Machine Learning for Image Analysis

Jean-Yves Ramel

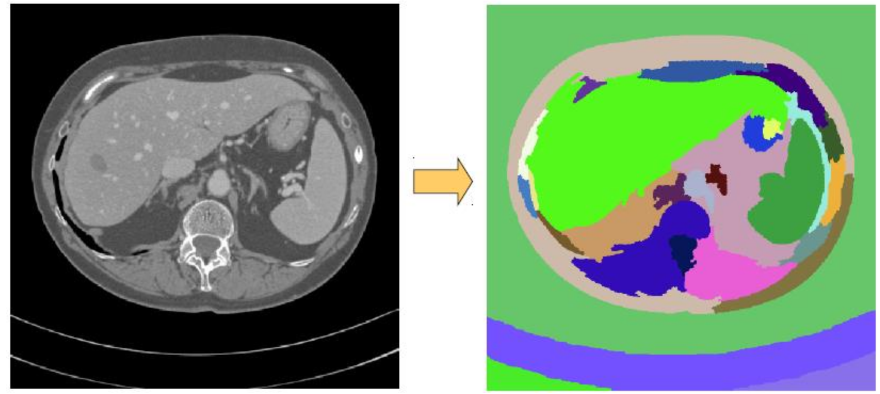
ramel@univ-tours.fr

LABORATOIRE D'INFORMATIQUE FONDAMENTALE ET APPLIQUÉE DE TOURS

Laboratoire d'Informatique Fondamentale
et Appliquée de Tours

What is segmentation ?

- Image decomposition into regions corresponding to meaningful objects
  - 1 object = 1 label ➔ Pixel tagging
  - 1 object = Homogeneity criteria ?
  - Same color = same region



1 region = 1 label = 1 color
(for human visualisation)

- Easy for human
  - Use of a priori knowledge
  - Looking to the whole image for a more global interpretation
- 3 main categories of segmentation methods
  - Region based approach : associate similar pixels to obtain homogenous regions
  - Contour based approach: search for adjacent dissimilar pixels to obtain the frontiers between homogenous areas
  - Today, mainly Machine Learning based approaches

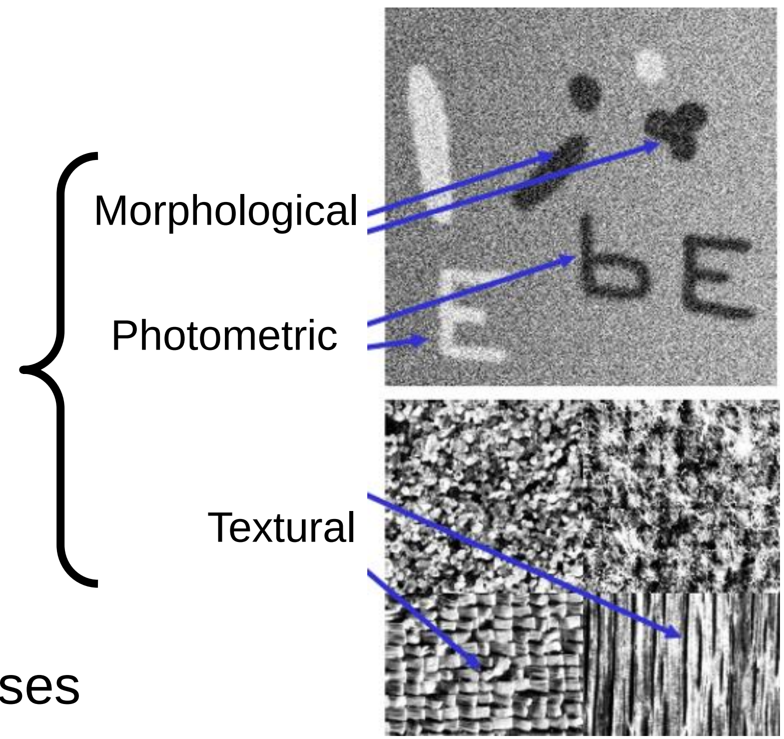
Machine learning based approaches

Object description using numerical features

Initial data (pixels, regions, …)

Feature selection

Classification



Object classification

➔Grouping similar objects into clusters or classes

- Unsupervised Classification

  We just have the objects described by features

  → Grouping similar objects into homogenous clusters

Supervised Classification ➔ **Machine Learning**

A training/learning set is available to define, learn **a classifier** (class recognizer)

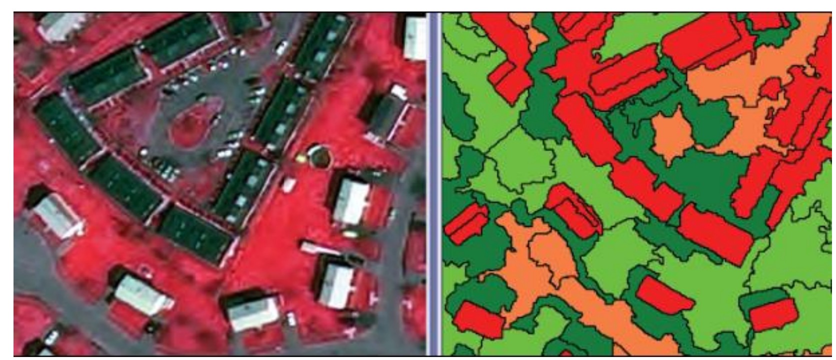
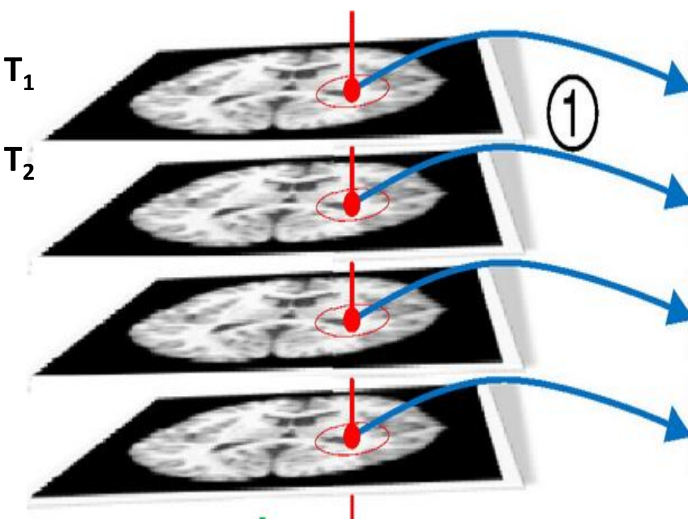


Image Analysis with Machine learning

Principle 1

- Images are no more seen as image, **BUT**
- Pixels or ROIs become objects → Each object has to be described by features
- 1 object ⇨ p features ⇨ 1 Vector = 1 Point $\in R^p$ ➔ n objects = 1 table
- Statistics, Data analysis and Machine Learning tools become usable

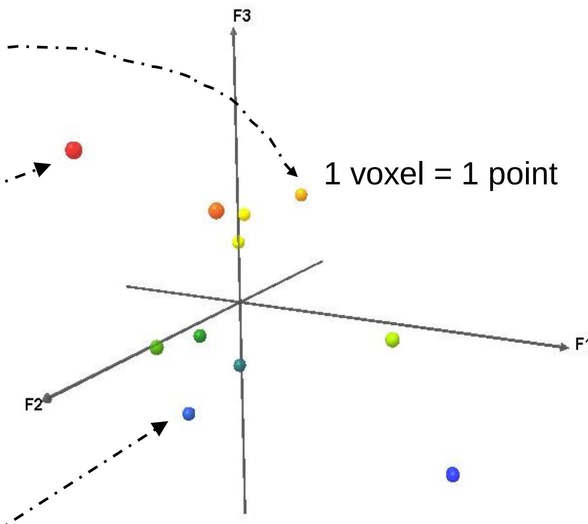
1 pixel / Voxel with several modalities



F_1
F_2
F_3
…

$Voxel_1 = [F_1\ F_2\ F_3\ …]$
$Voxel_2 = [F_1\ F_2\ F_3\ …]$
$Voxel_3 = [F_1\ F_2\ F_3\ …]$
$Voxel_4 = [F_1\ F_2\ F_3\ …]$
$Voxel_5 = [F_1\ F_2\ F_3\ …]$
…..
$Voxel_N = [F_1\ F_2\ F_3\ …]$

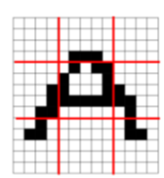
Table of Voxels



1 table = 1 cloud of points

- An other examples (OCR)

V=(0,3,0,4,12,4,3,0,3)



V=(6,10,0,12,4,0,10,10,0)

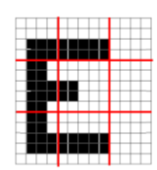


Image Analysis with Machine learning

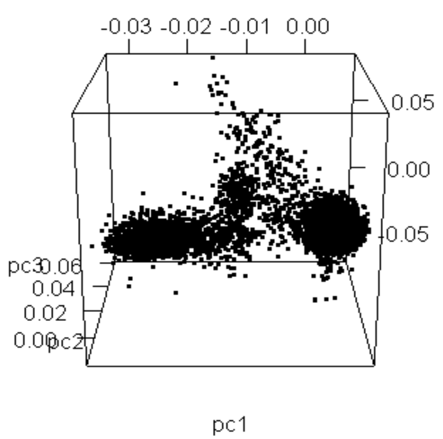
Principle 1

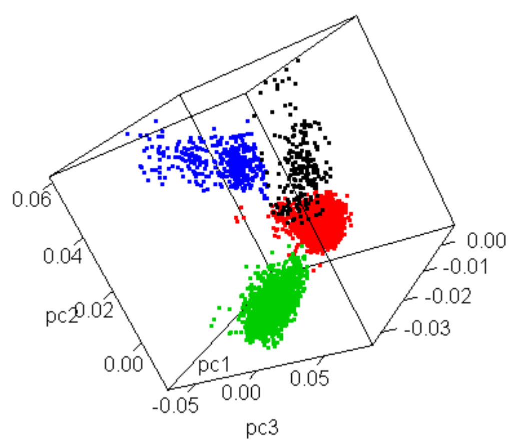
- Images are no more seen as image, **BUT**
- Pixels or ROIs become objects → Each object has to be described by features
- 1 object ⇨ p features ⇨ 1 Vector = 1 Point $\in R^p$ ➔ n objects = 1 table
- Statistics, Data analysis and Machine Learning tools become usable

$Voxel_1 = [F_1\ F_2\ F_3\ \ldots]$
$Voxel_2 = [F_1\ F_2\ F_3\ \ldots]$
$Voxel_3 = [F_1\ F_2\ F_3\ \ldots]$
$Voxel_4 = [F_1\ F_2\ F_3\ \ldots]$
$Voxel_5 = [F_1\ F_2\ F_3\ \ldots]$
.....
$G= [F_1\ F_2\ F_3\ \ldots]$

$Voxel_1 = [F_1\ F_2\ F_3\ \ldots]$ [Tumor]
$Voxel_2 = [F_1\ F_2\ F_3\ \ldots]$ [Normal]
$Voxel_3 = [F_1\ F_2\ F_3\ \ldots]$ [Tumor]
$Voxel_4 = [F_1\ F_2\ F_3\ \ldots]$ [Tumor]
$Voxel_5 = [F_1\ F_2\ F_3\ \ldots]$ [Normal]
$G_G= [F_1\ F_2\ F_3\ \ldots]$ ↑
$G_R= [F_1\ F_2\ F_3\ \ldots]$ Label

$Voxel_1 = [F_1]$
$Voxel_2 = [F_1]$
$Voxel_3 = [F_1]$
$Voxel_4 = [F_1]$
$Voxel_5 = [F_1]$
.....
$G= [F_1]$





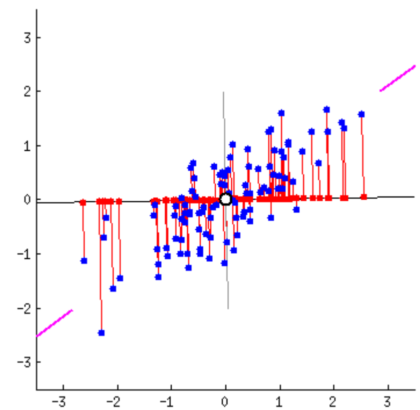
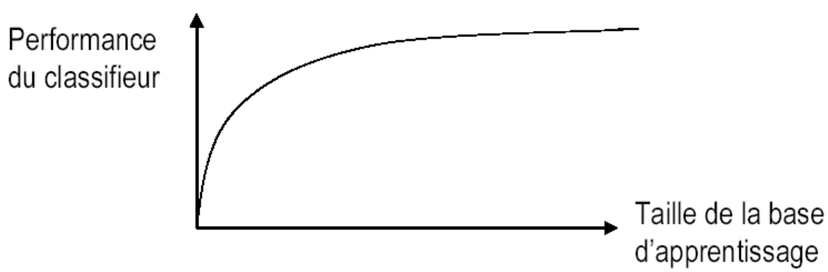


Image Analysis with Machine learning

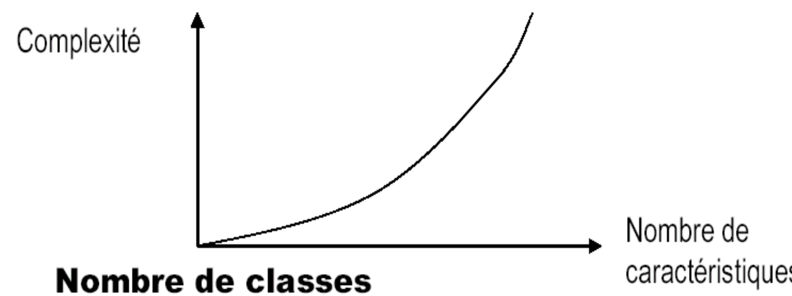
Principle 2: Data (labeled) becomes the Graal ➔ Big data / data science

- 1 object ⇨ p features ⇨ 1 Vector = 1 Point $\in R^p$
- **Dataset** = Table of Vectors with or without a label (class)
- **Selected Features are crucial** (expertise, …)
  - Stable and discriminative
  - And…
- If a label is available for each object, we speak of Learning / Training sets

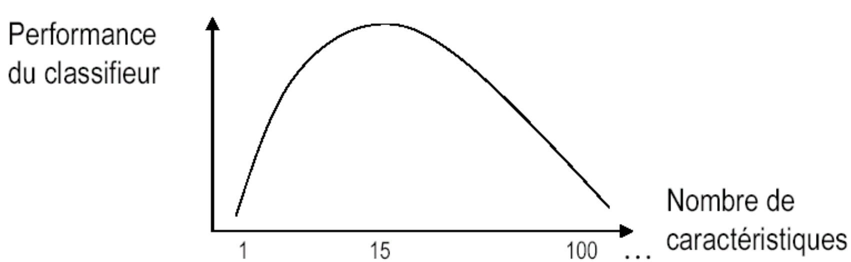
Statistique suffisante



Complexité vs nombre de caractéristiques



Malédiction de la dimensionalité



Nombre de classes

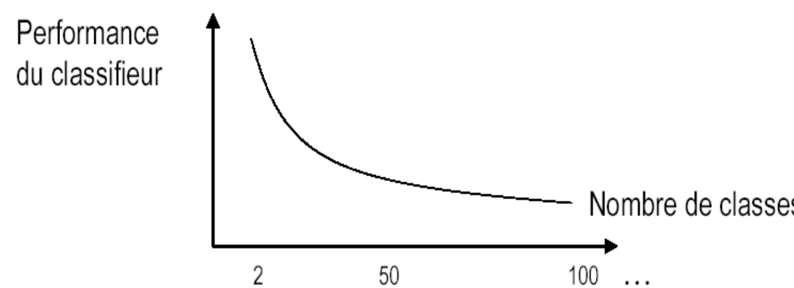


Image Analysis with Machine learning

How computers can recognize objects?

- We need a **large set of (labelled) examples** similar to the patterns to be recognized ➔ a training set
- We need a list of **stable and discriminative** features (shape, color, size,…) used to describe the patterns (labelled ones and unknown one)

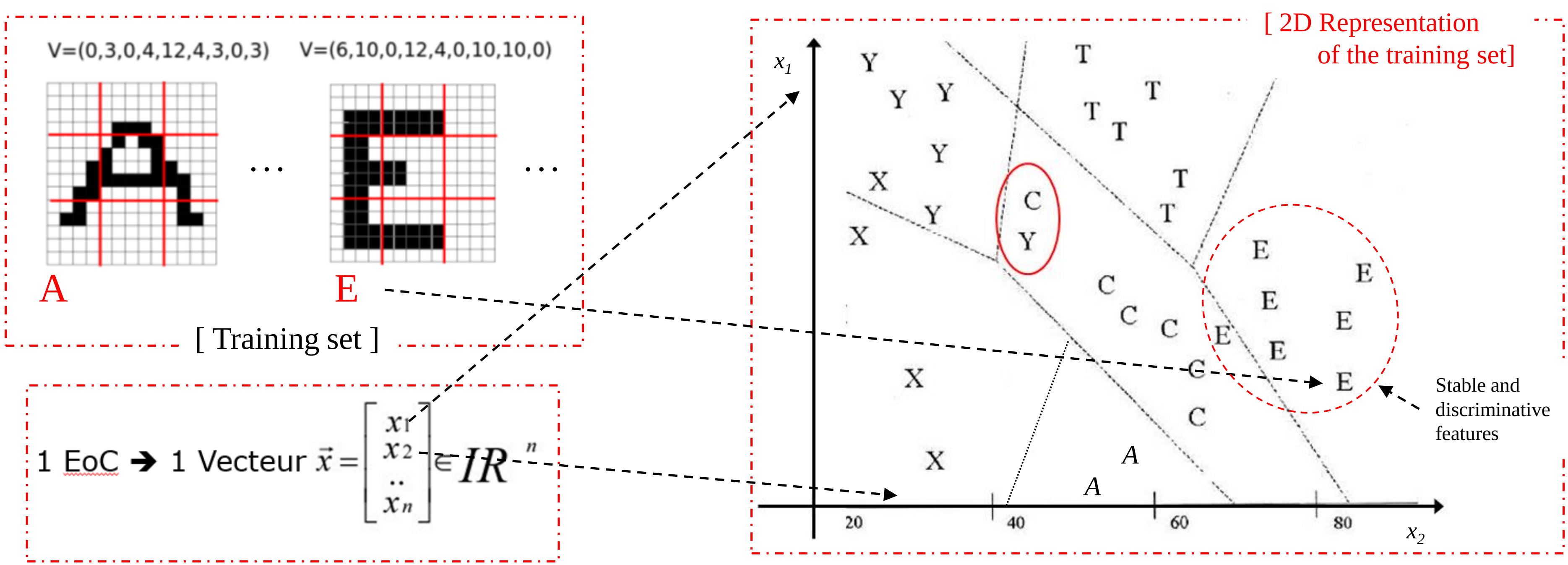
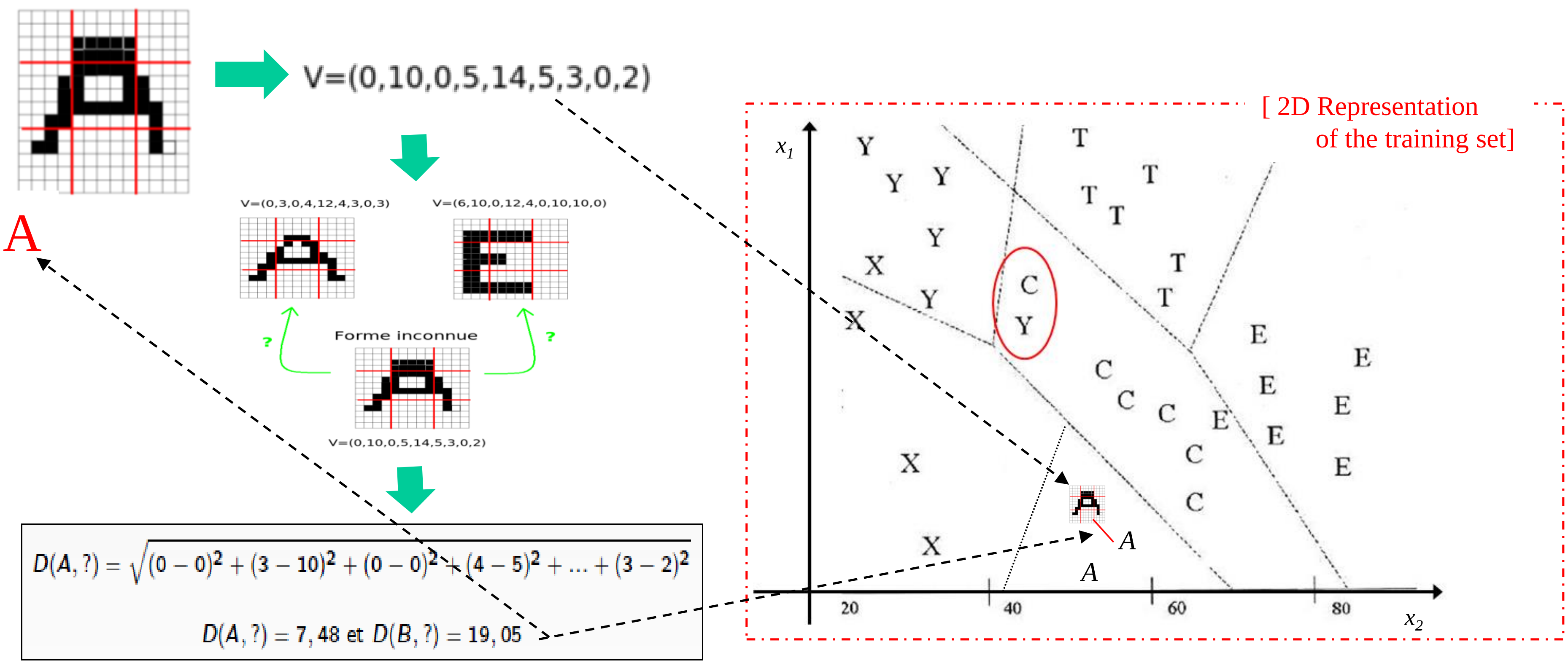


Image Analysis with Machine learning

How computers can recognize objects?

