EPFL

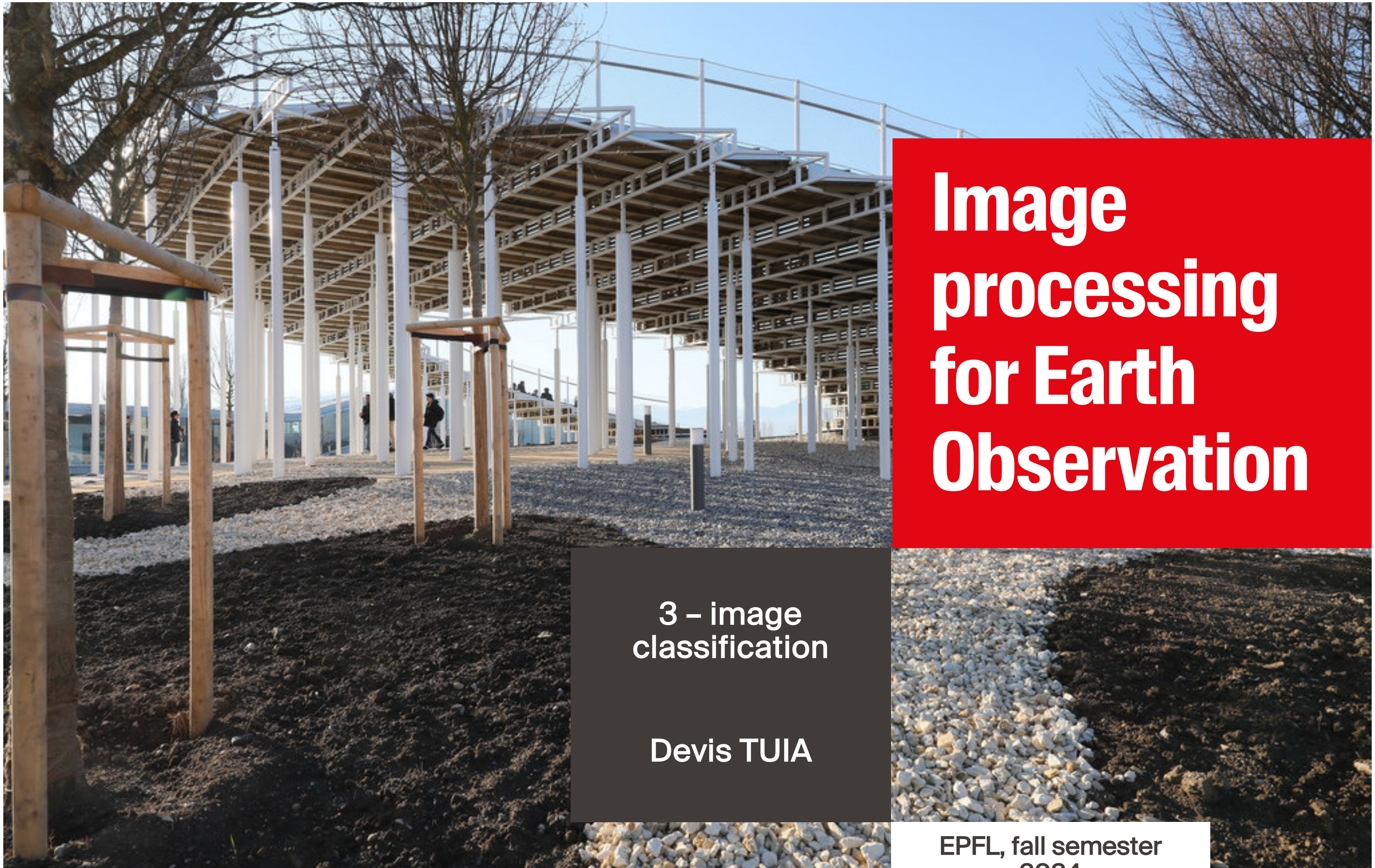


Image processing for Earth Observation

3 – image classification

Devis TUIA

EPFL, fall semester 2024

École polytechnique fédérale de Lausanne

Content (6 weeks)

- W1 General concepts of image classification / segmentation
  Traditional supervised classification methods (RF)
- W2 Traditional supervised classification methods (SVM)
  Accuracy measures

- W3 Elements of neural networks
- W4 Convolutional neural networks
- W5 Convolutional neural networks for semantic segmentation
- W6 Sequence modeling, change detection

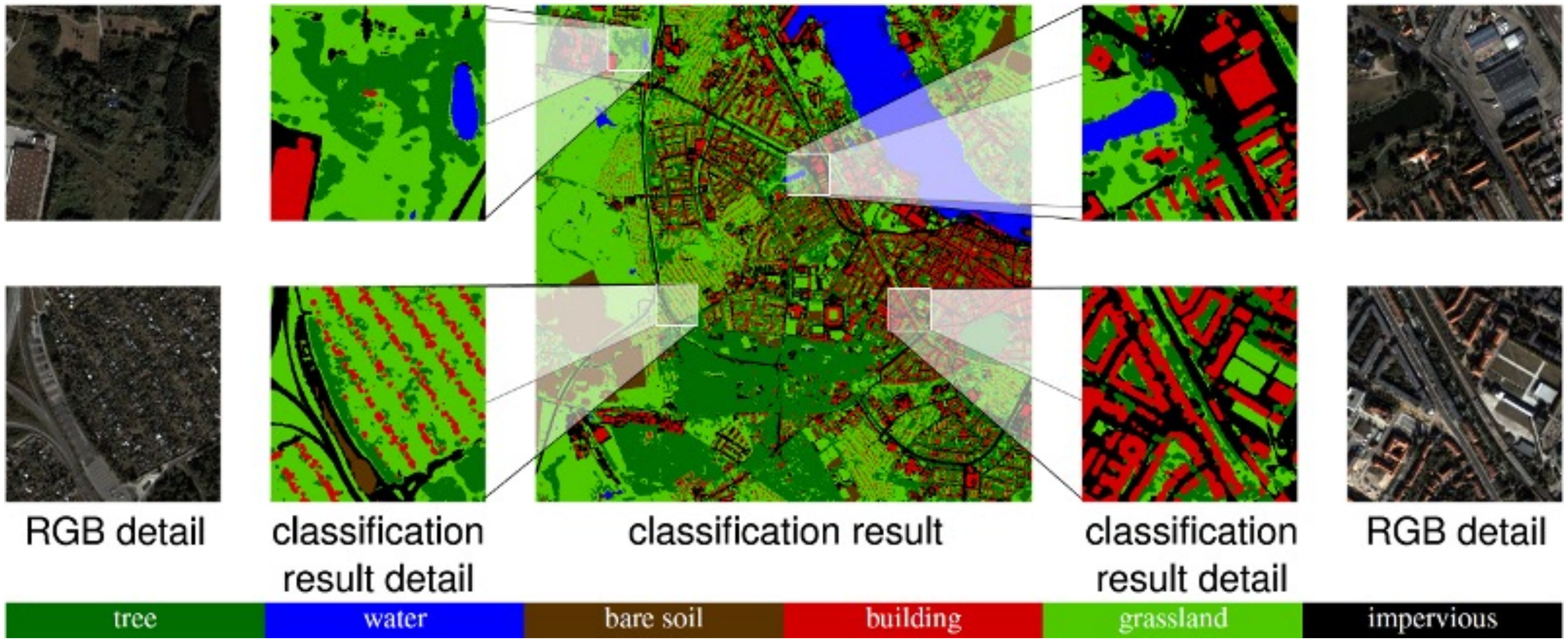
What do you see in this image?

