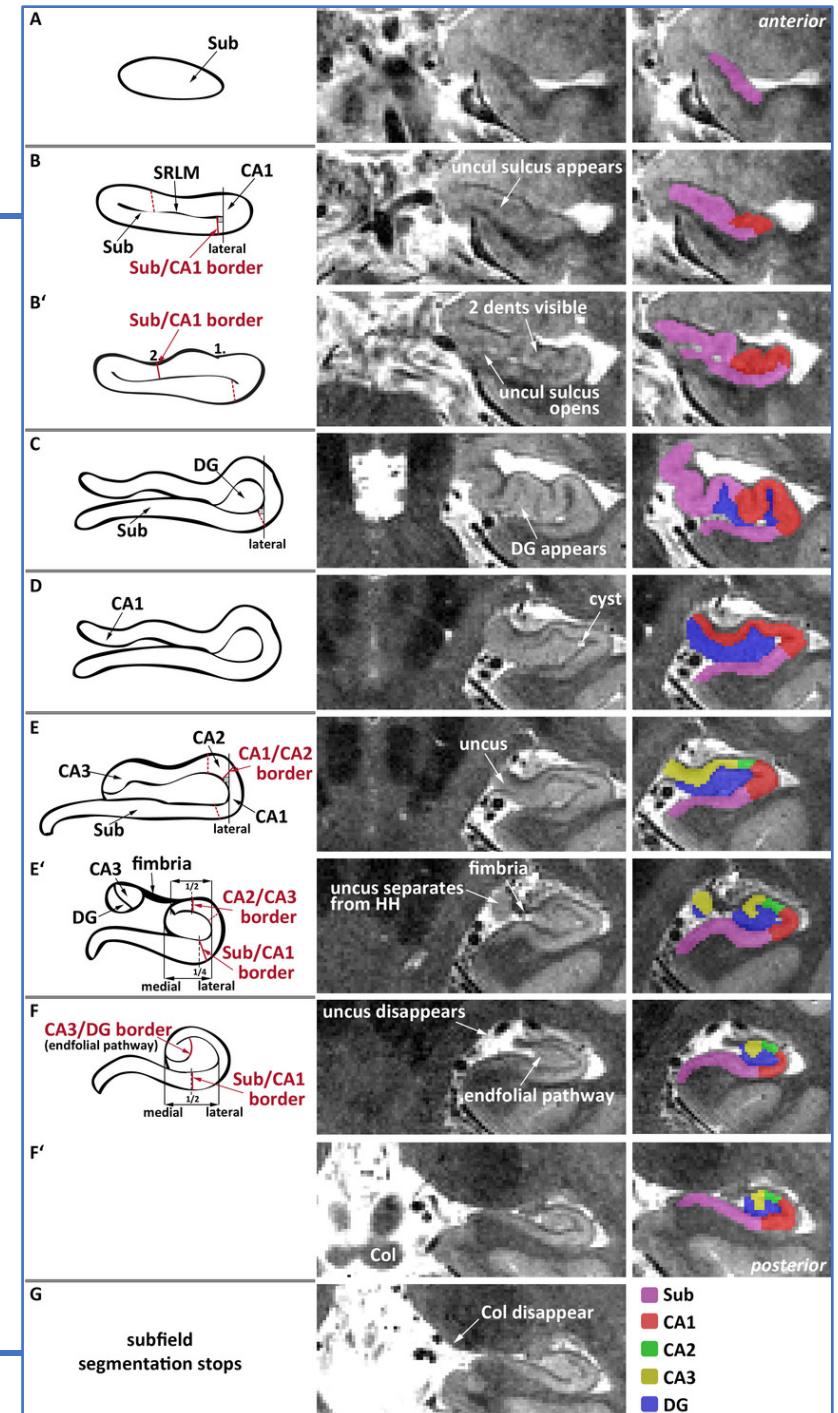
Image segmentation

- Image segmentation: the image is partitioned into its parts or regions (segments)
- Segmentation of tissues, organs, lesions or other **regions of interest (ROIs)** out of a medical image is often a useful step to extract meaningful information related to shape or texture of the object of interest

For example, in the study of Alzheimer's Disease (AD), the hippocampus is one of the first regions of the brain to suffer damage; memory loss and disorientation are included among the early AD symptoms.