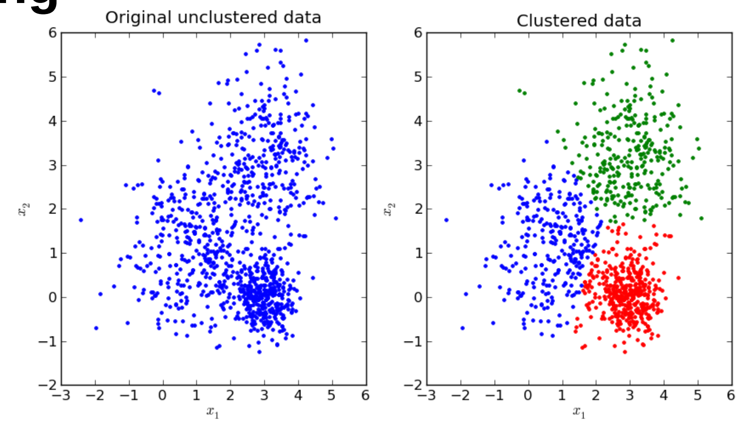
- We need a powerful **model of classifier**
- We need a powerful **similarity measure** (dissimilarity, distance, metric) to compare objects together

Unknown object



Machine learning based approaches

Unsupervised classification : k-means clustering



- No tagged data available ⇨ learning impossible
- We look for k classes starting from k centers (G_i)

Objectif : minimising the intra-class variance

Algorithm:

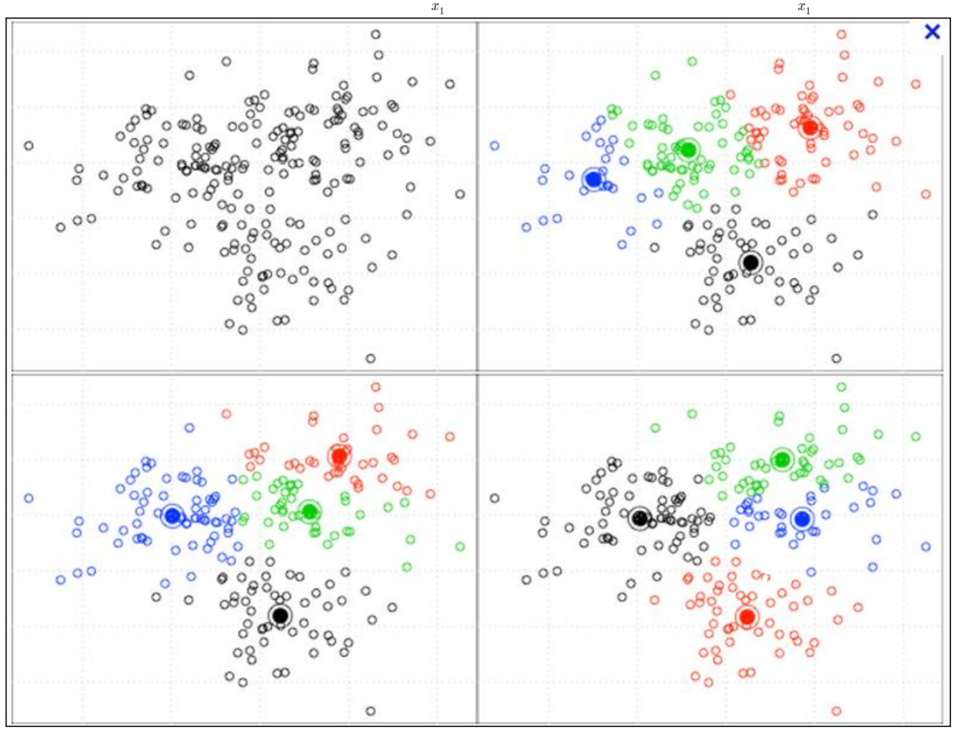
1 - Choose K centers randomly

2 – Repeat :

a/ Allocate each x to the closest center Gi

b/ Compute the new Gi until stabilization

Dependent from initialization →



Machine learning based approaches

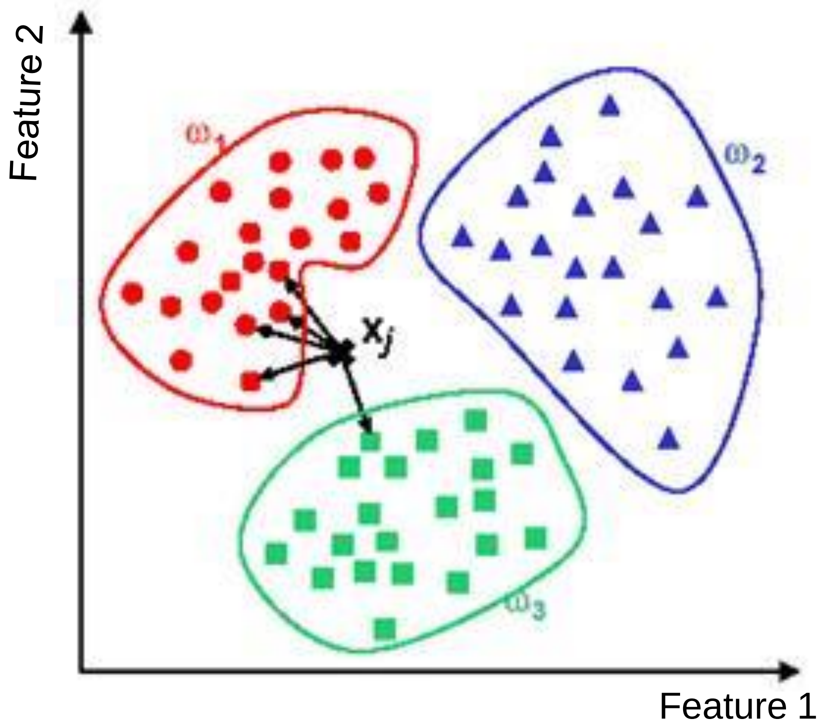
Clustering on images

Group together pixels by color, automatic segmentation: k-means, k = 2

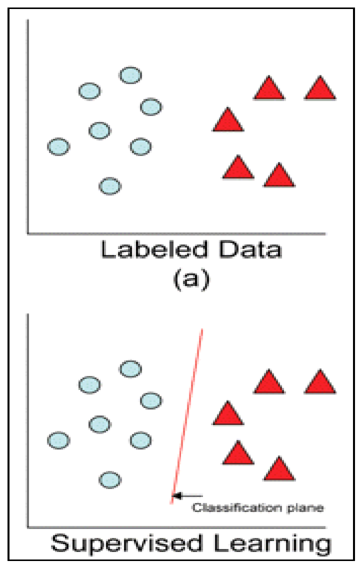
Machine learning based approaches

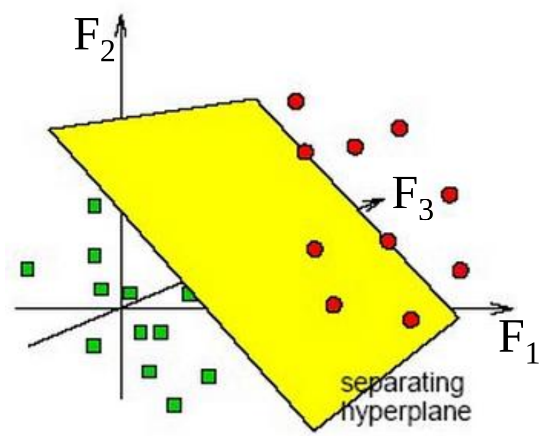
Supervised classification : k-Nearest-Neighbors (kNN)

- We have a training set with feature vectors tagged with the corresponding classes (w_i)
- The unknown vector X_j is classified with/inside the most represented class among its k nearest neighbors



Or other sophisticate techniques for classifier definition = ML





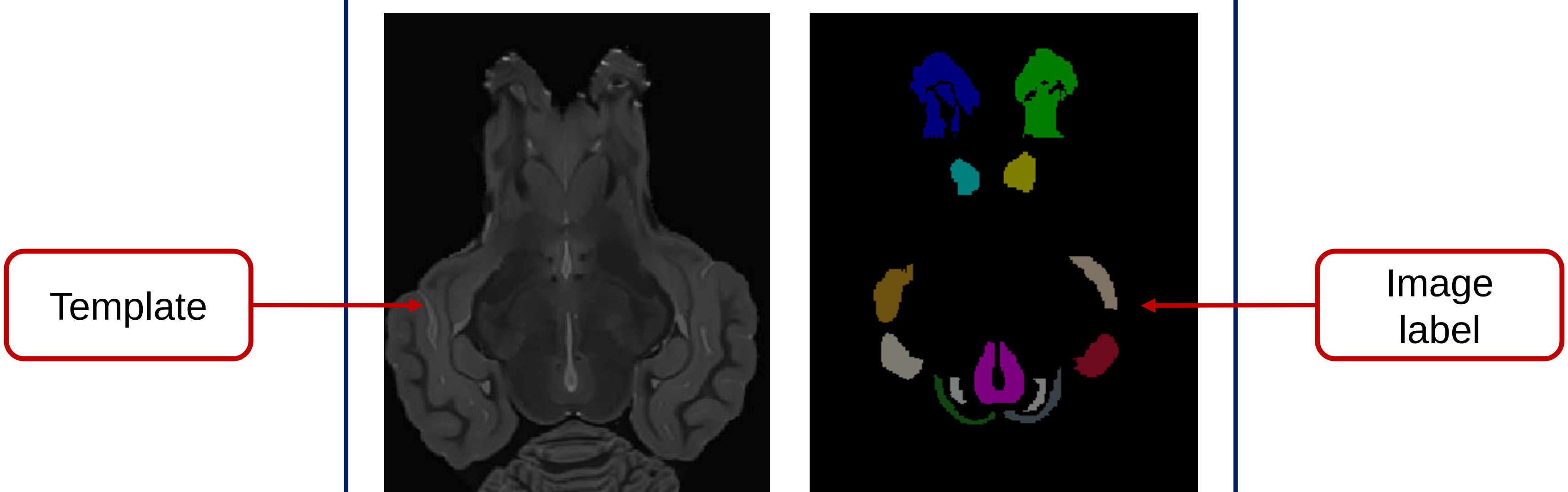
Probabilistic Approaches (Atlas)

Volumic Atlas

Spatial information on the position and intensity of anatomical structures

Different types of atlases :

- Simple



Atlas topologique simple de cerveau de brebis

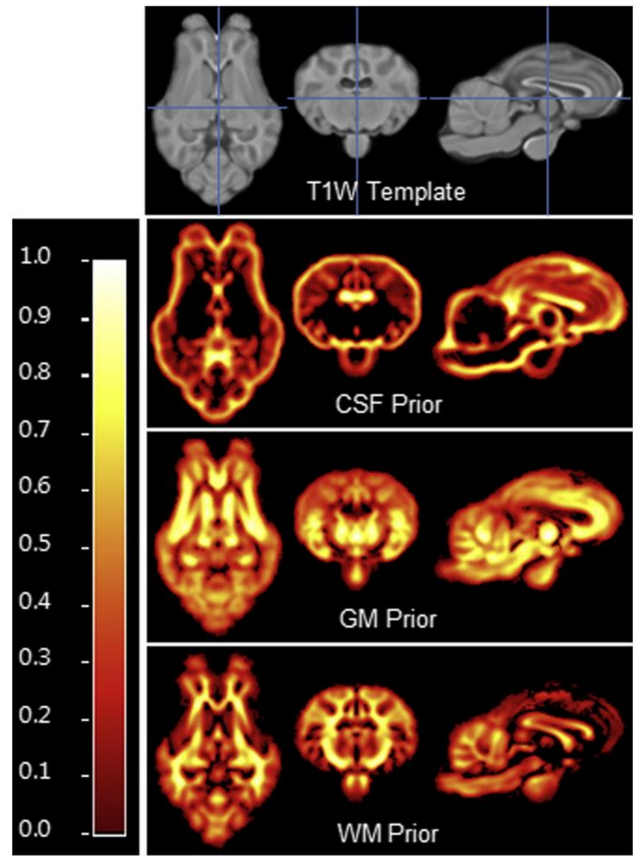
Probabilistic Approaches (Atlas)

Volumic Atlas

Spatial information on the position and intensity of anatomical structures

Different types of atlases :

- Simple
- Probabilistic



Atlas probabiliste des tissus du cerveau de brebis [Ella *et al.*, 2015]

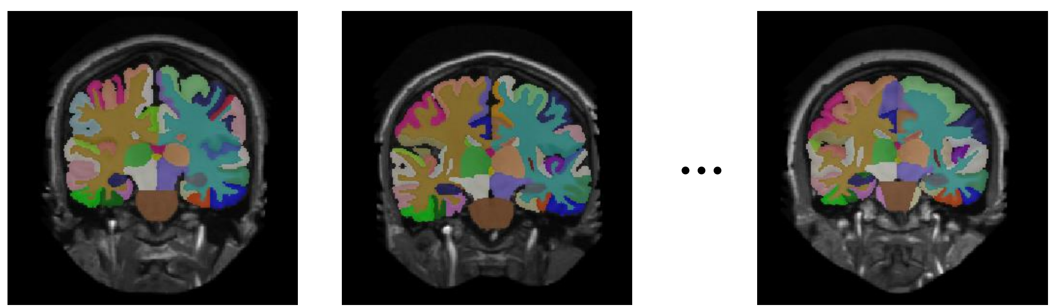
Probabilistic Approaches (Atlas)

Volumic Atlas

Spatial information on the position and intensity of anatomical structures

Different types of atlases :

- Simple
- Probabilistic
- Multi-Atlas



Multi-Atlas of Human Brain used during MICCAI'12 contest

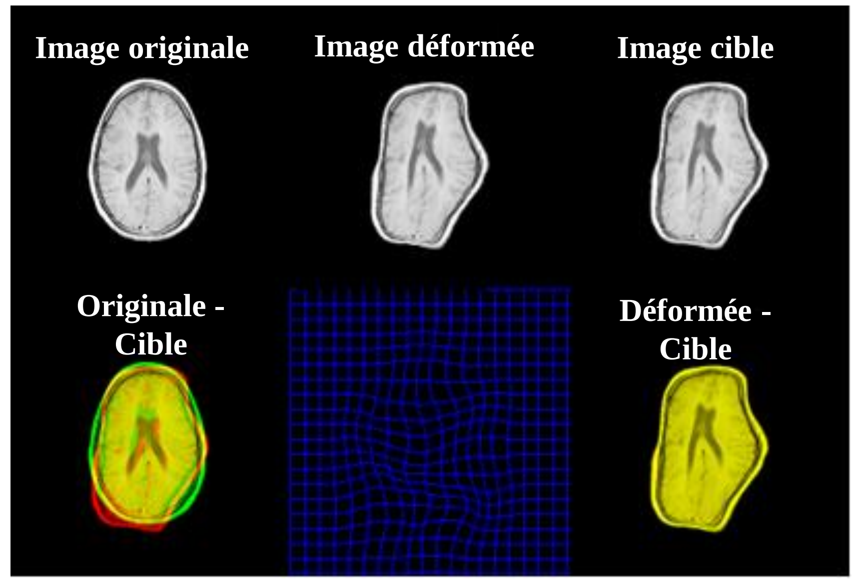
Probabilistic Approaches (Atlas)

Using Atlases to segment images

Registration methods are needed when using atlases

- Transformations
  - Rigid
  - Refined
  - Non-linear (TPS, B-spline, etc ...)

- Metrics
  - Sum of squared differences
  - Mutual information
  - Cross-correlation



Registration of 2 brain Images [3]

Label propagation

- Merging information and decision
- Voting method, globally weighted [Artaechevarria et al., 2008], locally weighted [Isgum et al., 2009]
- Joint label fusion [Wang et al., 2013], Generative model [Sabuncu et al., 2010], etc.
- Mix of Gaussian Markov Models [Bricq et al., 2006], Markov Random Field [Scherrer et al., 2009], etc.

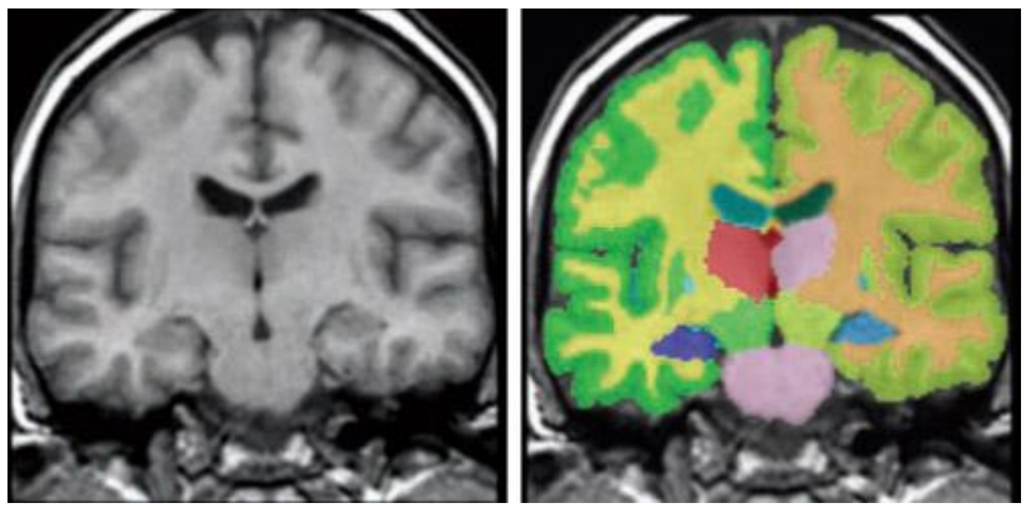
Probabilistic Approaches (Atlas)

Synthesis

- Principle: Pixels are random variables representing the probability of belonging to each class
- Then, estimation of the maximum likelihood (MAP, EM, ...)
  - Markov chain: 2D represented in 1D
  - Markov Fields: Only neighbors count

Advantages disadvantages

- Exploitation of a priori knowledge (atlas)
- Learning phase
- Slow



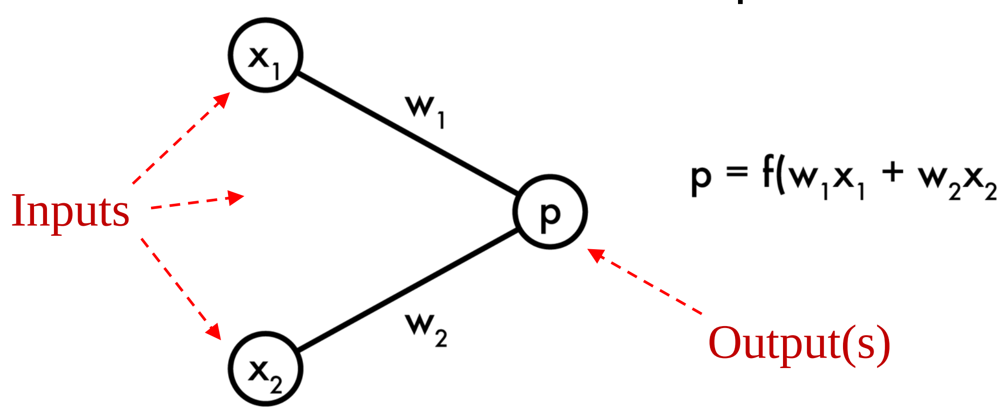
Overcoming the limitation of ML…

What is feature engineering?

- Arguably the **core** problem of machine learning (especially in practice)
- ML work well if there is a clear relationship between the inputs (features representing objects: x_i) and the outputs (p: the function you are trying to model / learn)

Classifier = Linear model

- Compute a probability for each possible class according the selected features
- Linear model means output = linear combination of the inputs
- Learning the weights

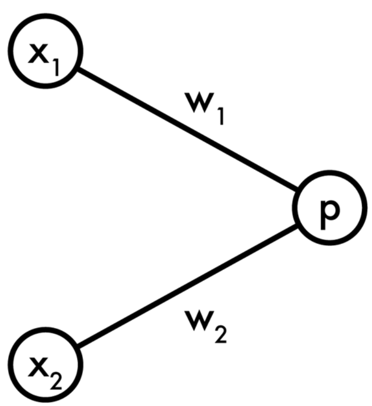


Overcoming the limitation of ML…

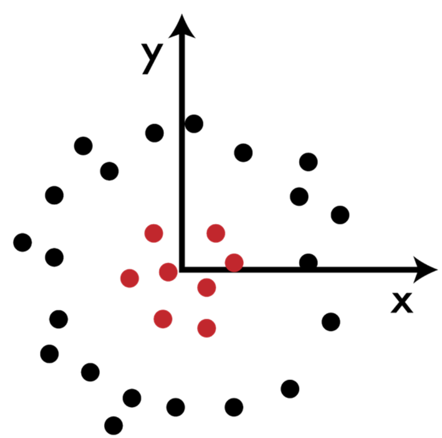
What is feature engineering?

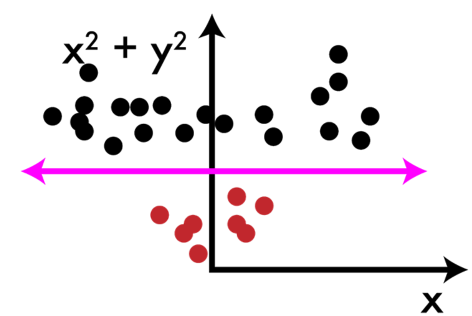
- Even with more sophisticate models, generally, feature engineering is just coming up with combinations of the features we already have

- Cannot learn transformations of features, only use existing features. Human must create good features



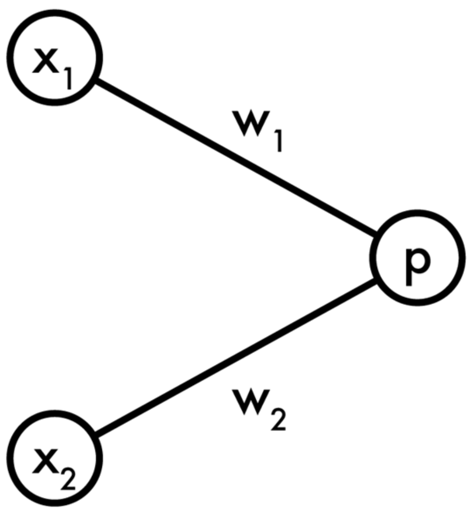
$$p = f(w_1x_1 + w_2x_2)$$





Overcoming the limitation of ML…

What if we added more processing?

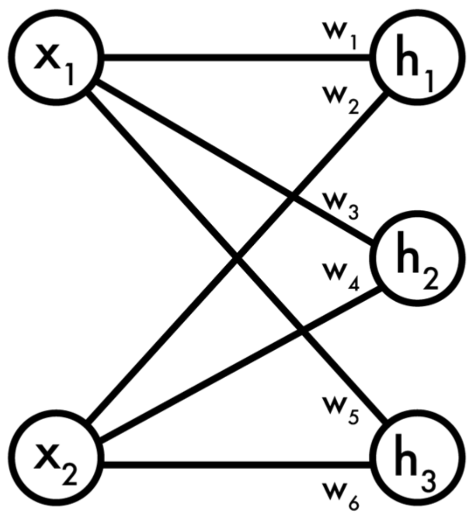


$$p = f(w_1 x_1 + w_2 x_2)$$

Overcoming the limitation of ML…

What if we added more processing?

Create “new” features using old ones. We’ll call **H** our *hidden layer*

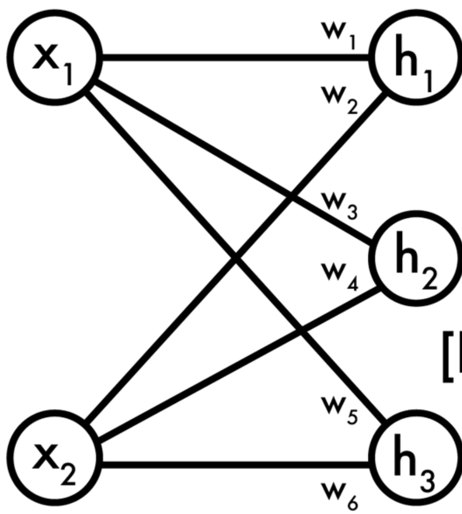


$h_1 = \varphi(w_1 x_1 + w_2 x_2)$
$h_2 = \varphi(w_3 x_1 + w_4 x_2)$
$h_3 = \varphi(w_5 x_1 + w_6 x_2)$

Overcoming the limitation of ML…

What if we added more processing?