MRSC-21 dataset
(http://rodrigob.github.io/are_we_there_yet/build/semantic_labeling_datasets_results.html)

- Your brain detects objects naturally
- It was trained to do it
- It does it automatically

- It also provides you with a semantic interpretation, a class

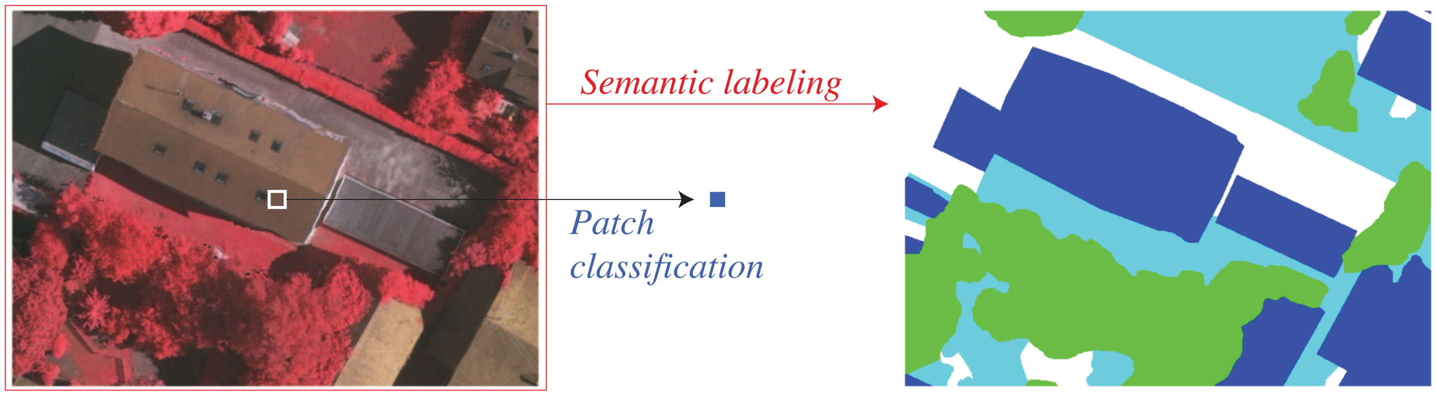
We want to achieve the same with Earth observation images, automatically



Source: inf-cv.uni-jena.de

pixel classification vs semantic segmentation

- Classifiers so far predict one value per unit of support (pixel, patch, …)
- When we predict per pixel, we do semantic segmentation (or labeling)



M. Volpi and D. Tuia. Dense semantic labeling of subdecimeter resolution images with convolutional neural networks. *IEEE Trans. Geosci. Remote Sens.*, 55(2):881–893, 2017.

Did you say classification / semantic seg.?

- In many applications, the complexity of the image information content has to be reduced
- A certain generalization can provide a clearer information
- In our case, each pixel is:

$$\text{image} \rightarrow \text{classification map}$$

$$R^b \qquad\qquad N$$

Real numbers in b dimensions

Integer numbers

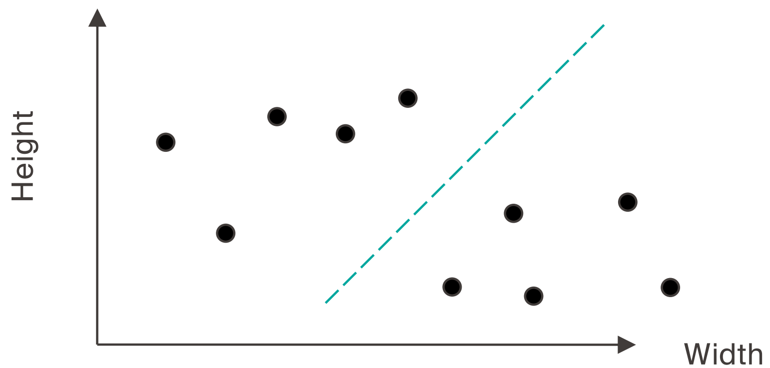
Did you say classification / semantic seg.?

- We want to reduce each pixel, being a multidimensional information, into a single value corresponding to a class.
- The output is a thematic map
- A class can be
  - A land use type
  - A land cover type
  - A level of damage
  - A type of change
  - …

A taxonomy

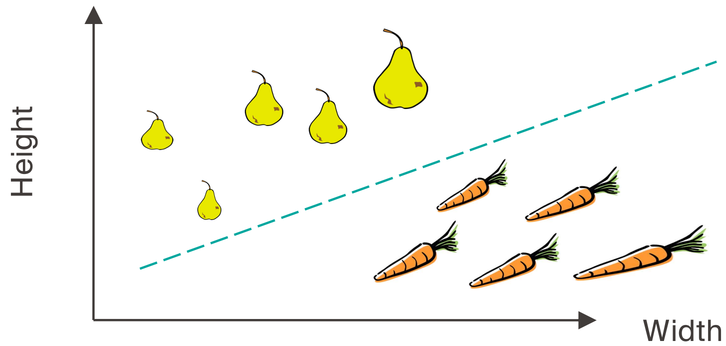
Covered in this course

Unsupervised
- Class information is not available
- Decision function defined by data similarity only



- Clustering
- Segmentation
- Saliency detection
- ...

Supervised
- Class information is provided by examples from the user
- User intervention!



- Supervised classification
- Semantic segmentation
- Object detection
- Instance segmentation
- ...

But not manually!

- We want this to be automatic (as much as possible)
- Interpreting manually the entire image is too time consuming and costly
- So we have to teach the machine how to do it
  - How to recognize similar objects?
  - Easy for us, we have a fantastic processing unit (our brain)
  - For a machine, a pixel is only a vector of numerical values.

Machine learning !

The truth behind images

For you

100	134	79	191	78	3	34	12	132	62

100	134	79	191	78	3	34	12	132	62

100	134	79	191	78	3	34	12	132	62
23	121	78	156	100	21	22	134	156	100
29	34	90	121	121	231	134	99	189	86
12	127	95	232	23	230	212	90	198	67
11	134	78	132	100	233	245	78	212	34
190	156	121	145	77	230	7	56	200	230
189	234	231	111	78	34	8	3	255	100
40	45	67	89	245	234	222	21	245	1

OR

100	100	100
23	24	101
29	151	123
12	165	111

⋮

100	76	212
1	24	156

For the computer

The truth behind images

For you

100	134	79	191	78	3	34	12	132	62

100	134	79	191	78	3	34	12	132	62

100	134	79	191	78	3	34	12	132	62
23	121	78	156	100	21	22	134	156	100
29	34	90	121	121	231	134	99	189	86
12	127	95	232	23	230	212	90	198	67
11	134	78	132	100	233	245	78	212	34
190	156	121	145	77	230	7	56	200	230
189	234	231	111	78	34	8	3	255	100
40	45	67	89	245	234	222	21	245	1

OR

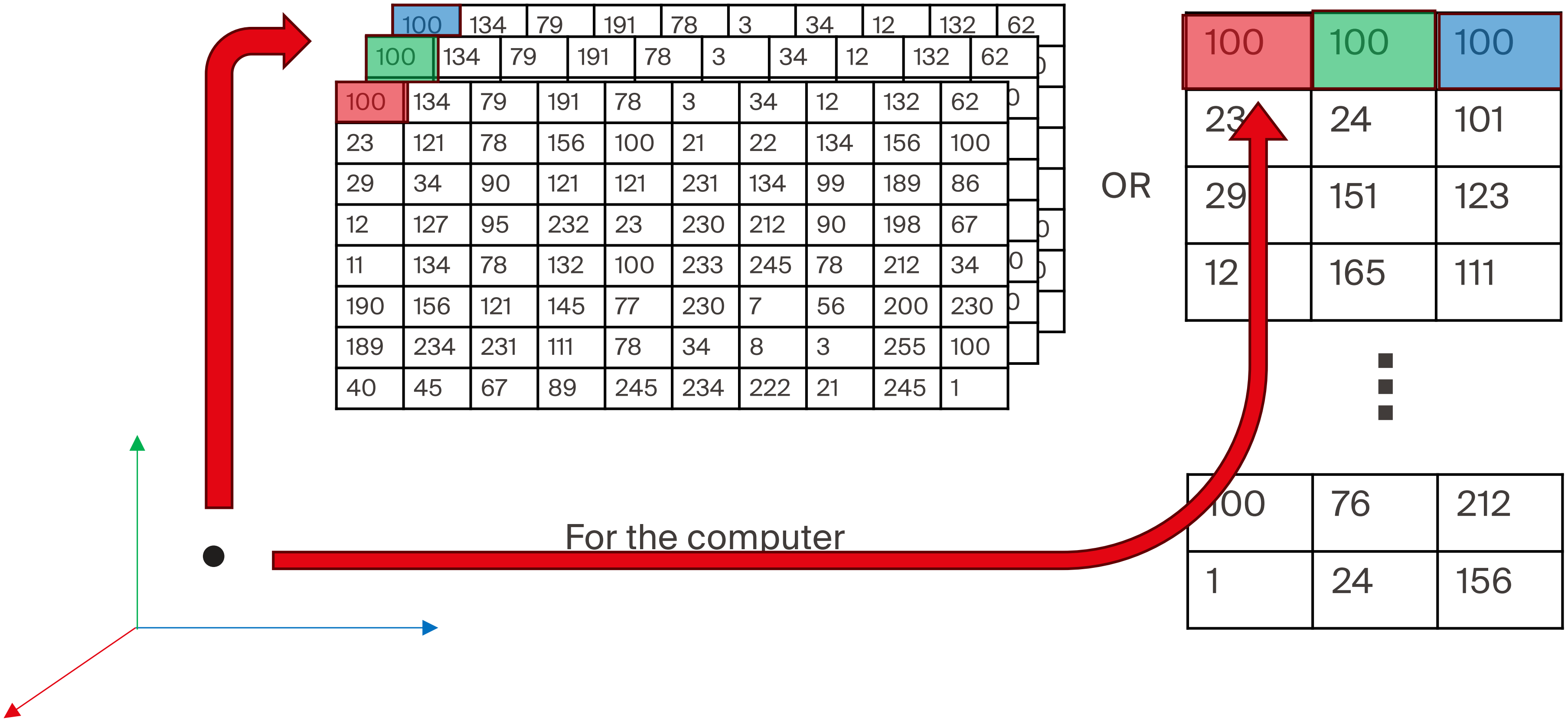
100	100	100
23	24	101
29	151	123
12	165	111

⋮

100	76	212
1	24	156

For the computer

The truth behind images



Machine learning?

