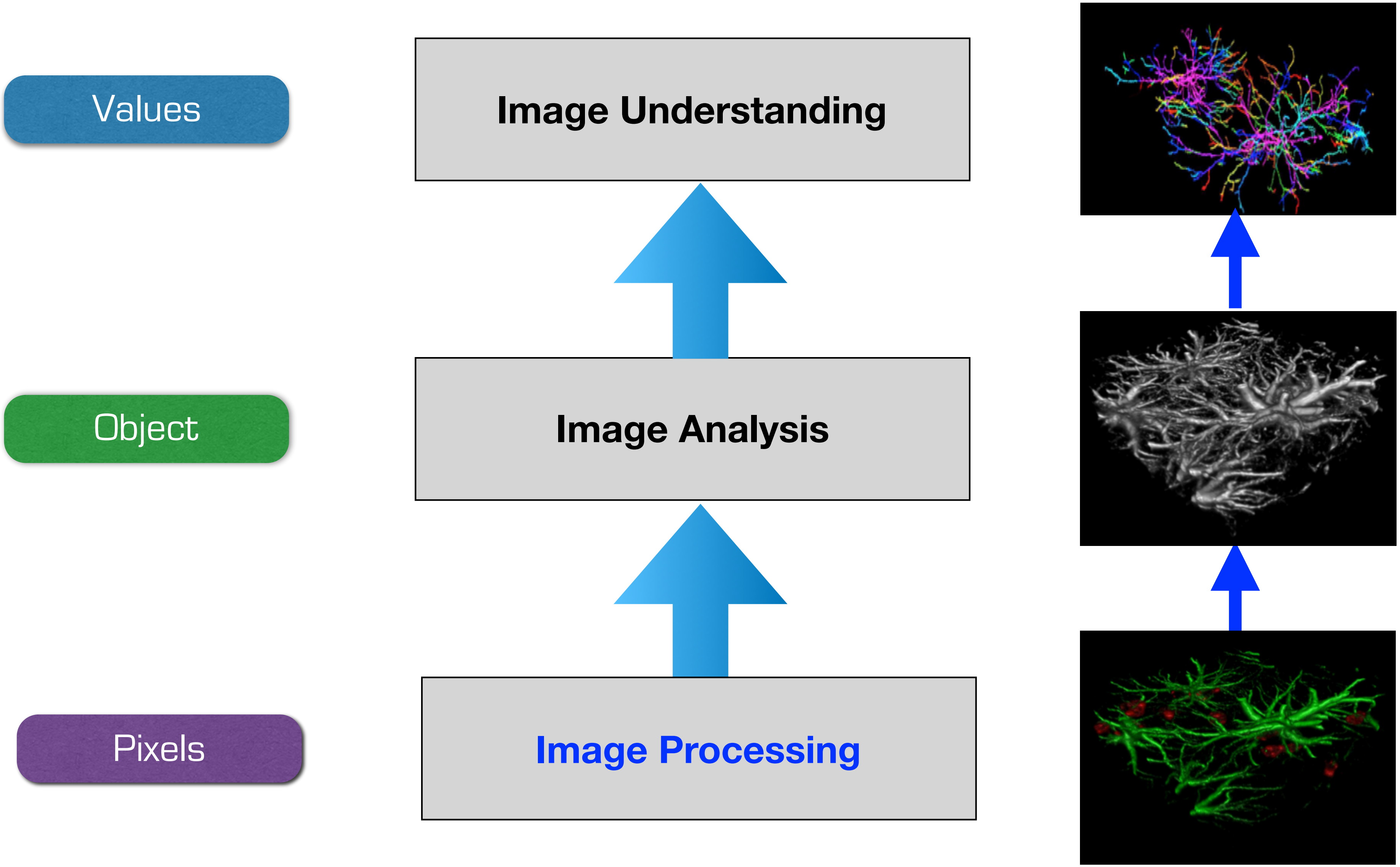
Image segmentation

Vannary Meas-Yedid
BioImage Analysis Unit
Institut Pasteur

vmeasyed@pasteur.fr

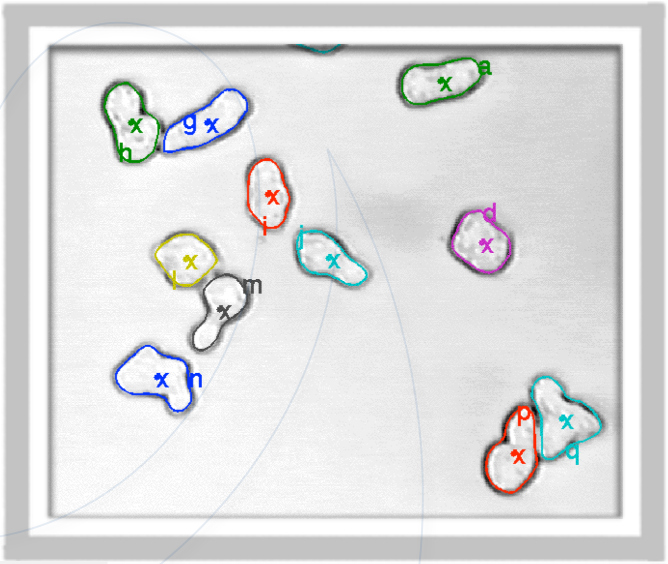
Introduction

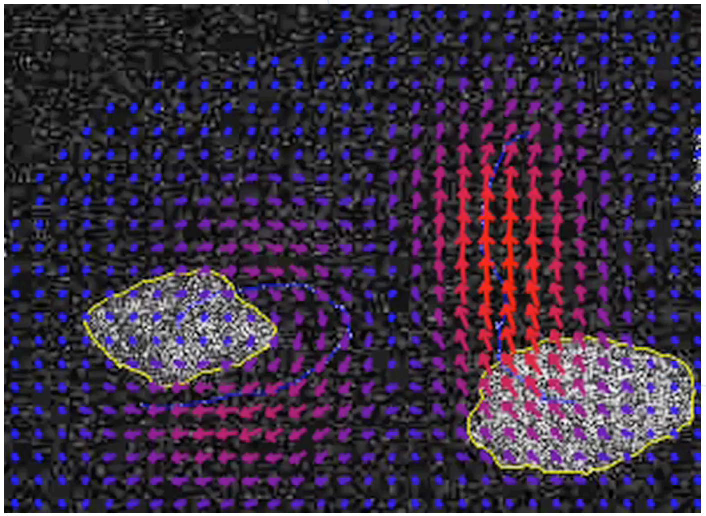


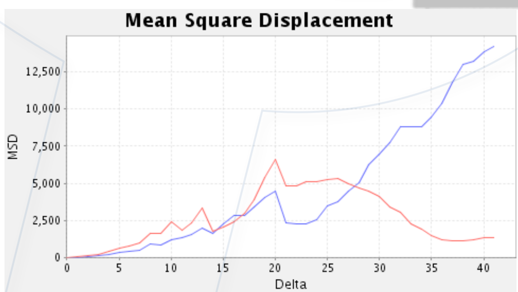
Segmentation: what for?

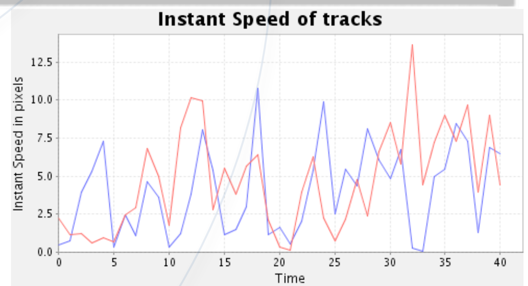
The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze

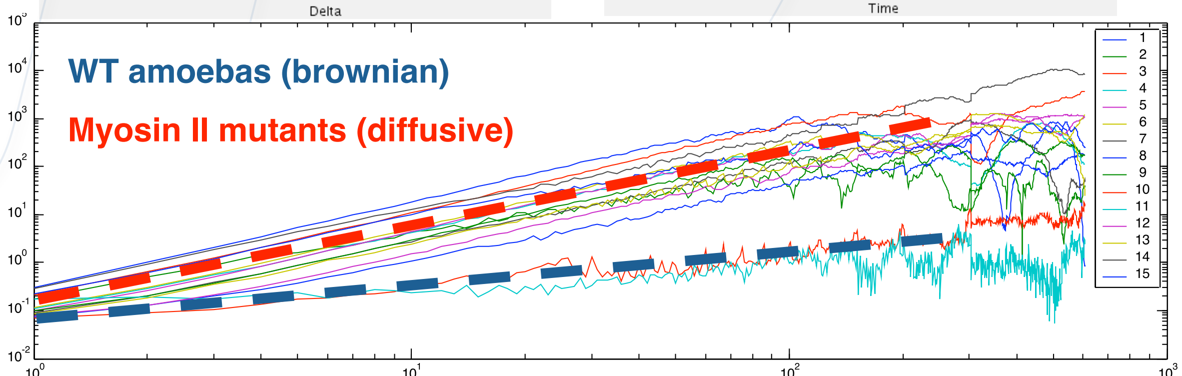
- Region of Interest detection
- Object recognition
- Object quantification
- Object tracking
- Compression











Segmentation: what for?

Partition an image into multiple homogeneous segments components, corresponding to real world objects

Formally (Horowitz 75)

X: the image domain *I*

P: predicate defined on the set of parts of X

Segmentation X: $(S_i)_{i=1,\ldots,n}$, sub-sets of X such as:

$$1.\ X = \cup_{I=1}^{n} S_i\ (Partition\ X)$$

$$2.\ \forall i \in 1, ..., n, S_i\ is\ connected\ and\ P(S_i) = true$$

$$3.\ \forall j \in 1, ..., n, S_i\ is\ adjacent\ to\ S_j\ and\ i \neq j \implies P(S_i \cup S_j) = false$$

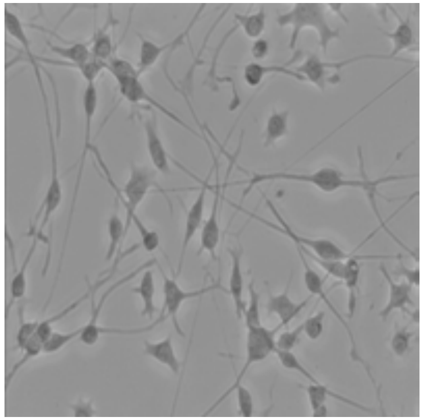
Some predicate examples

- $P(R) = \text{Vrai} \Leftrightarrow \sigma_R < 5$
- $P(R) = \text{Vrai} \Leftrightarrow \forall p \in R, |I(p) - \mu_R| < 10$

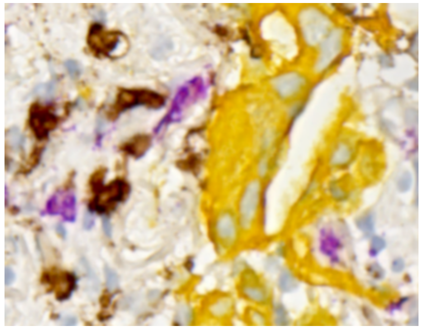
How to group pixels ?

Components share common properties

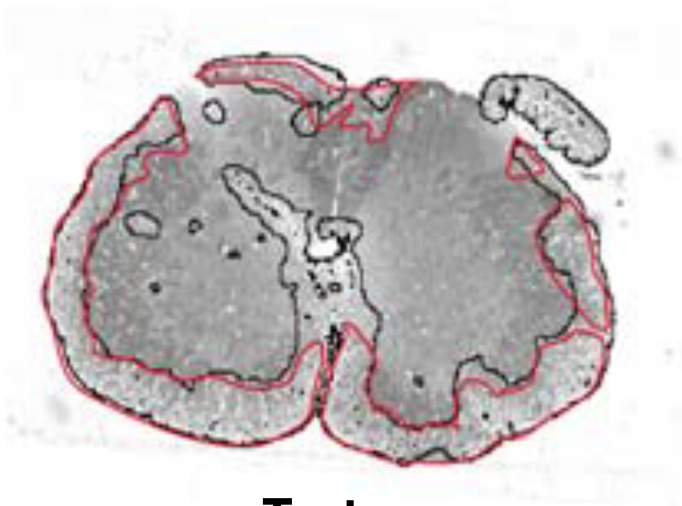
Similarity → **Region**



Intensity



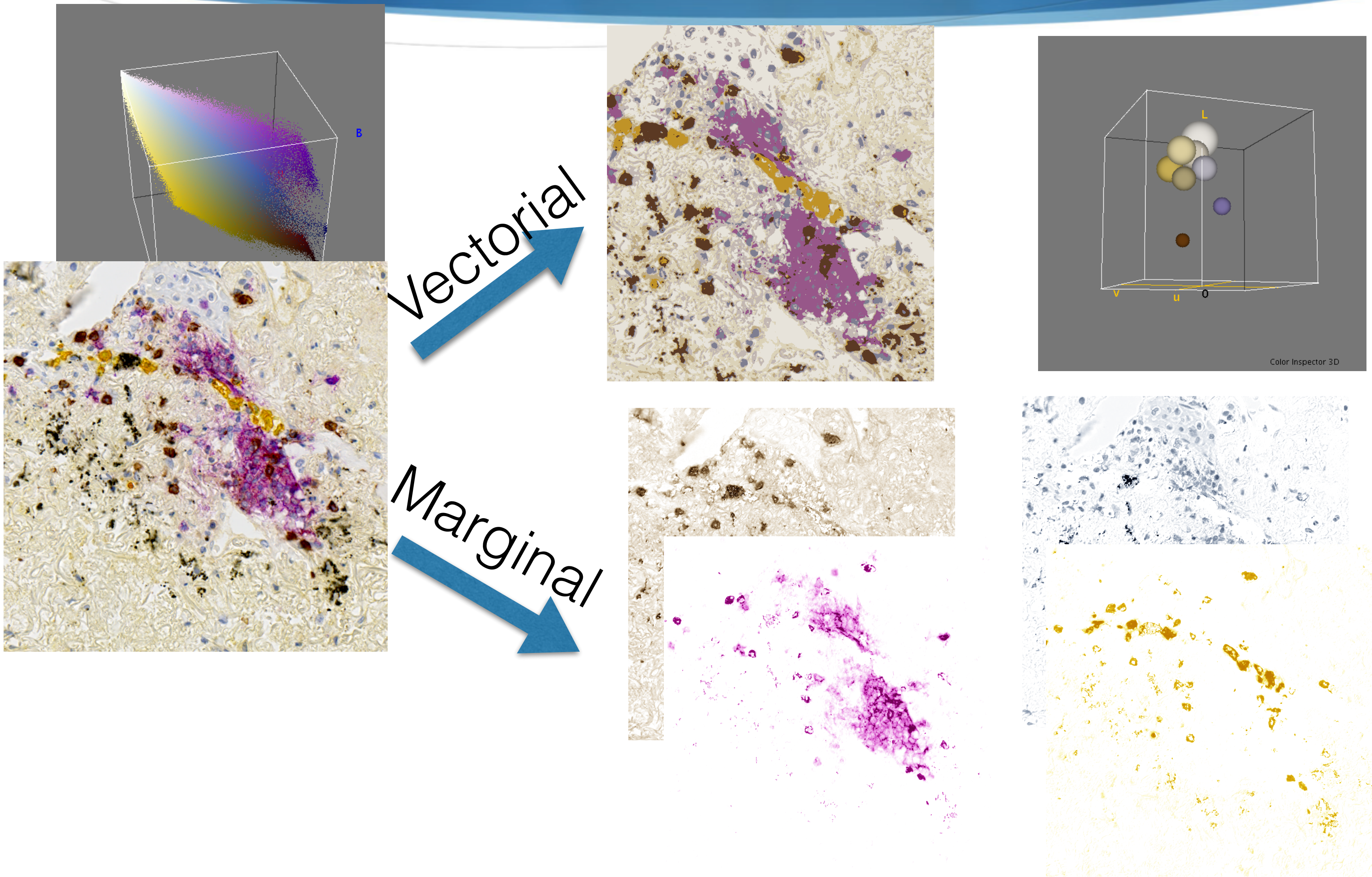
Color



Texture

Discontinuity → **Edge**

Color Processing



Color deconvolution, color unmixing, color decomposition, color separation

Introduction

Edge-based approach
- Derivative methods
- Optimal filters

Global approach: statistical methods
- Thresholding

Segmentation as clustering
- K-means, Fuzzy c-means
- Adaptive threshold

Region approach: geometric methods
- Region growing
- Split & merge
- Watershed

Energy minimisation approach
- Graph-based segmentation
- Superpixels
- Graph cuts
- Markov Random Fields, Conditional Random Fields
- Deformable models

Machine Learning
- Deep Learning: CNN

Edge detection

Contour: transition zone defined by its delineation

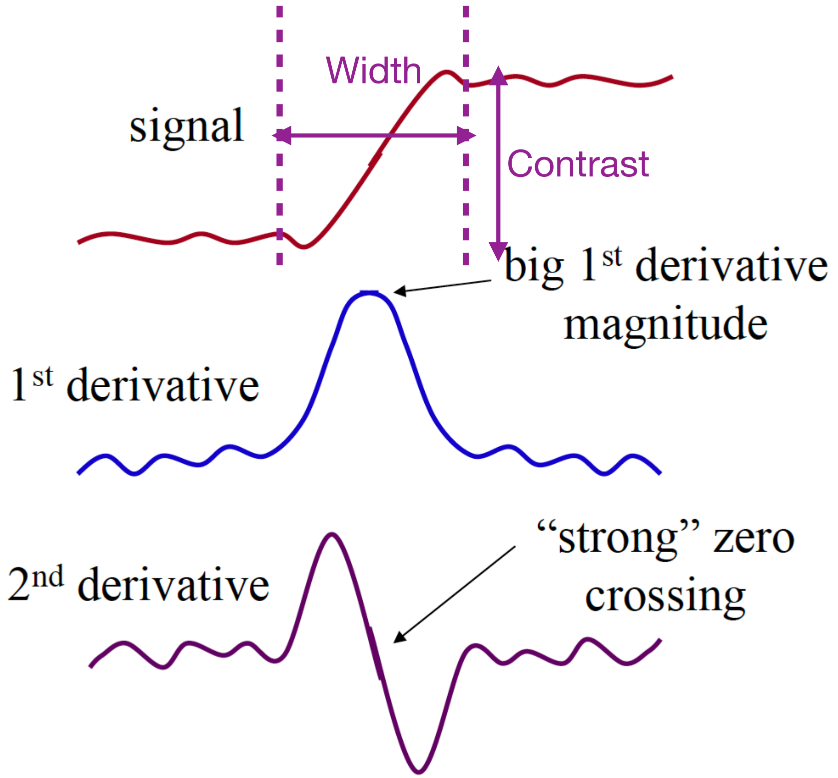
Defined by its width and contrast

Derivative approaches

- Filter of 1rst derivative order

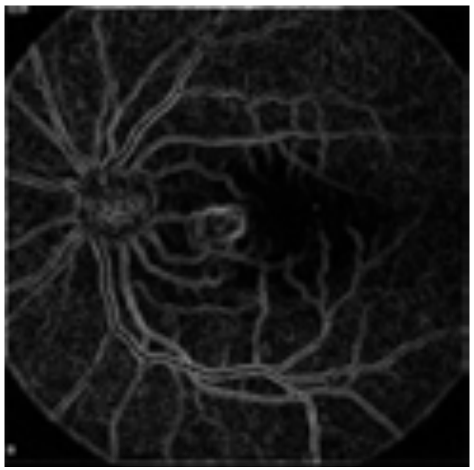
$$\begin{bmatrix} 1 & 0 \\ -1 & 0 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$$

gradient Roberts Prewitt Sobel



MIT OpenCourseWare (http://ocw.mit.edu)