As with linear model, **H** can be expressed in matrix operations



$$h_1 = \varphi(w_1x_1 + w_2x_2)$$
$$h_2 = \varphi(w_3x_1 + w_4x_2)$$
$$h_3 = \varphi(w_5x_1 + w_6x_2)$$

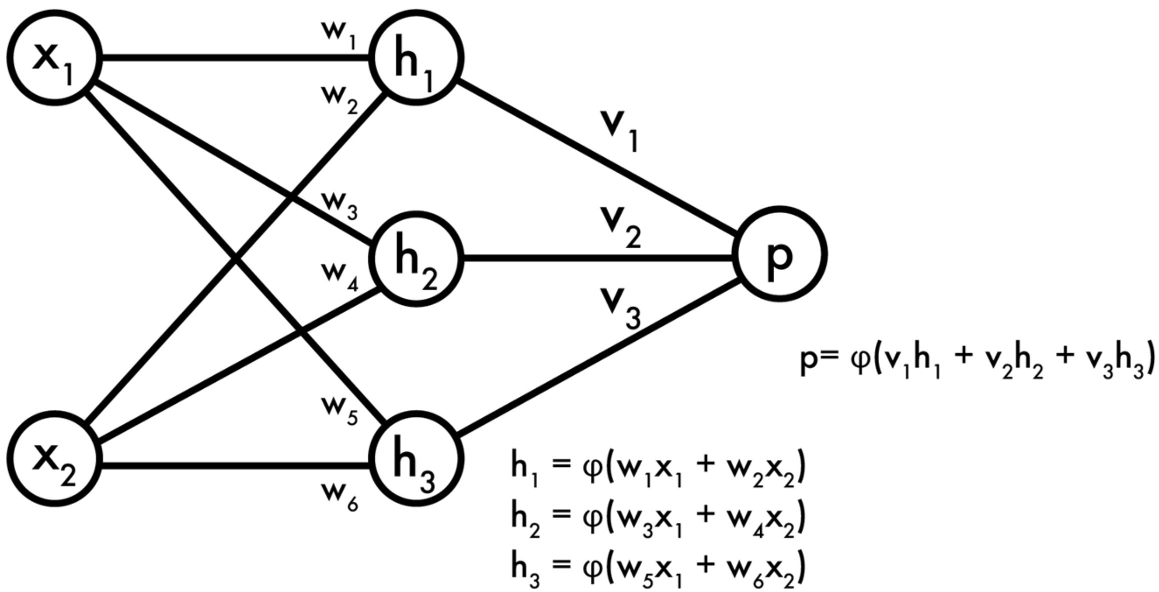
$$[h_1\ h_2\ h_3] = \varphi([x_1\ x_2]\begin{bmatrix} w_1 & w_3 & w_6 \\ w_2 & w_4 & w_5 \end{bmatrix})$$

$$\mathbf{H} = \varphi(\mathbf{Xw})$$

Overcoming the limitation of ML…

What if we added more processing?

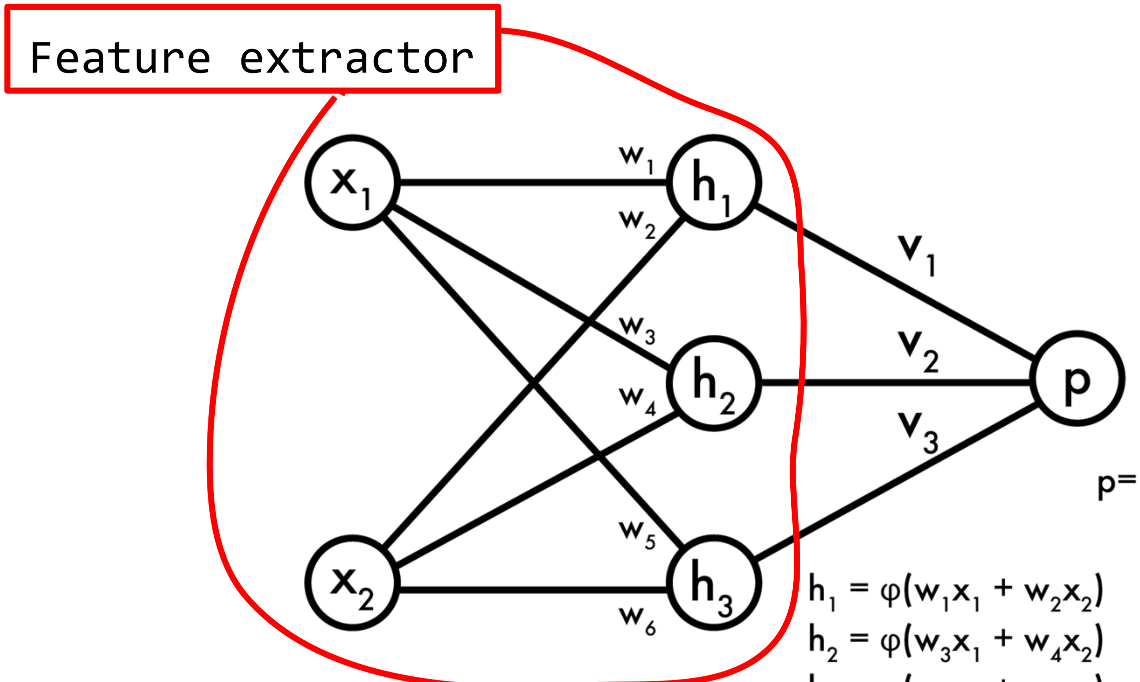
Now our prediction p is a function of our hidden layer



Overcoming the limitation of ML…

What if we added more processing?

Now our prediction p is a function of our hidden layer



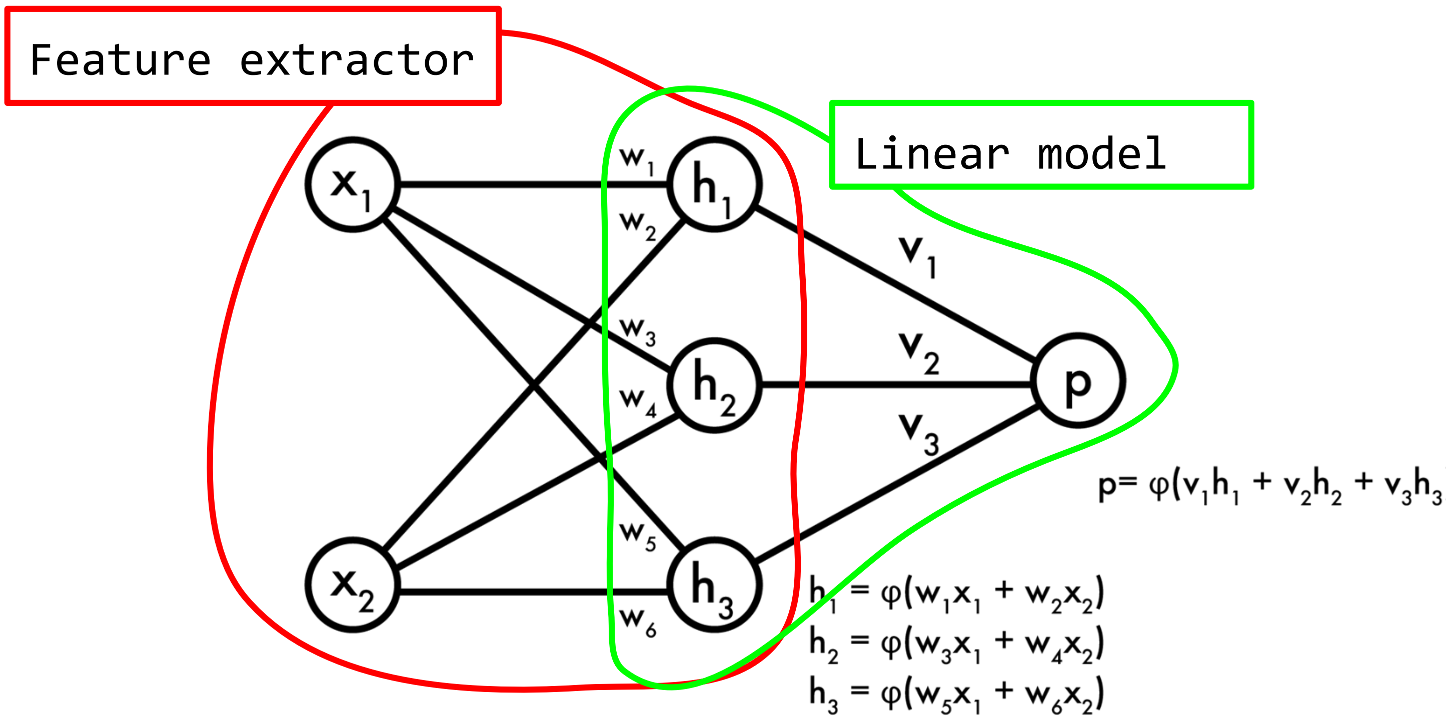
$p = \varphi(v_1h_1 + v_2h_2 + v_3h_3)$

$h_1 = \varphi(w_1x_1 + w_2x_2)$
$h_2 = \varphi(w_3x_1 + w_4x_2)$
$h_3 = \varphi(w_5x_1 + w_6x_2)$

Overcoming the limitation of ML…

What if we added more processing?

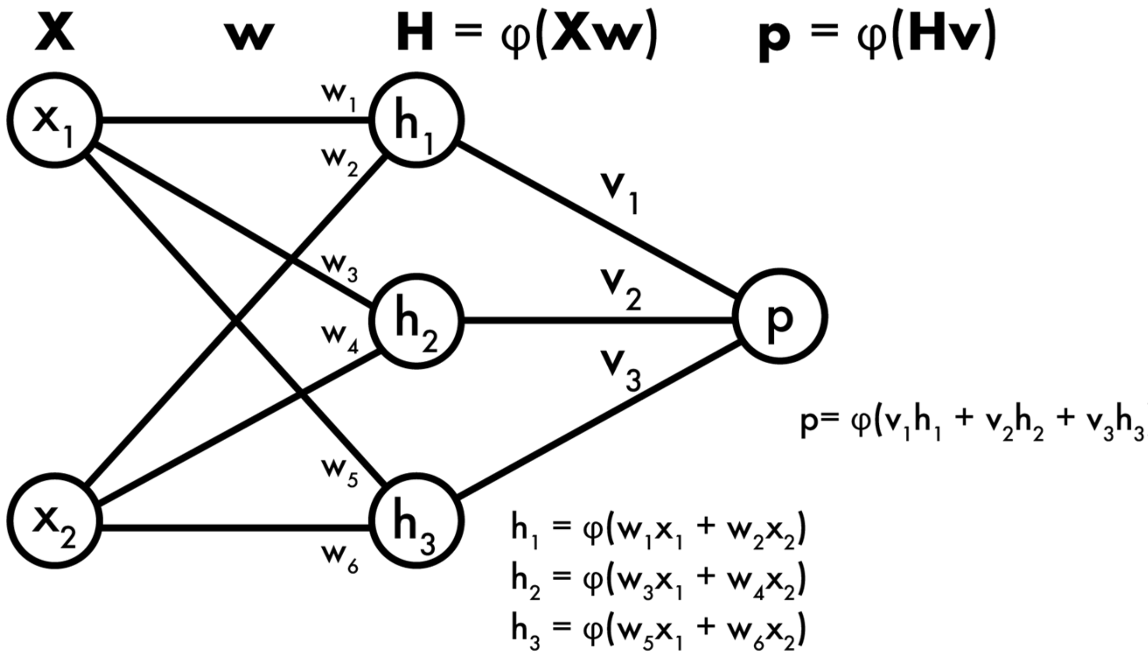
Now our prediction p is a function of our hidden layer



Overcoming the limitation of ML…

What if we added more processing?

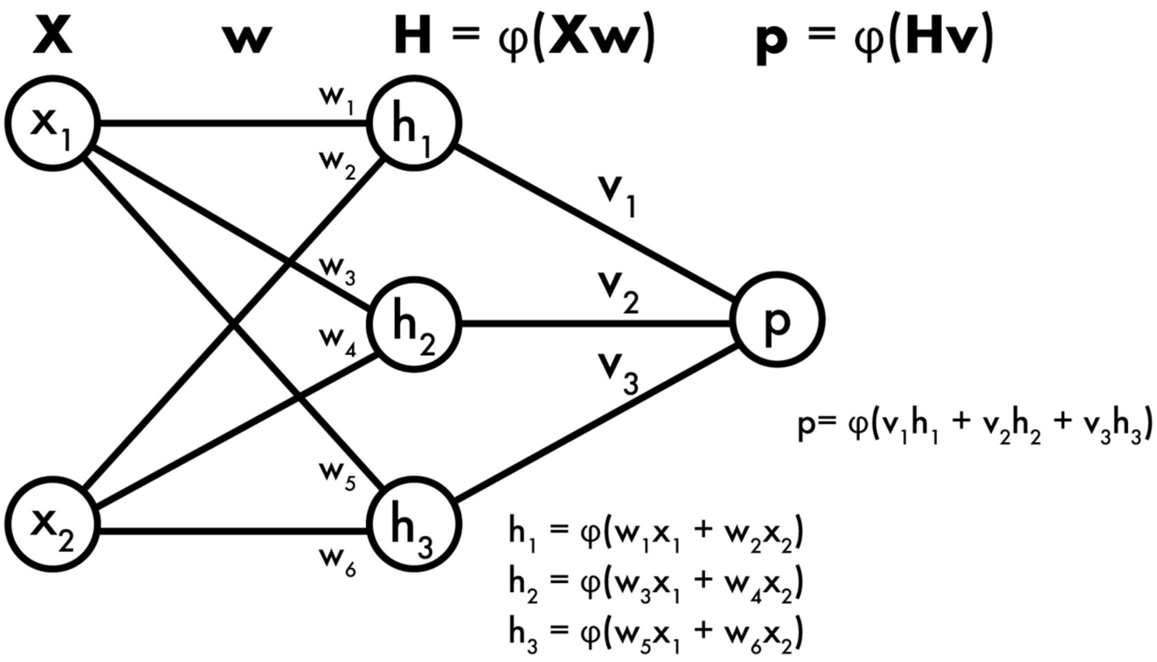
Can still express the whole process in matrix notation! Nice because matrix ops are fast



Overcoming the limitation of ML…

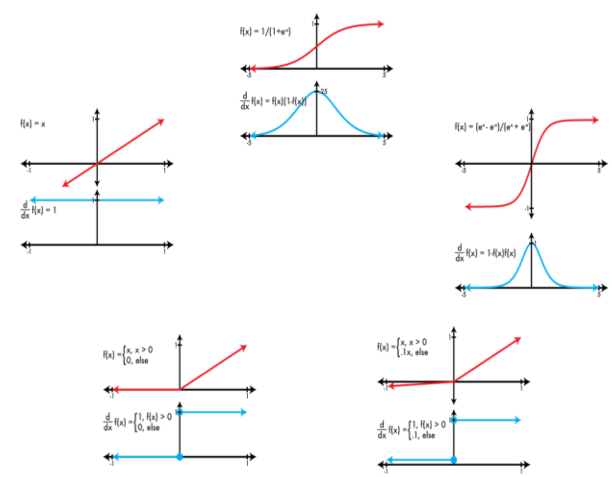
This is a neural network! (the starting point of Deep Learning)

- This one has 1 hidden layer, but can have **way** more
- Each layer is just **some functions φ** applied to linear combination of the previous layer



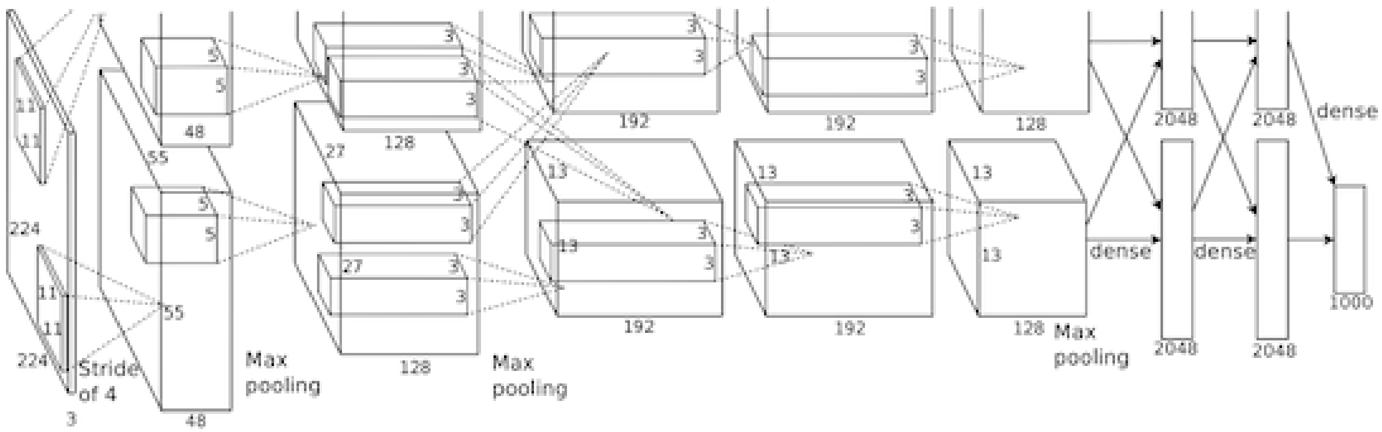
What about activation functions φ ?

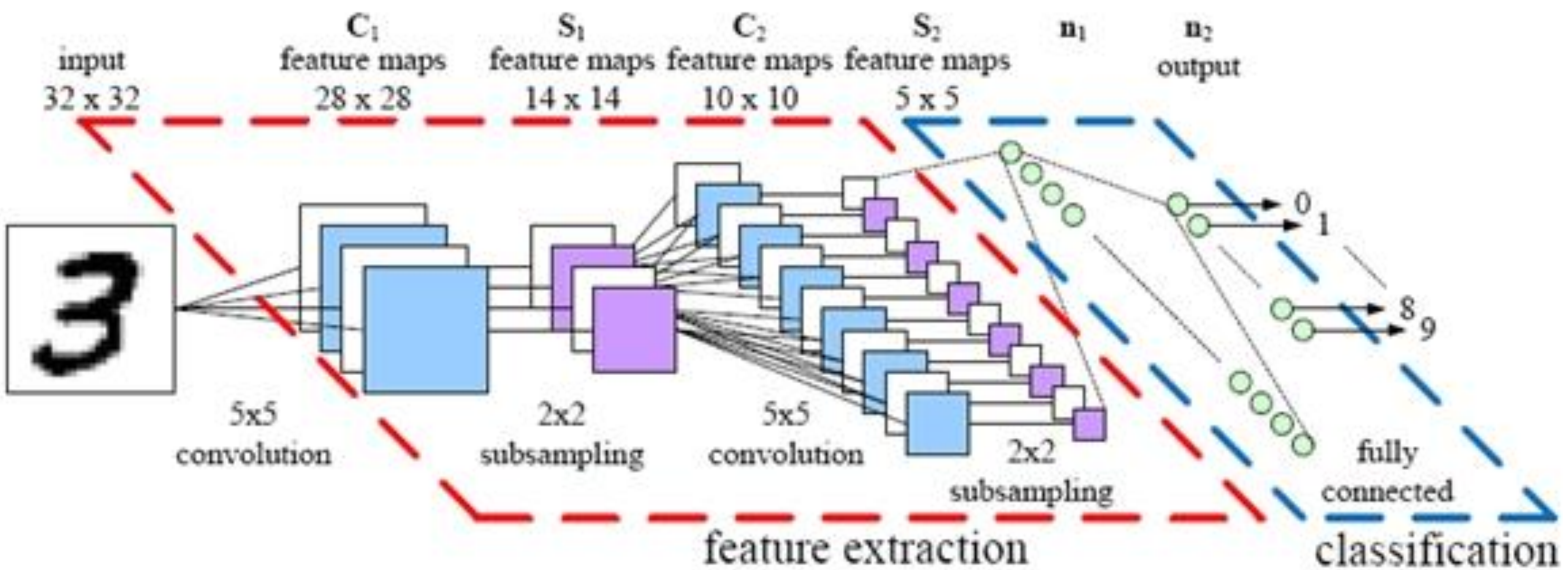
- So many possible options
- want them to be easy to take derivative



First idea of Deep Learning…

Deep Learning (Convolutional Neural Networks - CNN)





First idea of Deep Learning…

Too many weights!

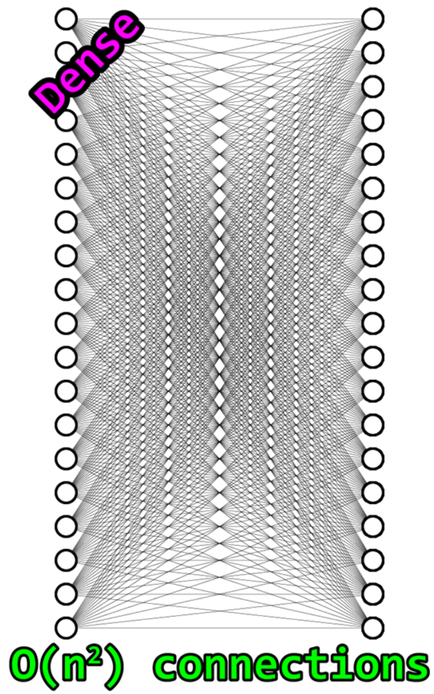
- Would rather have sparse connections

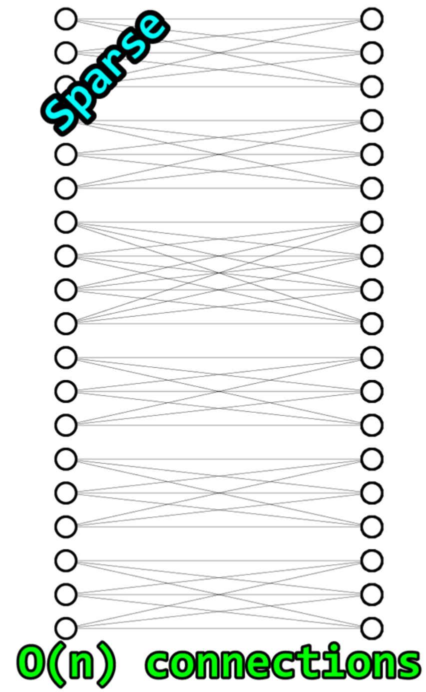
  Fewer weights
  Nearby regions - related
  Far apart - not related

- Convolutions!