D. Berron et al., A protocol for manual segmentation of medial temporal lobe subregions in 7 Tesla MRI, *NeuroImage: Clinical* 15(C) 2017



How to segment objects out of medical images

- **Manual segmentation:** ROIs are manually drawn by experts
 - Extremely time consuming, but necessary step to build the ground truth and/or templates for segmentation algorithm
- **Template matching:** Images can be warped to specific template images, which contain suitable ROI definitions
 - Applicable in processing almost-normal anatomy; not working in case of subject-specific alterations
- **Custom algorithms** can be set up to extract ROIs (tissues, organs, lesions), relying on image properties (e.g. object intensities, discontinuities in pixel values)

Some segmentation methods (and corresponding MATLAB functions)

Thresholding-based methods:

- Global/Multilevel image thresholding using Otsu's method ([imbinarize](#), [graythresh](#), [multithresh](#))

Use only histogram information

Clustering-based methods:

- K-means clustering based image segmentation ([imsegkmeans](#))

Edge-detection-based methods:

- finding edges in the input image ([edge](#)), then fill the region inside ([imfill](#))
- Active contours ([activecontour](#))

Exploit spatial relations between pixels

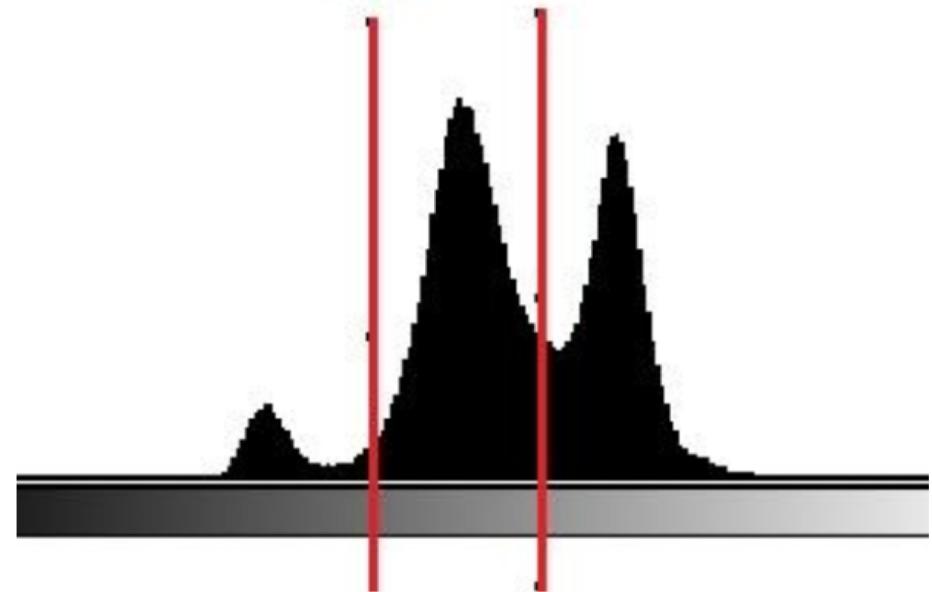
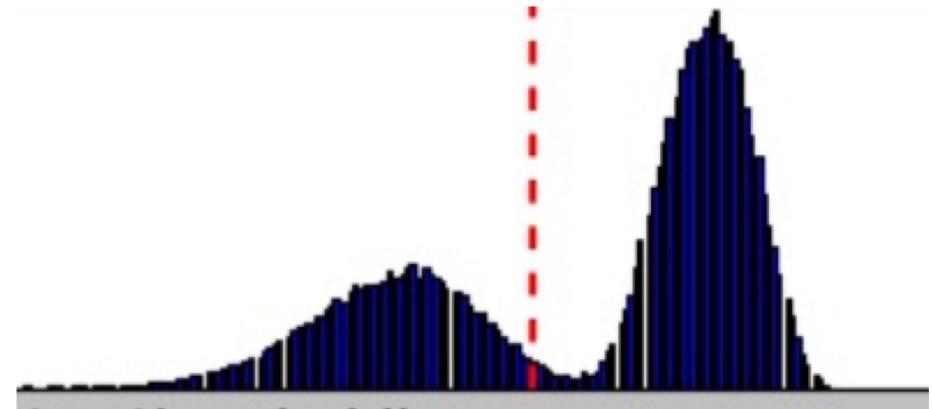
Region growing methods:

- Select contiguous image regions with similar gray values ([grayconnected](#))

... → Machine-Learning and Deep-Learning based methods

Global and Multilevel image thresholds using Otsu's method

- The Otsu algorithm, in the simplest form, performs automatic definition of a single intensity threshold that separate pixels into two classes, foreground and background.
- This threshold is determined by minimizing intra-class intensity variance, or equivalently, by maximizing inter-class variance.
- In the multilevel form, the $N-1$ thresholds partitioning N different image levels are provided.