- ML has been defined as "*learning from data*".

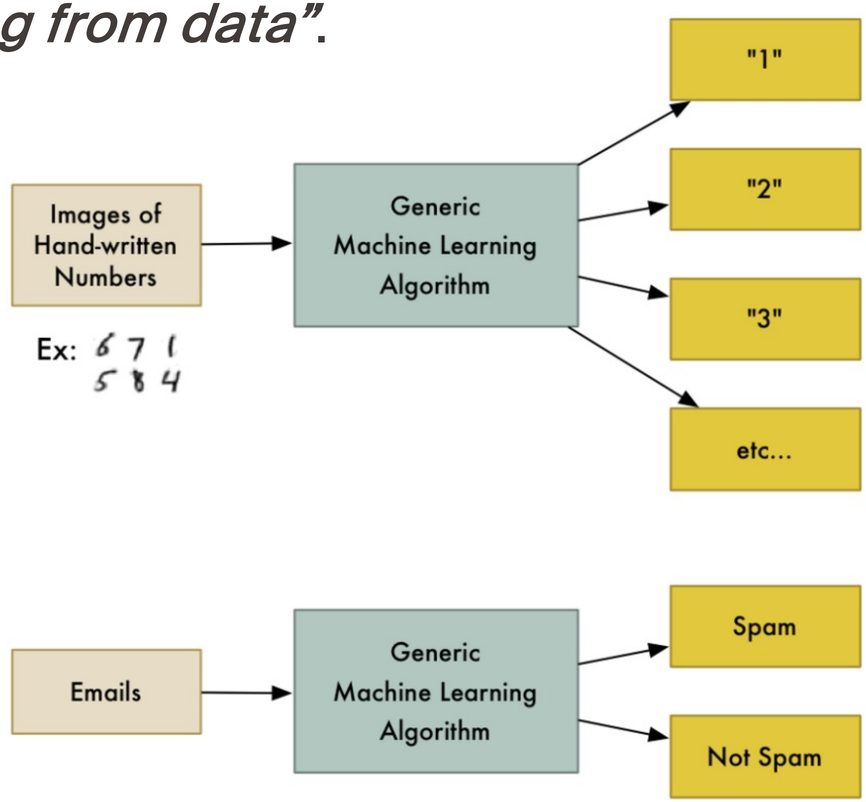
- **Learning how?**

  With generic algorithms

- We don't want to write specific code
- We want to feed data to the generic algorithm
- We leave the algorithm build its own logic linking inputs and the output (.... and then improve it with specific knowledge)

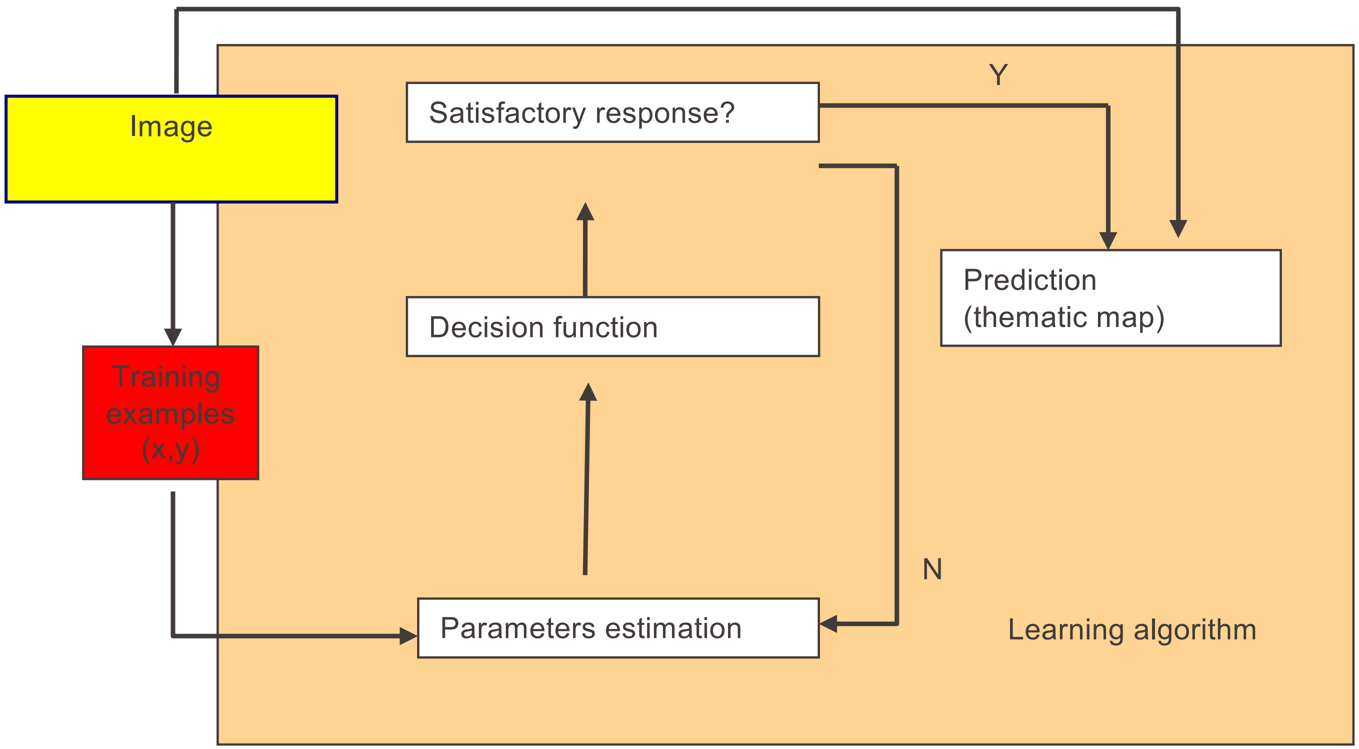
Machine learning?

- ML has been defined as "*learning from data*".
- **Learning how?**
  With <u>generic</u> algorithms



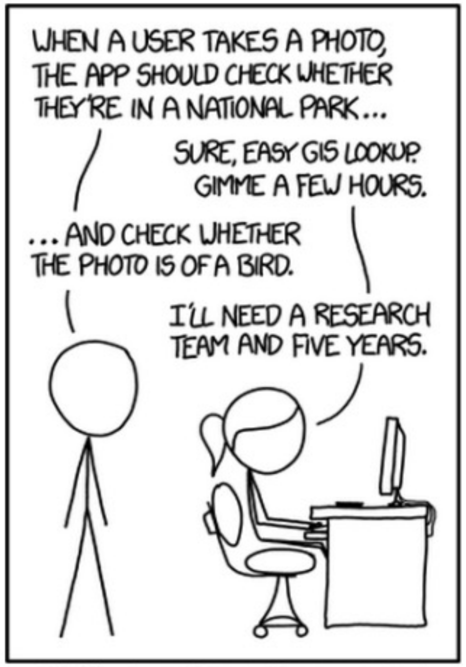
Source: https://medium.com/@ageitgey/machine-learning-is-fun-80ea3ec3c471#.7x1l6rhtr

Our generic algorithm for Supervised classification



Machine learning?