  Just weighted sums of
  small areas in image

- Weight sharing in different
  locations in image



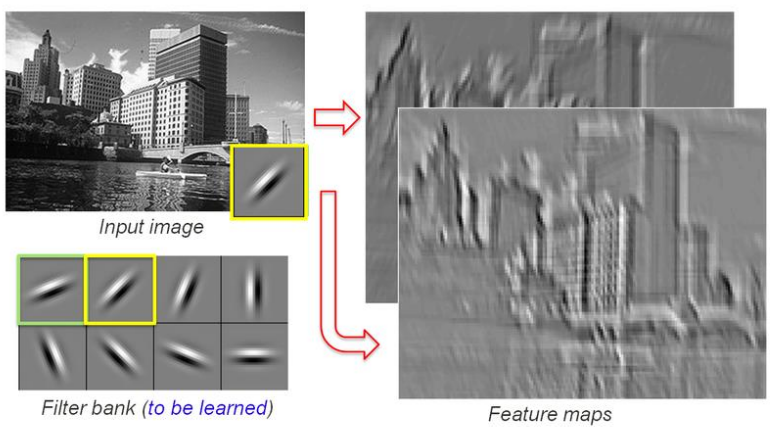


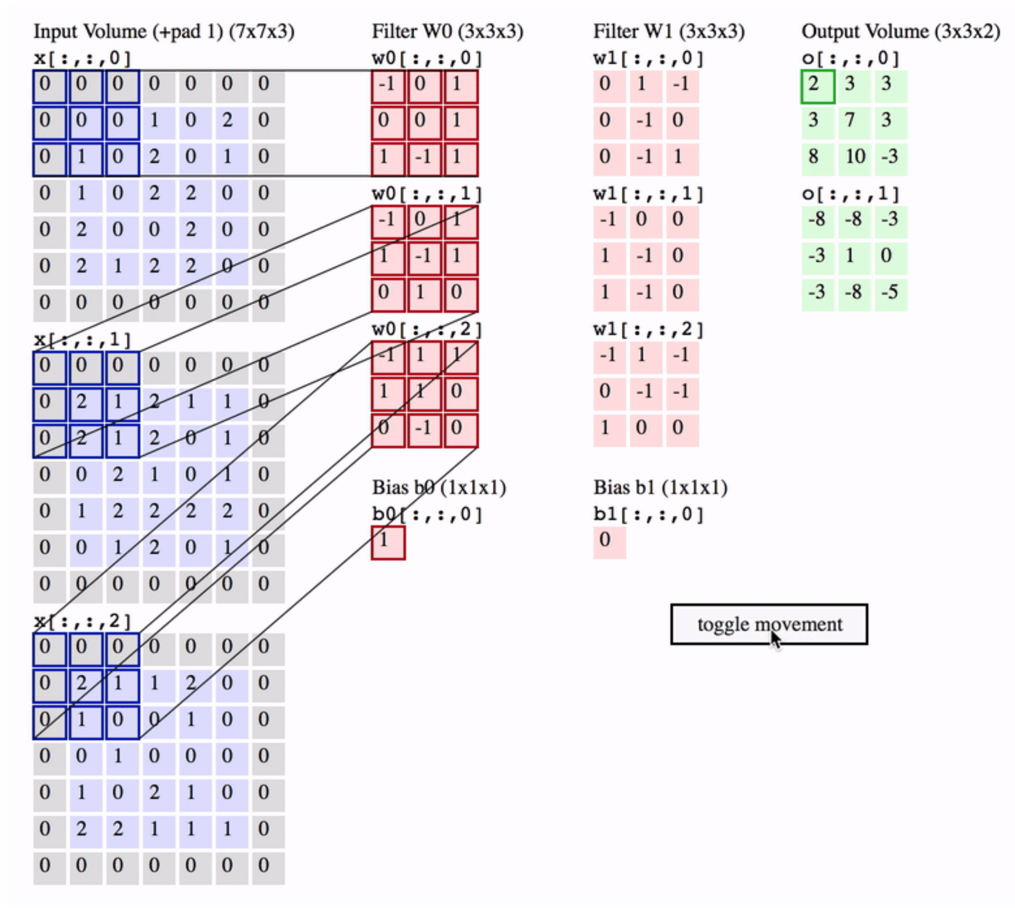
First idea of Deep Learning…

Convolutional Layer

- **Input:** an image
  Processing: convolution with multiple filters
  Output: an image, # channels = # filters

- Output still weighted sum of input (w/ activation)



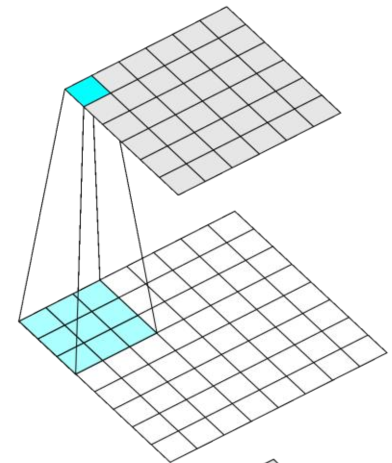


Example of a Conv layer with K = 2 filters, each with a spatial extent F = 3, moving at a stride S = 2, and input padding P = 1

First idea of Deep Learning…

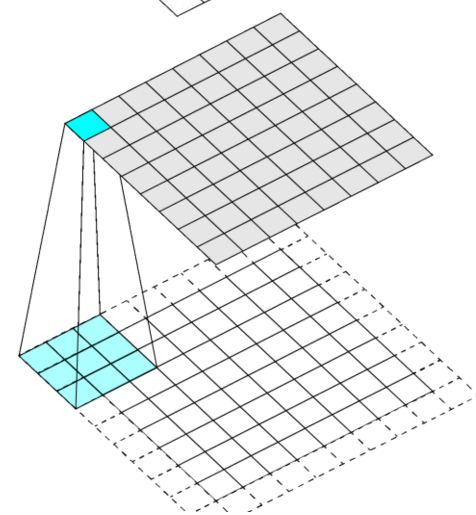
Padding

- Convolutions have problems on edges
- Do nothing: output a little smaller than input
- Pad: add extra pixels on edge



Stride

- How far to move filter between applications
- We've done stride 1 convolutions up until now, approximately preserves image size
- Could move filter further, downsample image



ReLU Layer

- Applies an elementwise activation function max(0,x)
  ➔ turns negative values to zeros

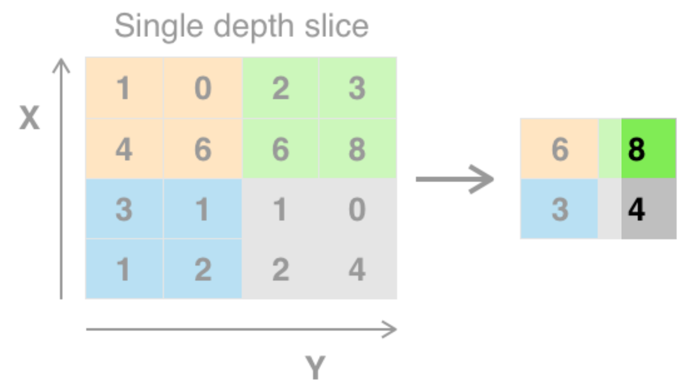
First idea of Deep Learning…

Pooling Layer

- **Input:** an image
  Processing: pool pixel values over region
  Output: an image, shrunk by a factor of the stride

- Hyperparameters:
  - What kind of pooling? Average, mean, max, min
  - How big of stride? Controls downsampling
  - How big of region? Usually not much bigger than stride

- Most common: 2x2 or 3x3 maxpooling, stride of 2



Max pooling

Fully Connected Layer

- The standard neural network layer where every input neuron connects to every output neuron

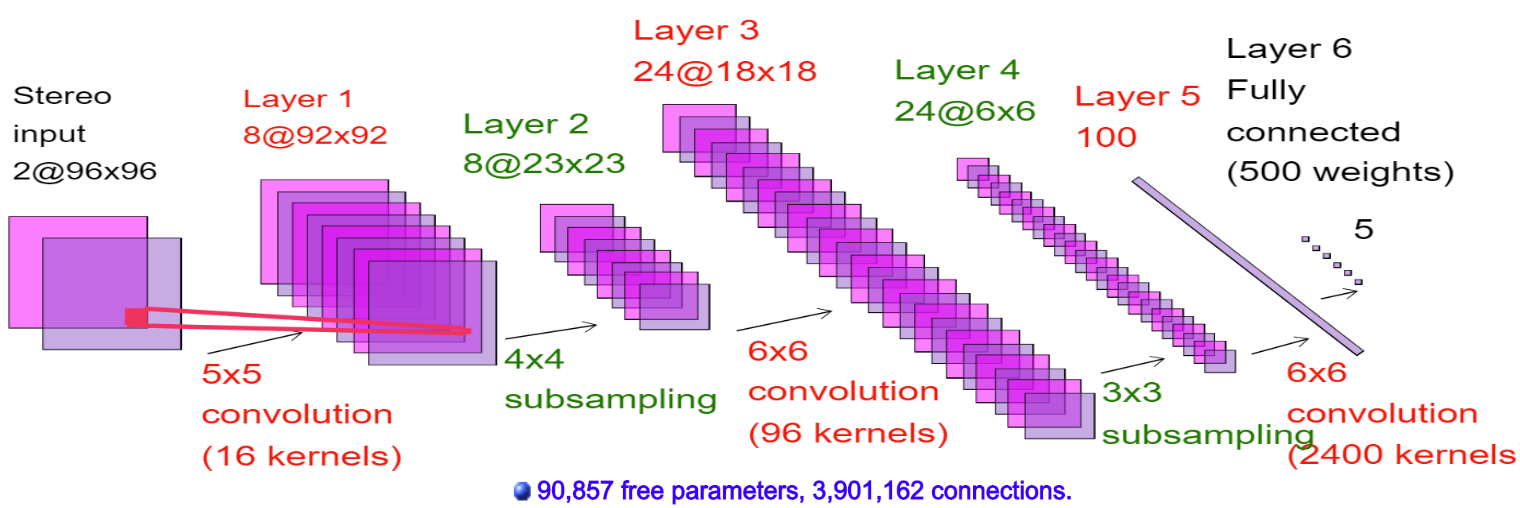
- Often used to go from image feature map -> final output or map image features to a single vector

- Eliminates spatial information

First idea of Deep Learning…

Example of Convnet Building Blocks

- Convolutional layers:
  - Connections are convolutions
  - Used to extract features
- Pooling layers:
  - Used to downsample feature maps, make processing more efficient
  - Most common: maxpool, avgpool sometimes used at end
- Connected layers:
  - Often used as last layer, to map image features → prediction
  - No spatial information
  - lots of weights, no weight sharing ➔ Inefficient ?

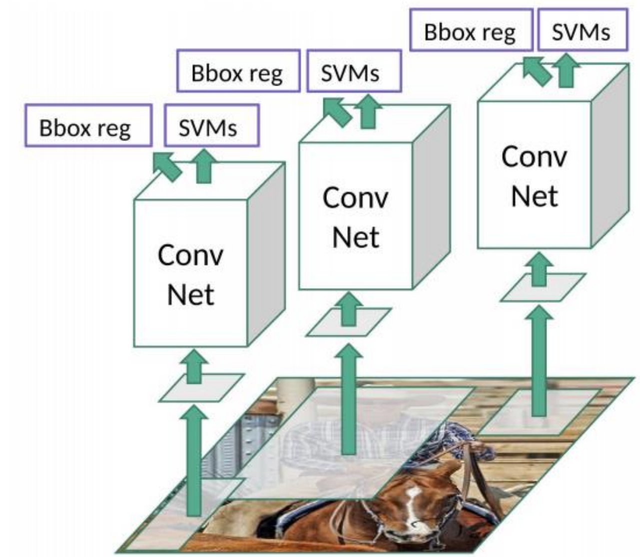


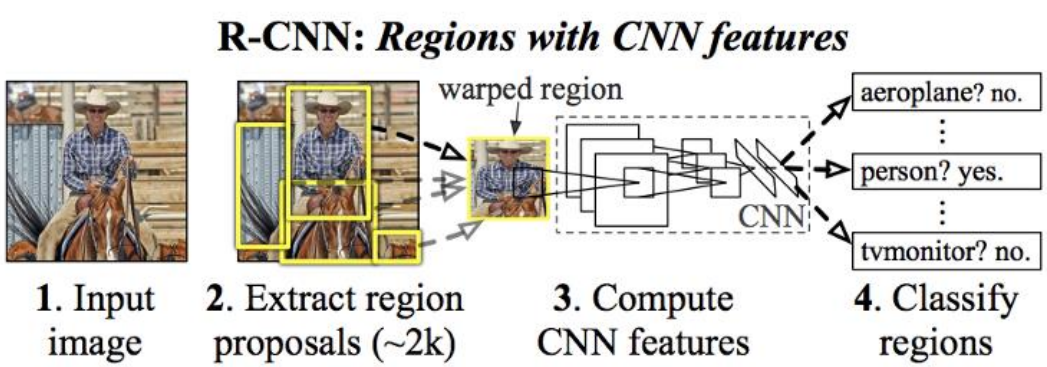
First idea of Deep Learning…

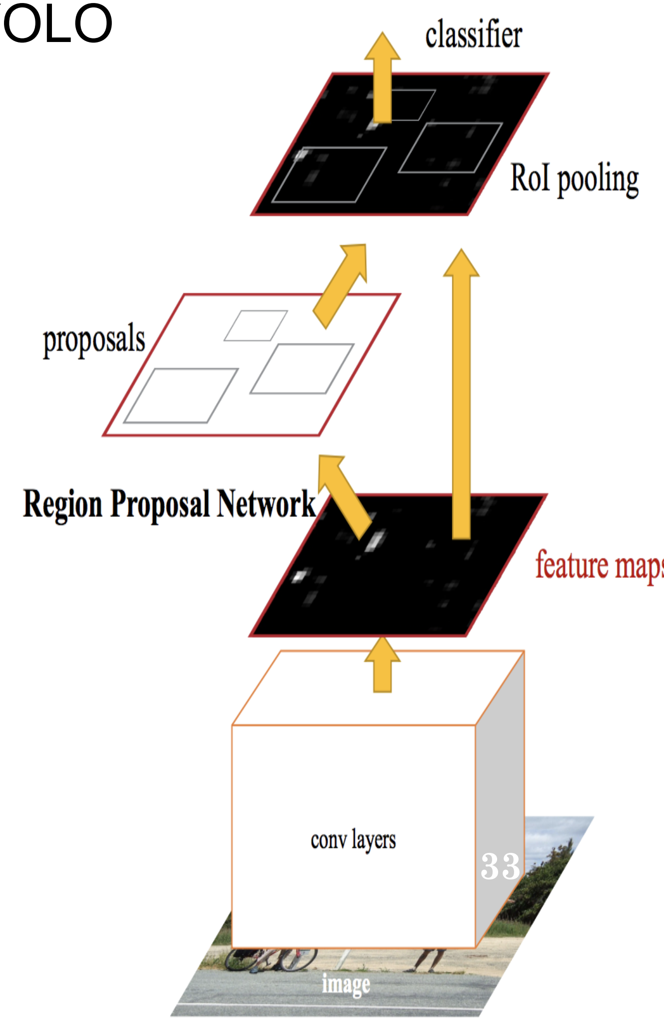
So many architectures & hyperparameters….

➔ Architecture engineering replaces Feature engineering

- CONVnet, R-CNN, Fast R-CNN, Faster R-CNN, YOLO





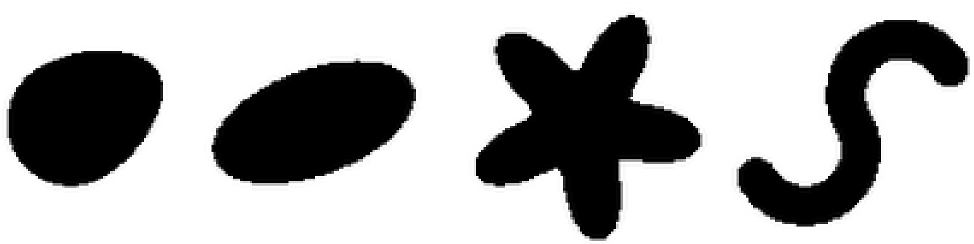


Binary Image Processing

Why so much attention?

- **Problem simplification (0 & 1)**
  - Background = white / Shapes = black
  - 1 object ➔ 1 region / 1 contour ➔ 1 black component (blob)
  - Analysis, characterization of objects from B&W data
  - Counting, measure, shape analysis are easier

- **Mathematic morphology and discrete geometry**
  - Many possible operations on binary images

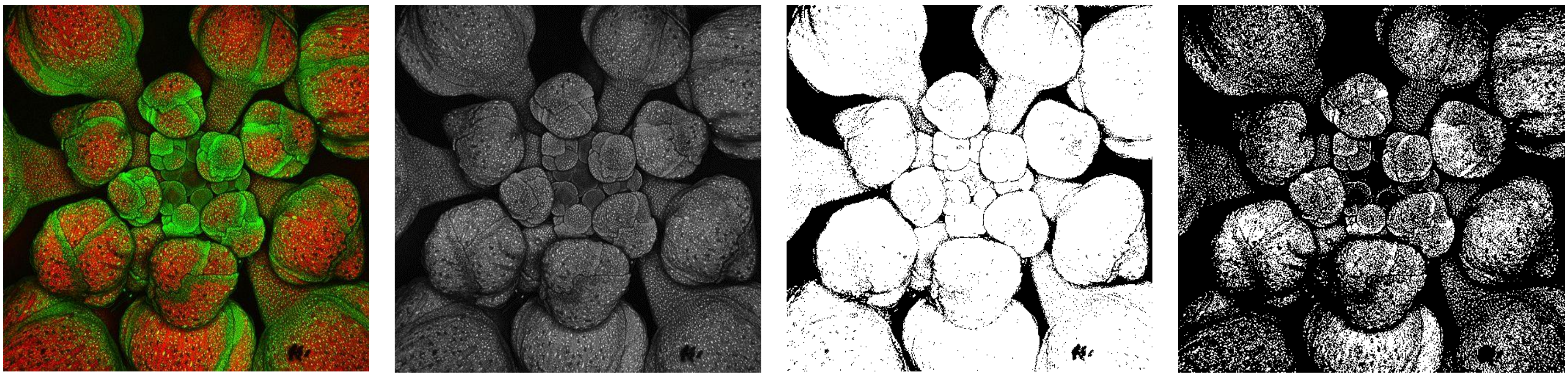


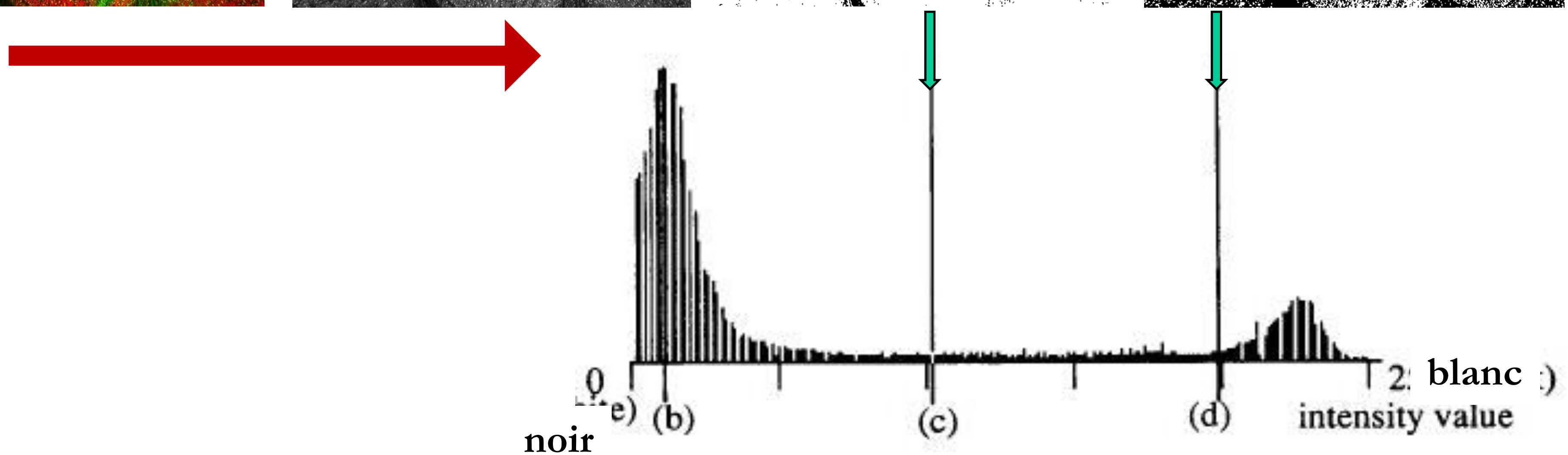
Binarisation = Thresholding

- **Manual Thresholding ➔** The easiest and the most often used
- There is a relation between the gray level and the probability of belonging to a shape

Binarisation with a global threshold

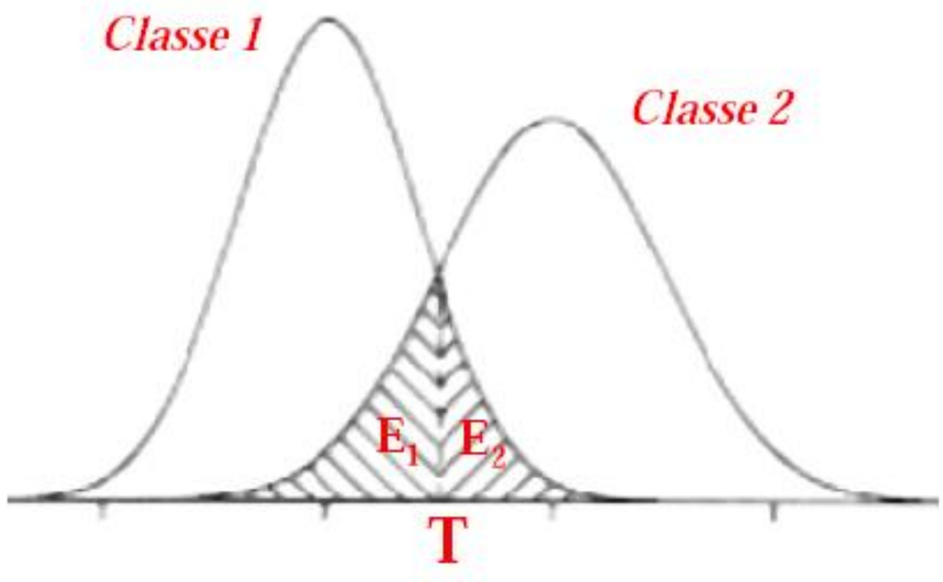
- How to choose a good threshold ➔ so many methods
  - It depends of the images and the objectives …





Automatic thresholding: OTSU

- Knowing there are only 2 classes of color in the image
  - We assume that the gray level distribution is composed by 2 Gaussians
  - We look for a threshold *t* that will minimize the intra-class variance



Automatic thresholding: OTSU

- For each possible value of ***t***, the intra class variance is computed

$$\sigma_w^2(t) = \omega_1(t)\sigma_1^2(t) + \omega_2(t)\sigma_2^2(t)$$

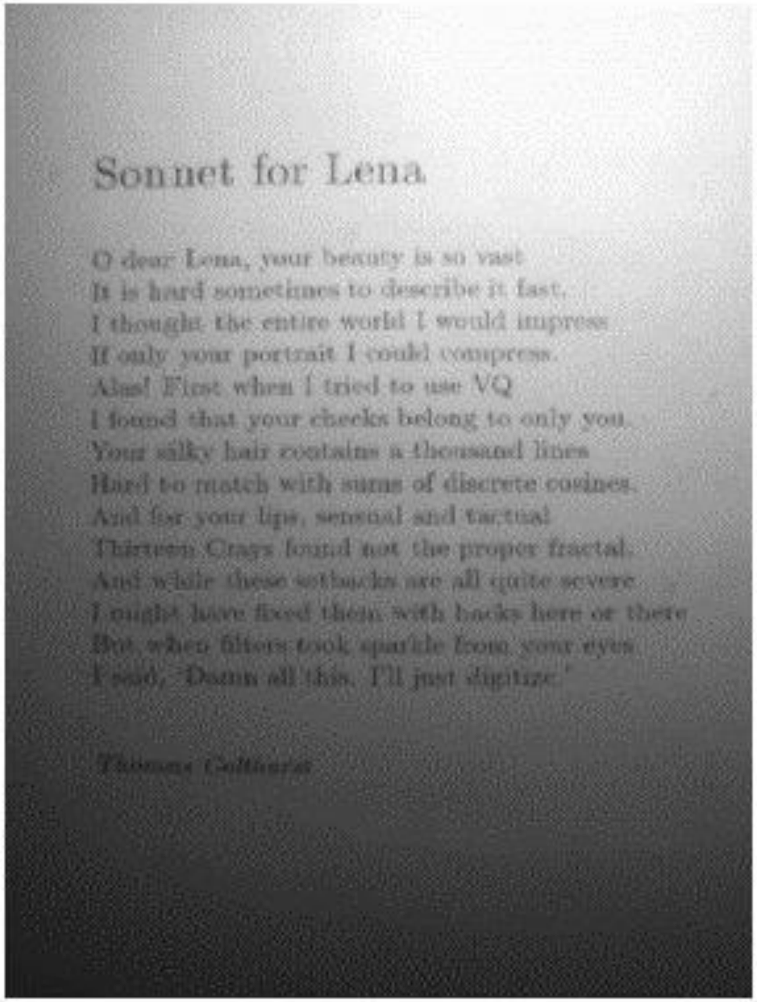
- The optimal threshold is the one for which σ^2_w is minimum
  - $\boldsymbol{w_i}$ is the probability (weight) of the class *i*
  - σ^2_i is the variance of the gray levels of the class *i*
- A more efficiant formulation uses the inter-class variance σ^2_b
  - probability and mean of the classes can be updated iteratively

$$\sigma_b^2(t) = \sigma^2 - \sigma_w^2(t) = \omega_1(t)\omega_2(t)\left[\mu_1(t) - \mu_2(t)\right]^2$$

1. Calculer l'histogramme et les probabilités de chaque niveau d'intensité
2. Définir les $\omega_i(0)$ et $\mu_i(0)$ initiaux
3. Parcourir tous les seuils possibles $t = 1 \ldots$ intensité max
    1. Mettre à jour ω_i et μ_i
    2. Calculer $\sigma_b^2(t)$
4. Le seuil désiré correspond au $\sigma_b^2(t)$ maximum.