Original



Sobel

From computer vision
Roberts 1965 Lincoln Lab
Horn1972 MIT AI
Marr and Hildreth 1980 MIT AI

Edge detection

- Filter of 2nd derivative order

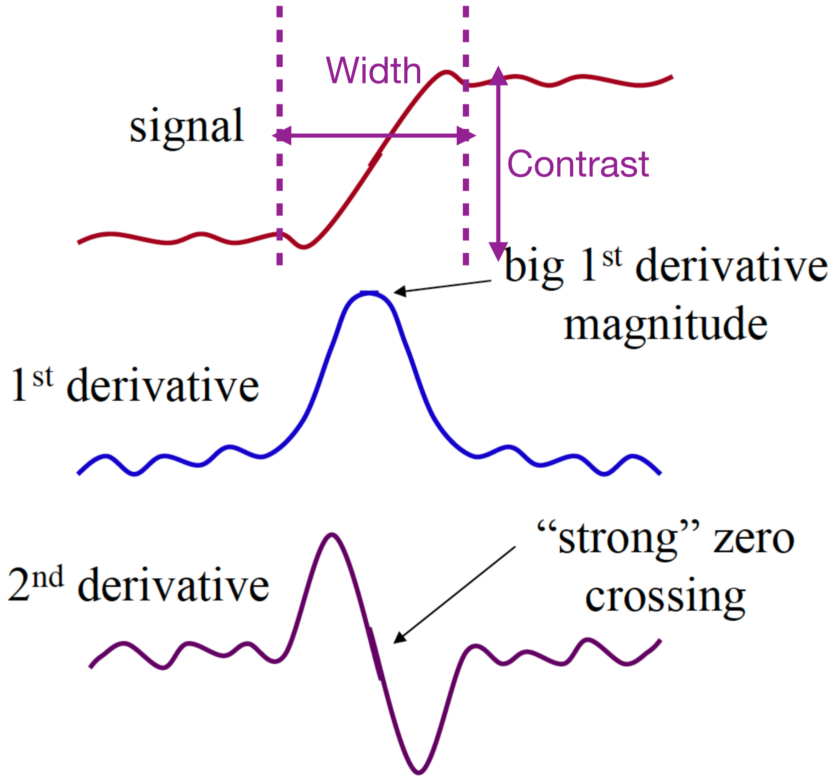
$$\frac{\partial^2 I(x,y)}{\partial x^2} \approx I(x+1,y) - 2I(x,y) + I(x-1,y)$$

$$\frac{\partial^2 I(x,y+1)}{\partial y^2} \approx I(x,y) - 2I(x,y) + I(x,y-1)$$

Discrete formulation

$$\Delta^2 = f(k+1,l) + f(k,l+1) + f(k-1,l) + f(k,l-1) - 4f(k,l)$$

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$



MIT OpenCourseWare (http://ocw.mit.edu)

From computer vision
Roberts 1965 Lincoln Lab
Horn1972 MIT AI
Marr and Hildreth 1980 MIT AI

D. Marr and E. C. Hildreth, Theory of Edge Detection, Proc. of the Roy. Soc. Lond. B, Vol 207, 1980.

Edge detection

Optimal approaches

Search for *h* optimal according to the following criteria:

- Good detection
- Good localisation
- Unicity

Canny's filter (1986)

1. Filter image with derivative of Gaussian
2. Find magnitude and orientation of gradient
3. Non-maximum suppression
4. Linking and thresholding (hysteresis):

Define two thresholds: low and high

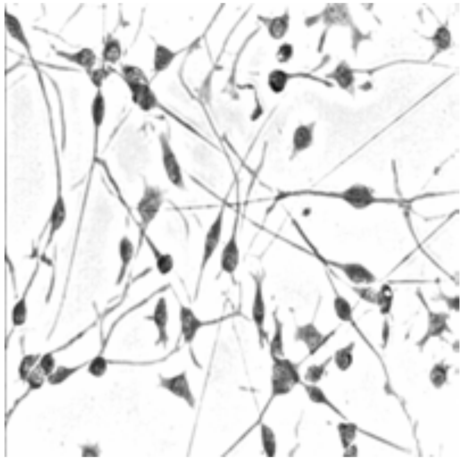
Use the high threshold to start edge curves and the low threshold to continue them

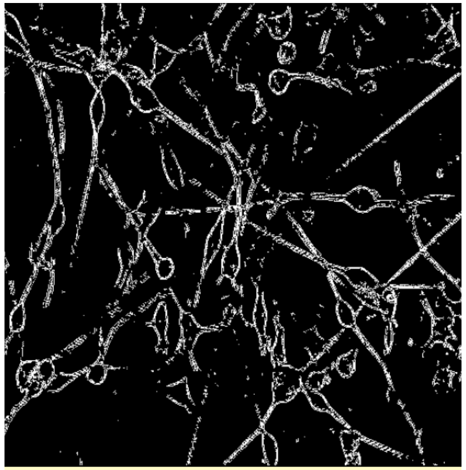
Variant: Deriche

J. Canny, A Computational Approach To Edge Detection, IEEE Trans. Pattern Analysis and Machine Intelligence, 8:679-714 (1986).

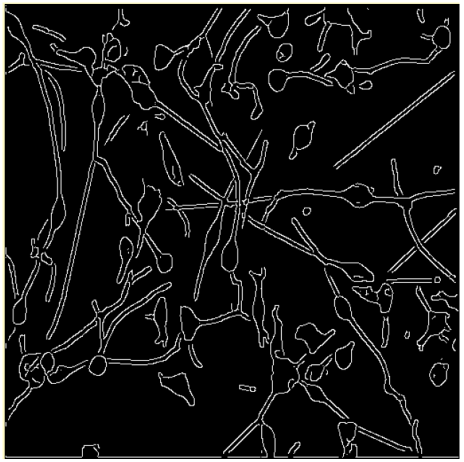
Edge detection

Cloppet 94

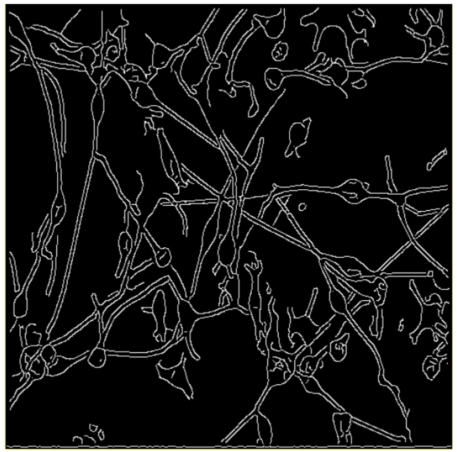




Laplacian



Deriche



Deriche+hysteresis thresholding

Histogram-based Thresholding

Threshold detection

- Test value
- Mean value
- Median value
- automatic value

M. Sezgin and B. Sankur, J. Electron. Imaging, vol. 13, no. 1, pp. 146–165, 2004.

Otsu's method (1979)

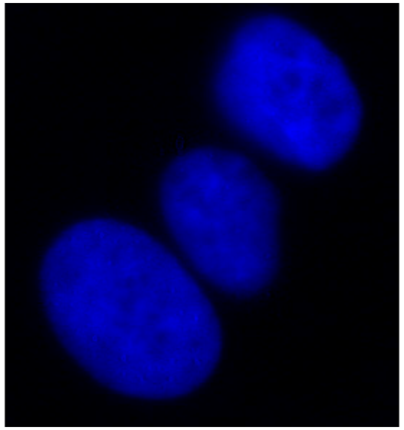
A threshold *t* defined two groups of pixels: C_1 and C_2

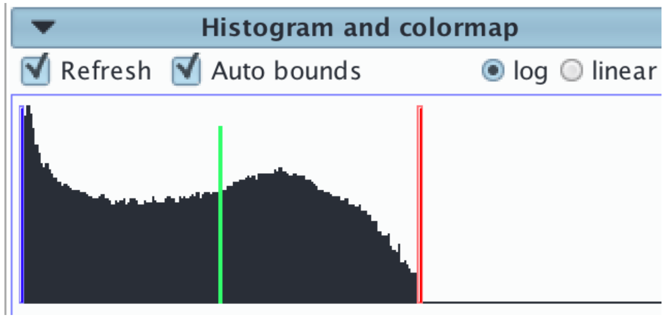
The threshold will minimise the variance of the intra class

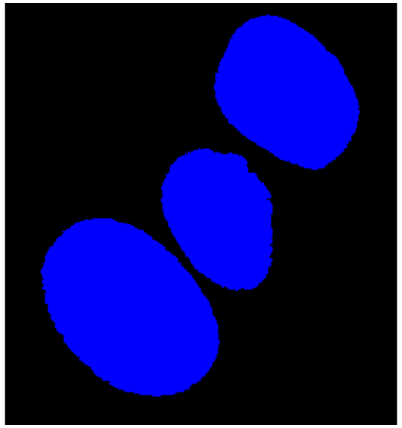
$$\sigma_\omega^2(t) = \omega_1(t)\sigma_1^2(t) + w_2(t)\sigma_2^2(t)$$

The weights $\omega_i(t)$ represent the probability to be in the i^{th} class

The σ_i^2 are the variance of the classes



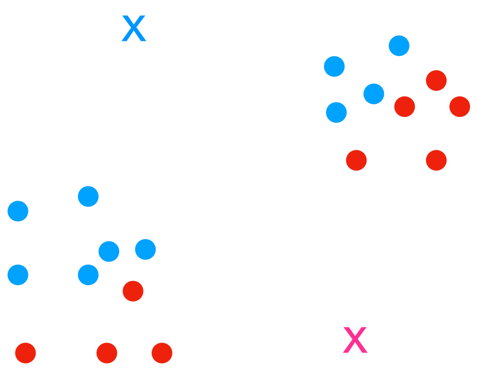




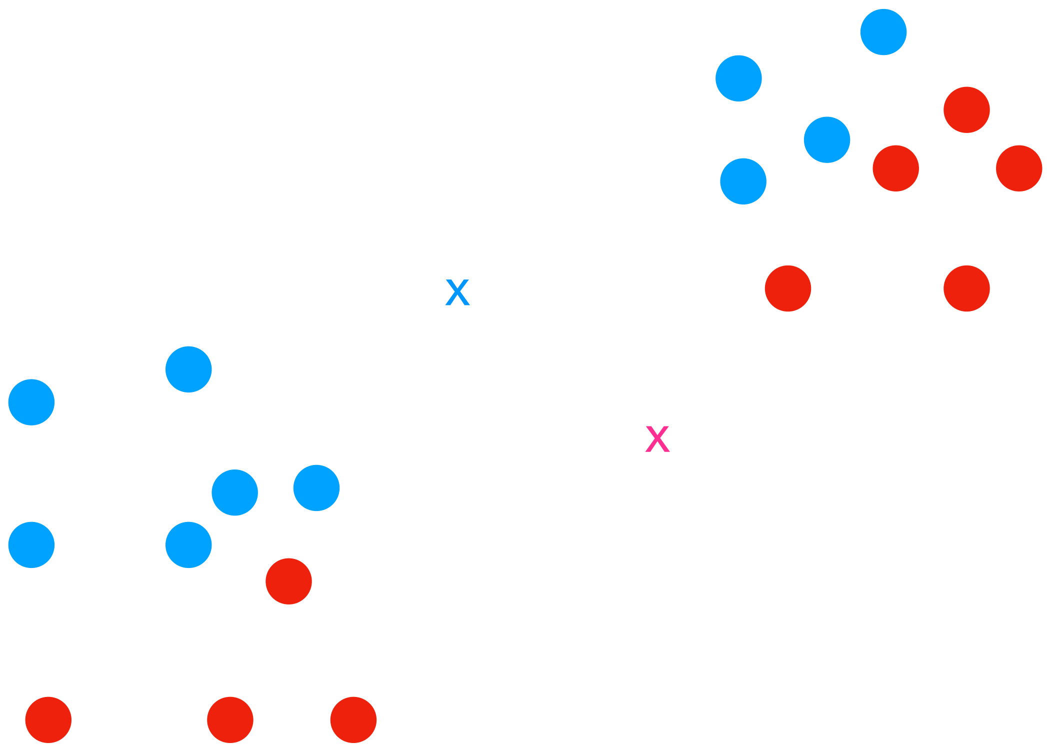
K-Means

Algorithm:

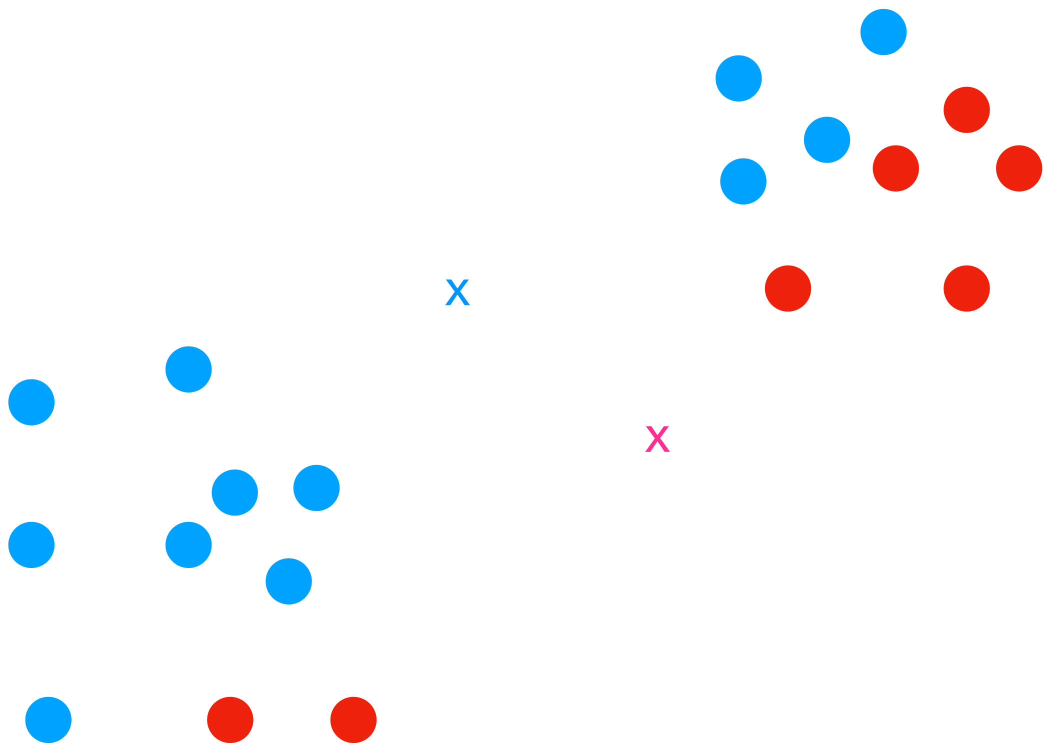
1. Randomly initialize the cluster centers, $c_1, \ldots, c_K$
2. Given cluster centers, determine points in each cluster
   - For each point p, find the closest c_i. Put p into cluster i
3. Given points in each cluster, solve for c_i
   - Set c_i to be the mean of points in cluster i
4. If c_i have changed, repeat Step 2