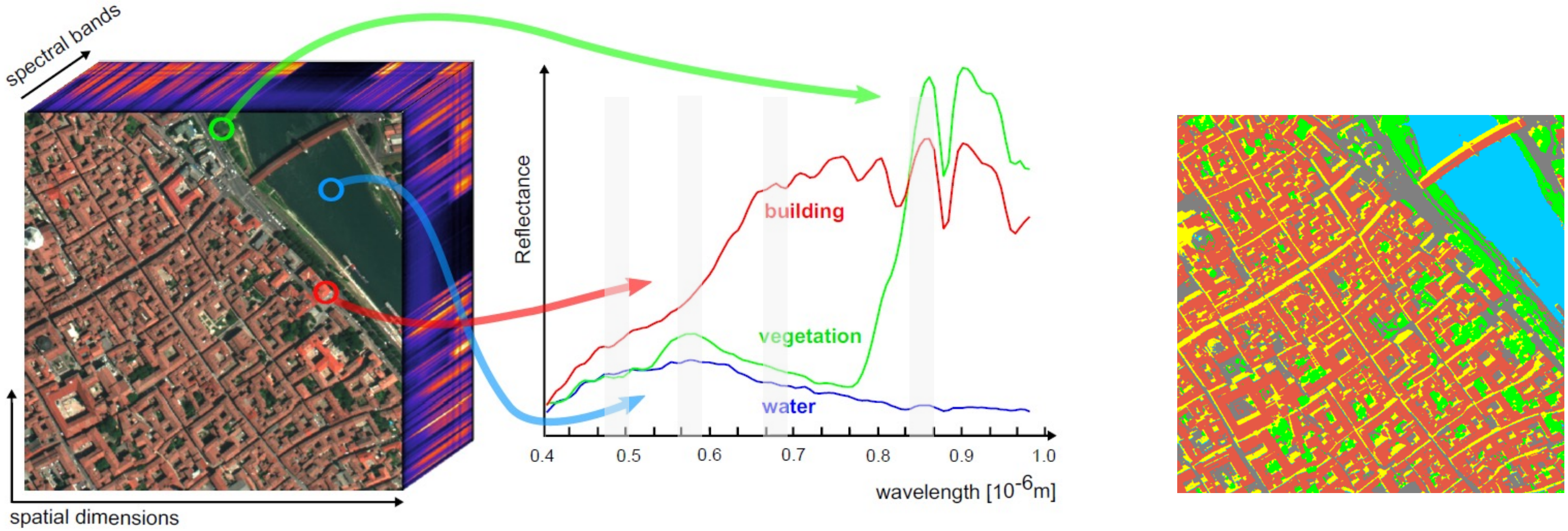
- ML has been defined as "*learning from data*".
- **Learning how?**
  With generic algorithms
- Does it work all the time?
  No. It's not magic.
- It works if we have
  - the right inputs
  - the right learning machine
  - sufficient training data



IN CS, IT CAN BE HARD TO EXPLAIN THE DIFFERENCE BETWEEN THE EASY AND THE VIRTUALLY IMPOSSIBLE.

Source xkcd

The three steps of classification



1. Selection of training examples

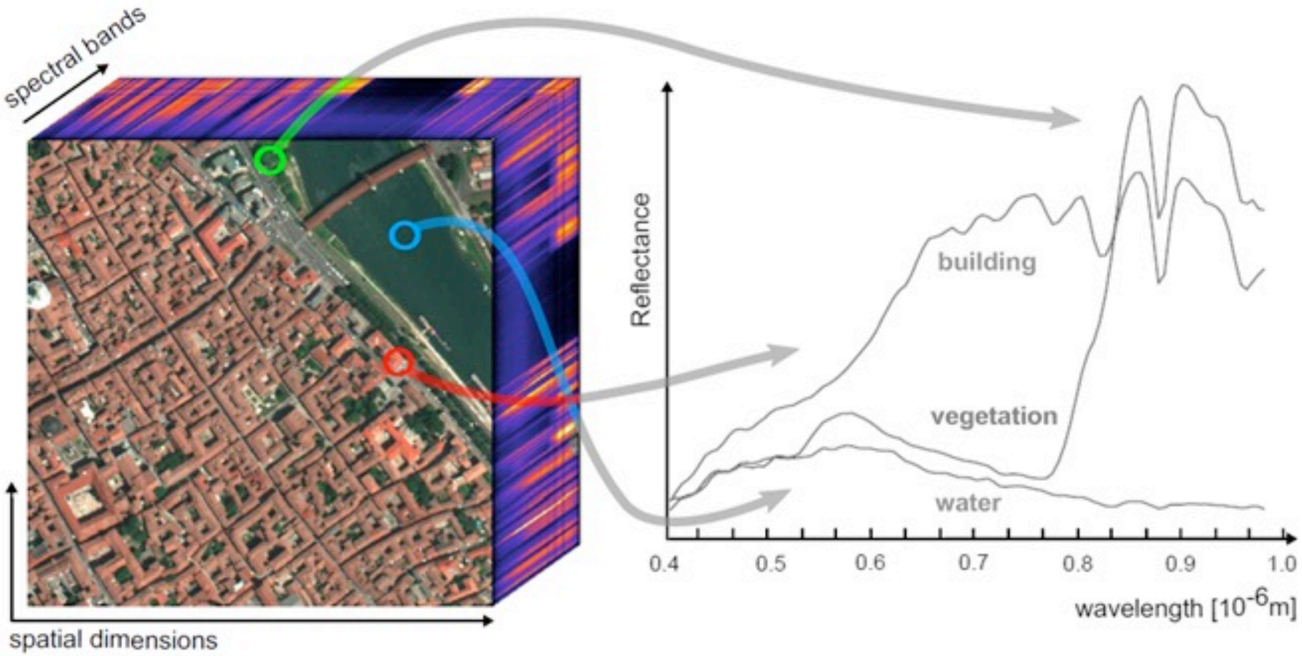
2. Establishment of a model

3. Prediction on the grid (image)

Learning phase

Prediction phase

The three steps of classification



1. Selection of training examples

2. Establishment of a model

3. Prediction on the grid (image)

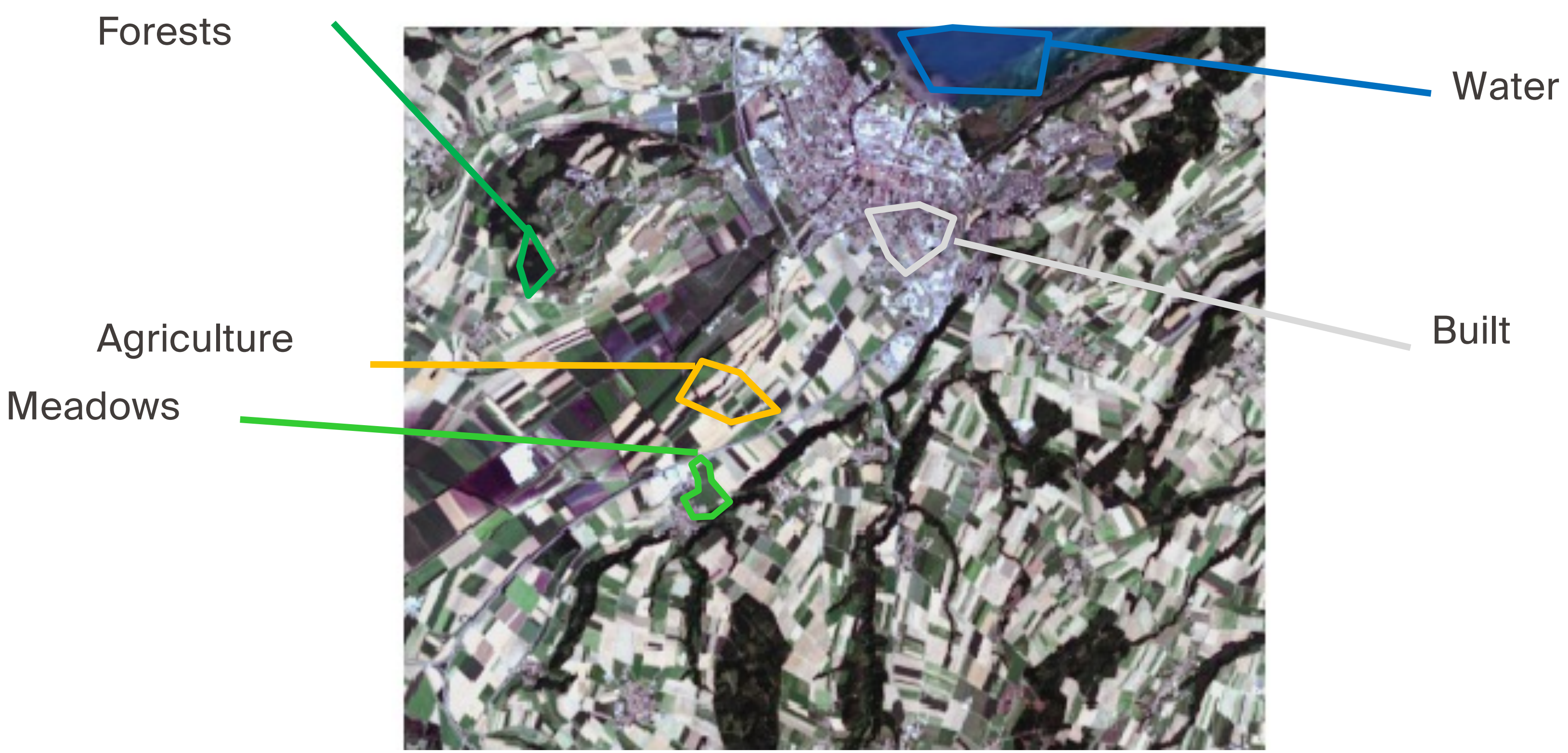
Selecting good examples

- A model “learns” data dependencies from examples
- Examples, also called training data, are
  - Given by a user (supervised classification)
    - From image interpretation
    - From in-situ groundtruthing

  - Drawn from the image by a given criterion (e.g. randomly, samples in low density regions, etc.)