Problem with Global Thresholding

- **A solution ?**



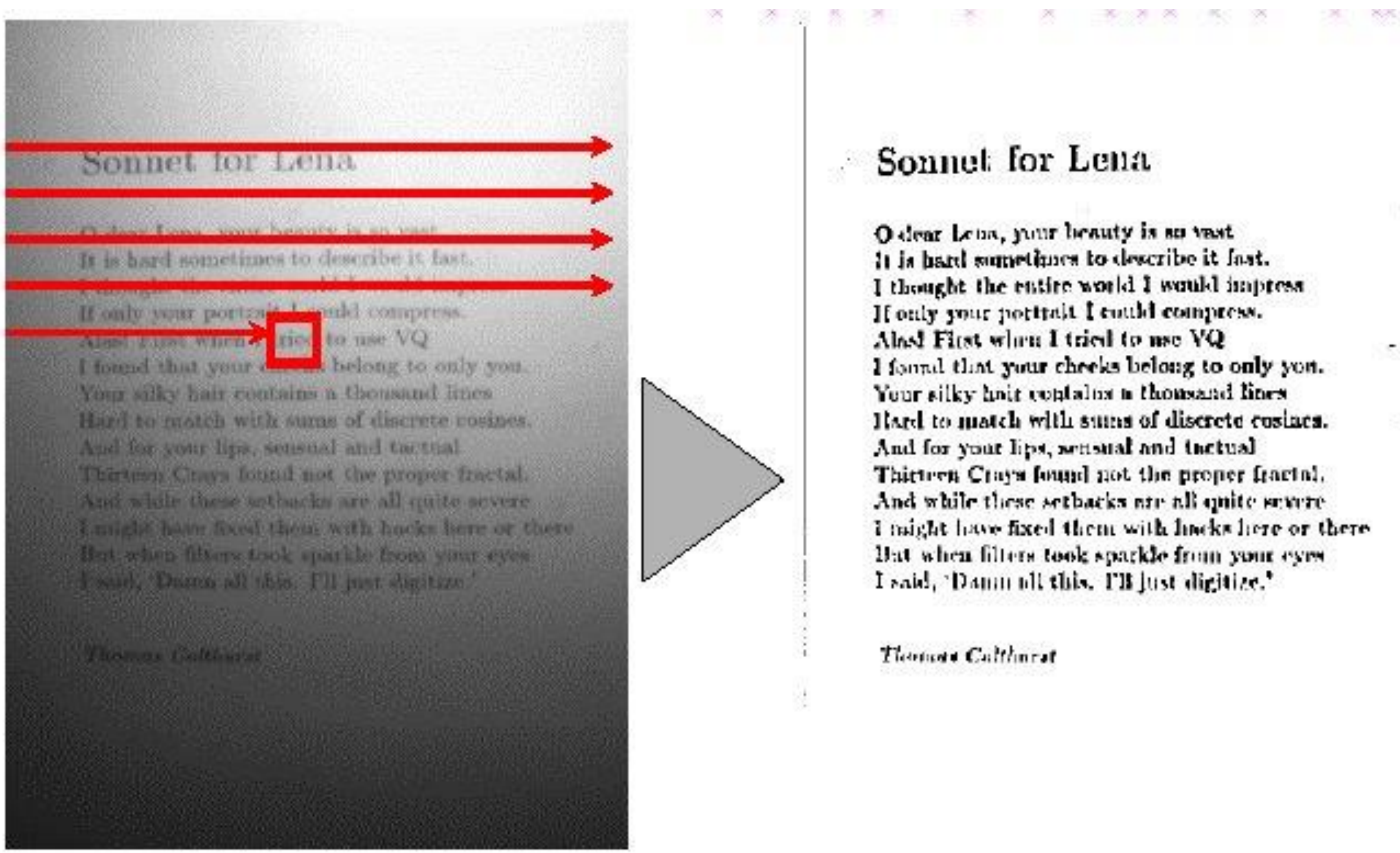
Sonnet for Lena

O dear Lena, your beauty is so vast
It is hard sometimes to describe it fast.
I thought the entire world I would impress
If only your portrait I could compress.
Alas! First when I tried to use VQ
I found that your cheeks belong to only you.
Your silky hair contains a thousand lines
Hard to match with sums of discrete cosines.
And for your lips, sensual and tactual
Thirteen Crays found not the proper fractal.
And while these setbacks are all quite severe
I might have fixed them with hacks here or there
But when filters took sparkle from your eyes
I said, 'Damn all this. I'll just digitize.'

Thomas Colthurst

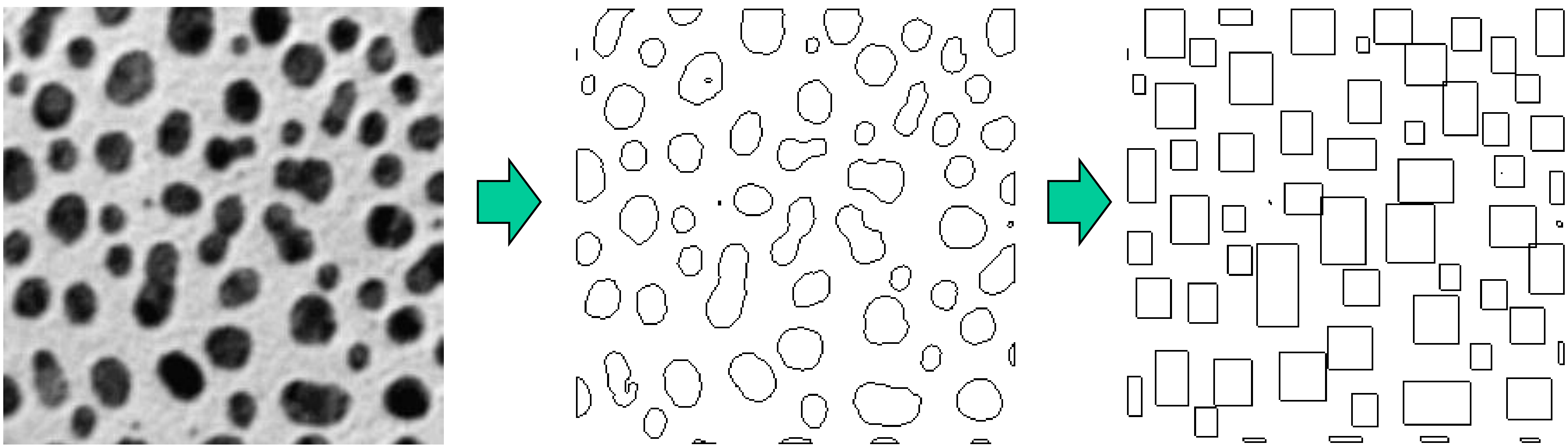
Adaptive local Thresholding

- A local threshold is defined and used for each pixel according to it neighborhood (sometime difficult to select)
- **Niblack :** $S = m + ks^2$ avec k= -0,2 | m : mean et s : standard deviation



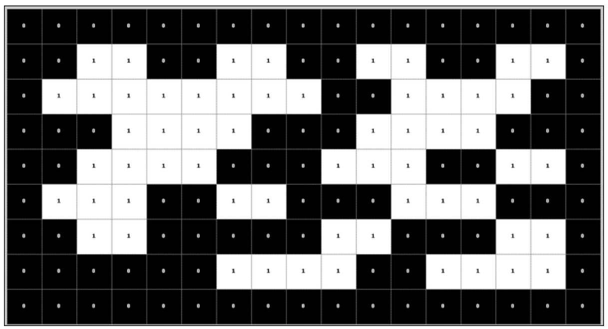
Background-Foreground segmentation

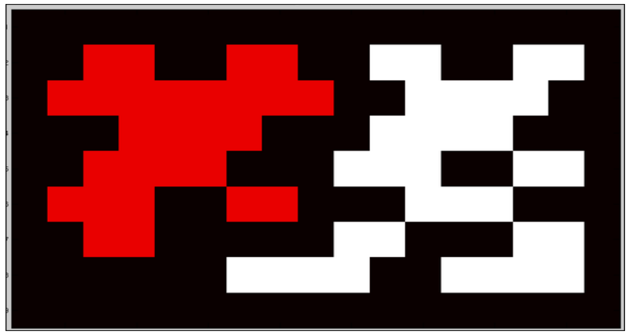
- **How to analyze black shapes on a white background?**
  - Shape localization
  - Counting
  - Describing, charaterizing
  - Classifying



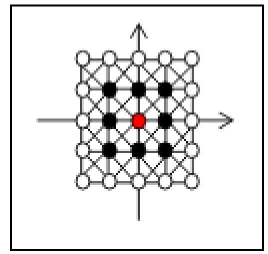
Connected component extraction

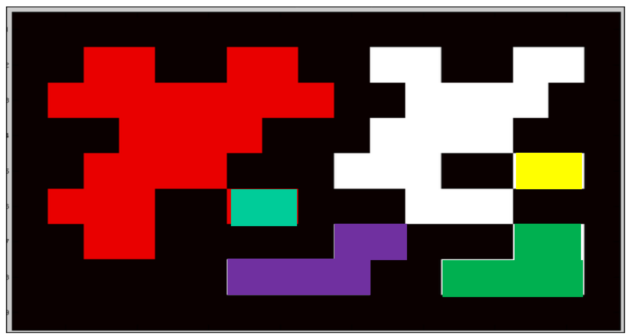
- **Notion of Connected component**
  - A set S of pixels is a 4-connected component if and only if, for all pairs of pixels P and Q, a 4-path $p_1, p_2, p_3, \ldots p_n$ with $p_1 = P$ and $p_n = Q$ and all $p_i \in S$.
  - A set S of pixels is a 8-connected component if and only if, for all pairs of pixels P and Q, a 8-path $p_1, p_2, p_3, \ldots p_n$ with $p_1 = P$ and $p_n = Q$ and all $p_i \in S$.



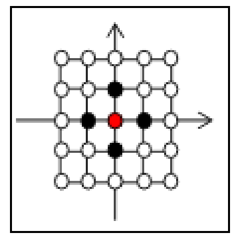


8-connexity





4-connexity



Connected component extraction

- **A two pass algorithm**

```
. . . . . . . . . . . . . . .
. . . * * . . . * * * . . . .
. . . * * . . . * * * . . . .
. . * * * * . * * * * . . . .
. . . * * * * * * * * . . . .
. . . . . * * * * . . . . . .
. . . . . . * * * . . * . . .
. . . . . . * * * . . * * . .
. . * * . . * * * . . * * . .
. . * * . . . . . . . . . . .
. . . . . . . . . . . . . . .
```

(b) Original binary image.

```
. . . . . . . . . . . . . . .
. . . 1 1 . . . 2 2 2 . . . .
. . . 1 1 . . . 2 2 2 . . . .
. . 1 1 1 1 . 2 2 2 2 . . . .
. . . 1 1 1 ? * * * * . . . .
. . . . . * * * * . . . . . .
. . . . . . * * * . . * . . .
. . . . . . * * * . . * * . .
. . * * . . * * * . . * * . .
. . * * . . . . . . . . . . .
. . . . . . . . . . . . . . .
```

(c) Labeling in progress.

```
. . . . . . . . . . . . . . .
. . . 1 1 . . . 2 2 2 . . . .
. . . 1 1 . . . 2 2 2 . . . .
. . 1 1 1 1 . 2 2 2 2 . . . .
. . . 1 1 1 1 1 1 1 1 . . . .
. . . . . 1 1 1 1 . . . . . .
. . . . . . 1 1 1 . . 3 . . .
. . . . . . 1 1 1 . . 3 3 . .
. . 4 4 . . 1 1 1 . . 3 3 . .
. . 4 4 . . . . . . . . . . .
. . . . . . . . . . . . . . .
```

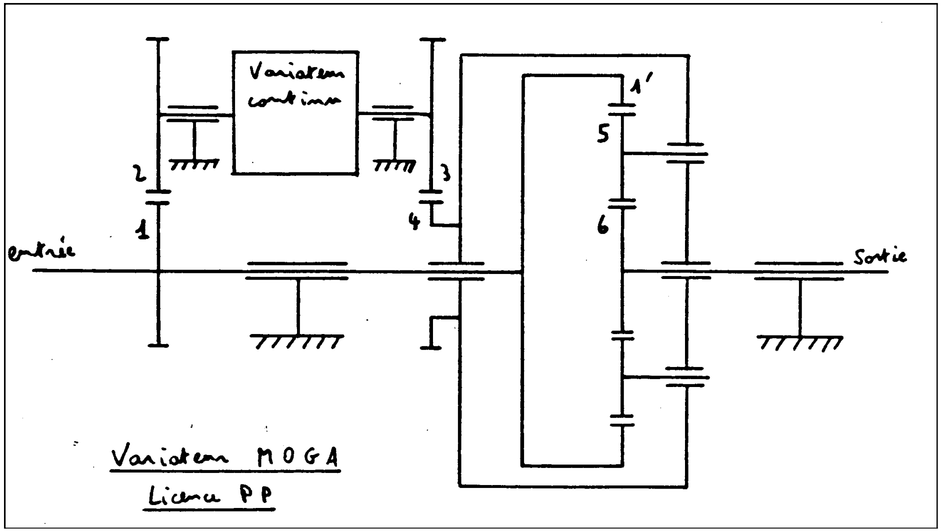
(d) Scanning completion.

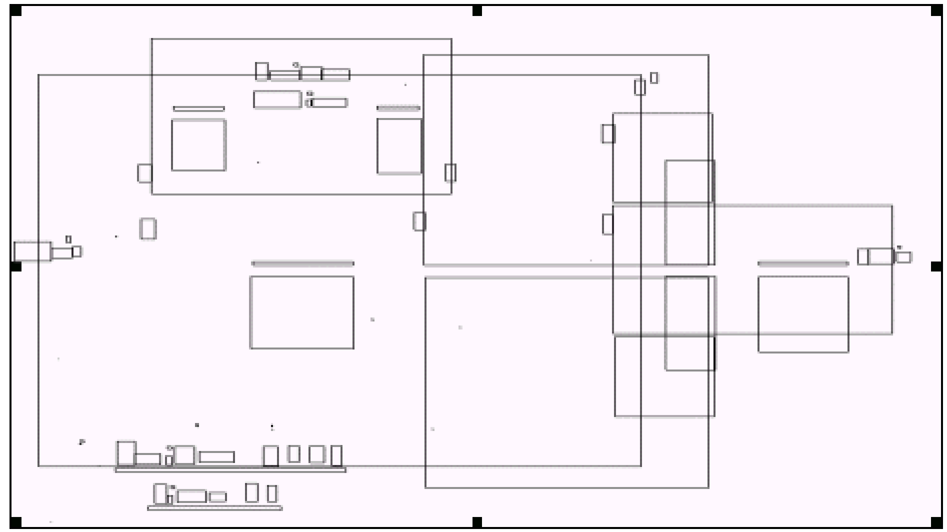
```
. . . . . . . . . . . . . . .
. . . 1 1 . . . 1 1 1 . . . .
. . . 1 1 . . . 1 1 1 . . . .
. . 1 1 1 1 . 1 1 1 1 . . . .
. . . 1 1 1 1 1 1 1 1 . . . .
. . . . . 1 1 1 1 . . . . . .
. . . . . . 1 1 1 . . 2 . . .
. . . . . . 1 1 1 . . 2 2 . .
. . 3 3 . . 1 1 1 . . 2 2 . .
. . 3 3 . . . . . . . . . . .
. . . . . . . . . . . . . . .
```

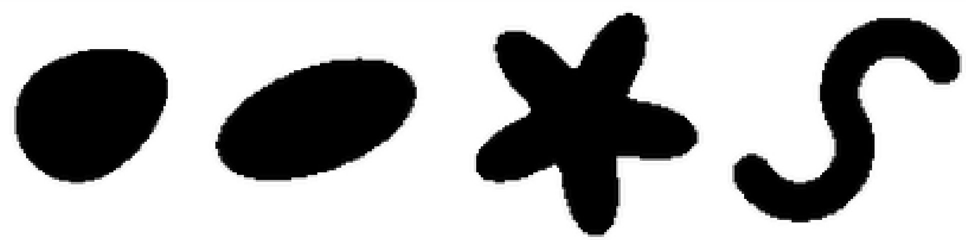
(e) After label unification and reassignment.

Connected component Analysis

- Studying their size and shape



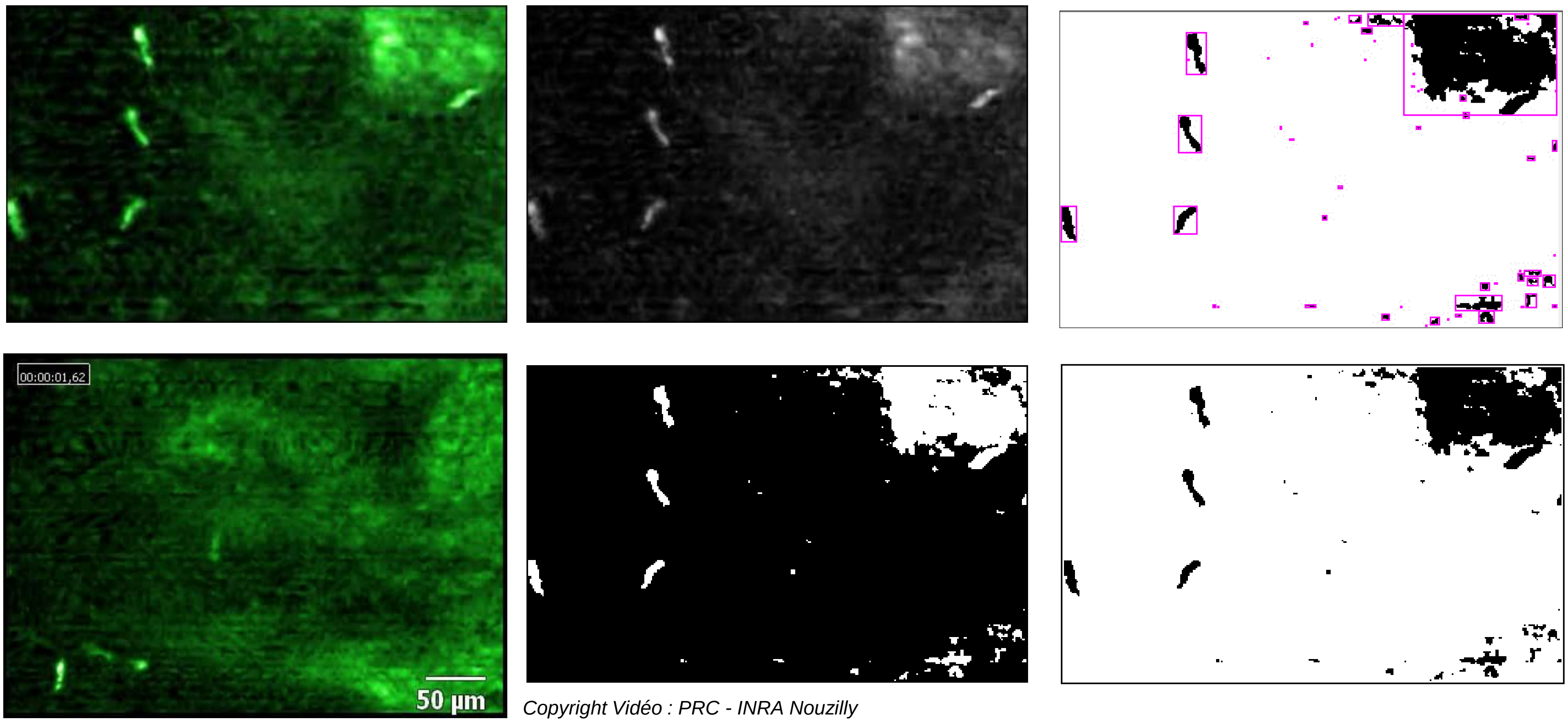




Circularité	0.966	0.834	0.395	0.280
Compacité	0.973	0.961	0.685	0.556
Allongement	1.29	1.99	1.00	2.63
Tortuosité	1.06	1.06	1.04	1.26

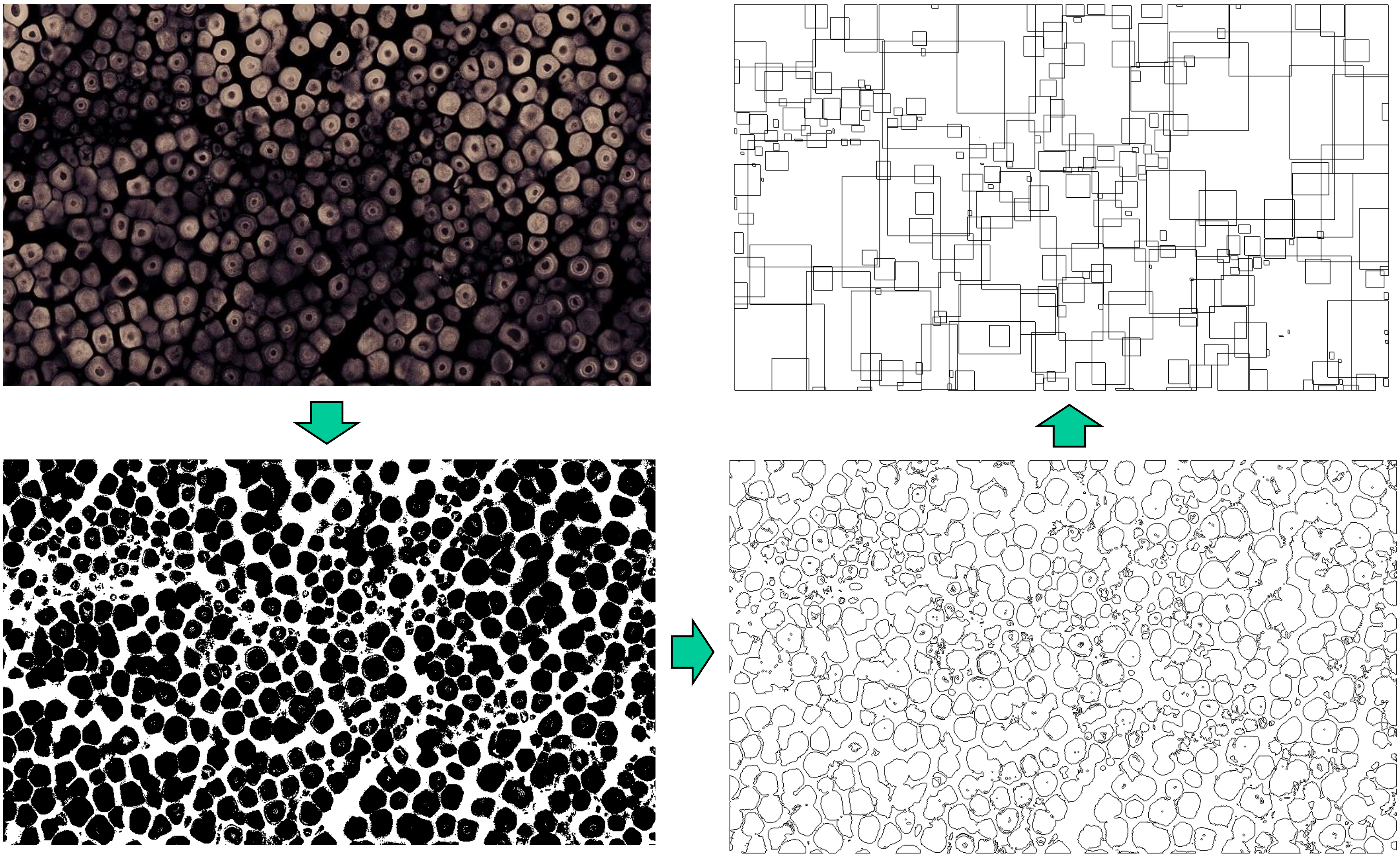
Connected component Analysis

- Exploitation of the connected components
  - How ?
  - Image analysis sequence ?



Copyright Vidéo : PRC - INRA Nouzilly

Connected Components and connexity !