Clustering methods: the K -means algorithm

- The image is iteratively partitioned in K clusters (distinct, non-overlapping subgroups of pixels).
- The objective is to make the inter-cluster elements as similar as possible while also keeping the clusters as different (far) as possible

Algorithm steps:

- Definition of the number of clusters K
- Initialize centroids randomly selecting K data points for the centroids
- Compute the sum of the squared distance between data points and all centroids.
- Assign each data point to the closest cluster (centroid).
- Compute the centroids for the clusters by taking the average of the all data points that belong to each cluster.

Objective function

$$J = \sum_{i=1}^m \sum_{k=1}^K w_{ik} \|x^i - \mu_k\|^2$$

K is the number of clusters;
 w_{ik} is 1 for points in the k^{th} cluster, 0 outside;
 m is the total amount of data points.

Expectation-Maximization.

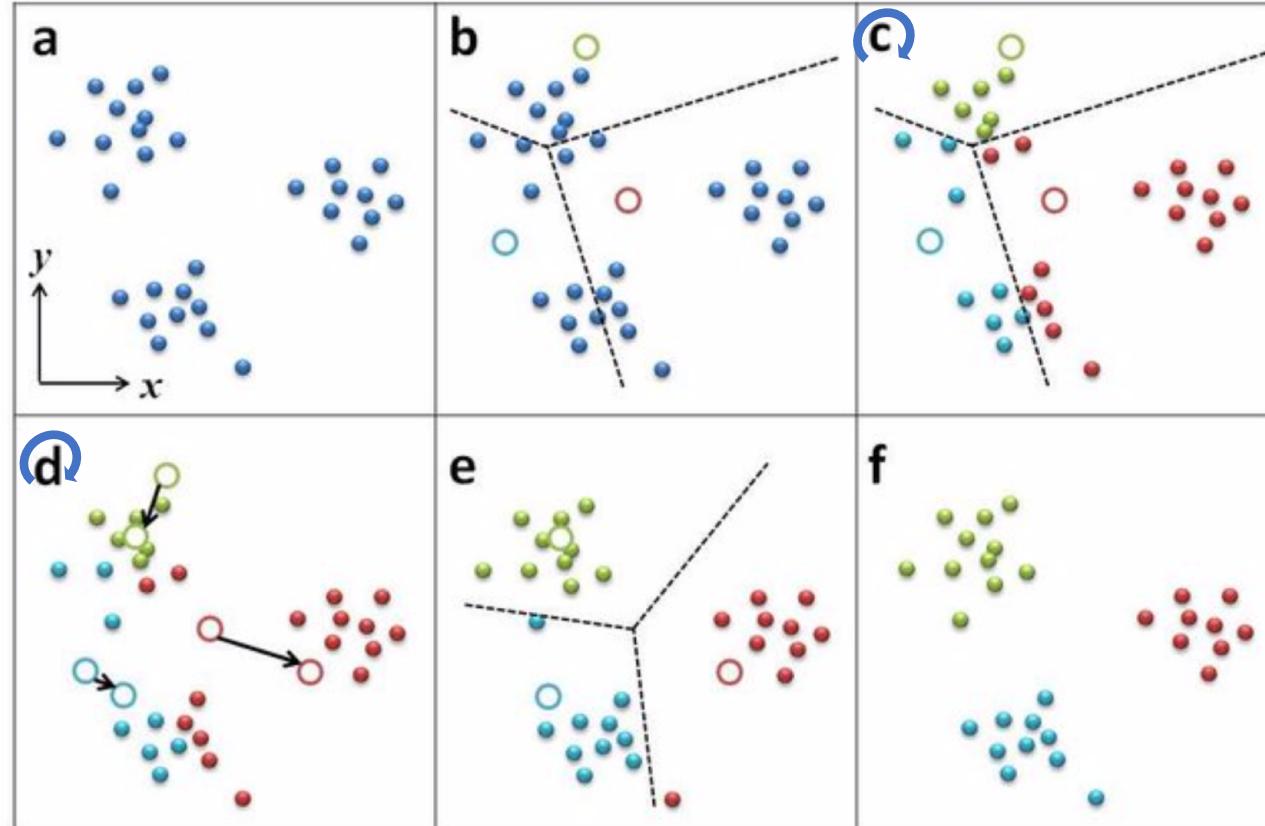
The **E**-step is assigning the data points to the closest cluster.