EPFL

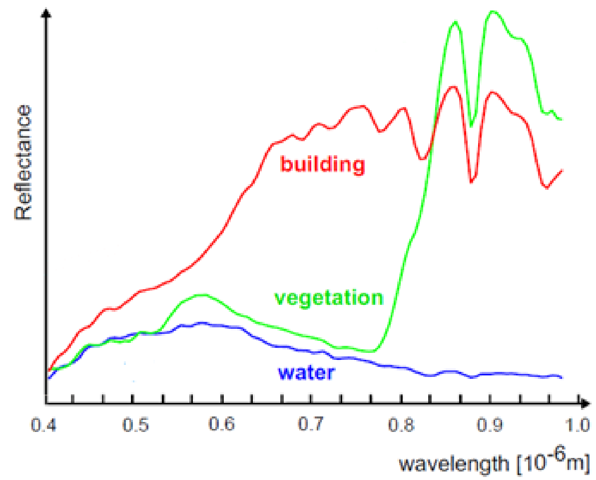
Selection of training examples by the user



Spectral signatures

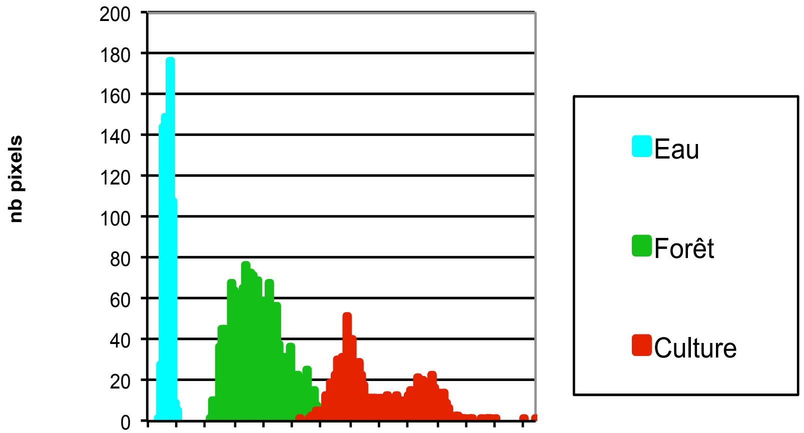
- The spectral signature is the response of a type of surface (a class) in radiance or reflectance
- We want it to be
  - Representative of the class
  - Different from the other classes

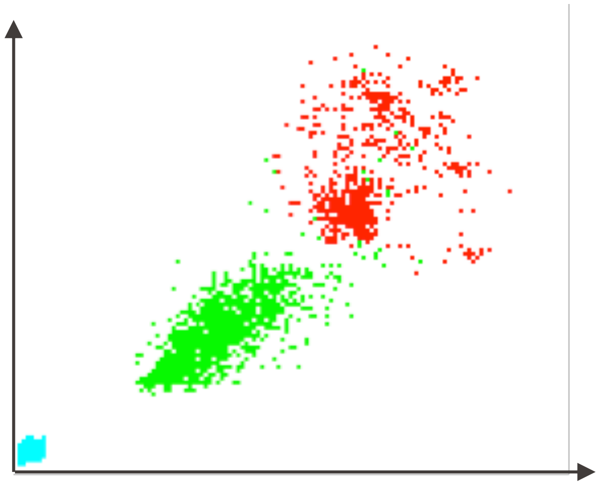
- In the following we discuss everything at the pixel level, but all kind of descriptors can be used (see course on spatial info)



Discriminative signatures

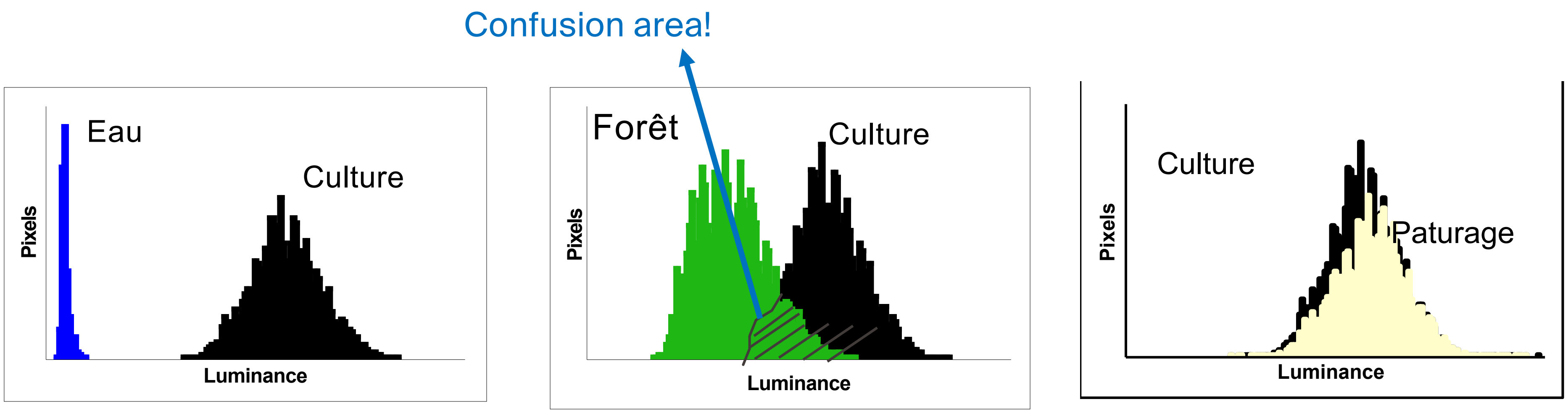
- Here is the distribution of three classes, in one (left) and two (right) dimensions





Discriminative signatures

- The samples selected and the bands used must be discriminative for the problem at hand!
- Below examples of a band and four different thematic classes



Discriminative

Partially discriminative

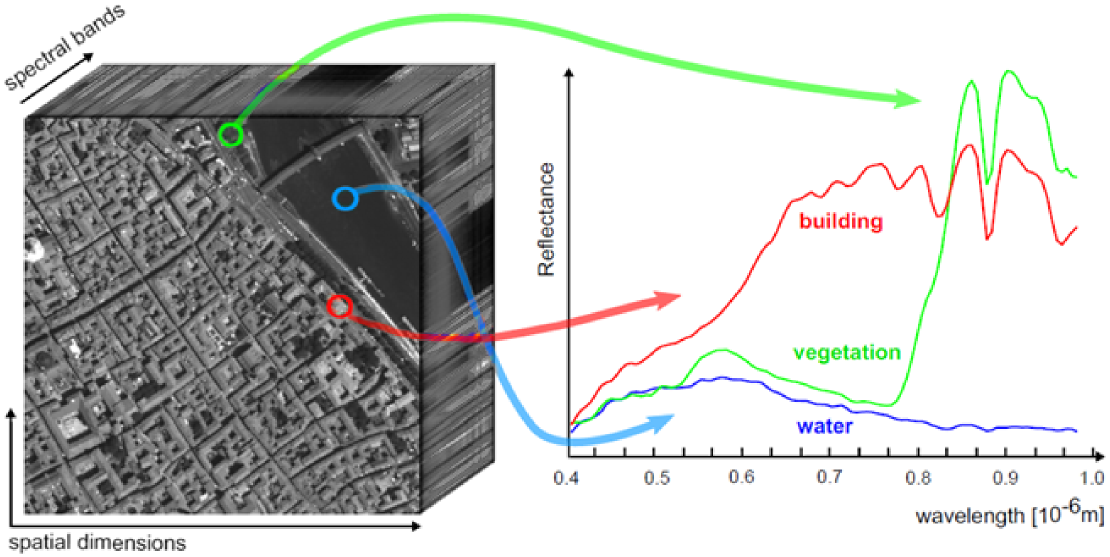
Not discriminative
The band is useless for these classes