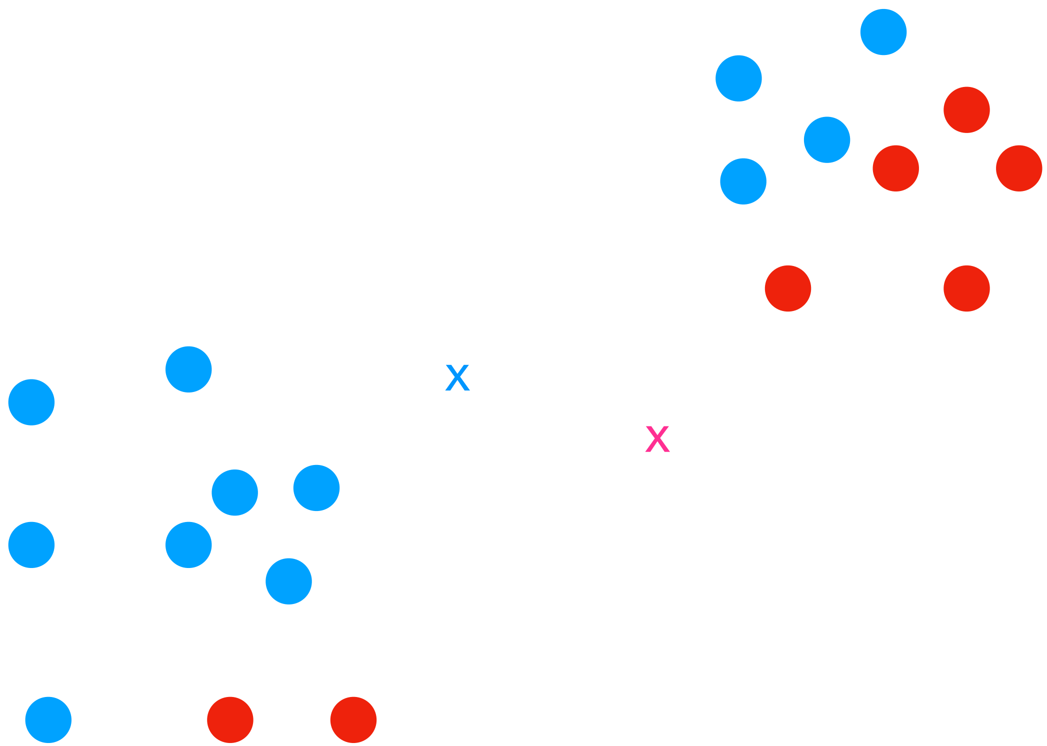
Clustering



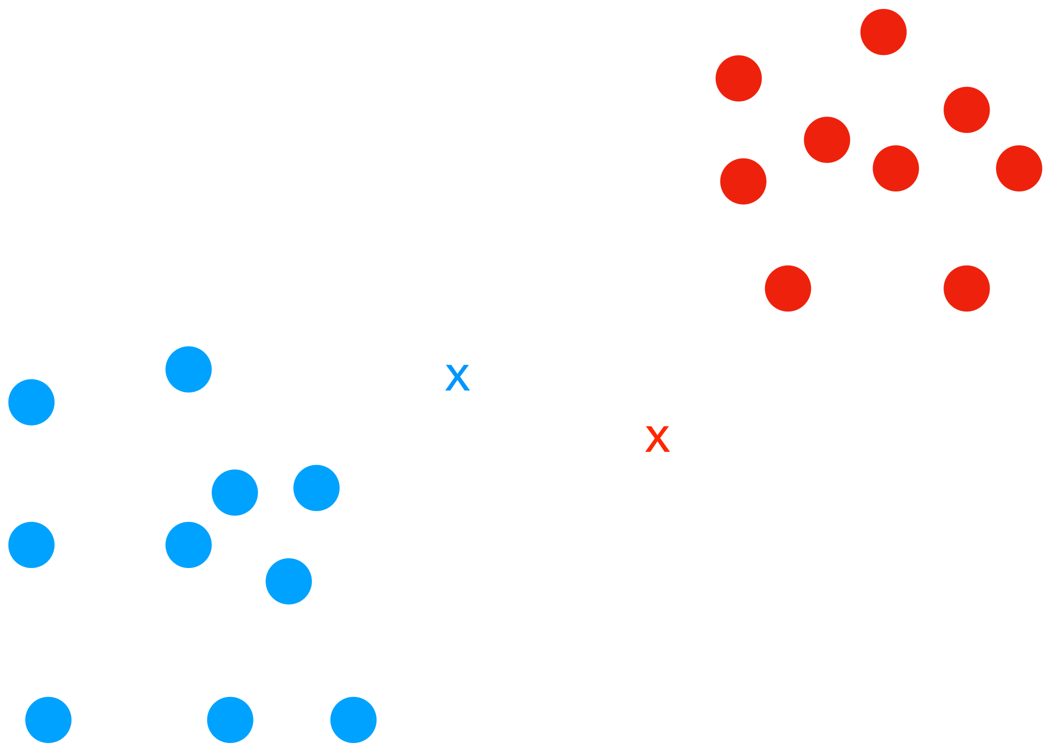
Clustering



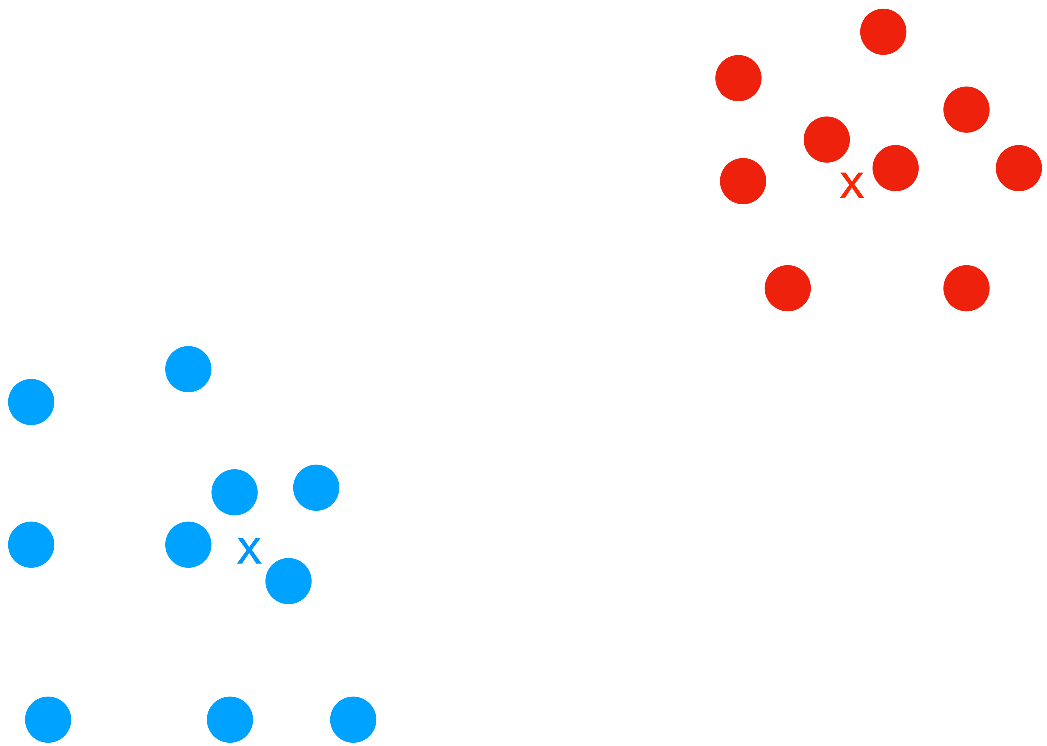
Clustering



Clustering



Clustering



Summary of Global Approaches

Pros:
Simple, fast
Adapted to multimodal histograms

Cons:
Number of classes must be known
Choose the threshold
No spatial information

Region growing

- Start from seeds
- Compute mean μ_R and standard deviation σ_R
- We add to R all neighbour pixels that are similar to *R*

$$|I(x) - \mu_R| < \text{seuil}$$

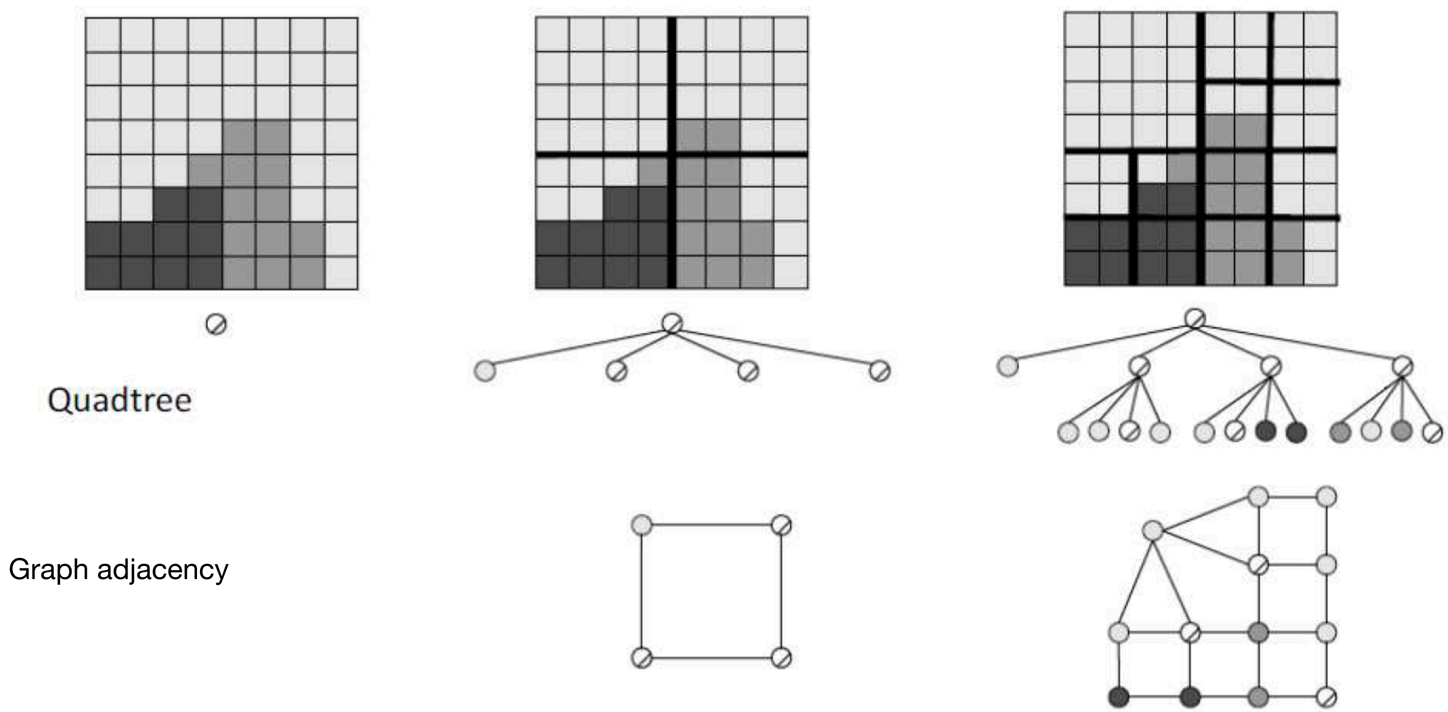
ou bien

$$\begin{cases} \min\{|I(x) - I(y)|; y \in R \cap V(x)\} < \text{seuil} \\ |I(x) - \mu_R| < \sigma_R \end{cases}$$

Split & Merge

Pavlidis and Horowitz 1974

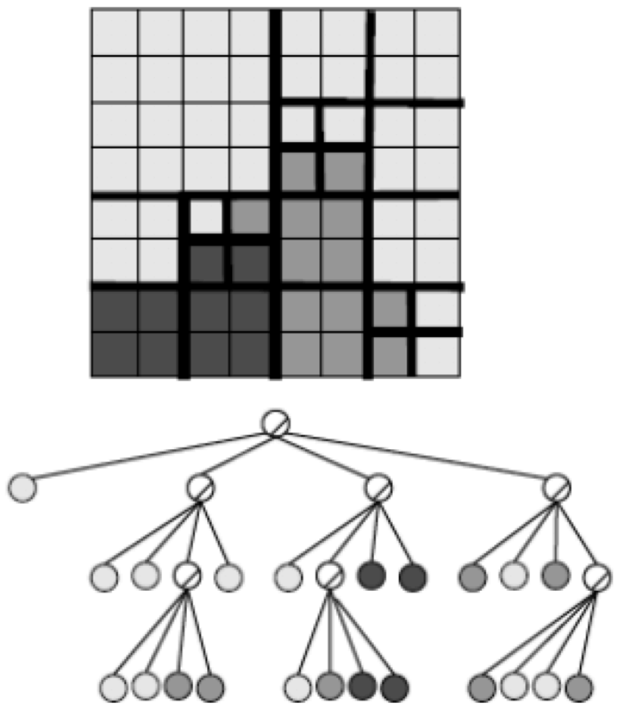
Split step

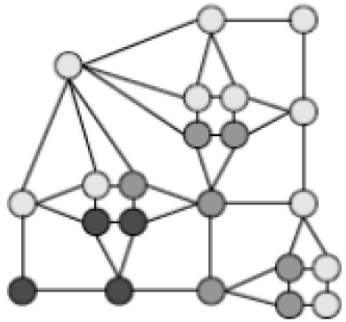


Split & Merge

Pavlidi and Horowitz 1974

Split step results in
Over-segmentation

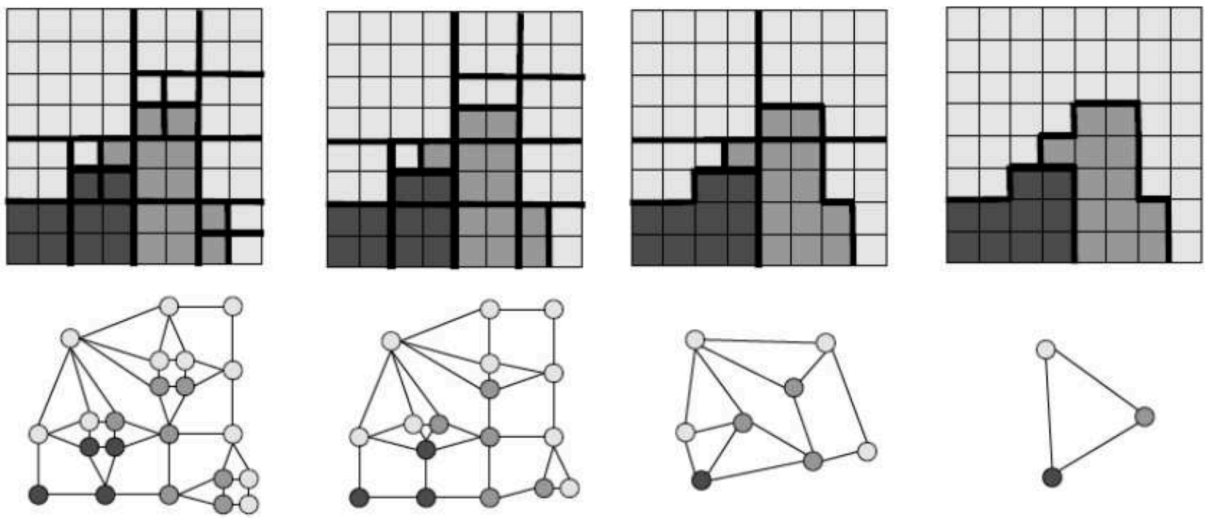




Split & Merge

Pavlidis and Horowitz 1974

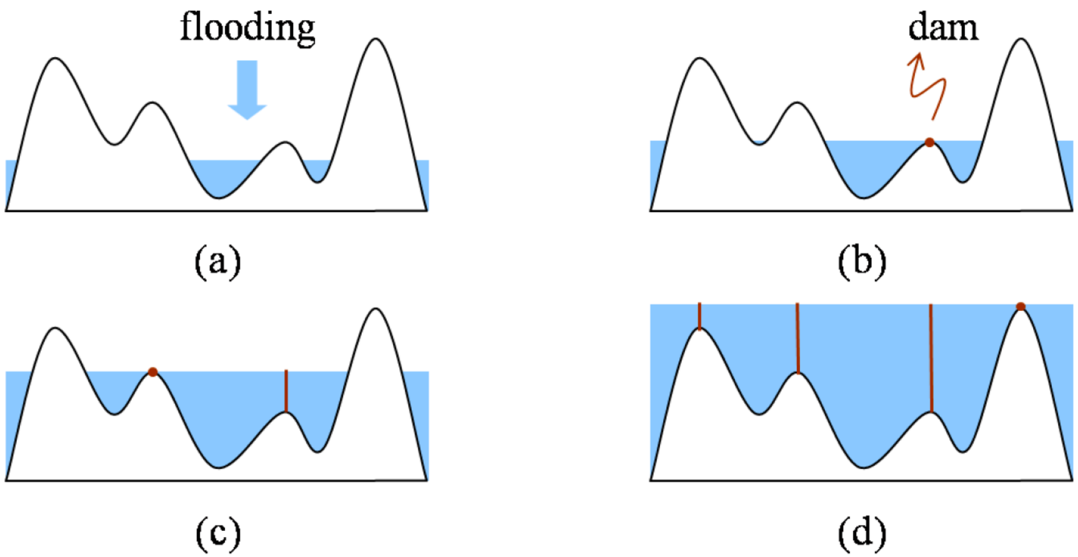
Merge step



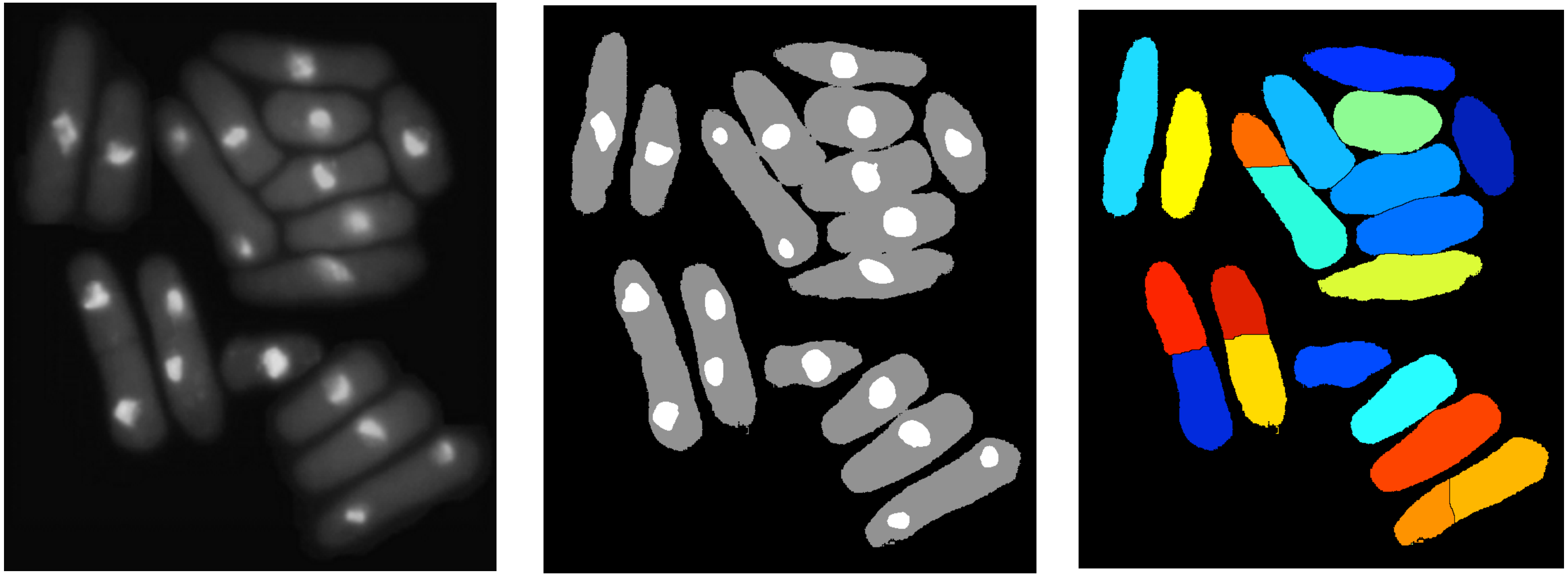
Pro: Hybrid method (local/global)
Cons: square structures in the resulted image
Sensitive to the order of the processing