The **M**-step is computing the centroid of each cluster.

The optimization consists in two steps, i.e. minimizing J first with respect to w_{ik} and keep μ_k fixed. Then, minimizing J with respect to μ_k and keep w_{ik} fixed.

Clustering methods: the K-means algorithm

A schematic illustration of the K-means algorithm for two-dimensional data clustering



(a) The data points (solid blue circles) to be clustered in a 2D feature space. There are three clusters, so $K = 3$. The K-means algorithm is capable of assigning each data point into one of the clusters.

(b) For random locations of the cluster centers (empty circles), each data point can be associated with the closest center.

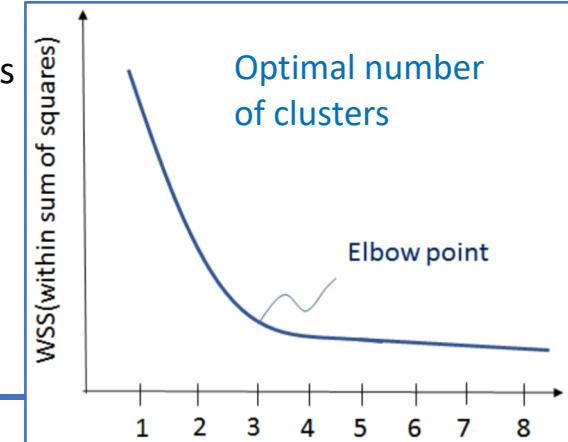
(c) The 2D space is divided into three regions through three decision boundaries, each containing the corresponding data points whose closest center is within. The data points currently in the regions with the aqua, green, and red centers are assigned to the corresponding clusters, respectively.

(d) Each center moves to the centroid of the data points currently assigned to it (movements shown by the black arrows).

(e) The updated cluster assignments of the data points are obtained according to the new center locations.

The steps in (c) and (d) are repeated until convergence is achieved.