Reminder: discriminative and well-selected makes life easier to the classifier

- Remember ?

$$\mathbf{x}_i = [\mathbf{x}_i^{\text{av}} \quad \mathbf{x}_i^{\text{std}} \quad \mathbf{x}_i^{\text{entr}} \quad \mathbf{x}_i^{\text{hist}} \quad \mathbf{x}_i^{\text{bow}} \quad \ldots]$$

The three steps of classification



1. Selection of training examples

2. Establishment of a model

3. Prediction on the grid (image)

Machine learning loves similarity measures

- To decide the class membership of a pixel, we need a model telling us that a pixel belongs to a class by

$$y_i^* = \arg\max_{c \in C} p(y_i = c|\mathbf{x})$$

- All classification models need two base ingredients:
  - A similarity measure, returning how much pixels “look alike”
  - A decision function, taking the decision

- Both functions are interrelated: similarity is used to take the decision

Machine learning loves similarity measures

- To take decisions, most ML models use similarity functions
- ~inverse of distances

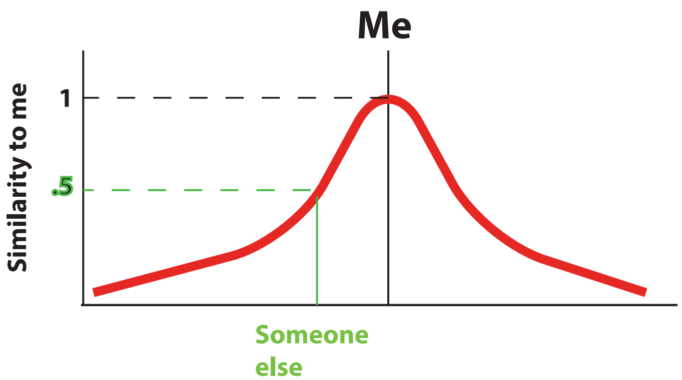
- It is a function that scores high if two objects look alike and low if they don't

vs.

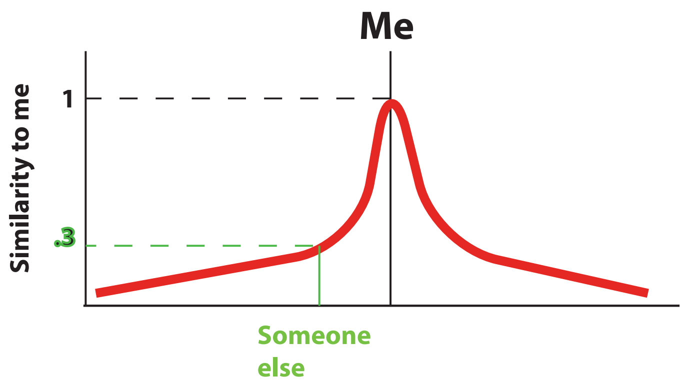
Gaussian similarity

- The γ parameter will decide how much similarity decreases with feature distance.

$$K(me, se) = \exp(-2\gamma^2 (me - se)^2)$$



Large bandwidth (large gamma)

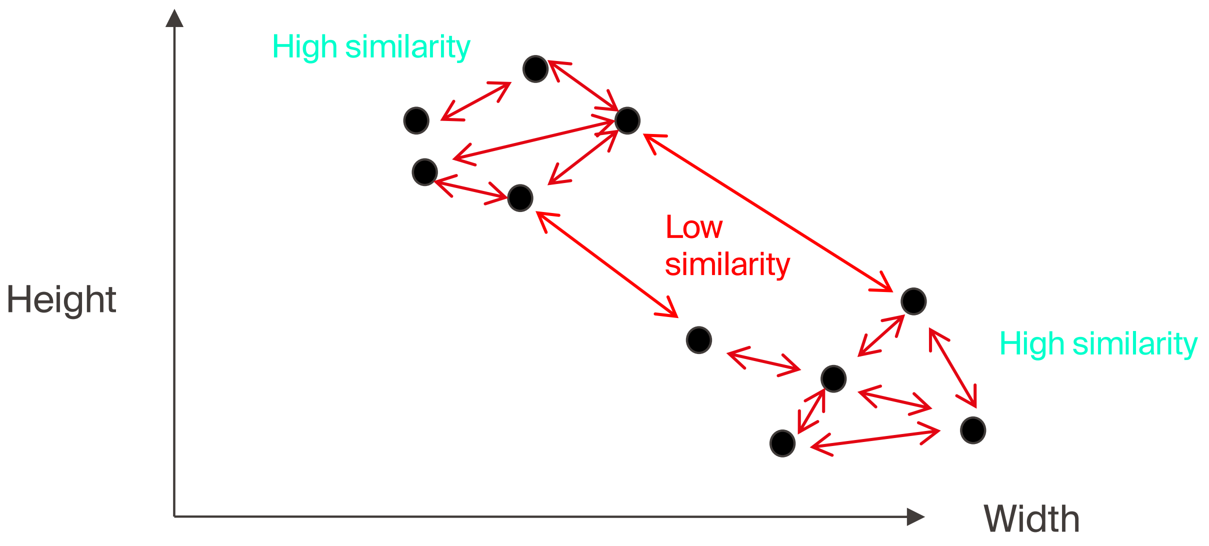


Small bandwidth (small gamma)

me = me
se = someone else

What we would like to obtain

- In a first approximation, it can be seen as the inverse of distance



Similarity functions

- There are many similarity functions
- The more classical are the Euclidean distances.

Euclidean
norm

- They define a similarity $s(\mathbf{x}_i, \mathbf{x}_j) = \exp\big(-d(\mathbf{x}_i, \mathbf{x}_j)\big)$

  - Euclidean distance: $$d^e(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\sum_{b=1}^{B} \left(\mathbf{x}_i^{(b)} - \mathbf{x}_j^{(b)}\right)^2} = ||\mathbf{x}_i - \mathbf{x}_j||$$

  - Manhattan distance: $$d^m(\mathbf{x}_i, \mathbf{x}_j) = |\mathbf{x}_i - \mathbf{x}_j|$$

Similarity functions

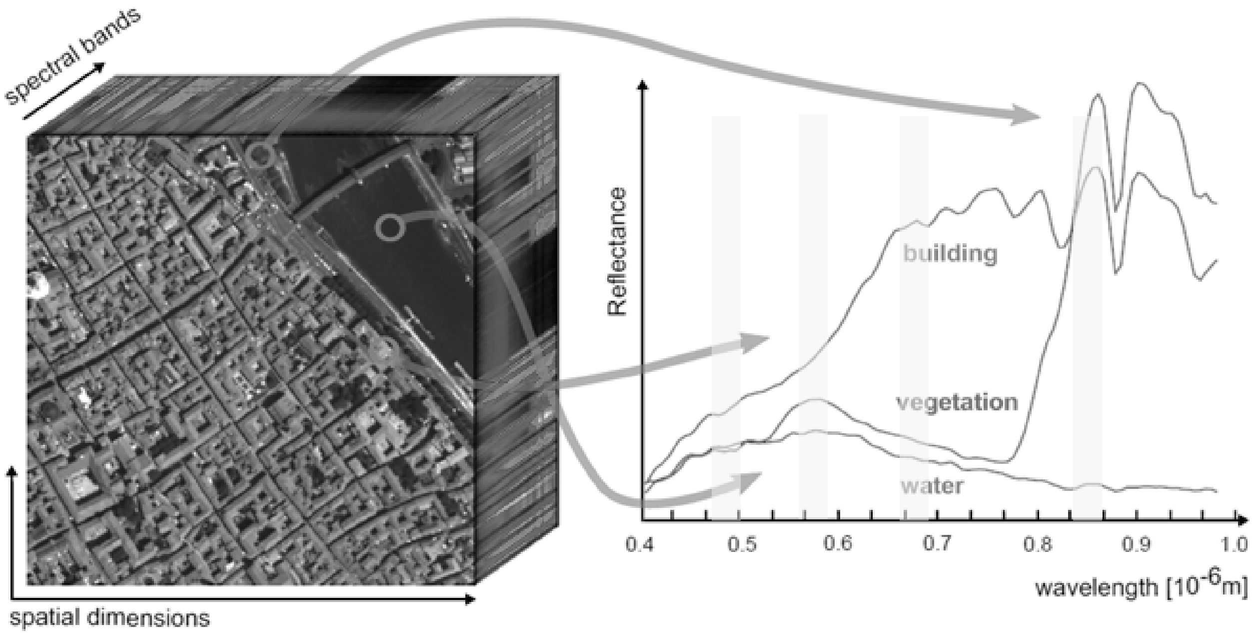
- There are many similarity functions
- The more classical are the Euclidean distances.