Watershed segmentation

Beucher and Lantuejoul 1979



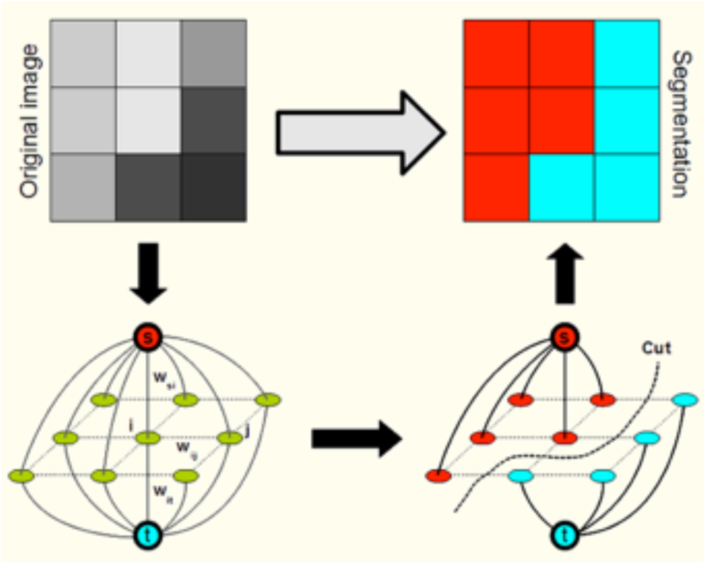
Watershed segmentation



Graph-based segmentation

Markov

Graph-Cut



Goal: Minimize the energy function

$$E(L) = \sum_{p \in \mathcal{P}} D_p(L_p) + \sum_{(p,q) \in \mathcal{N}} V_{p,q}(L_p, L_q),$$

Interpreted as

$$E(A) = \lambda \cdot R(A) + B(A)$$

where

$$R(A) = \sum_{p \in \mathcal{P}} R_p(A_p) \quad \textit{(regional term)}$$

$$B(A) = \sum_{\{p,q\} \in \mathcal{N}} B_{p,q} \cdot \delta_{A_p \neq A_q} \quad \textit{(boundary term)}$$

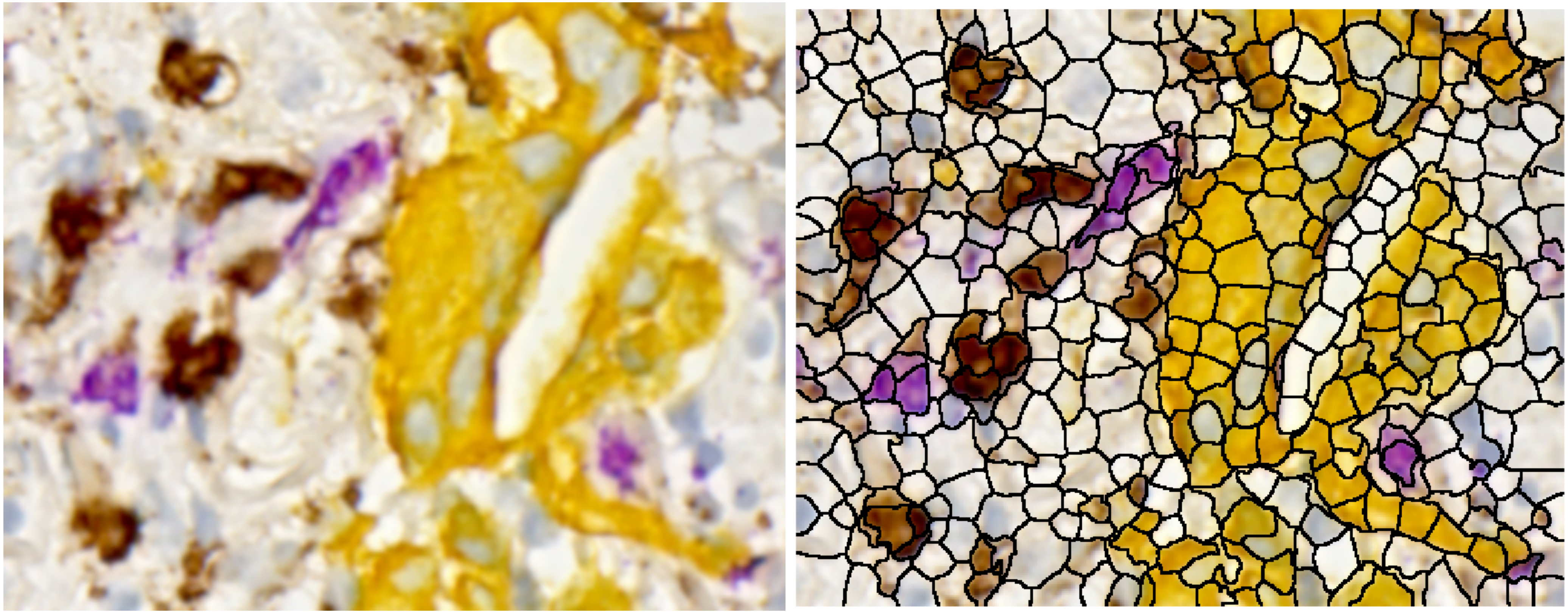
Wu et Leahy (1993)
Boykov, Y., Funka-Lea, G. 2006. Graph Cuts and Efficient N-D Image Segmentation. In *International Journal of Computer Vision* 70(2), 109–131.

Technique:

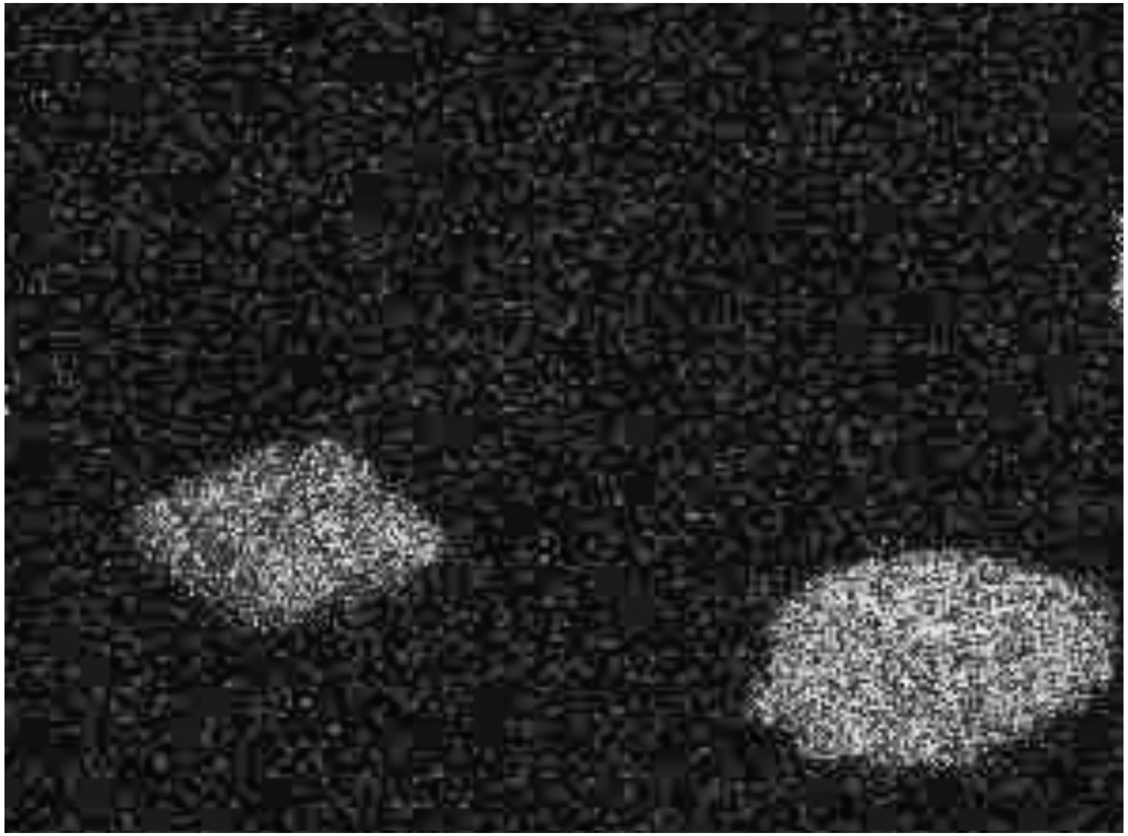
Minimum cut / maximum flow (Ford-Fulkerson algorithm)

Graph-based segmentation

Superpixels
SLIC
SCALP http://www.labri.fr/perso/rgiraud/research/scalp.php



1. Pixel classification
   (automated threshold)
2. Post-processing
   (math. morphology)
3. Objects extraction

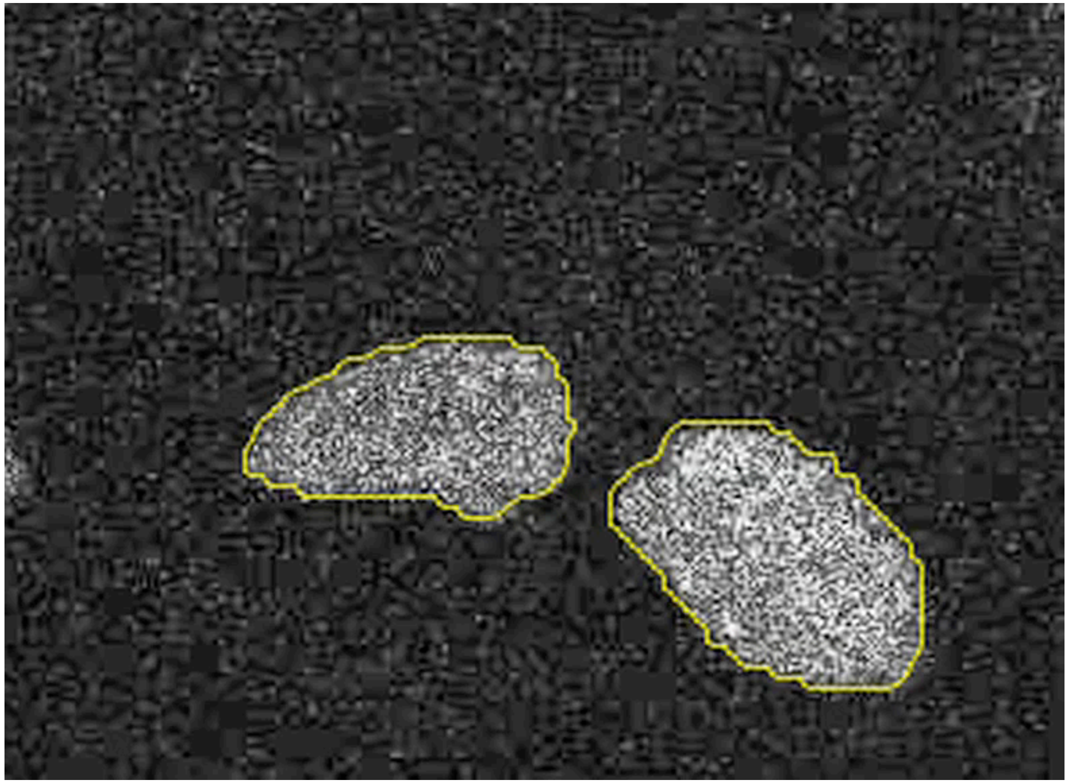


touching objects get confused...

Deformable models

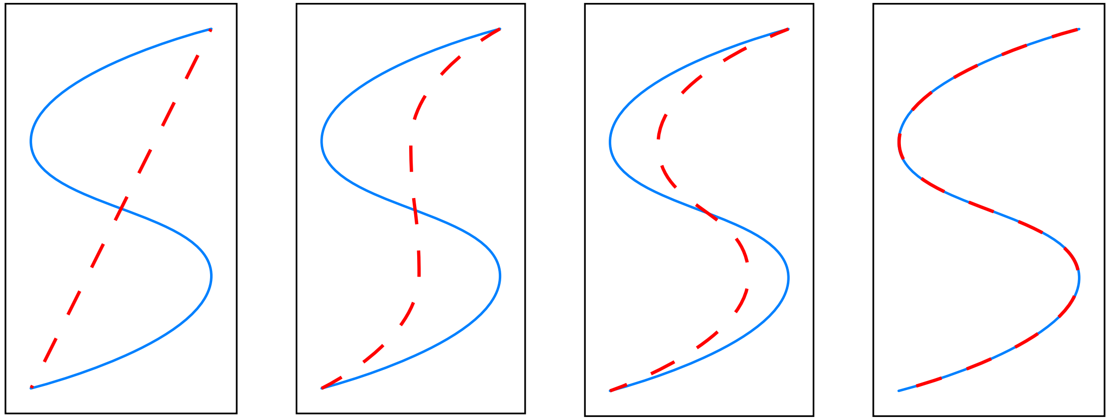
1) Define an initial contour and a cost functional
2) Let the contours move until stabilisation

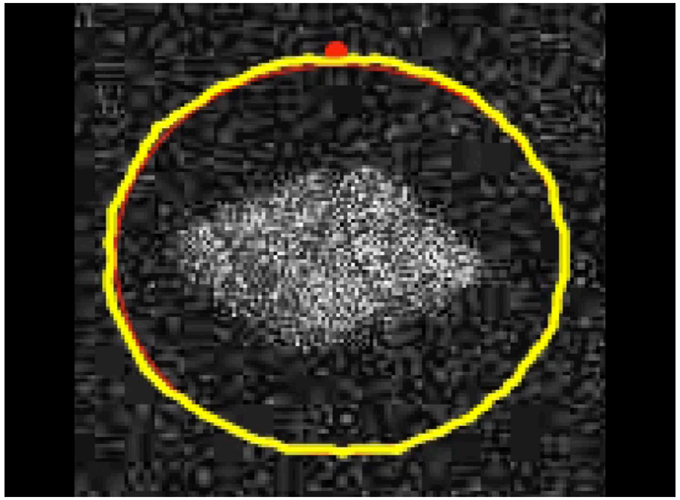
preserve the identification of touching objects

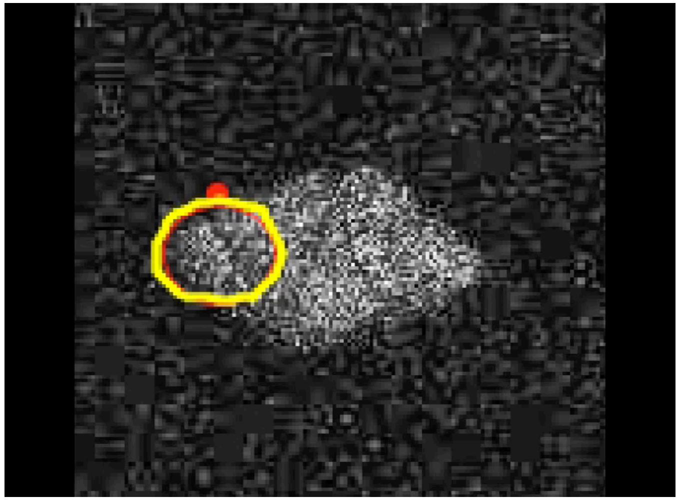


General idea

Deform an initial contour toward features of interest



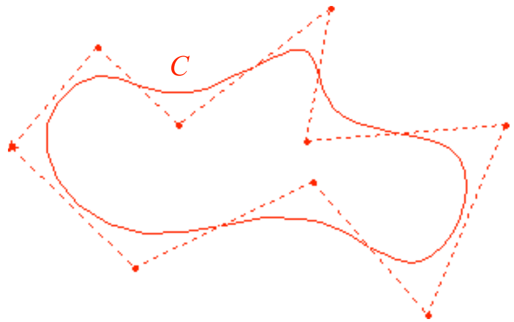


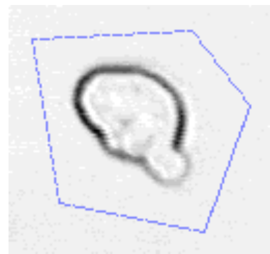


from the outside... ... or from anywhere

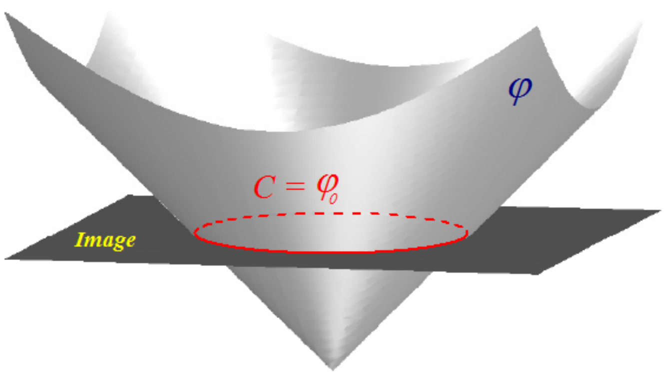
2 representations

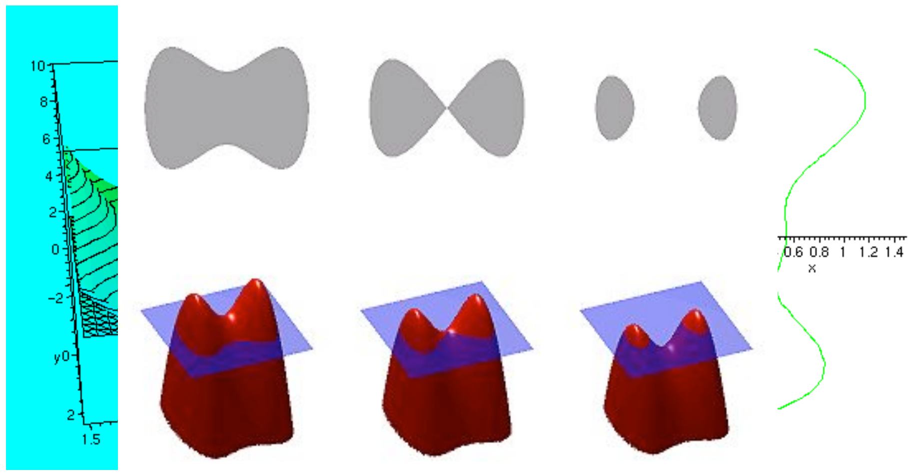
- parametric contours





- implicit contours





S1

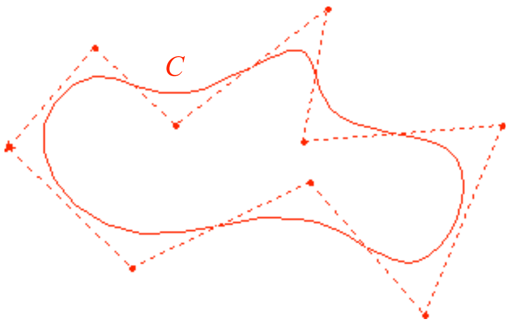
Curves representation (1)