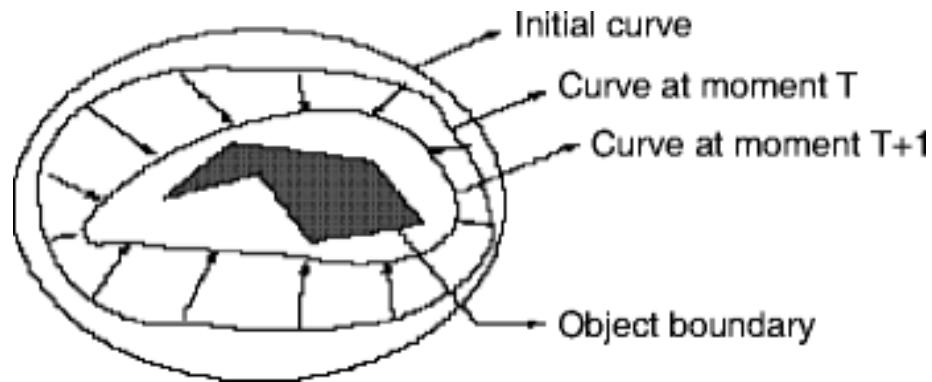
(f) The final cluster assignments



Chen and Lai, <https://arxiv.org/abs/1611.01849>

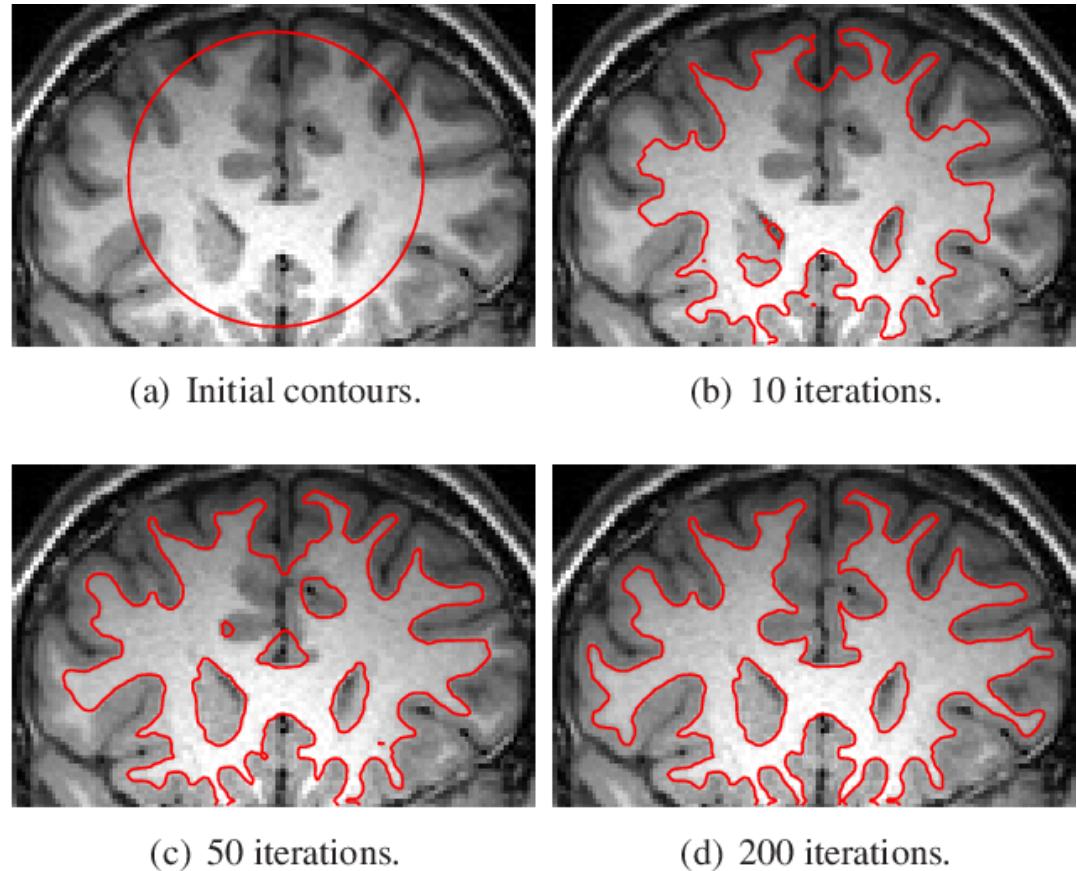
Active contour models/snakes

- In active contour models a contour algorithm, also called *snake*, moves to find object boundaries.
- The *snake* is an energy minimizing spline* guided by external constraint forces and influenced by image forces that pull it toward features such as lines and edges.