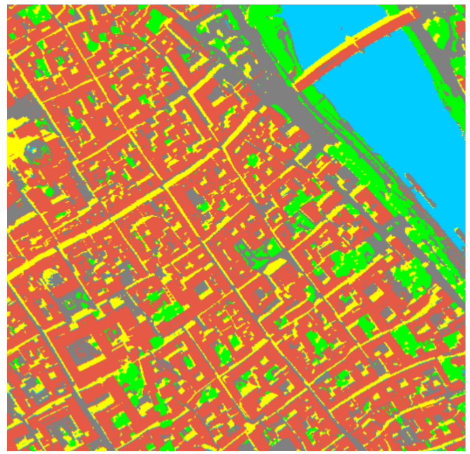
- They define a similarity $s(\mathbf{x}_i, \mathbf{x}_j) = \exp\big(-d(\mathbf{x}_i, \mathbf{x}_j)\big)$

- How they use similarity will define the classifier.

- But in the end, they all $y_i^* = \arg\max_{c \in C} p(y_i = c | \mathbf{x})$

The three steps of classification





1. Selection of training examples

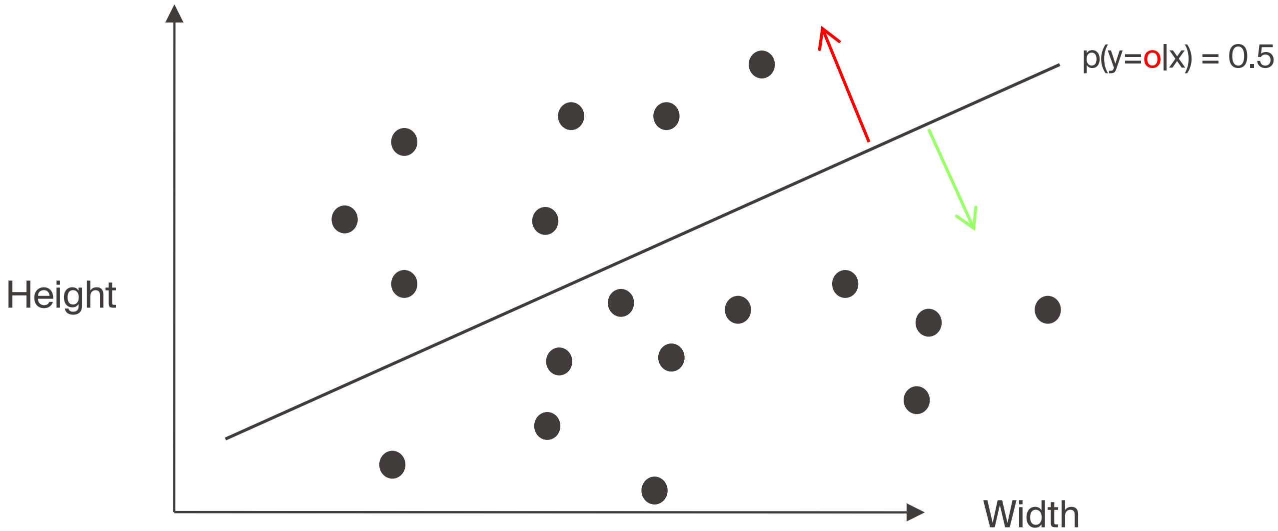
2. Establishment of a model

3. Prediction on the grid (image)

Prediction

- The decision function assigns all the pixels to the classes

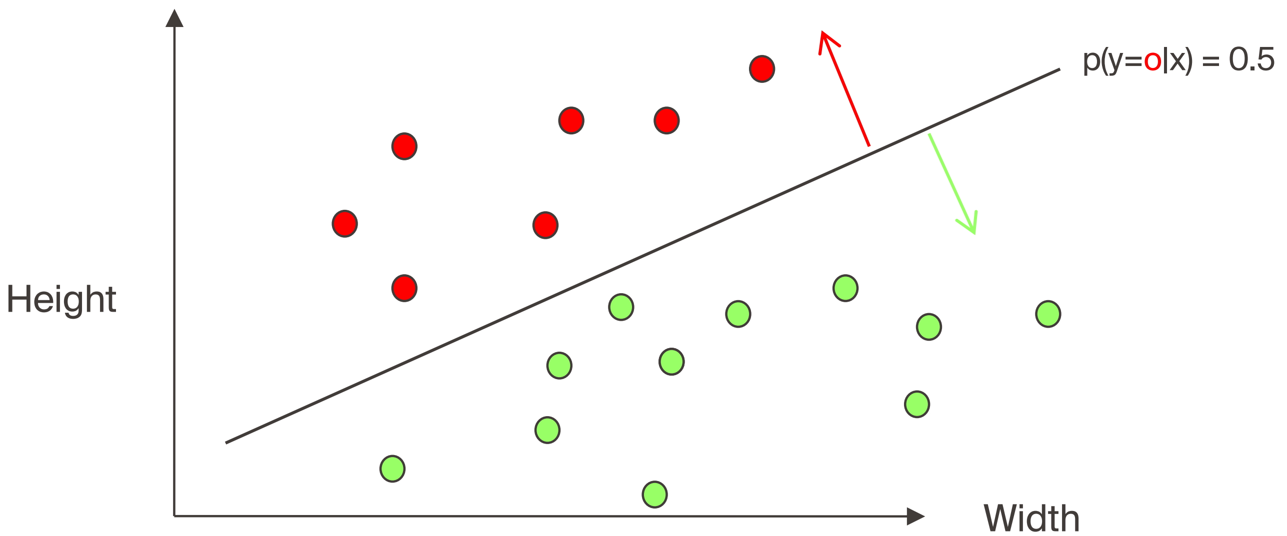
$$y_i^* = \arg\max_{c \in C} p(y_i = c|\mathbf{x})$$



Prediction

- The decision function assigns all the pixels to the classes

$$y_i^* = \arg\max_{c \in C} p(y_i = c | \mathbf{x})$$



Supervised classification approaches: a taxonomy

- Parametric methods: assumptions about the distribution of classes ("generative approach")
  - Gaussian Maximum Likelihood (GML) → Remote sensing course ENV-341

- Nonparametric methods: no assumptions (or less strict) about the distribution of classes, focus on modelling the class separation ("discriminative approach")
  - K-NN (lazy learner) → in a sec
  - Decision trees and random forests → today, next course
  - Support Vector Machine (SVM) → next week
  - Neural Networks (typically convolutional neural networks CNNs) → in two weeks

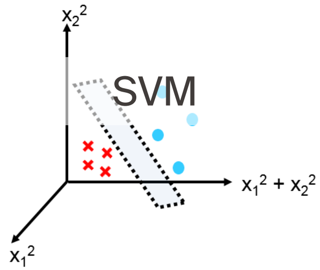
Nonparametric classifiers: how to chose?

Training data

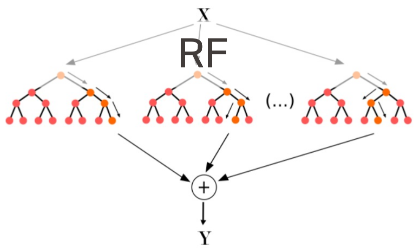
features?

samples?

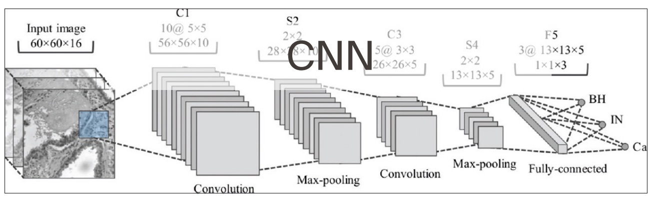
features low
samples limited



features high
samples limited



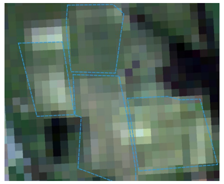
features high
samples large



Nonparametric classifiers: how to chose?