Explicit representation

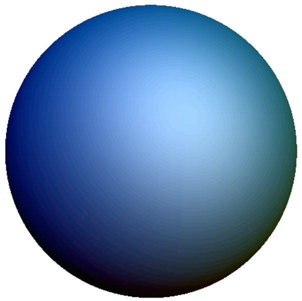
The contour is a ***parametric curve*** defined on the image domain

$$\Gamma : (p_1, \ldots, p_n) \in [0,1]^n \longmapsto \Omega \subset \mathbb{R}^n$$

Examples



$$\begin{aligned} \mathcal{C} : \quad & p \in [0,1] && \longmapsto \Omega \subset \mathbb{R}^2, \\ \mathcal{S} : \quad & (p,q) \in [0,1]^2 && \longmapsto \Omega \subset \mathbb{R}^3 \end{aligned}$$



Geometric representation: usually spline / B-spline functions

Light and efficient representation (very popular in CG / CAD fields)

Parametrization issues (number of control points, spline degree and parameters)

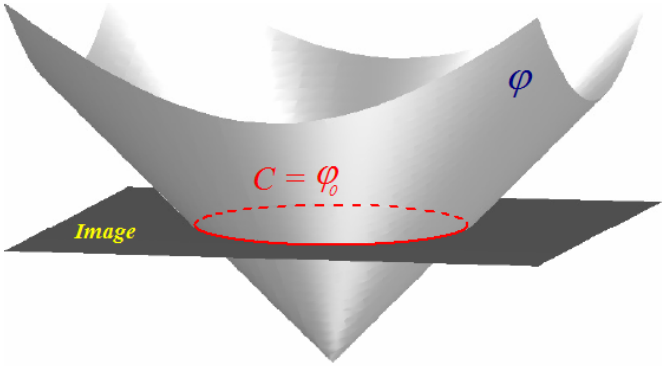
Rigid topology (cannot represent curve fusions or divisions)

Curves representation (2)

Implicit representation

The contour is embedded in a higher-dimensional ***level set*** function

$$\varphi : (x_1, \ldots, x_n) \in \Omega \subset \mathbb{R}^n \longmapsto \mathbb{R}$$

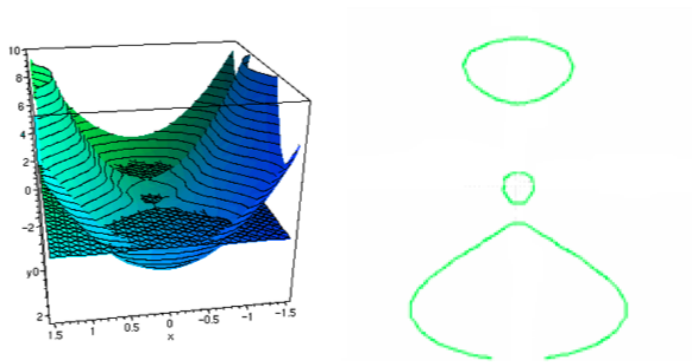


Convention: $\begin{cases} \Gamma = \varphi_0 = \{x_{1\ldots n},\ \varphi(x_{1\ldots n}) = 0\} \\ \varphi < 0 \quad \text{inside } \Gamma \\ \varphi > 0 \quad \text{outside } \Gamma \end{cases}$

Implicit topology (popular in fluid dynamics)

Indirect contour manipulation

Computationally more demanding



Problem formulation (1)

How to deform the contour ?

- The features of interest may be anywhere (near, far, inside, outside)
- Noise is on the way
- How to stop the deformation ?

... a classical inverse problem !

Typical solution: re-write the problem as an energy-minimizing one

$$\hat{\Gamma} = \arg\min_{\Gamma} E(\Gamma)$$

How to write E ?
How to deform the contour such that E decreases ?

Snake

$$E(\mathcal{C}) = \int_0^1 |\nabla I(p)|^2 + \alpha \left|\frac{\partial \mathcal{C}}{\partial p}\right|^2 + \beta \left|\frac{\partial^2 \mathcal{C}}{\partial p^2}\right|^2 dp$$

Physical approach of the contours (Kass, Witkin, Terzopoulos 88)

- An initial contour move under the action of several force actions :

$$E[(C)(p)] = \alpha \int_0^1 E_{int}(C(p))dp \; + \; \beta \int_0^1 E_{img}(C(p))dp \; + \; \gamma \int_0^1 E_{con}(C(p))dp$$

- The **internal energy** control the regularity et l'elasticity of the courbe
- The **image energy** guide the active contour toward the desired features of the image (gradient, dark/bright region, etc.)
- The **external energy** allow taking into account the *a priori knowledge* defined by the user (shape of the structures to segment)
- The equilibrium of these terms is reached when the potential of the cost function is minimum

The original «snake» model (Kass *et al.*, 1988)

- Standard 2D parametric case following image gradients

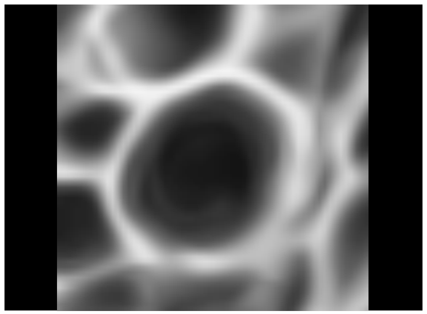
$$E(\mathcal{C}) = \int_0^1 |\nabla I(p)|^2 + \alpha \left| \frac{\partial \mathcal{C}}{\partial p} \right|^2 + \beta \left| \frac{\partial^2 \mathcal{C}}{\partial p^2} \right|^2 dp$$

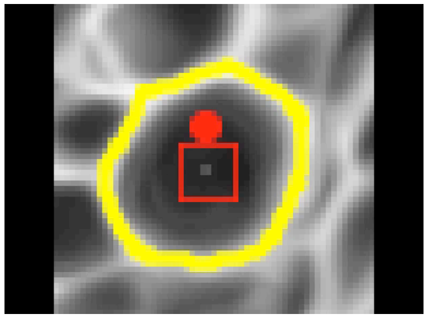
Extensions

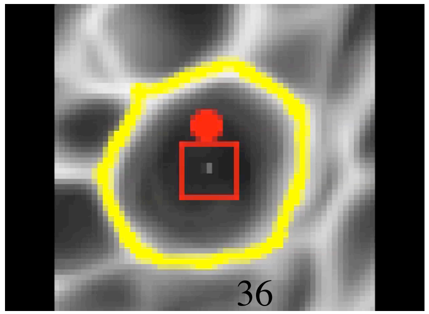
Parametric case:
- Balloon snakes ('91): a constant force {in/de}flates the curve
- Geometric active contours ('93): riemannian formulation
- Gradient vector flows ('97): gradients are propagated to reach distant objects

Level set case:
- Geodesic active contours ('93): analogous to the parametric model

Examples


original image


snakes


balloon snakes

Problem formulation (2)

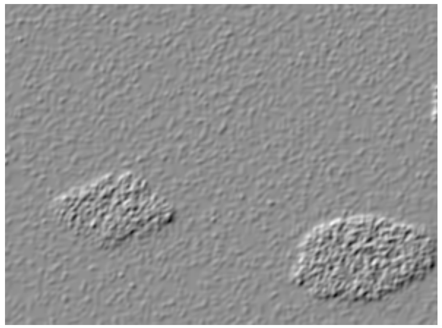
How to write E ?

- E should measure the fitting error between Γ and the image features

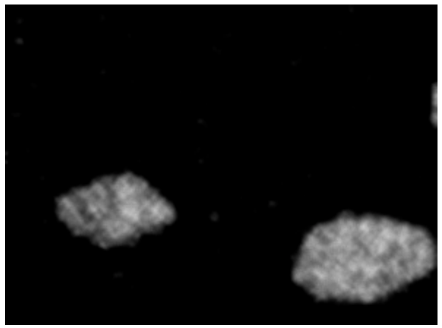
$$E(\Gamma) = E_{\text{image}}(\Gamma, I)$$

- E_{image} is called the «data attachment term»
- Which image information can be expressed here ?

- ***Edge-based:*** the target boundary is defined (at least partially) by high gradients (*wrt*. intensity, color, etc.)



- ***Region-based:*** the image is formed of multiple homogeneous (though distinct) regions (*wrt*. intensity, texture, total variation, etc.)

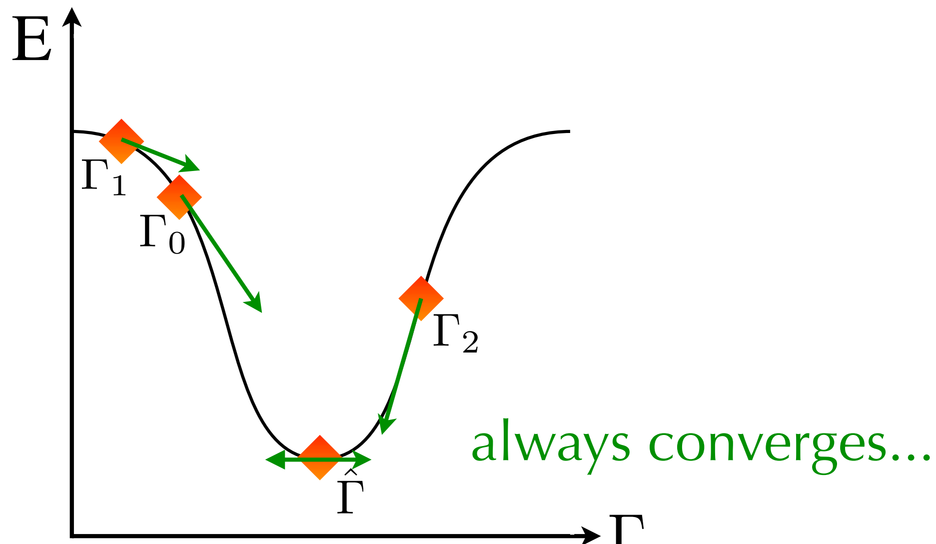


Problem formulation (3)

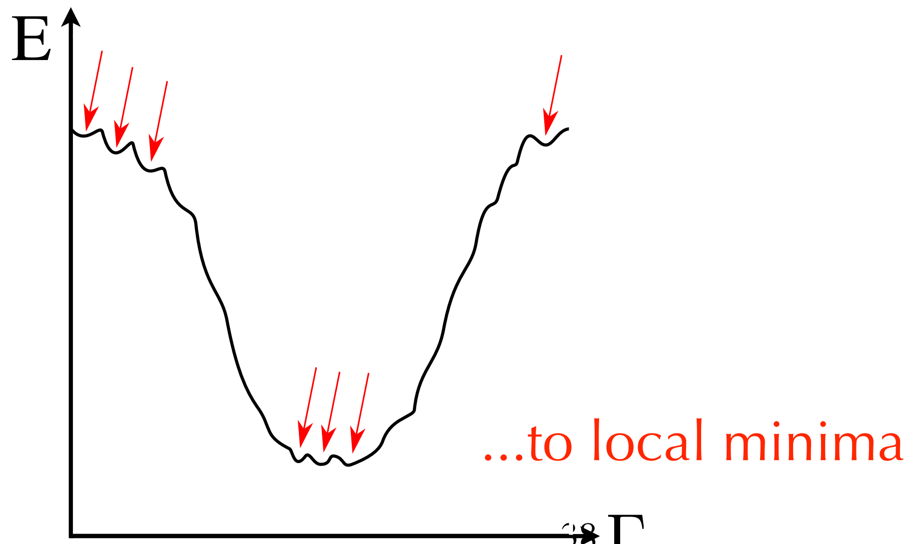
How to deform the contour such that E decreases ?

- The shape of E is unknown (or at least not computable for all possible Γ)
- Real question: How to «walk» in Γ space to find the lowest energy ?
- Popular solution: gradient descent (Euler-Lagrange)
  - assume E differentiable *wrt*. Γ
  - for any given Γ, compute $dE/d\Gamma$ to figure out where Γ should «walk»
  - if $dE/d\Gamma = 0$, Γ «should» be optimal

Simplistic example (1D Γ space):



«pseudo-ideal» case



«pseudo-real» case

Problem formulation (4)

How to deform the contour such that E decreases ?

- Minimizing E is *an **ill-posed problem*** and ***must be regularized***
- The final solution is now approximate but ***more stable*** to local minima
- The general energy becomes:

$$E(\Gamma) = E_{\text{image}}(\Gamma, I) + E_{\text{regul}}(\Gamma)$$

- ***Parametric models*** usually use Tikhonov's method (1963):
  - Assume smoothness of the solution by minimizing it's derivative(s)
  - Example in the case of a 2D curve:

$$E(\mathcal{C}) = \int_0^1 f_I(\mathcal{C}) + \alpha \left| \frac{\partial \mathcal{C}}{\partial p} \right|^2 + \beta \left| \frac{\partial^2 \mathcal{C}}{\partial p^2} \right|^2 + \ldots dp$$

data attachment — stretching energy — bending energy

- ***Level set models*** rely on fluid dynamics and mean curvature motion:
  - Assume smoothness of the level set by minimizing it's curvature

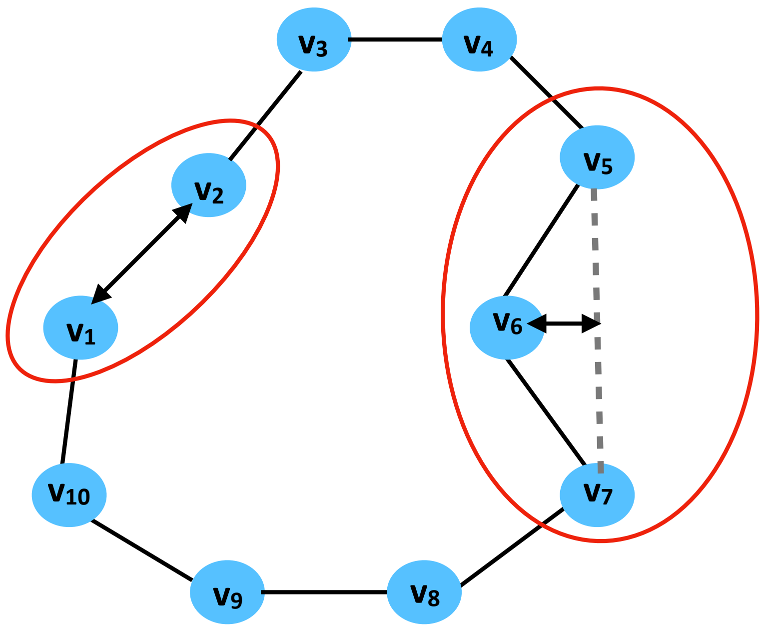
$$\frac{\partial \varphi}{\partial t} = |\nabla \varphi| F(\kappa) = |\nabla \varphi| F\left(\nabla \cdot \frac{\nabla \varphi}{|\nabla \varphi|} \right)$$

F is the speed function containing both data attachment and regularization

Internal energy

Elastic energy

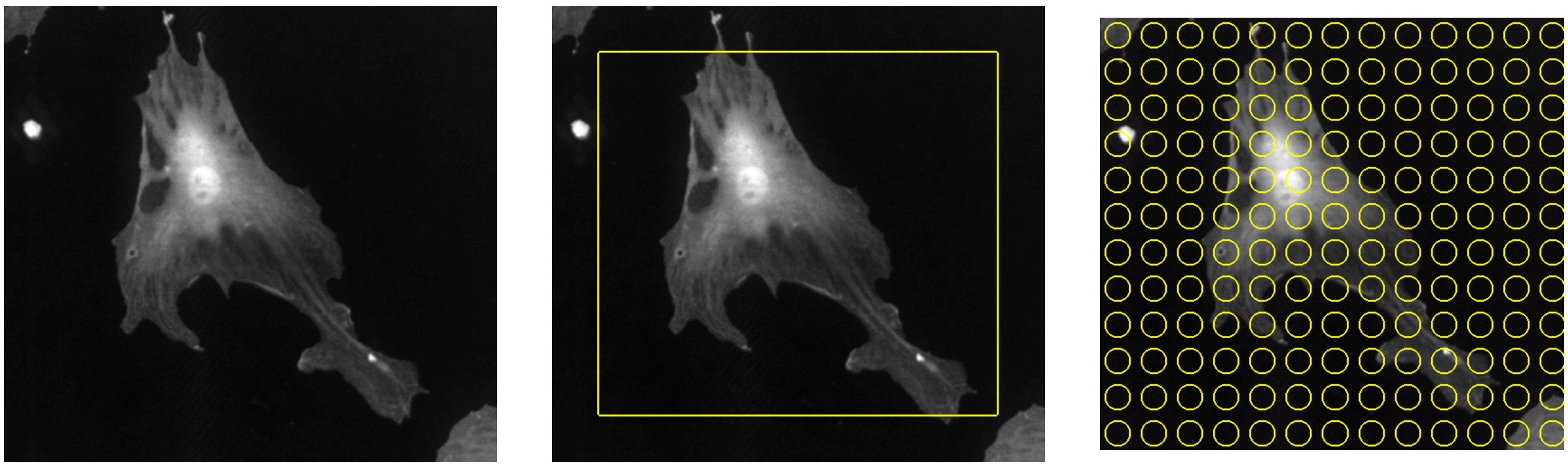
$$\frac{dv}{ds} \approx v_{i+1} - v_i$$



Bending energy
(Stiffness)

$$\frac{dv^2}{ds^2} \approx (v_{i+1} - v_i) - (v_i - v_{i-1}) = v_{i+1} - 2v_i + v_{i-1}$$