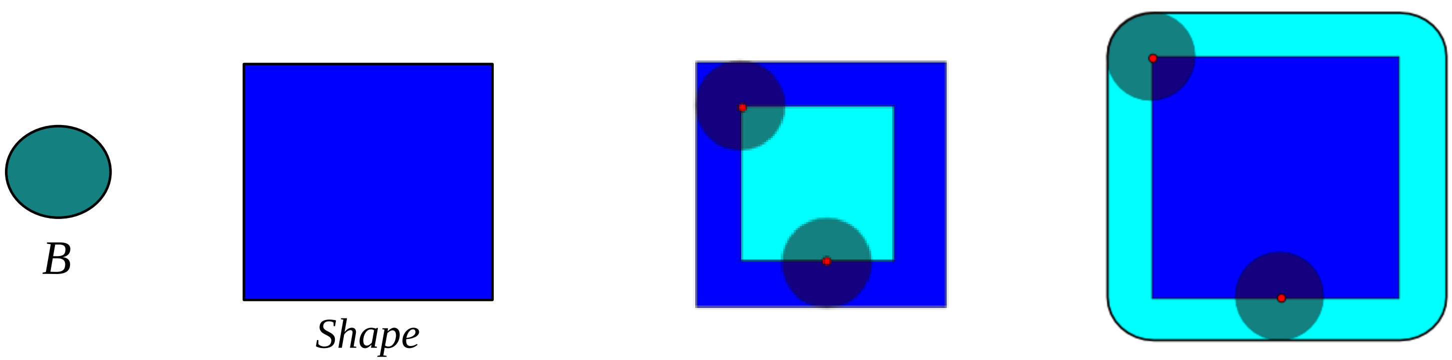
Mathematical / Binary morphology

Binary morphology and structuring element

- The basic idea in binary morphology is to probe an image with a simple, pre-defined shape, drawing conclusions on how this shape fits or misses the shapes in the image.
- This simple "probe" is called the structuring element, and is itself a binary image

Some examples of widely used structuring elements (denoted by *B*):

- In R^2, B is an open disk of radius r, centered at the origin.
- In Z^2, B is a 3x3 square, that is, B={(-1,-1), (-1,0), (-1,1), (0,-1), (0,0), (0,1), (1,-1), (1,0), (1,1)}.
- In Z^2, B is the "cross" given by: B={(-1,0), (0,-1), (0,0), (0,1), (1,0)}.



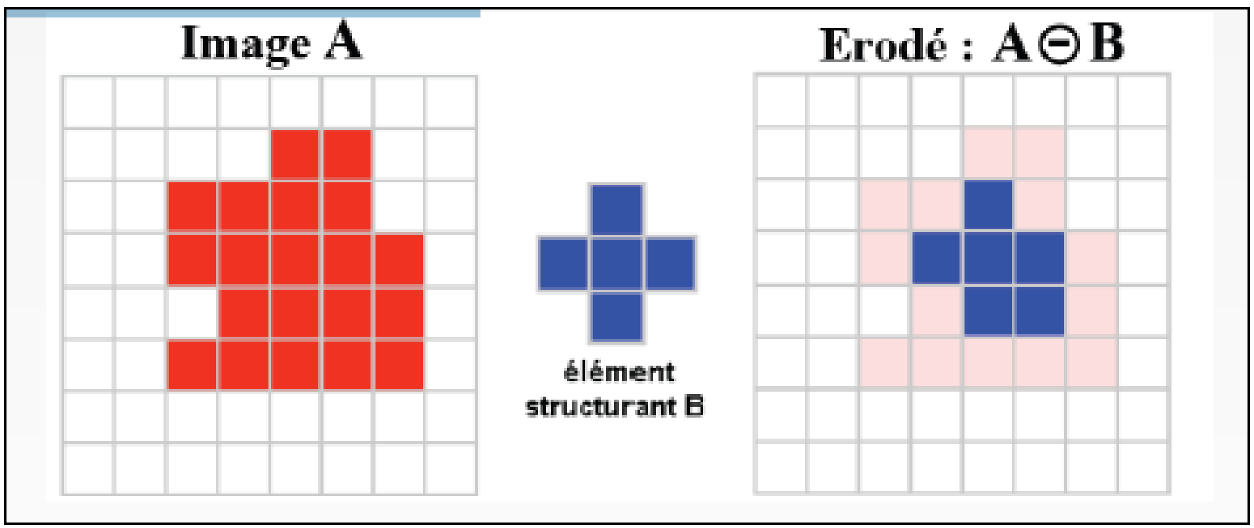
Basic operations: Erosion

Let B_x be the center of the structuring element B that is put on the pixel X of the image

Algorithm :

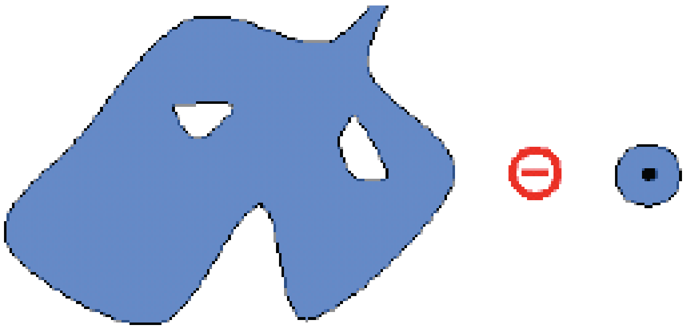
- B_x is positioned on each pixel X of the object A
- IF all pixels of B are inside the object A THEN

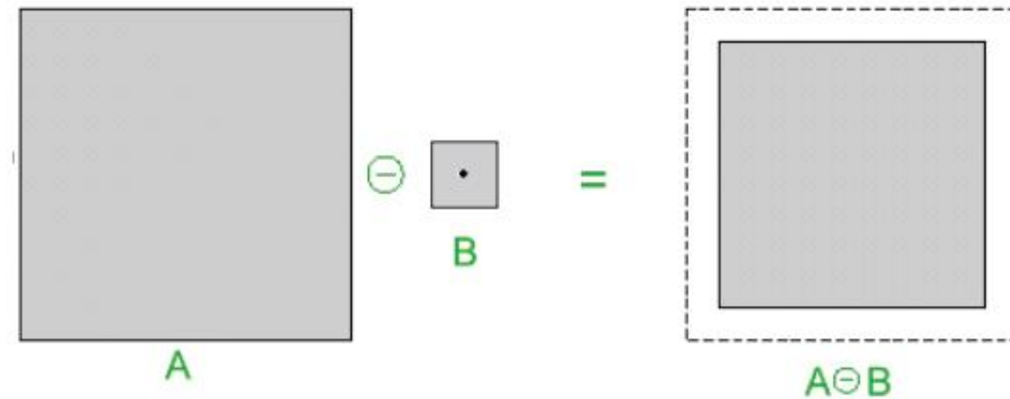
  B_x is set to kept (as part of the eroded object)

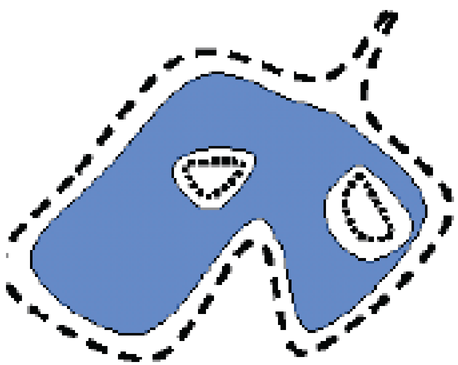


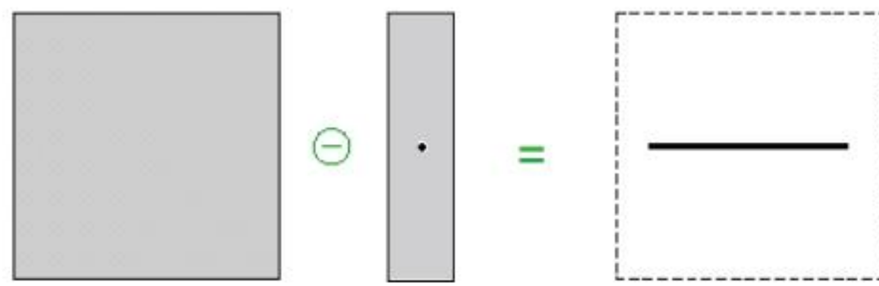
Basic operations: Erosion

- Examples









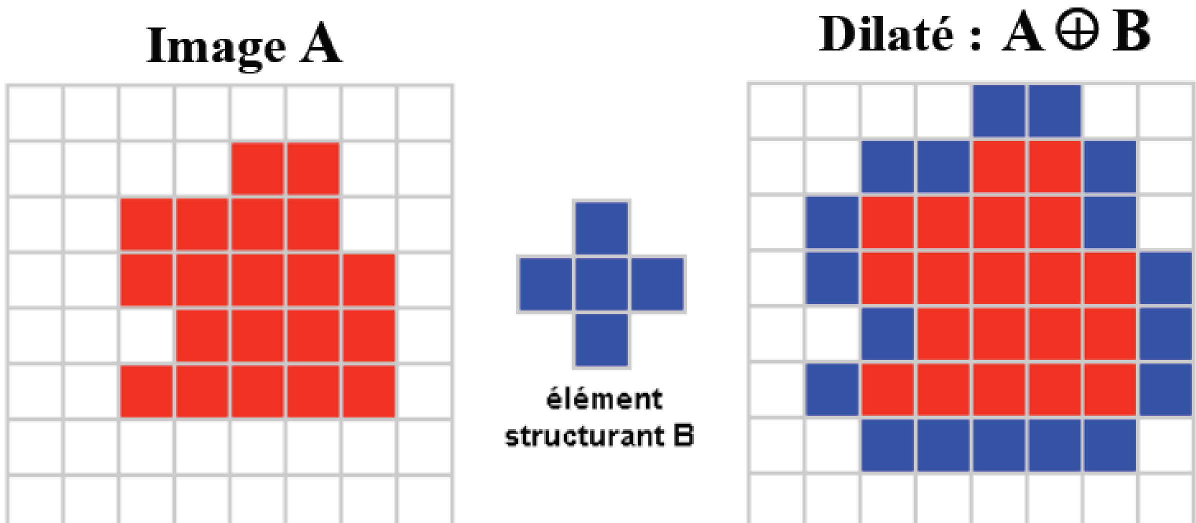
Basic operations: Dilatation

Let B_x be the center of the structuring element B that is put on the pixel X of the image

Algorithm :

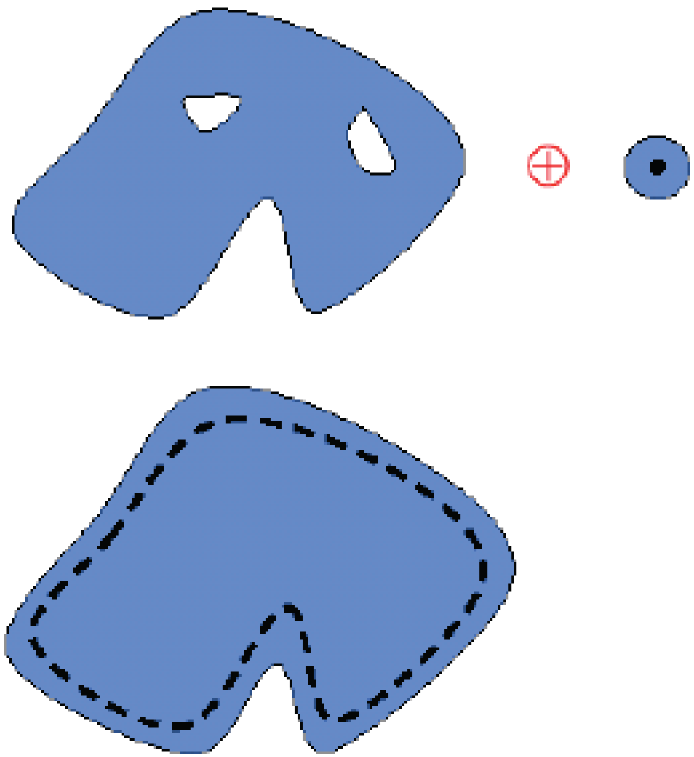
- B_x is positioned on each pixel X of the object A
- IF $B \cap A \neq \emptyset$ THEN

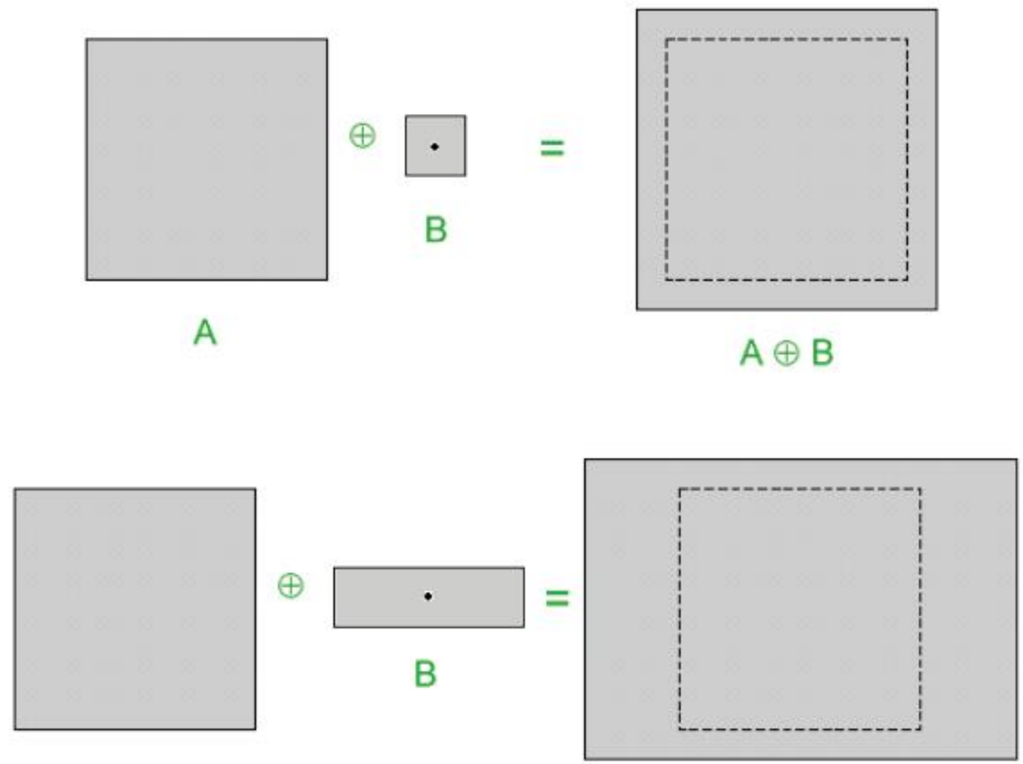
  B_x is set to kept (as part of the dilated object)