- It locks onto nearby edges and localizes them accurately

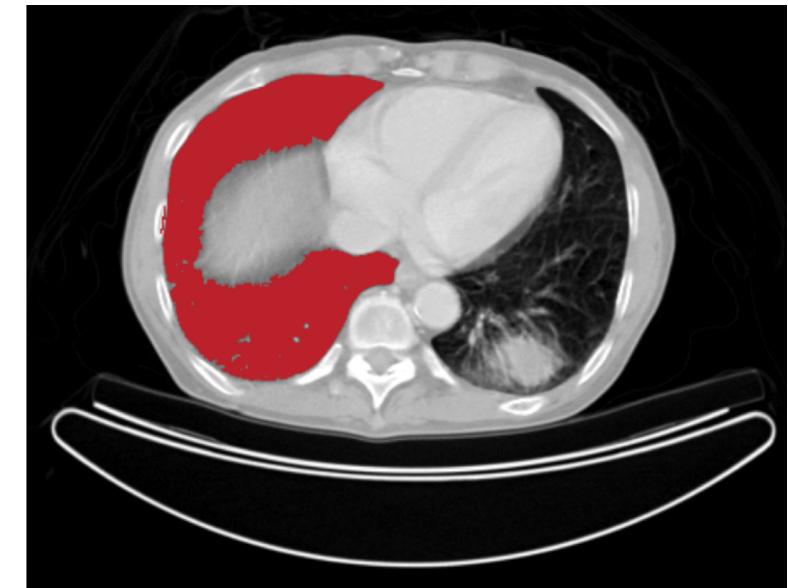
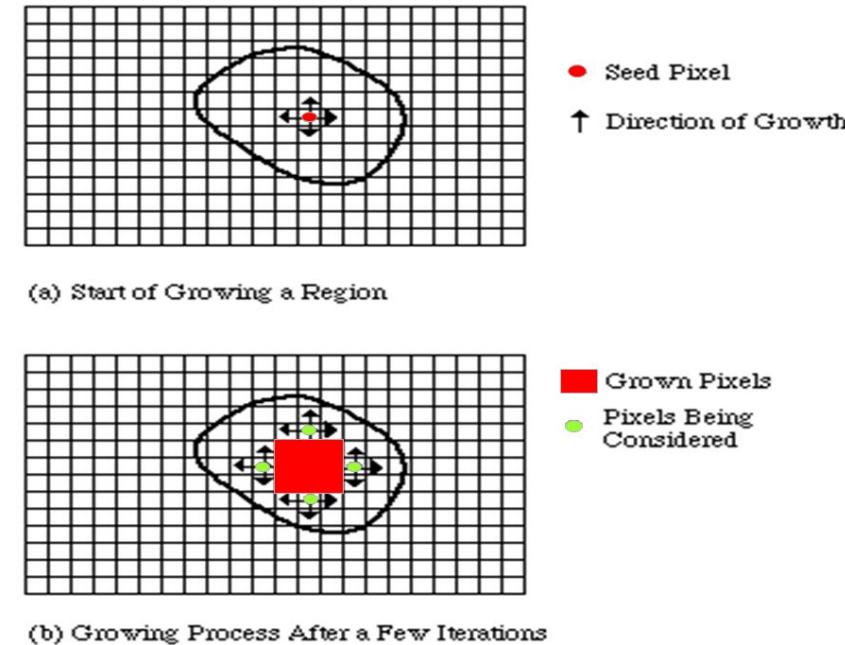
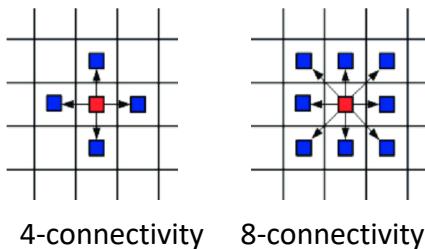
* A spline is a function, consisting of a set of polynomials connected together, whose purpose is to interpolate in a range a set of points, so that the function is continuous at least up to a given order of derivatives at any point in the interval.



Chunming Li, Chiu-Yen Kao, John C. Gore, Zhaohua Ding
Implicit Active Contours Driven by Local Binary Fitting Energy, 2007
IEEE Conference on Computer Vision and Pattern Recognition