Initialisation issues



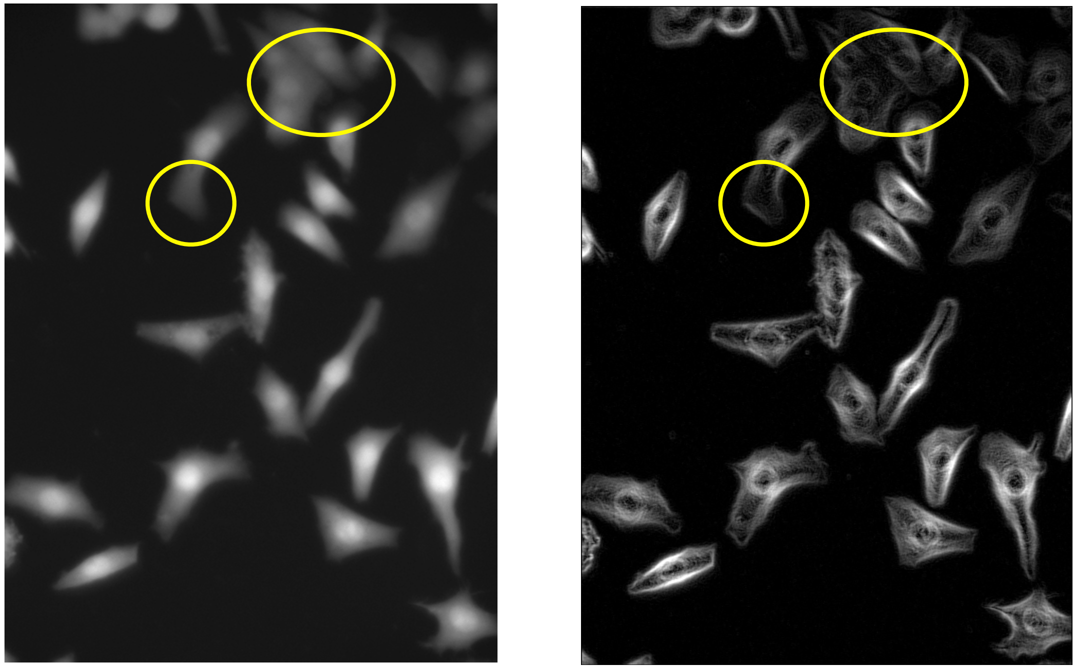
Segmentation models
Edge-based models

Limitations

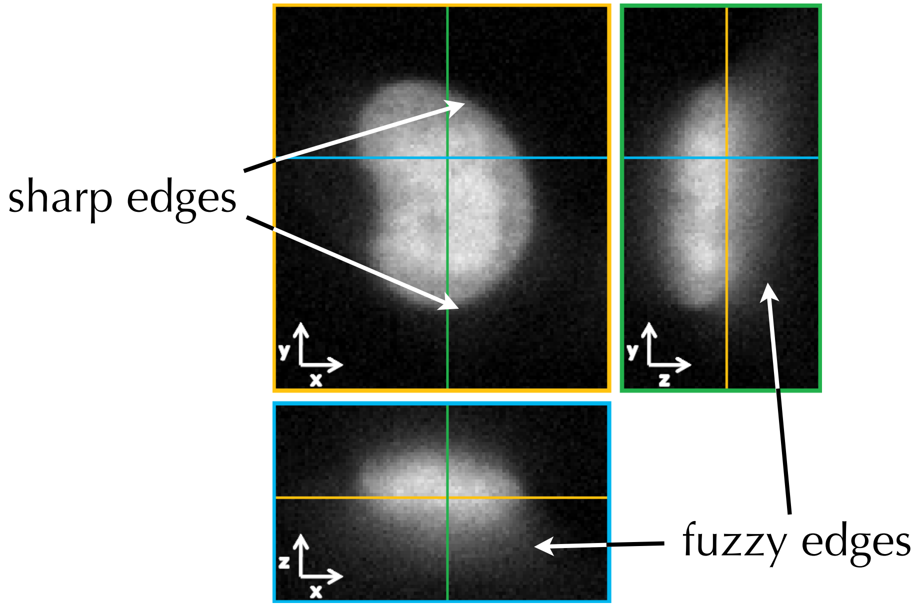
- Edge information is not suitable in all applications...

2D fluorescence microscopy

3D fluorescence microscopy



original image | edge map



Ill-defined boundaries (due to imperfect staining, low SNR, low resolution etc.)
Region-based information is required here

Segmentation models
Region-based models

An old but powerful model: the Mumford-Shah functional ('89)

- General functional for image denoising and segmentation

$$E(u, K) = \underbrace{\alpha \int_\Omega |u - u_0|^2 \, d\omega}_{\text{data attachment}} + \underbrace{\cancel{\beta \int_{\Omega \setminus K} |\nabla u|^2 \, d\omega}}_{\text{smoothness of } u} + \underbrace{\gamma \, \mathcal{H}^{d-1}(K)}_{\text{smoothness of } K}$$

denoised signal (u)

discontinuity set (K)

- For segmentation, we consider that the solution is piecewise constant
- The $(n+1)$-region case (objects + background) simplifies to

$$E(c_{0..n}, K) = \alpha \sum_{i=0}^{n} \int_{\mathcal{R}_i} |c_i - u_0|^2 \, d\omega + \gamma \, \mathcal{H}^{d-1}(K)$$

- Finally, assuming K is a set of closed contours Γ_1 to Γ_n we get

$$E(c_{0..n}, \Gamma_{1..n}) = \alpha \sum_{i=0}^{n} \int_{\mathcal{R}_i} |c_i - u_0|^2 \, d\omega + \gamma \sum_{i=1}^{n} \mathcal{L}(\Gamma_i)$$

A parametric implementation: diffusion snakes ('00)

- «one-curve-two-phase» simplification

$$E(c_1, c_2, \mathcal{C}) = \frac{\alpha}{2}\int_{\mathcal{R}_1} |c_1 - I|^2 \, d\omega + \frac{\alpha}{2}\int_{\mathcal{R}_2} |c_2 - I|^2 \, d\omega + \gamma \int_0^1 \left|\frac{\partial \mathcal{C}}{\partial p}\right|^2 dp$$

$$\frac{\partial \mathcal{C}}{\partial t} = \alpha\Big[(c_2 - I)^2 - (c_1 - I)^2\Big]\vec{\mathcal{N}} + \gamma \frac{\partial^2 \mathcal{C}}{\partial p^2}$$

A level set implementation: active contours without edges ('01)

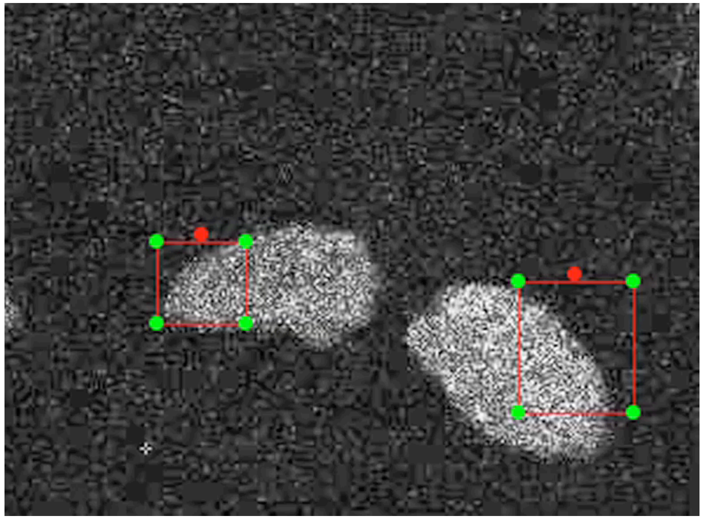
- «two-phase» simplification

$$\frac{\partial \varphi}{\partial t} = |\nabla \varphi| \left[\alpha\Big((1 - H(\varphi))(c_2 - I)^2 - H(\varphi)(c_1 - I)^2\Big) + \gamma\, \nabla \cdot \frac{\nabla \varphi}{|\nabla \varphi|}\right]$$

Segmentation models
A note on multi-object segmentation (and tracking)

What about multiple objects ?

- all models are inherently multi-object ready... or ?



video sequence
(2 time points)

well...
sure !

- the energy functional does not incorporate semantic information
- contours should be coupled to handle contacts (also true for segmentation...)

Segmentation models

Coupled deformable models

A parametric solution: coupled parametric active contours ('05)

- start from a fully multi-phase problem (image has *n +1* regions)...
- ... and add another energy term to penalize contour overlap

$$E(c_{0..n}, \mathcal{C}_{1..n}) = \underbrace{\gamma \sum_{i=1}^{n} \left[\int_0^1 \left| \frac{\partial \mathcal{C}_i}{\partial p} \right|^2 dp \right]}_{\text{regularization}} + \underbrace{\alpha \sum_{i=0}^{n} \int_{\mathcal{R}_i} |c_i - I|^2 \, d\omega}_{\text{data attachment}} + \underbrace{\sum_{j>i} \int_{\mathcal{R}_i \bigcap \mathcal{R}_j} d\omega}_{\text{overlap penalty}}$$

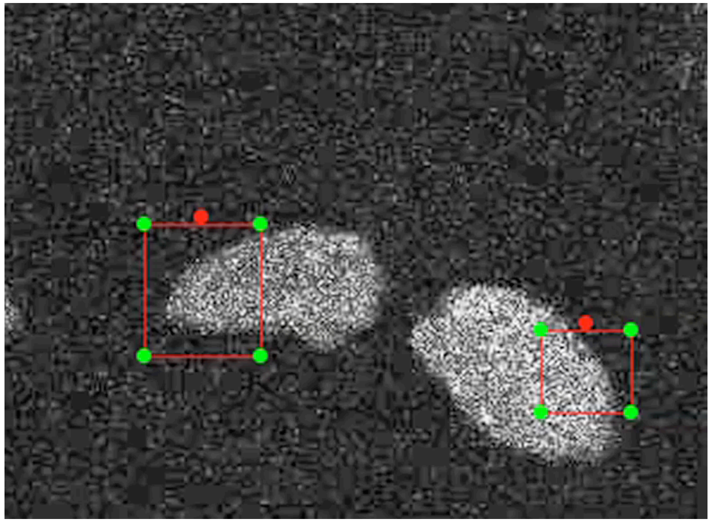
$$\frac{\partial \mathcal{C}_i}{\partial t} = \gamma \frac{\partial^2 \mathcal{C}_i}{\partial p^2} - \alpha \left[(c_i - I)^2 - (c_0 - I)^2 + \overline{\eta \sum_{j>i} \mathbf{in}_j(\mathcal{C}_i)} \right] \vec{\mathcal{N}}$$

A level set solution: coupled active contours / active surfaces ('04)

$$\frac{\partial \varphi_i}{\partial t} = |\nabla \varphi_i| \left[\gamma \nabla \cdot \frac{\nabla \varphi}{|\nabla \varphi|} + \alpha \left(\prod_{j=1}^{n} \left((1 - H(\varphi_j))(c_0 - I)^2 \right) - H(\varphi_i)(c_i - I)^2 \right) - \eta \sum_{j>i} H(\varphi_i) H(\varphi_j) \right]$$

What about multiple objects ?

- the energy is now «aware» of the objects

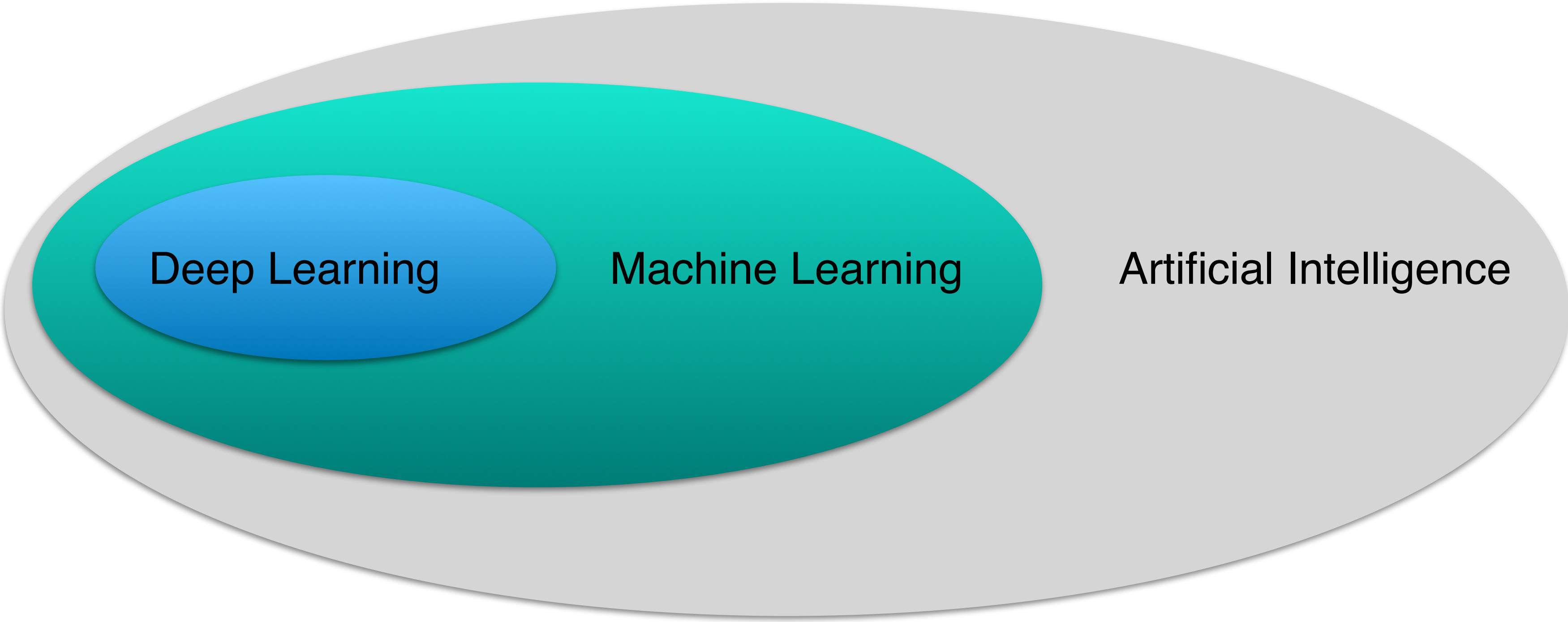


video sequence
(2 time points)

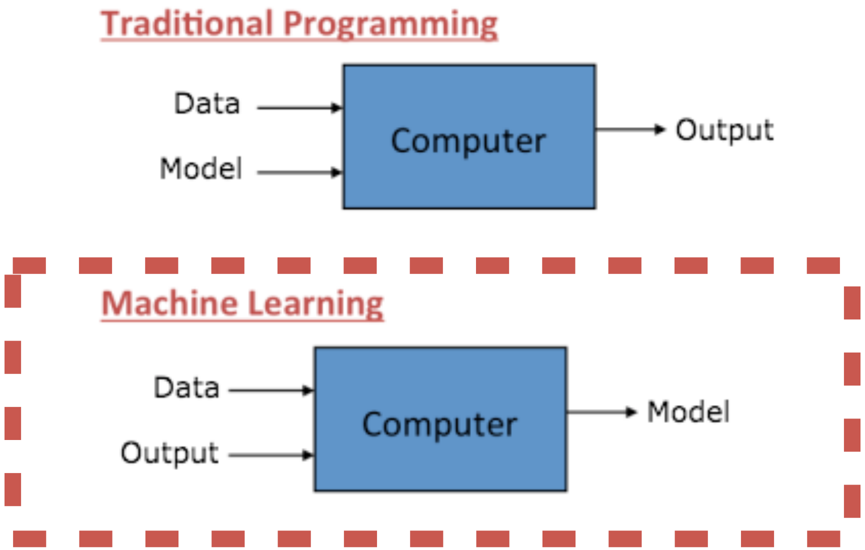
Machine Learning
Deep Learning

Deep learning: a family of computational techniques using (deep) artificial neural networks to learn complex input-output relations between numerical data

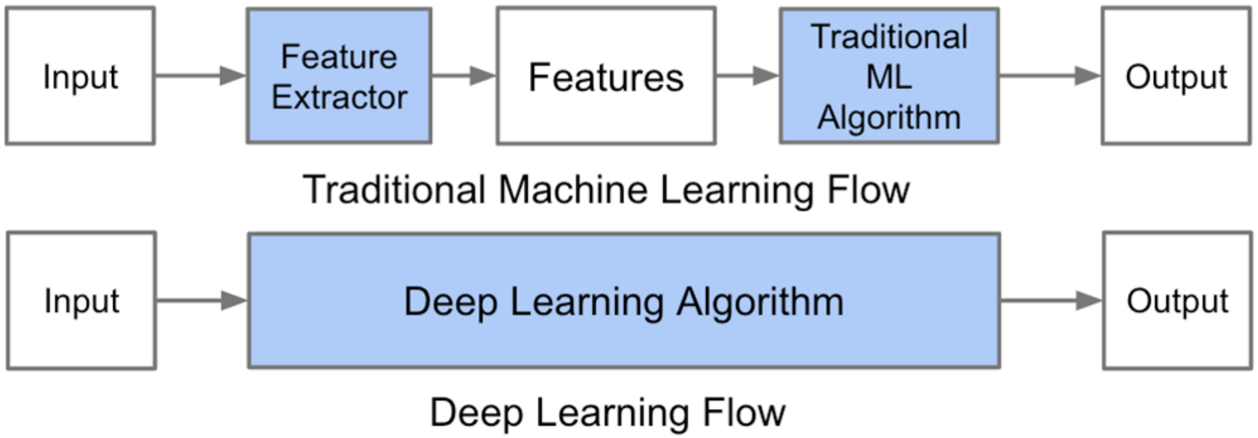
Lecun, Bengio and Hinton, Nature 2015
Games Alpha Go: Silver et al. Nature 2016



Concept



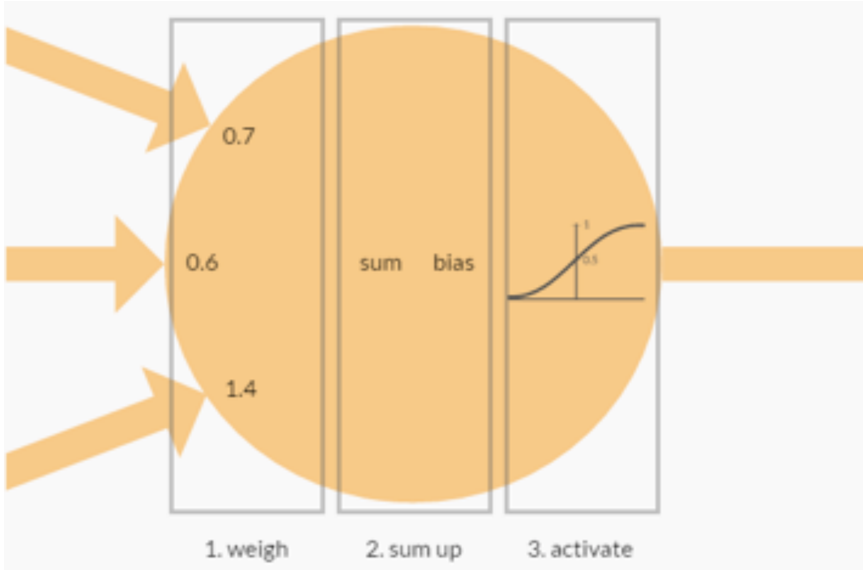
- inspired by human reasoning
- object detection, handwriting recognition, medical diagnostic
- decrease in human supervision from instruction-driven programming to DL
- fueled by GPU and Big Data