Basic operations: Dilatation

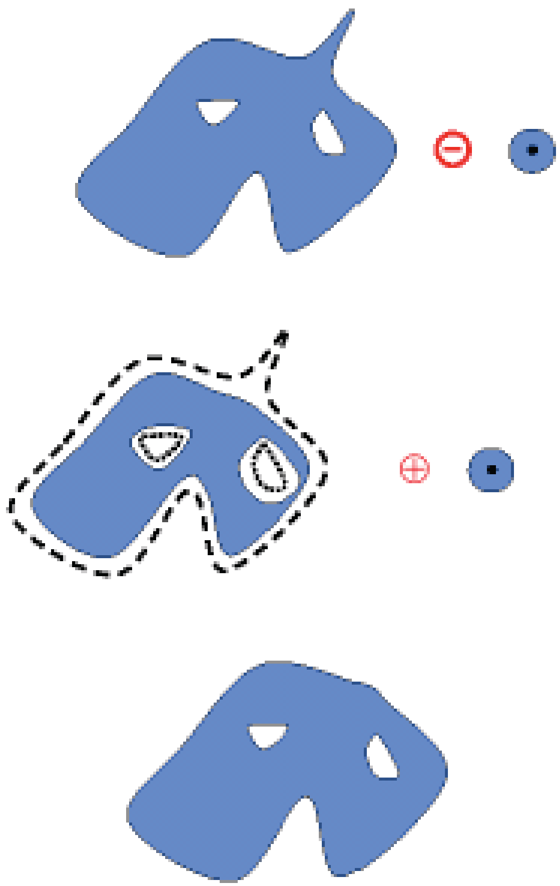
- Examples

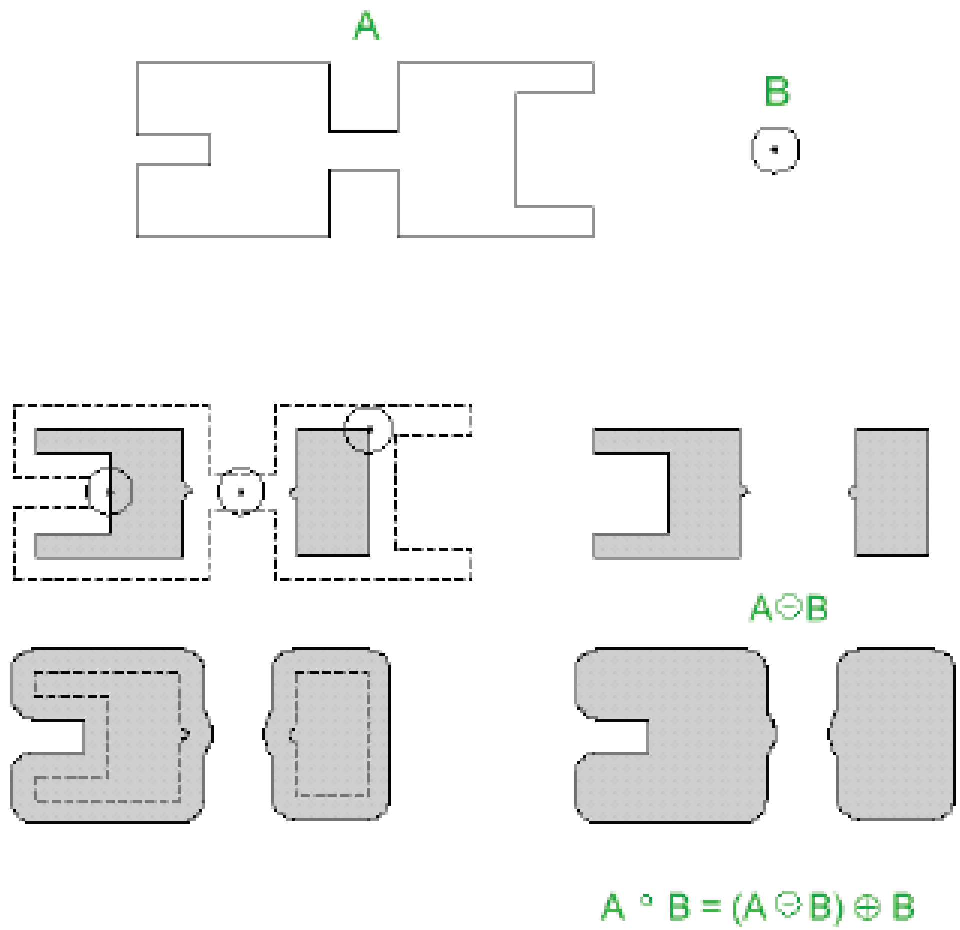




Basic operations: Opening

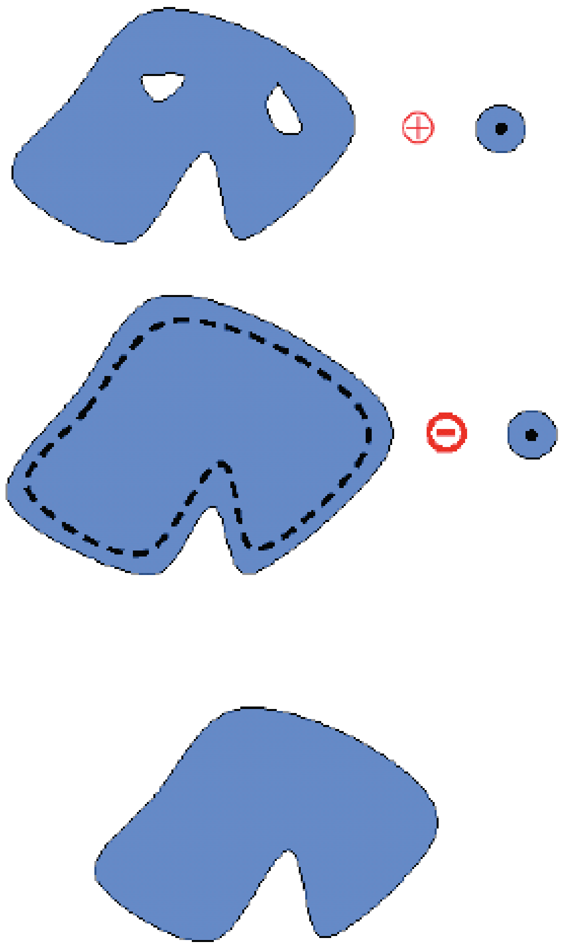
- A v B = (A - B) + B

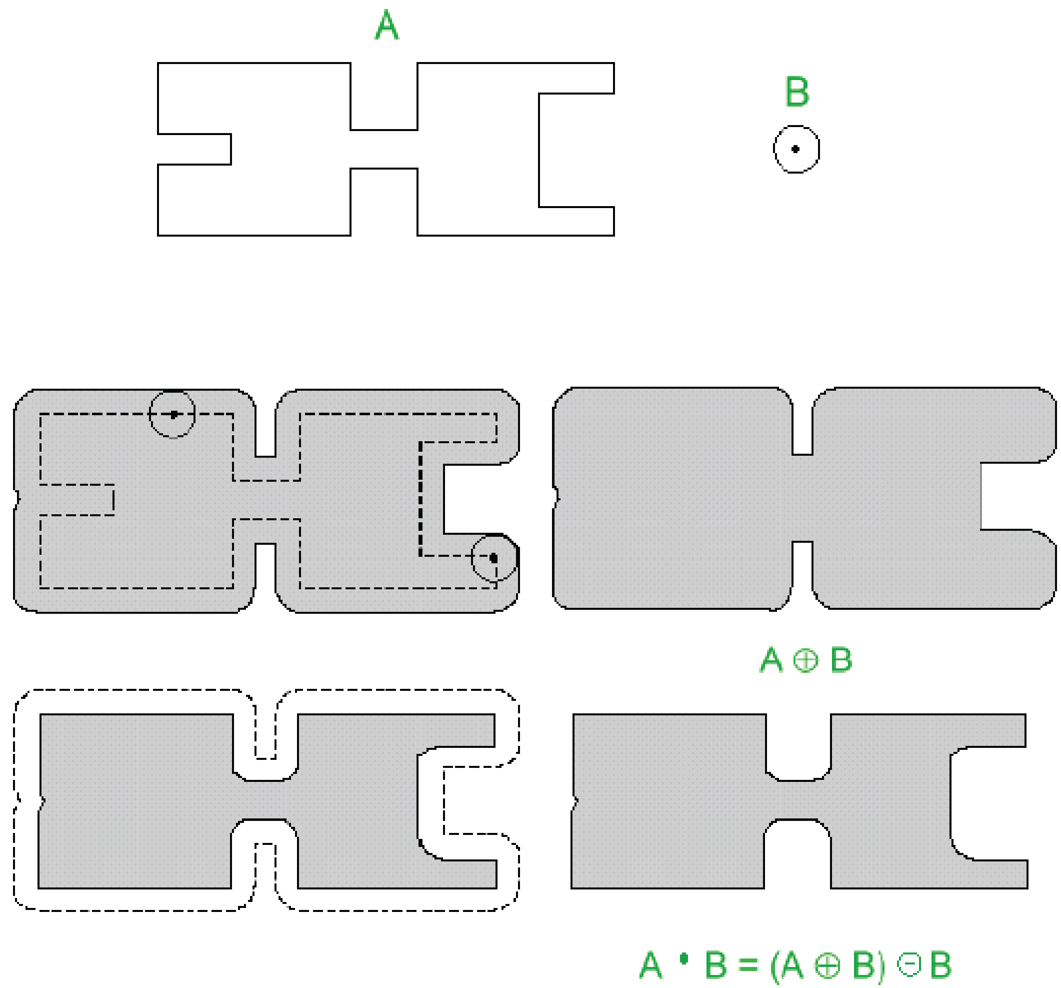




Basic operations: Closing

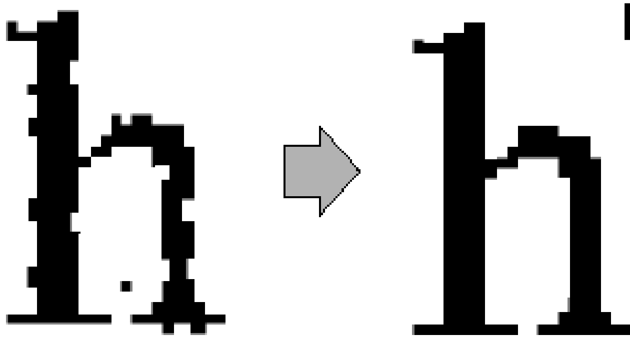
- A B = (A + B) - B





Basic operations: quality improvement

Using binary patterns

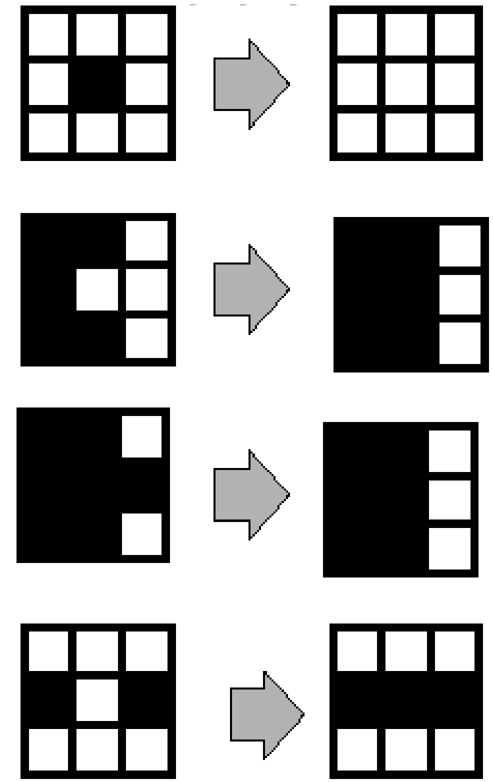


IF THEN

Noise removal

Hole filling

Connexity reparation



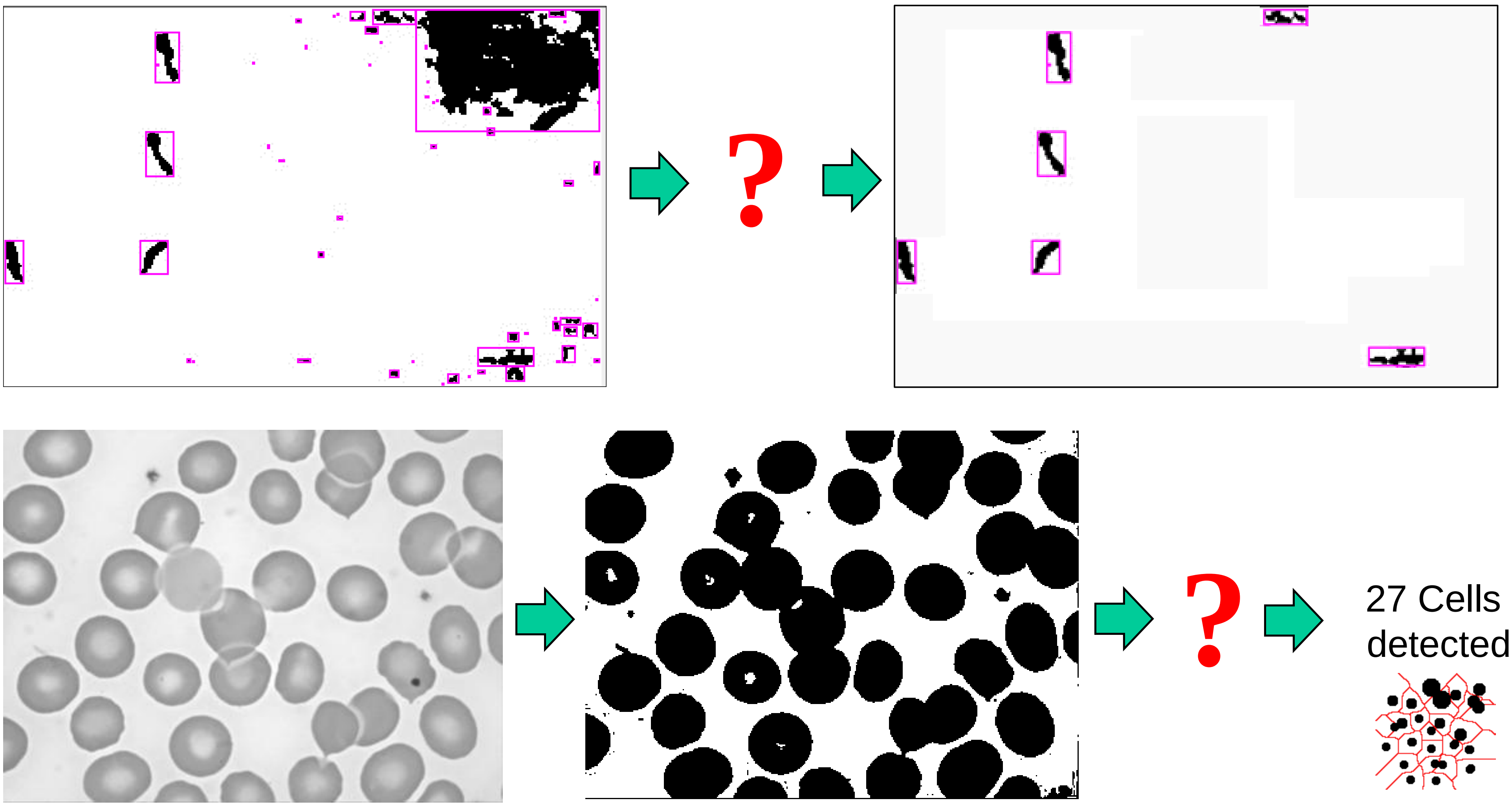
Ect…

Basic operations: quality improvement

Basic operations: quality improvement

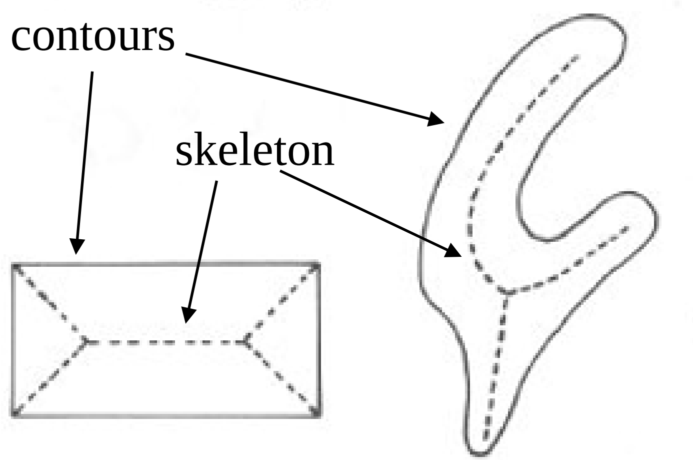
Basic operations: quality improvement

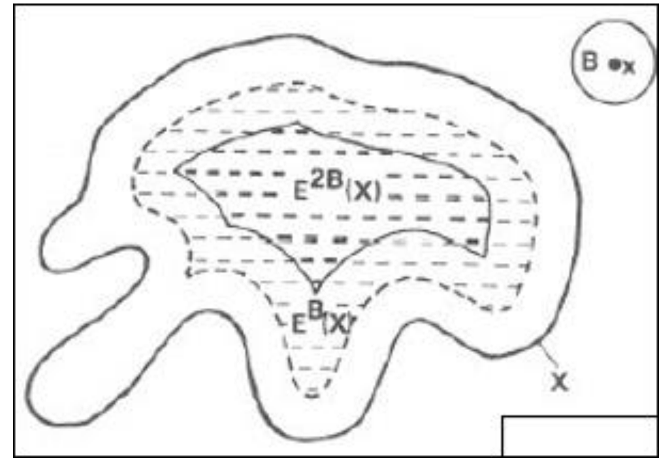
- Image analysis sequence ?

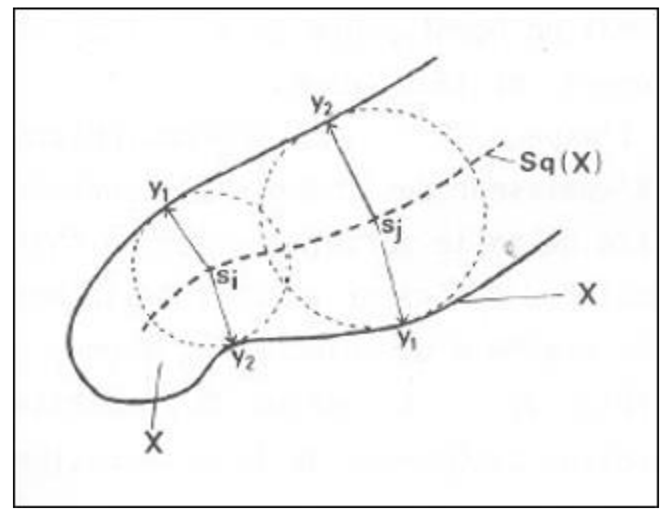


More sophisticate: Skeleton

- Skeleton is defined by the set of points located at equal distance from the border of the shape
- Union of the centers of the maximal spheres that can be included into the shape
- Computed by successive erosions

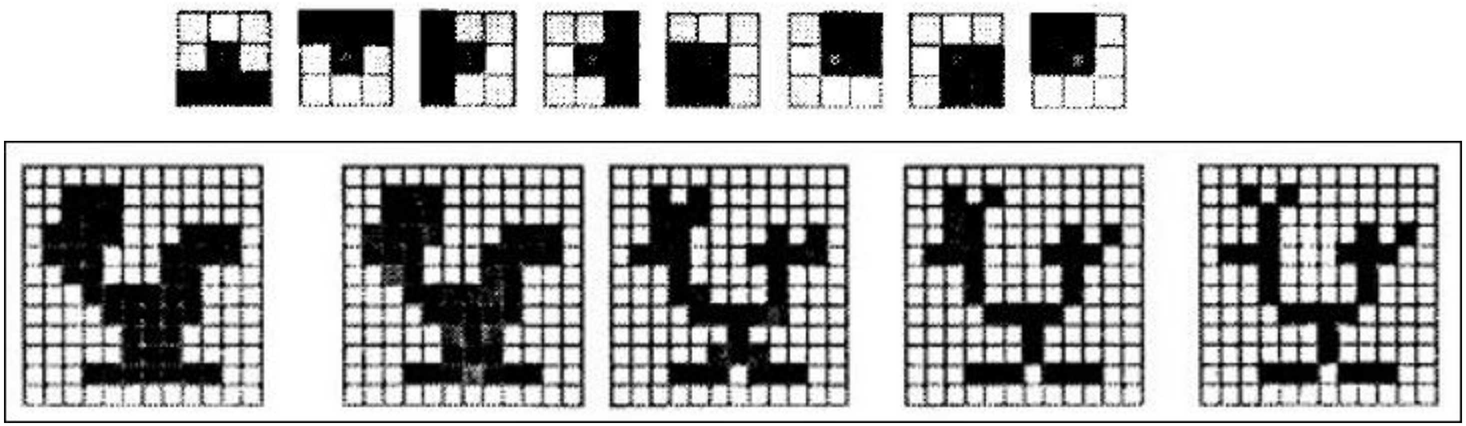




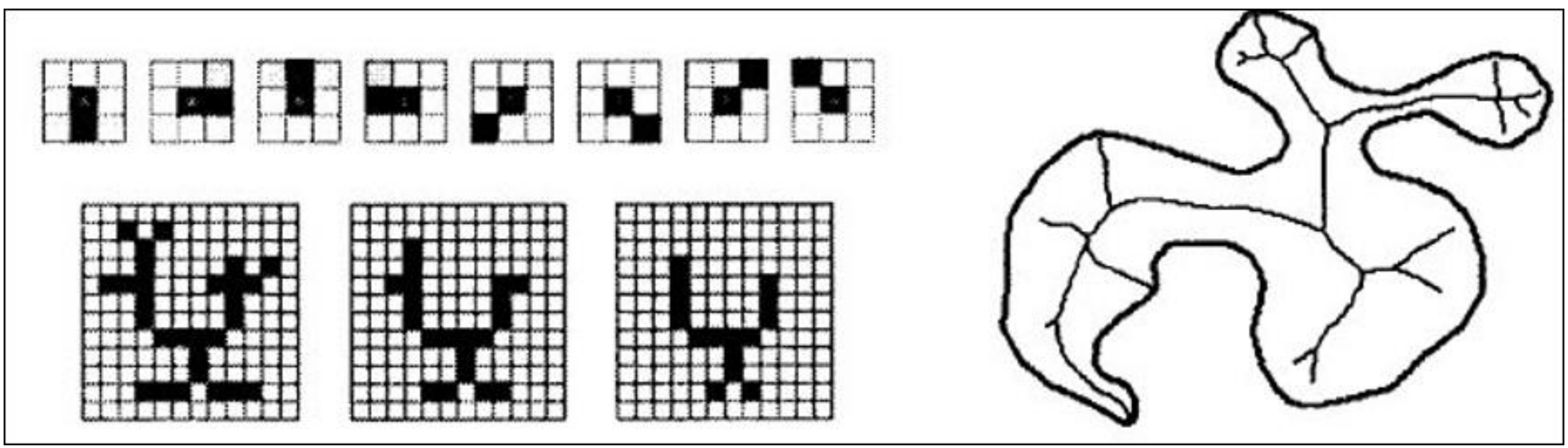


More sophisticate: Skeleton

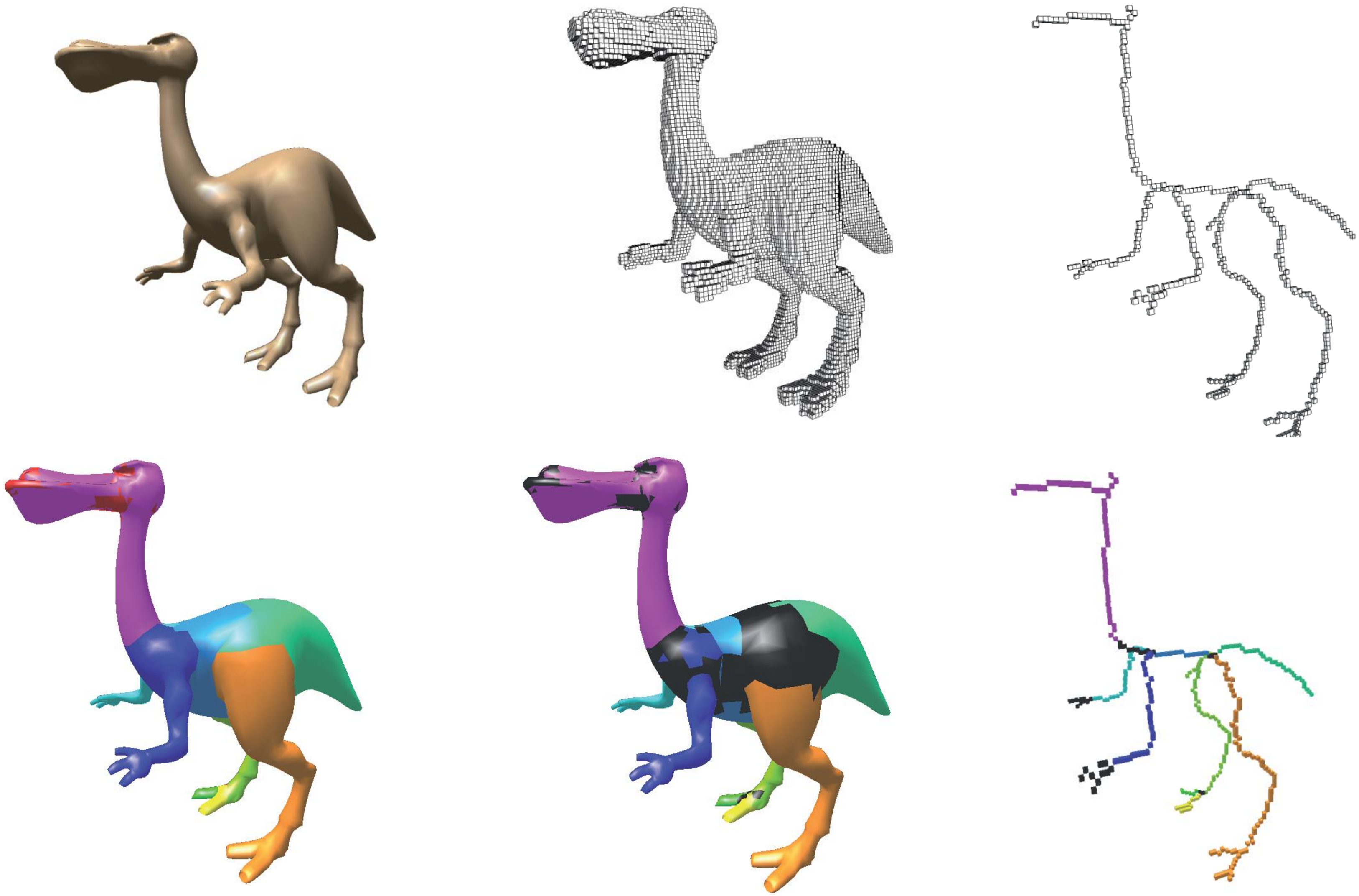
- Computation of the skeleton by successive erosions with different masks



- Small branches have to be removed by using specific masks

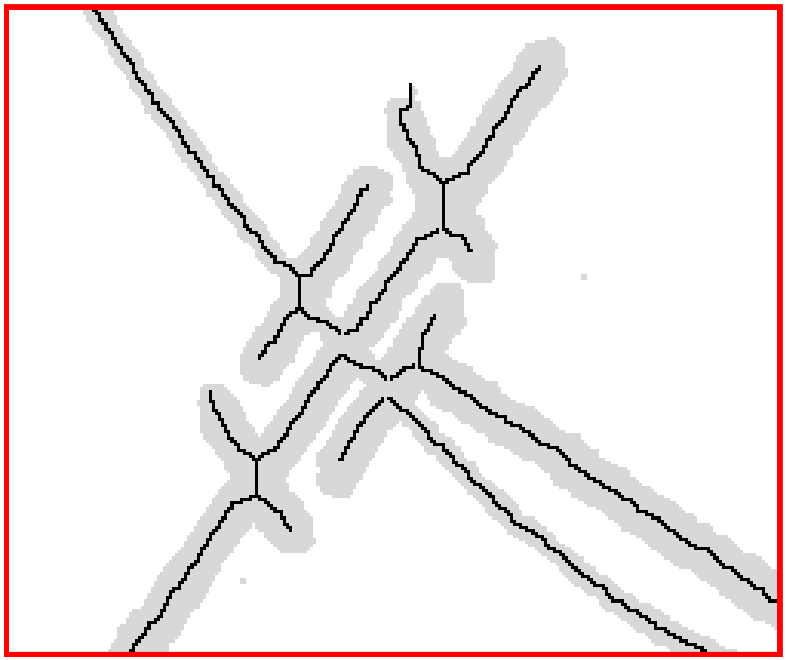


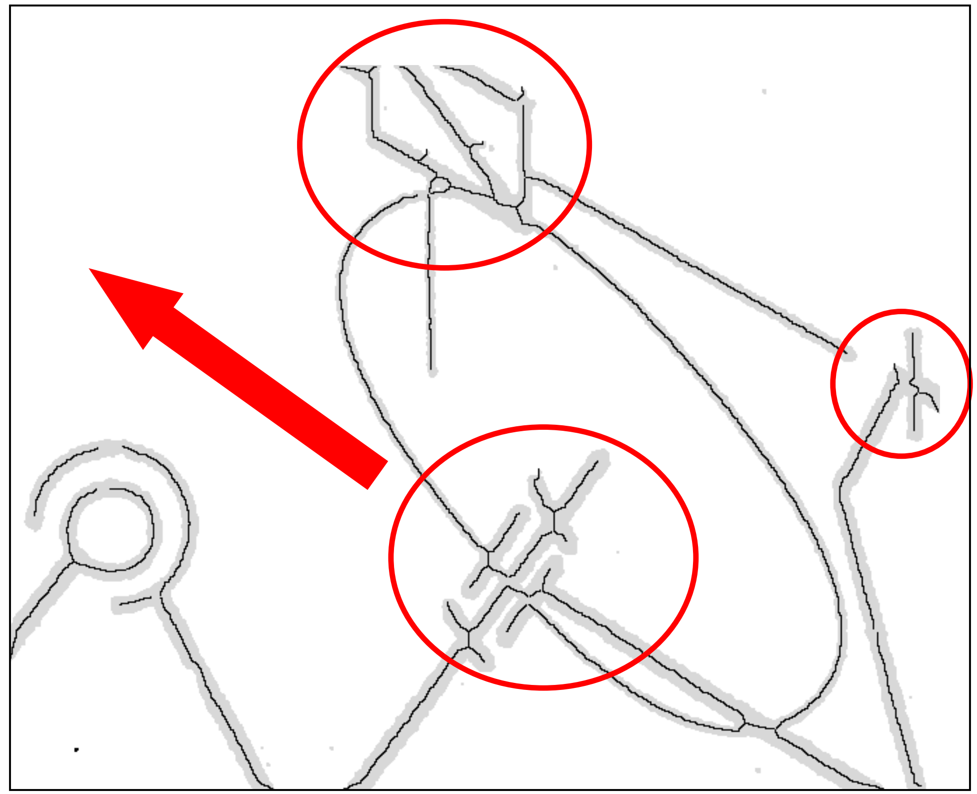
Possible also in 3D



But not so easy to analyse . . .

- Possible wrong representations of junctions and crossing (noise)
- Noise, barbules, …



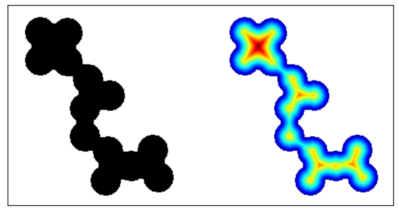


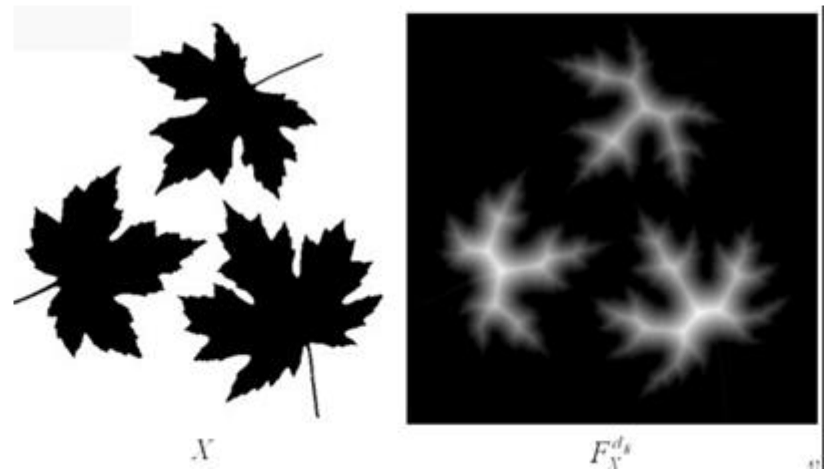
Distance map and Watershed

Euclidian Distance Map (EDM)

- Function that associated to each pixel the distance to the background pixels

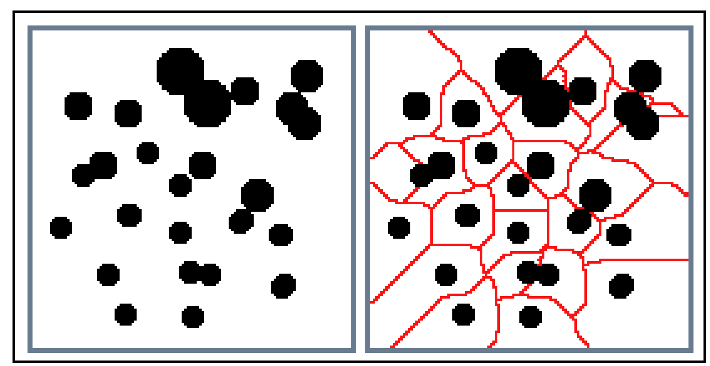
$$
\begin{array}{cccc}
F_X^d : & \mathbf{Z}^2 & \rightarrow & \mathbf{N} \\
 & p & \mapsto & d(p, X^c)
\end{array}
$$





Binary Watershed based on EDM

- Based on EDM, finds the ultimate eroded points (UEPs)
- Dilate each of the UEPs as far as possible
  - either until the edge of the particle is reached,
  - or the edge touches a region of another (growing) UEP.