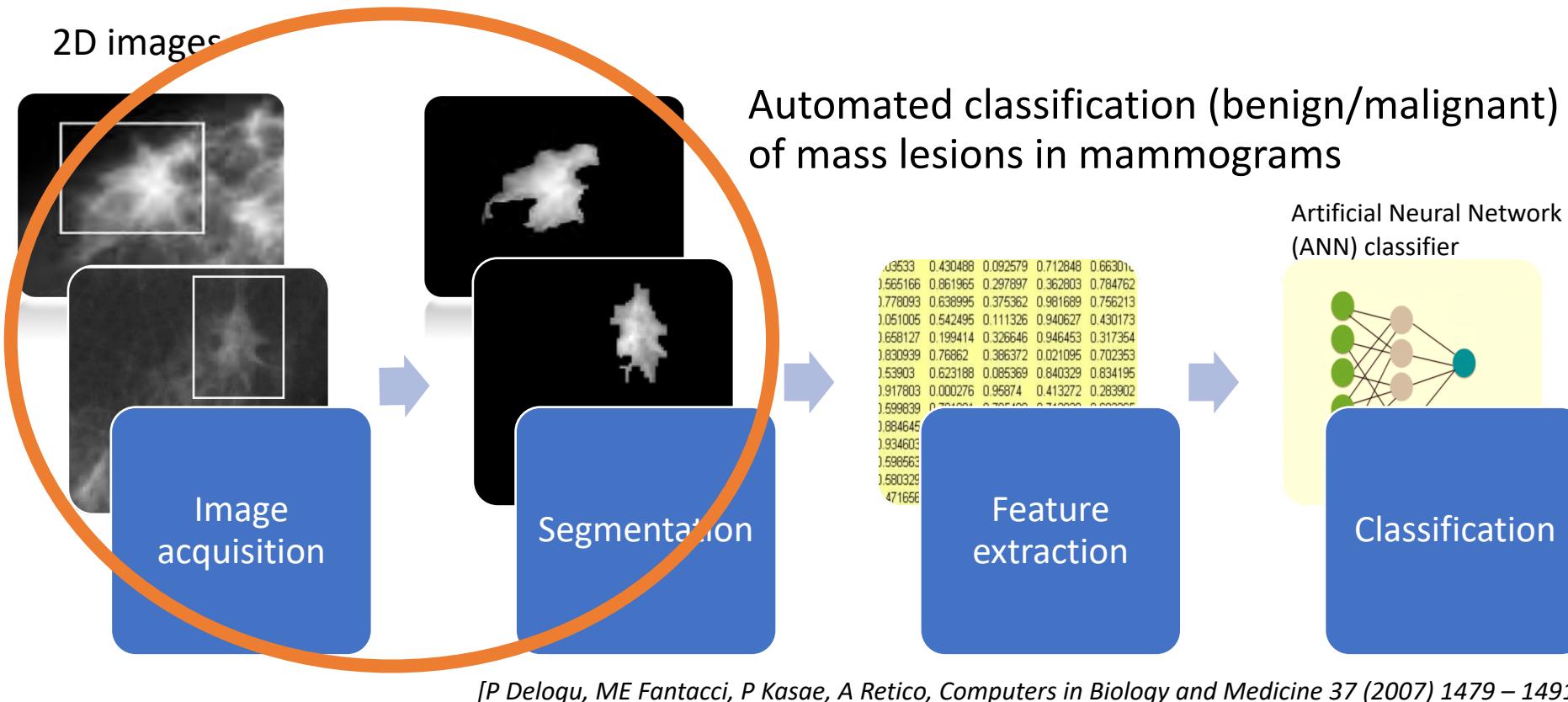
Region growing methods

- It requires the selection of an initial seed point.
- Region is **grown** from the seed pixel by adding in neighboring pixels that are *similar*, increasing the size of the region.
- When the growth of one region stops, another seed pixel is chosen, which does not yet belong to any region and the procedure starts again.
- *Similarity conditions* for pixels of the same regions have to be defined, e.g. pixel intensity should be in a certain range, or pixel intensity variance in a neighborhood should be in a certain range (to define the pixel neighborhood a connectivity measure should be defined, e.g. 4- or 8-connectivity in 2D, etc.)



Typical image analysis pipeline for assisted diagnosis

Example: 1) Object segmentation; 2) Hand-crafted feature extraction; 3) Machine Learning classification



See demo code: `mass_segment.m`; Data available on <https://pandora.infn.it/public/cmepda/DATASETS> and on https://drive.google.com/drive/folders/1YqK7ZkM-P2lrqfD7Pj-SCmjz-GWd_1-Y

Image types in MATLAB (Image Processing Toolbox)

- **Binary images:** m -by- n logical array. Array values of 0 and 1 are interpreted as black and white.
- **Indexed images:** m -by- n numeric matrix whose elements are direct indices into a color map. Each row of the color map specifies the red, green, and blue components of a single color.
- **Greyscale images (intensity images):** m -by- n numeric array whose elements specify intensity values.
- **Truecolor images (RGB images):** m -by- n -by-3 numeric array whose elements specify the intensity values of one of the three color channels. For RGB images, the three channels represent the red, green, and blue signals of the image