Neural Network

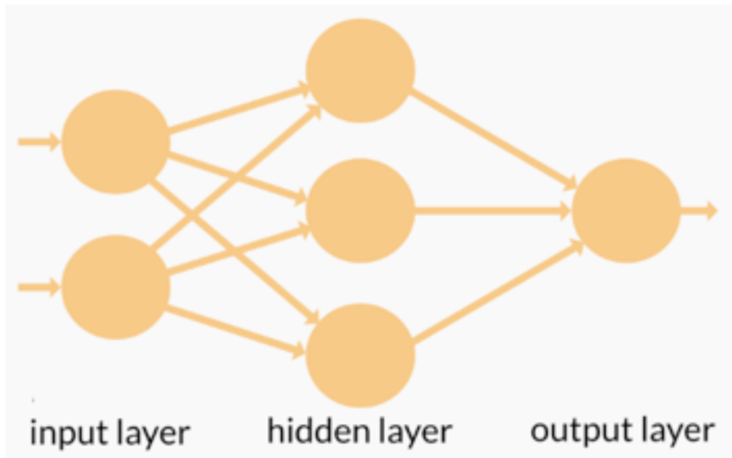
Neuron

- input ≡ *dendrites* x_i
- processing unit ≡ *cell body*: weight, sum up, activate
- output ≡ *axon* $\hat{y}$



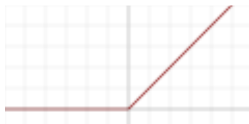
Forward propagation:

$$\hat{y} = f\left(\sum_i w_i x_i + b\right)$$



Activation functions:

Heaviside (step) Sigmoid (soft step) ReLU

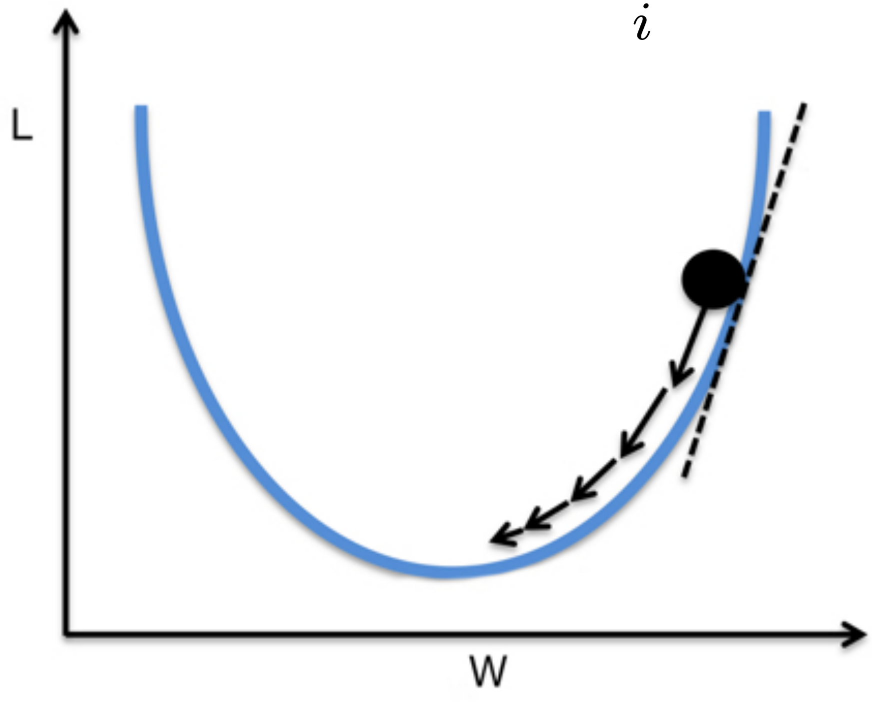


$$f(x) = \frac{1}{1 + e^{-x}} \qquad f(x) = max(0, x)$$

Learning Mechanism

- intelligence ≡ weights
- minimize error between ground truth and prediction
- **loss function**: MSE, cross entropy

$$\mathcal{L}(y, \hat{y}) = -\sum_i y_i - log\hat{y}_i$$



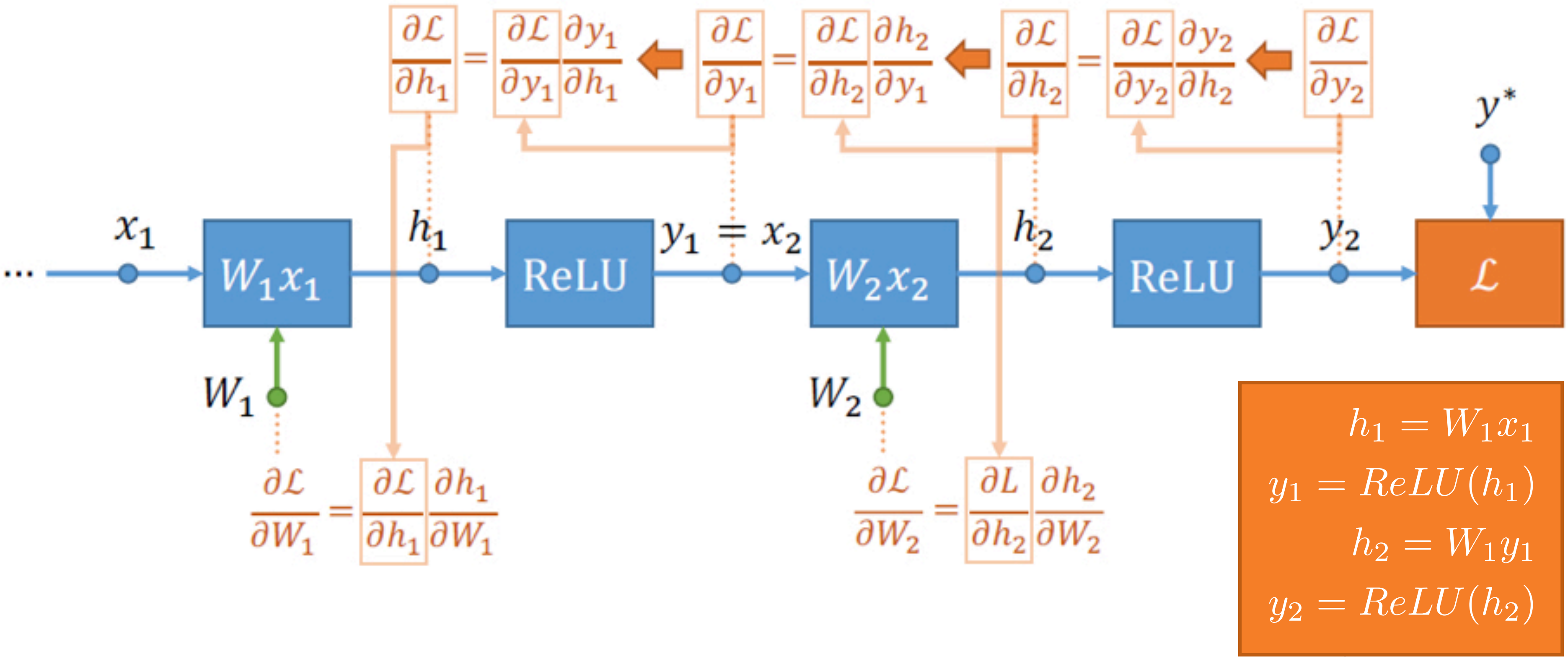
- **gradient descent**: move in negative of the gradient to find the optimal weight that minimizes the loss function learning rate

$$W \leftarrow W - \eta \frac{\partial \mathcal{L}}{\partial W}$$

Deep Learning

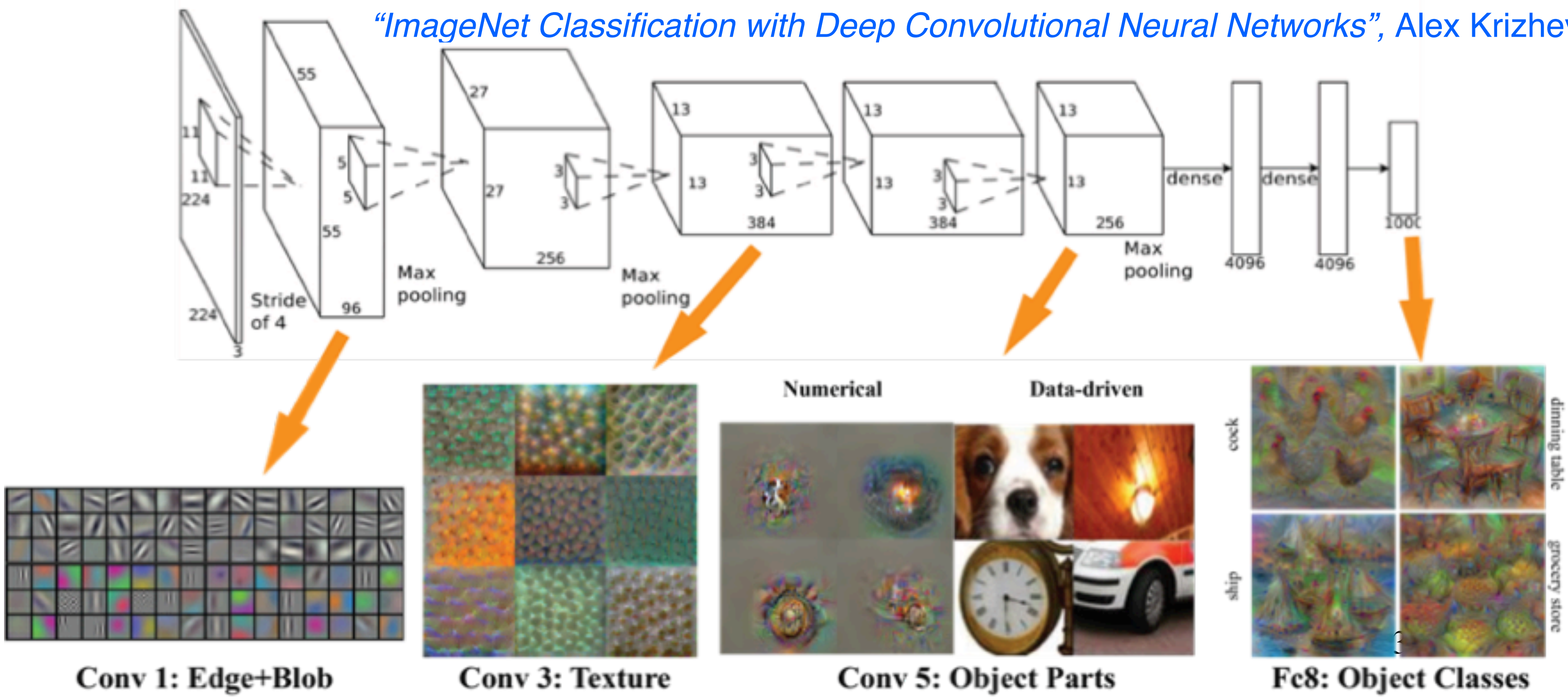
Learning Mechanism

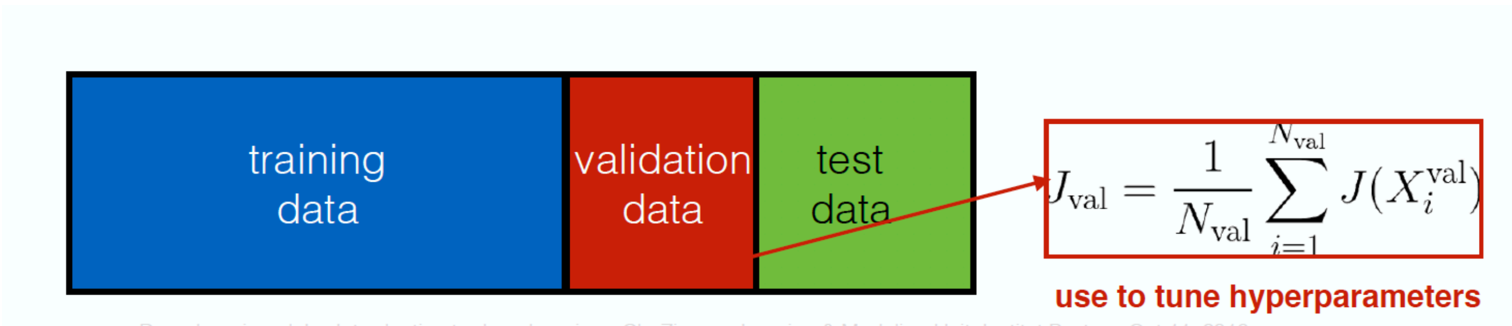
- **back-propagation:** distribute error to all neurons in a path proportionally to their contribution to the error
- chain rule: derivative of composition of functions $\frac{\partial c}{\partial a} = \frac{\partial c}{\partial b}\frac{\partial b}{\partial a}$



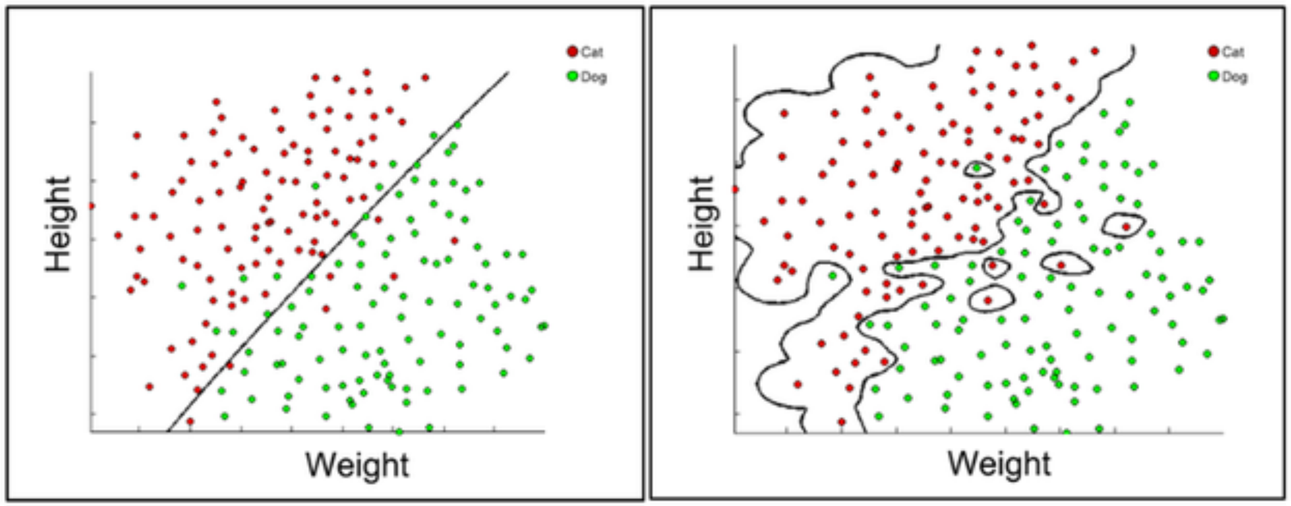
Convolutional Neural Networks

- NN for computer vision
- visual cortex: neurons sensitive to specific regions of the field of view
- look for specific patterns in different sections of the image
- hierarchical feature extractors



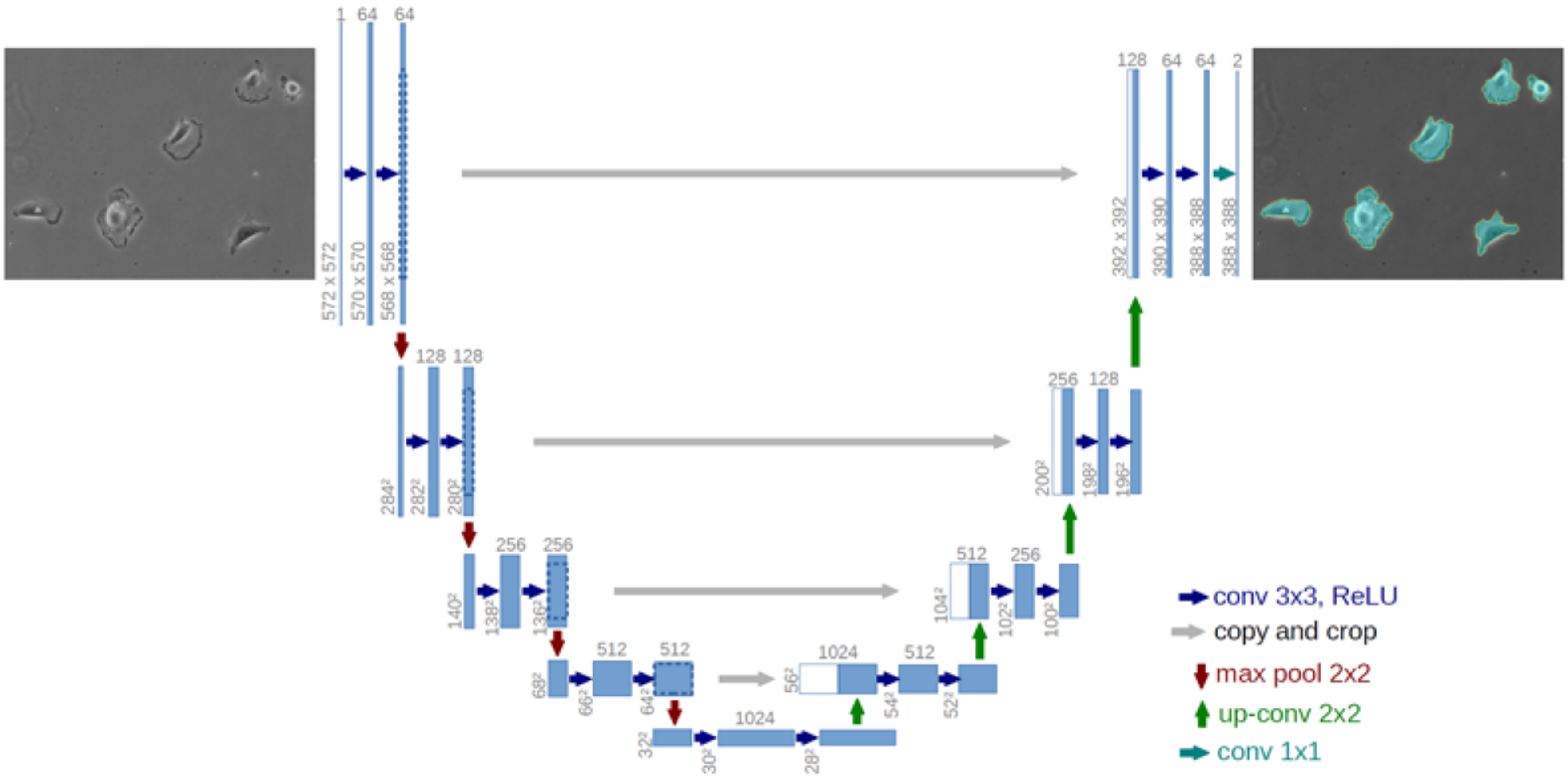


Be careful of overfitting !