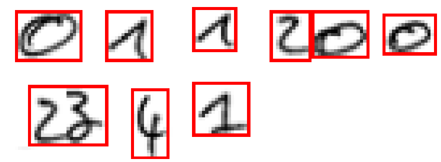


- Automatically cutting particles that touch
- Work best for smooth convex objects that don't overlap too much.

A full sample (simplified)

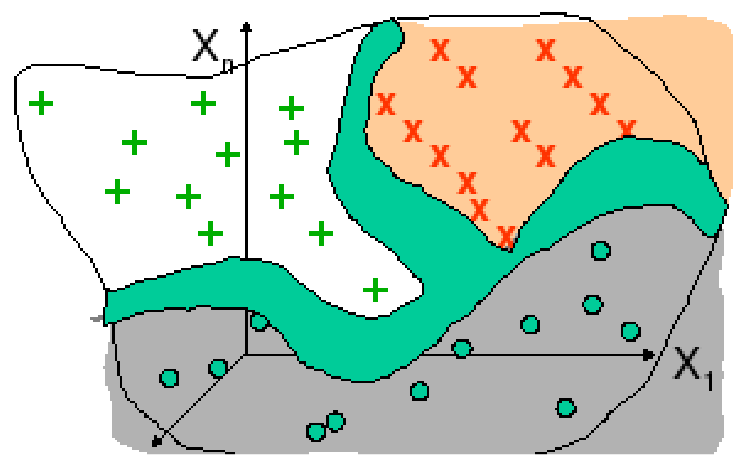
1. Image acquisition

2. Pre-processing :
   - Filtering, noise removal, scale selection, ...
   - Segmentation of the ROI (objects)



3. Feature extraction : building a vectorial representation of the objects
   - V(2:3;□ 2; 1000; 50; :::; 45) 
   - V(□ 3; 10:2; 0; 20; :::;□ 4; 5 

4. Classification : select a label for each object from the vectorial representation



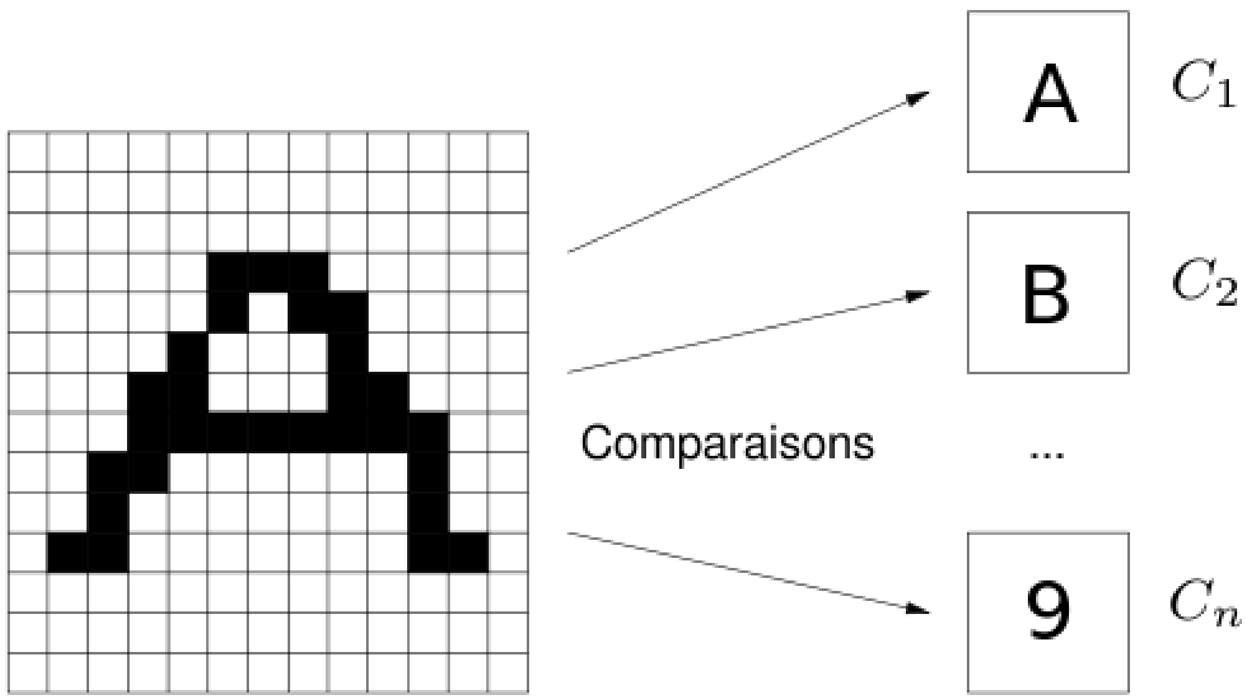
5. Post-processing : contextual verification to correct some errors

.

Feature selection (simplified)

Pixels can be considzered as features

- distance(C; C_i) = Σij | P (i; j) - P_i(i; j) |



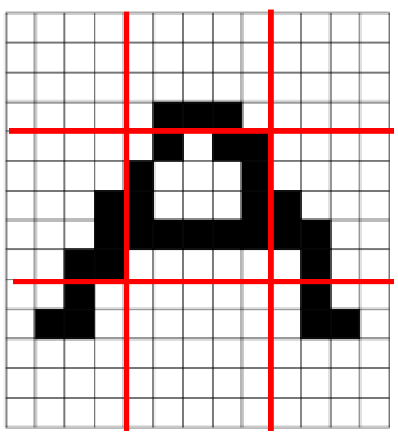
C : Unknow object

Training set (tagged models of object)

Feature selection (simplified)

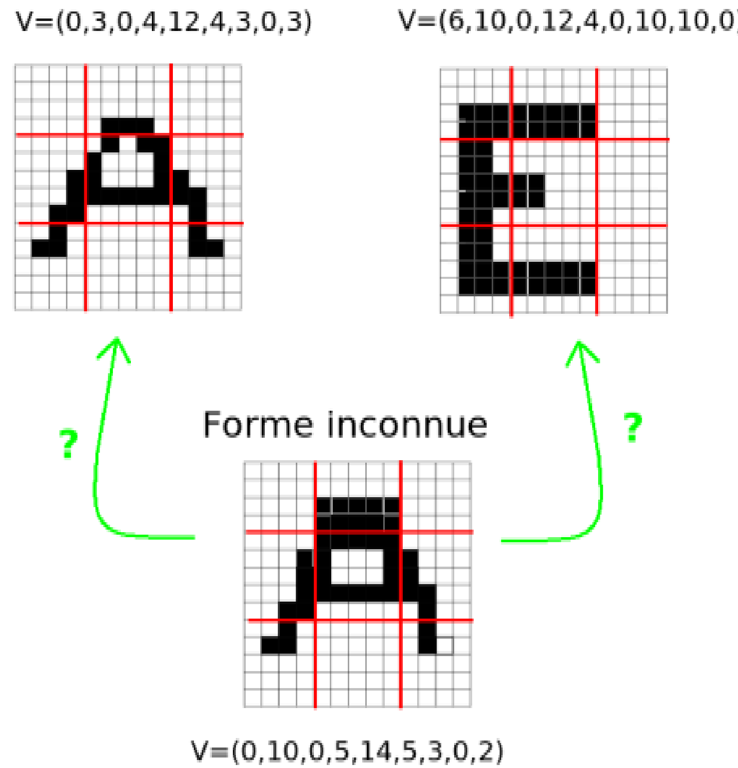
Zoning

- The image is splitted in n blocks
- For each block, some features are computed (number of black pixels)
- A new feature vector is obtained : $V = (Nb_1; Nb_2; :::; Nb_n)$



V=(0,3,0,4,12,4,3,0,3)

Classification (simplified)



The unknown object ? is identified as a « A » because min(D(A; ?);D(B; ?)) = D(A; ?)

$$D(A, ?) = \sqrt{(0-0)^2 + (3-10)^2 + (0-0)^2 + (4-5)^2 + \ldots + (3-2)^2}$$

$$D(A, ?) = 7,48 \text{ et } D(B, ?) = 19,05$$

That All for today…

Thank you

Support available

http://www.rfai.lifat.univ-tours.fr/PagesPerso/jyramel/PDF/MISS2019ramel.pdf

Approche Interactive

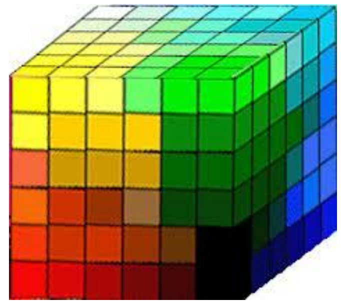
Modélisation du contenu de l'image

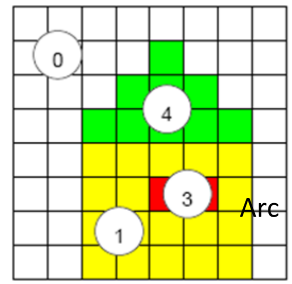
⇨ Structuration des données à l'aide d'un graphe d'adjacence

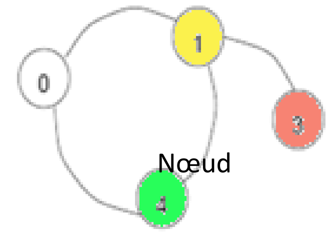
Nœud = Région

Attributs = Descripteurs de la région

- Liste des voxels de la région
- Position : G(x,y,z)
- Liste de caractéristiques F_j (les moyennes)
- Forme, couleur, modalités, …
- Label, annotation, …



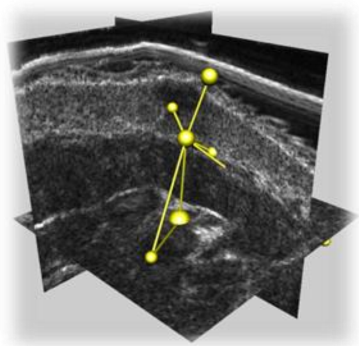


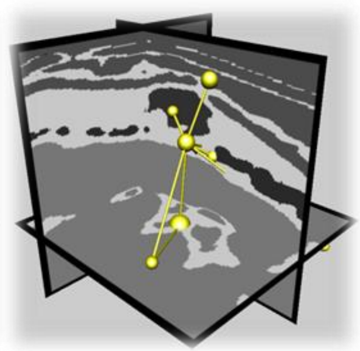


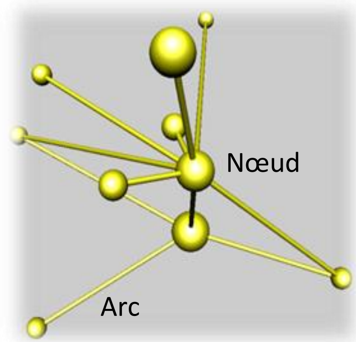
Arc = Relation entre Régions

Attributs = Descripteurs des relations

- Les 2 nœuds liés
- Aire de la surface de contact
- Type de relation, distance G_1, G_2
- Taux de ressemblance, …
- Label, annotation, …



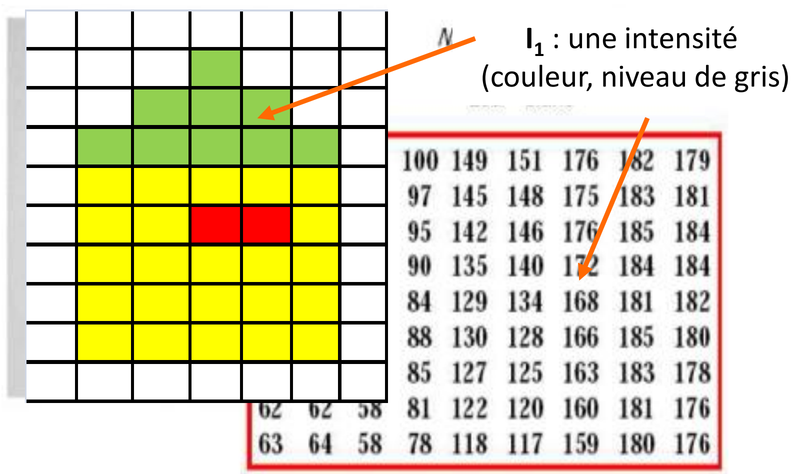




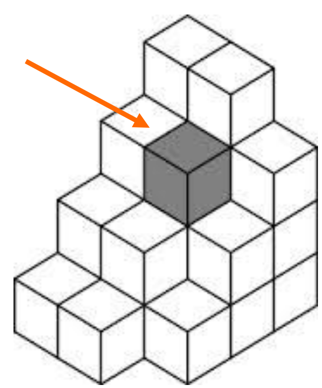
Approche Interactive

Modélisation du contenu des images

⇨ Caractérisation des contenus



Approche classique



1 voxel = 1 valeur

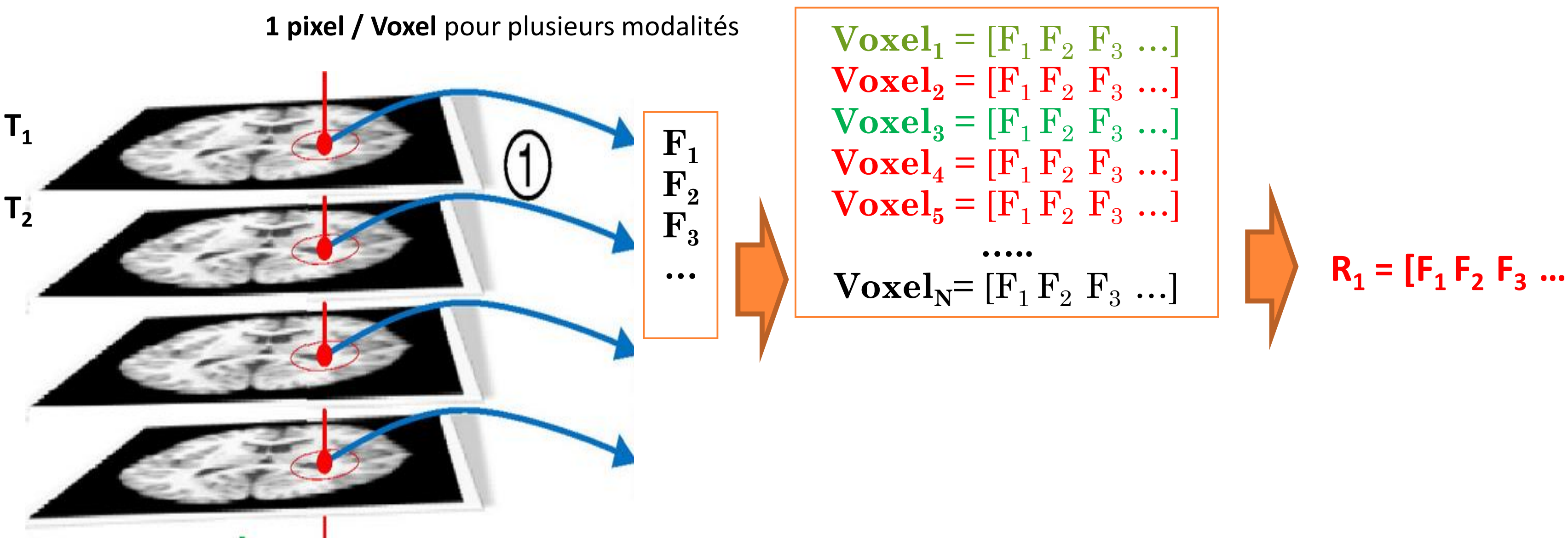
Caractérisation des voxels

- Recalage des acquisitions multimodales (T1, T2, …)
- 1 voxel = 1 liste de caractéristiques F_j plutôt qu'une couleur
- 1 voxel = 1 individu ⇨ **Voxel**$_i$ = [F_1 F_2 F_3 …]

Approche interactive

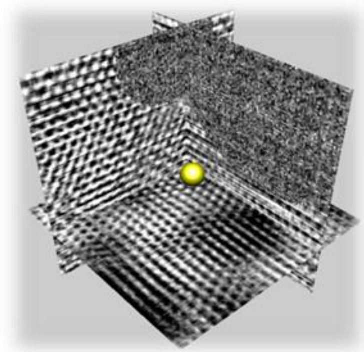
Caractérisation des régions

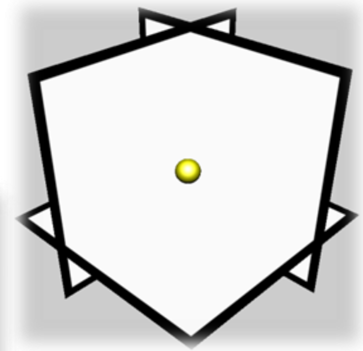
- Segmentation = regroupement des individus similaires en classes
- Perte de l'information de connexité (vs approche région, contours)
- Caractéristiques Région = Moyenne des caractéristiques des Voxels appartenant à la région

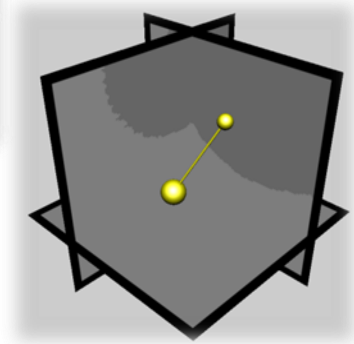


Approche Interactive

Segmentation interactive du contenu de l'image
⇨ Transformation successive du graphe (RAG)







Opération de Division

Fonction *Seg* correspond à un *K-means*

- Rapidité d'exécution
- Faible coût en mémoire
- Partitionnement efficace

Nouveau graphe :

$$G_{k+1} = Kmeans(U_{op}.V, G_k, F, U_{Seg})$$

U_{Seg} : *nombre de régions*

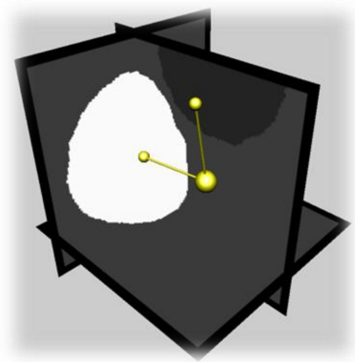
F : *caractéristiques choisies par l'utilisateur (attributs des nœuds)*

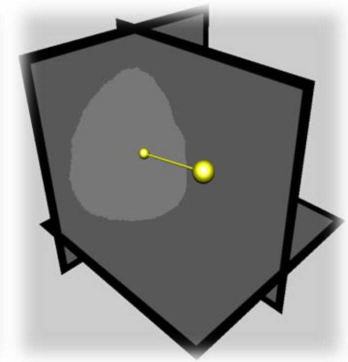
$U_{op}.V$: *identifiant du nœud à diviser*

Approche Interactive

Segmentation interactive du contenu de l'image

➔ Transformations successives du graphe (RAG)





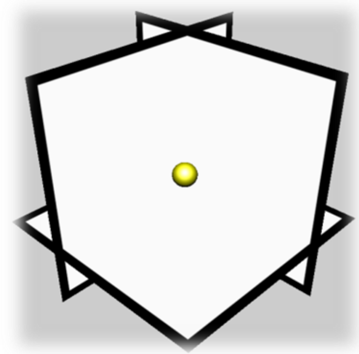
Opération de Fusion

Simple fonction de fusion *Merge*

- Nouveau graphe : $G_{k+1} = Merge(U_{op}.E, G_k)$

- Initialisation des attributs de V_{new} durant la fusion de V_1 et V_2

$$V_{new}.T_i = \frac{(V_1.T_i)(V_1.NV) + (V_2.T_i)(V_2.NV)}{V_1.NV + V_2.NV}$$



NV : nombre de voxels identifié par un nœud
$T = \{\bar{F}, \bar{G}\}$

Approche Interactive

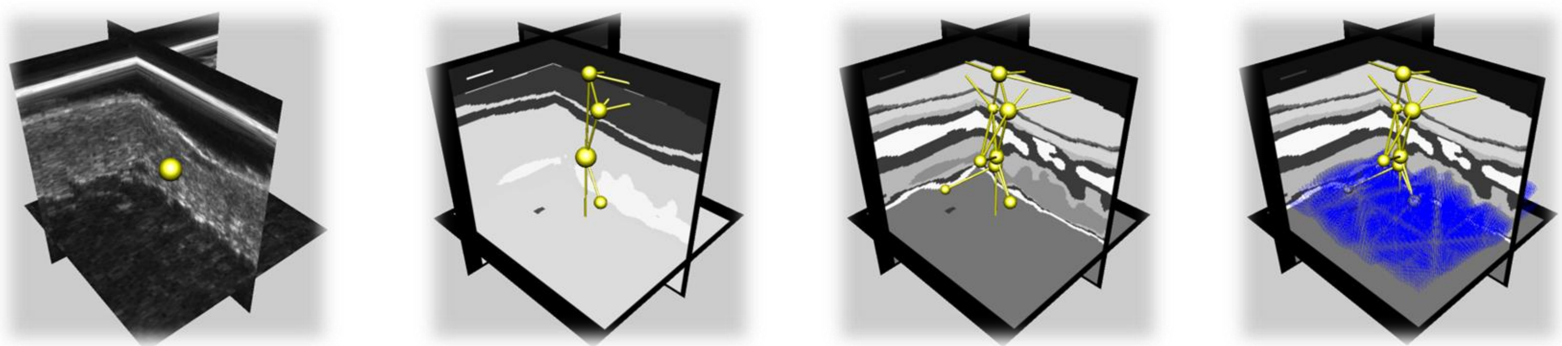
Segmentation interactive du contenu de l'image

➔ Transformations successives du graphe (RAG)

Initialisation

- GAR initialisé à partir d'un unique nœud = 1 région unique
- 1ère étape du processus de segmentation : Division
- Libre choix des opérations par la suite (Fusion, Division, Etiquetage, …)

Construction de scénarios de segmentation interactive

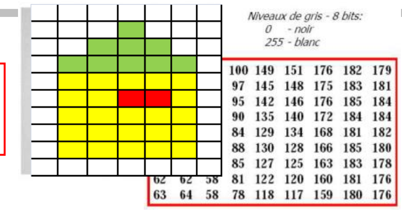


Autres opérations possibles : Etiquetage, simplification, suppression, …

ANALYSE D'IMAGES

Acquisition

Images, signaux 1D 2D 3D

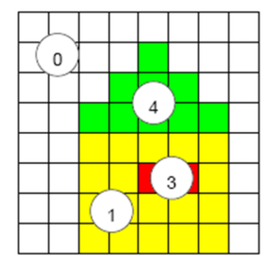


Prétraitements

Images, signaux améliorés

Segmentation / Description / Représentation

Régions d'intérêt, primitives de description



Caractérisation

Experts

Utilisateurs

Connaissances a priori

RECONNAISSANCE DES FORMES