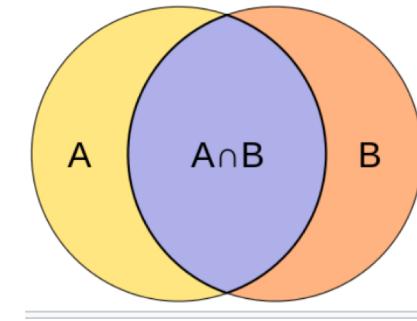
Multidimensional arrays

- The MATLAB cat function can be used to concatenate images:
 - 2D images can be concatenated in the 3rd dimension to create a 3-D array of size m -by- n -by- p . Each of the p images has size m -by- n , $\text{cat}(3, \text{Im}1, \text{Im}2)$
 - 3D images can be concatenated in the 4rd dimension, $\text{cat}(4, \text{Vol}1, \text{Vol}2)$
 - a sequence of 2-D RGB images, then concatenate the images along the fourth dimension to create a 4-D array of size m -by- n -by-3-by- p . Each of the p images has size m -by- n -by-3, $\text{cat}(4, \text{Im}1, \text{Im}2)$
 - ...

Segmentation similarity measures

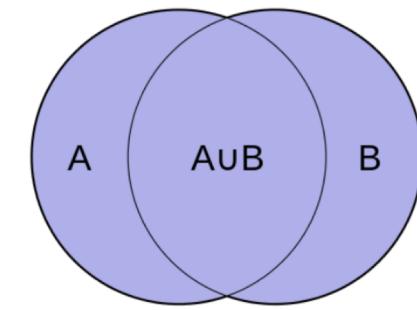
- **Jaccard similarity coefficient: Intersection over Union**

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \quad 0 \leq J(A, B) \leq 1.$$



- **Sørensen-Dice similarity coefficient**

$$DSC = \frac{2|X \cap Y|}{|X| + |Y|}$$



where $|X|$ and $|Y|$ are the cardinalities of the two sets (i.e. the number of elements in each set).
The Sørensen index equals twice the number of elements common to both sets divided by the sum of the number of elements in each set.
When applied to boolean data, using the definition of true positive (TP), false positive (FP), and false negative (FN), it can be written as

$$DSC = \frac{2TP}{2TP + FP + FN}.$$

It is different from Jaccard index which only counts true positives once in both the numerator and denominator.

Assignment: mass_segment.m

- Start from script_mass_segment.m (https://github.com/retico/cmepda_medphys/tree/master/L6_code)
- It contains the main steps of the segmentation algorithm developed in the paper by *P Delogu, ME Fantacci, P Kasa, A Retico, Computers in Biology and Medicine 37 (2007) 1479 – 1491*
- It calls the two custom functions [draw_radial_lines.m](#) and [max_var_points_interp.m](#)
- Try how the script_mass_segment.m works on more than one benign/malignant mass of the dataset available on <https://pandora.infn.it/public/cmepda/DATASETS> or on https://drive.google.com/drive/folders/1YqK7ZkM-P2lrqfD7Pj-SCmjz-GWd_1-Y (malignant masses are named as xxxx_1.png, whereas benign ones as xxxx_2.png)
- Modify the script to obtain a function (following the instructions at the end of the script) and obtain the mass_segment.m. Set it up so to pass all free parameters to the function to enable an optimization study.
- Use the mass_segment.m function in a new script following the instructions provided in [Lecture6_exercise1.pdf](#) with the aim of testing the reproducibility of the segmentation algorithm
- (I have also shared a run segmentation script (run_segmentation.m), which allows to run the mass_segment function on *.png files of a directory. It implements also the possibility to repeat the segmentation if the result is not satisfactory.)

A list of free SW for processing medical images

- PET, MRI, US, RX and CT images
 - 3DSlicer, <https://www.slicer.org>
 - ImageJ, <https://imagej.nih.gov/ij/index.html>
 - Itk-SNAP, <http://www.itksnap.org/pmwiki/pmwiki.php>
 - LIFEx, www.lifexsoft.org
 - OsiriX (only for Mac), <https://www.osirix-viewer.com>
- Brain MRI:
 - AFNI, <https://afni.nimh.nih.gov>
 - Freesurfer, <http://freesurfer.net>
 - FSL, <https://fsl.fmrib.ox.ac.uk/fsl/fslwiki>
 - Mango, <http://rii.uthscsa.edu/mango/mango.html>
 - SPM (requires MATLAB or R), <https://www.fil.ion.ucl.ac.uk/spm/software/>