Deep Learning

U- Net

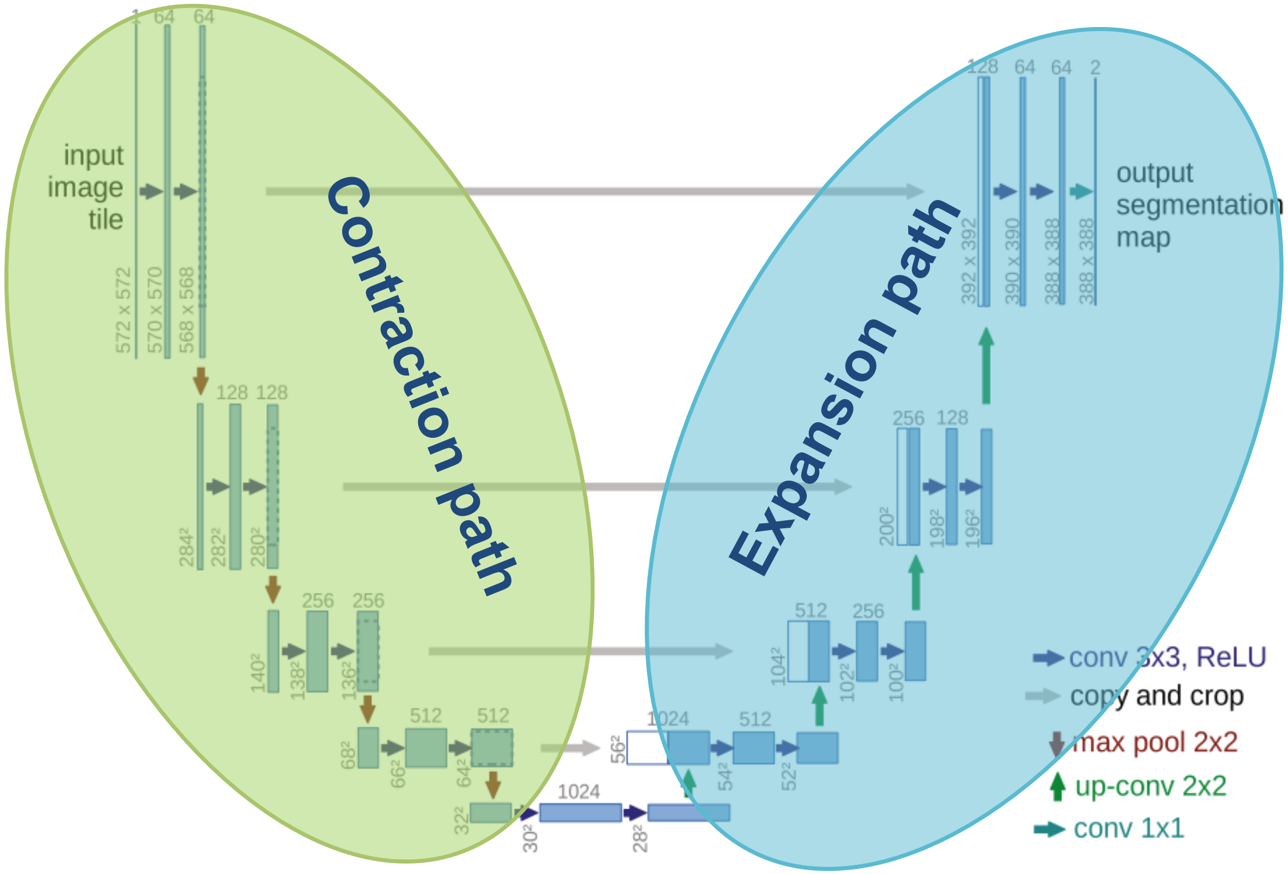


(Ronneberger et al, 2015)

Network architecture

↓ **Where**

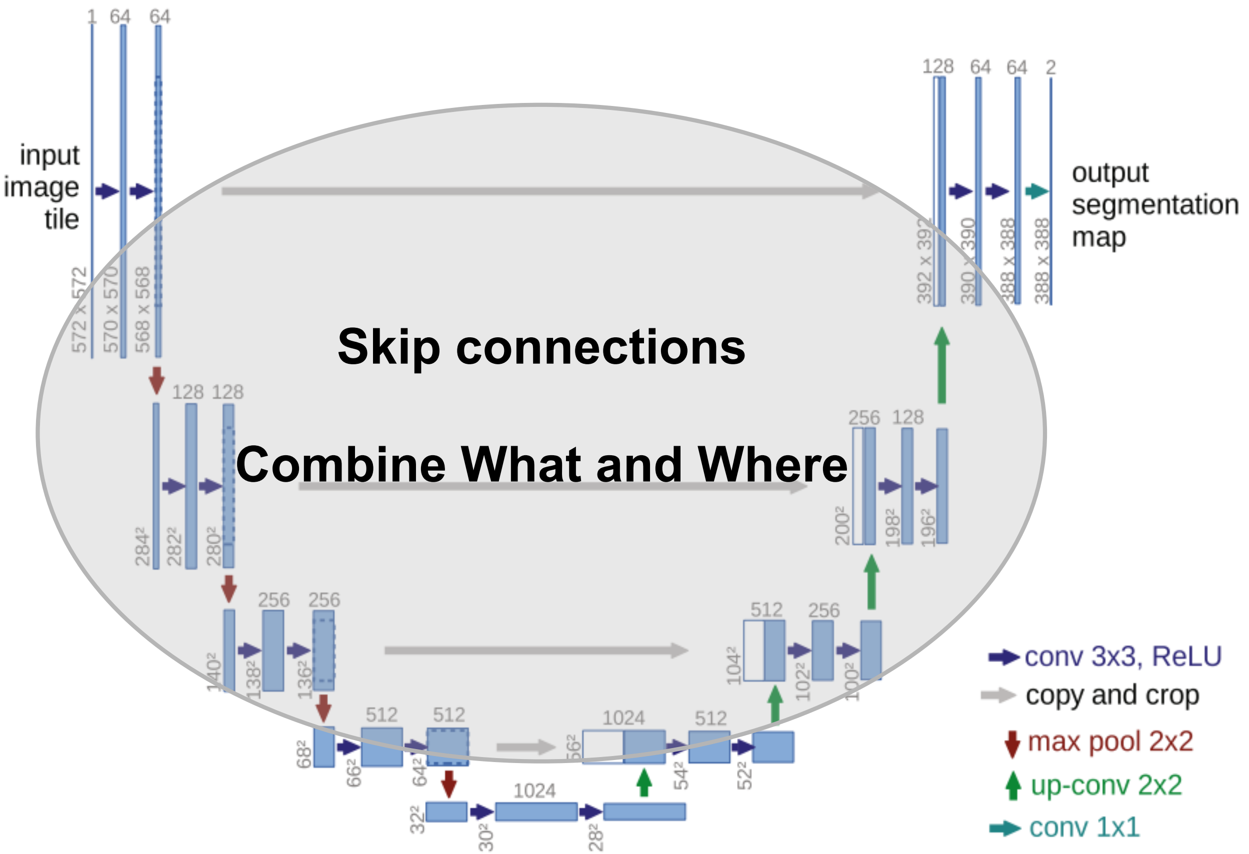
↑ **What**



↑ **Where**

↓ **What**

Network architecture

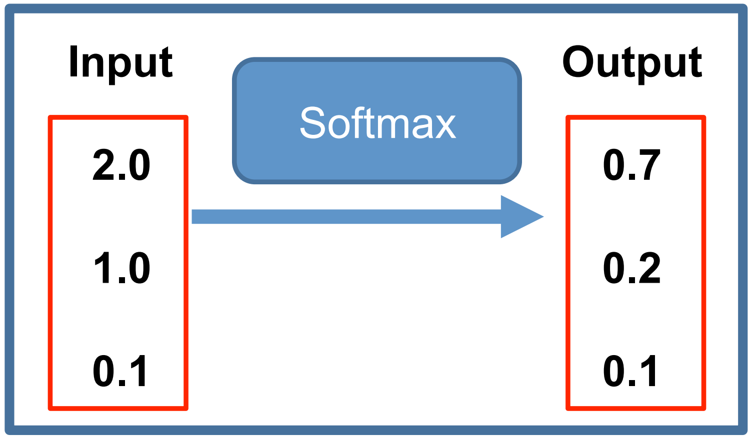


Training

Energy function = pixel-wise soft-max + cross entropy loss function

Soft-max (≈ probability distribution)

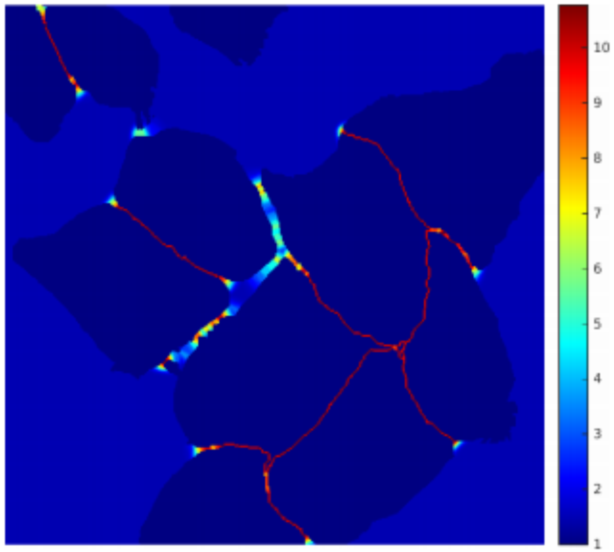
$$\sigma(x_j) = \frac{e^{x_j}}{\sum_i e^{x_i}}$$



Cross entropy loss function (≈ log error)

$$E = \sum_{\mathbf{x} \in \Omega} w(\mathbf{x}) \log(p_{\ell(\mathbf{x})}(\mathbf{x}))$$

Dissimilarity output / target



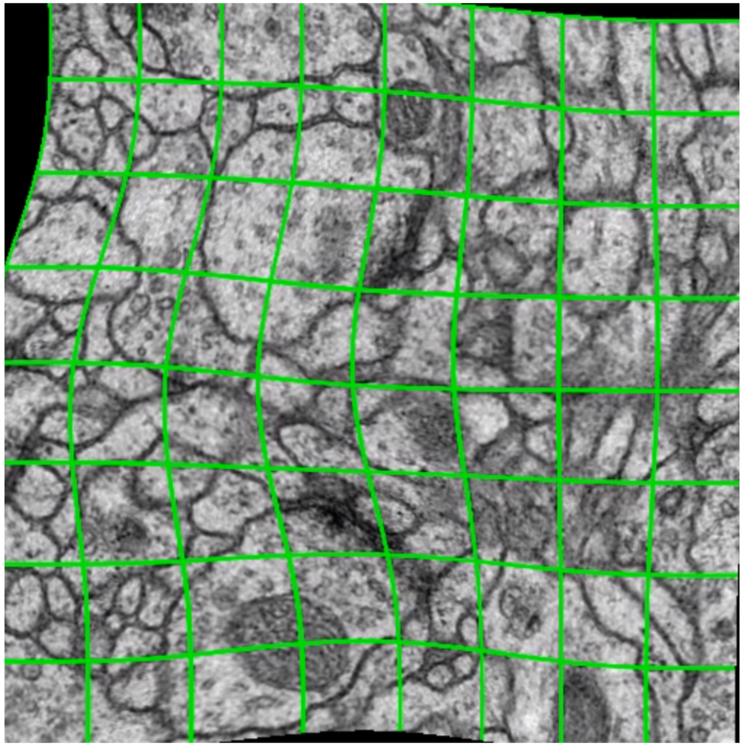
Data augmentation

Application of geometric transformations on images

↑ training dataset size

No need of new labelling

↑ robustness and invariance
- Shift
- Rotation



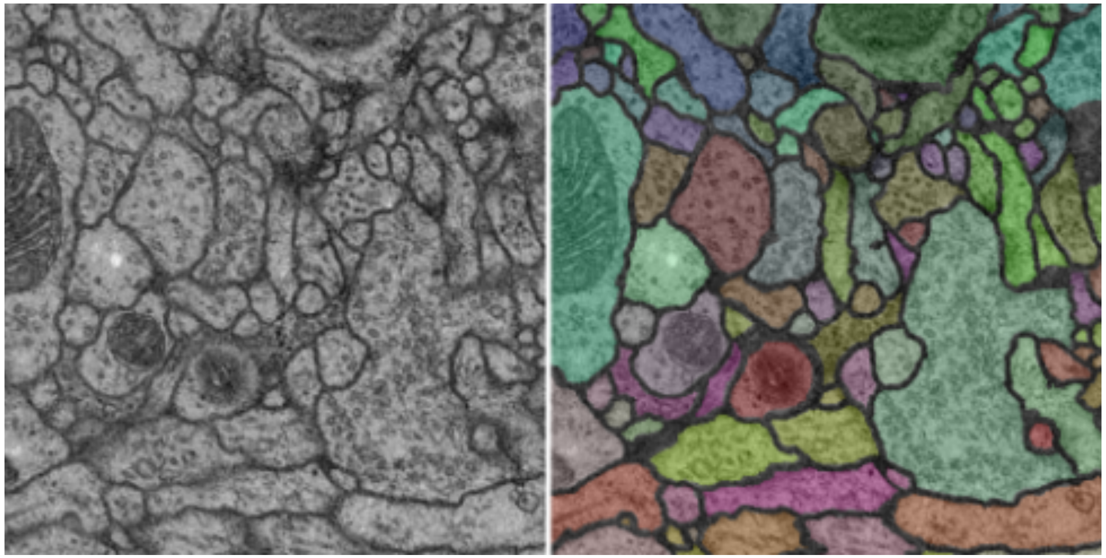
Deep Learning

ISBI EM challenge 2012

Training data:

30 images (512x512 pixels)

from serial section transmission EM of the Drosophila first instar larva ventral nerve cord (VNC).



Results:

warping error of 0.0003529 *(versus 0.000420*)*

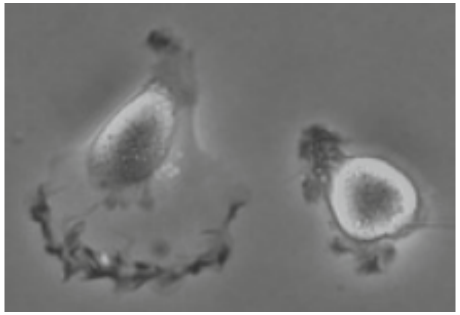
rand-error of 0.0382 *(versus 0.0504*)*

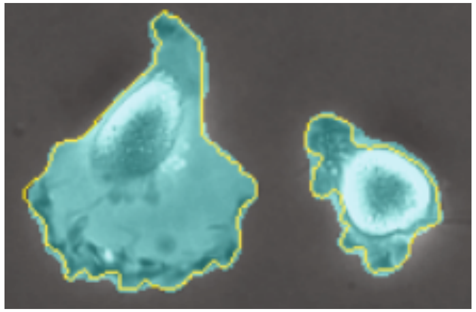
**Ciresan et al. network who won the 2012 challenge.*

Deep Learning

"PhC-U373" containing Glioblastoma-astrocytoma U373 cells on a poly-acrylamide substrate recorded by phase contrast microscopy.

(ISBI cell tracking challenge 2015)





Training data:

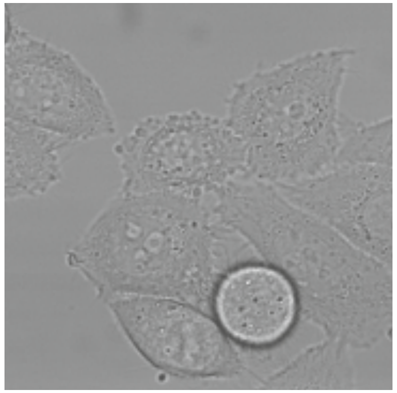
35 partially annotated training images.

Results:

Average IOU ("Intersection Over Union") of 92% *(versus 83%*).*

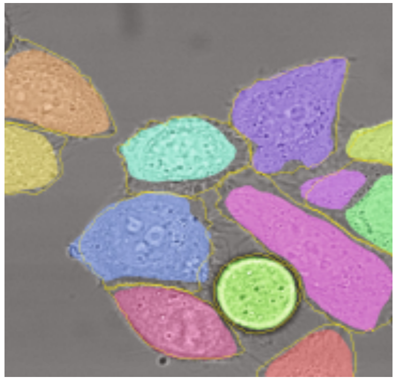
** Second best team in 2015*

Deep Learning



"DIC-HeLa" HeLa cells on a flat glass recorded by differential interference contrast (DIC) microscopy.
(ISBI cell tracking challenge 2015)

Training data
20 partially annotated training images.



Results
Average IOU ("Intersection Over Union") of 77.5% *(versus 46%*).*

The U-Net CNN outperformed previous state-of-the-art methods in 3 segmentation tasks in 2015
Downsampling, Upsampling and Skip paths allow to combine precise localization and use of context.
No pre or post-processing is required

Since 2015, other CNN have improved these results

Summary

Object well discriminated

Global methods: histograms

Localised Objects

Local methods: region growing

General case

Mixed methods

Split & merge

Geometrical
constrained by the
decomposition

Optimisation

Deformable models

Initialisation
Curve evolution

Markov Random Fields

Strategy of the path
And update

…..

Evaluation ?

Segmentation is **crucial** to image analysis and image understanding

A generic method adapted to any problem does not exist … yet

Deep learning ?