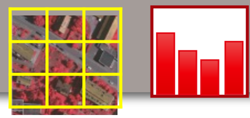
DATA SOURCE	FEATURE EXTRACTION	FEATURE SELECTION	CLASSIFICATION MODELS

HR

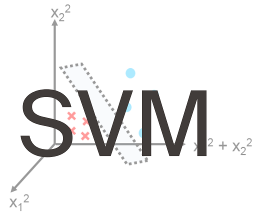


Pixel features (spectral/Indices, e.g NDVI)

Object (spatial) features (e.g. mean, std)

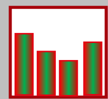


features: < 20
samples: < 1000



Hand-crafted features:

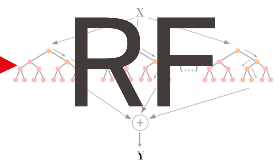
- colour histograms
- Histogram of Orientated Gradients (HOG)
- SIFT points (corners and edges)



Unsupervised features:

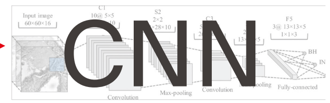
- BoVW, pLSA, LDA, SPM, LLC

features: 100-50'000
samples: < 100'000



Supervised features: Deep features from Convolutional Neural Networks (CNN)

features: 1000 – 10'000
samples: 100'000 – 10 Mio



VHR

Example: *k*-NN

- K nearest neighbors (KNN) assess class membership according to distance to neighbors in the feature space

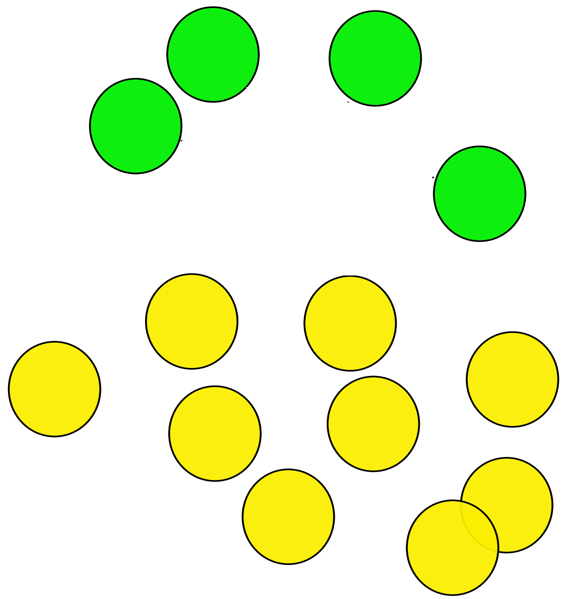
- *K*-NN does not make any assumption about classes

- It does not even model class distributions!

- It only assumes that
  - close elements are of the same class
  - classes are separated by a low-density region

Example: *k*-NN

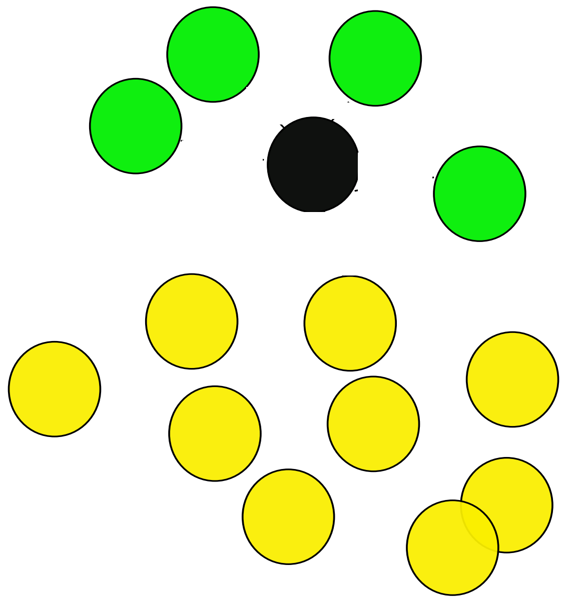
- We have this two classes problem.
- All you know is the distribution of these training samples



Example: *k*-NN

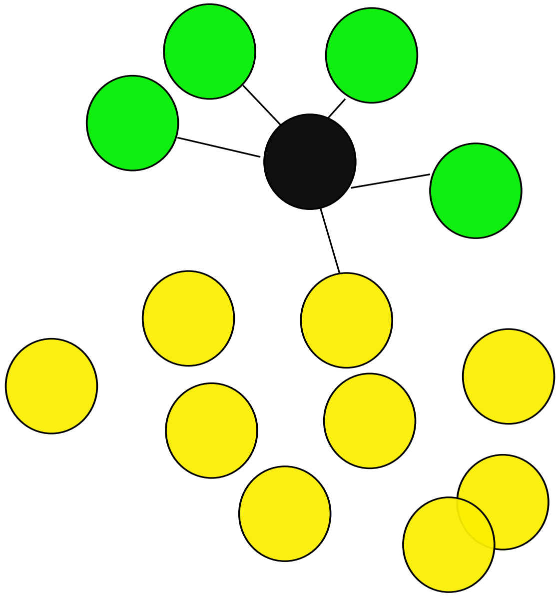
D. Tuia. ECEO

- We have this two classes problem.
- All you know is the distribution of these training samples
- What is the class of this new one?



Example: *k*-NN

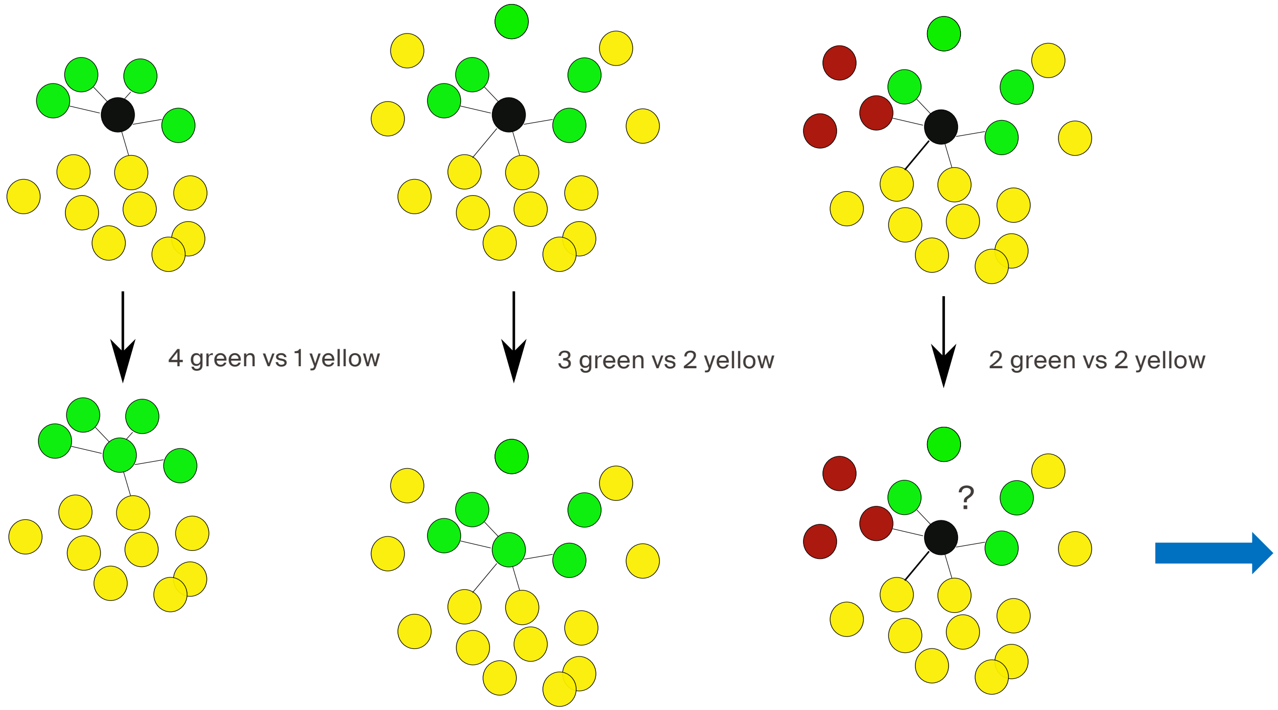
- We have this two classes problem.
- All you know is the distribution of these training samples
- What is the class of this new one?



Let's look at the k=5 nearest neighbors

Example: *k*-NN

- Principle (example with $k = 5$ neighbors)



When on a tie select random among winning classes

Lazy learners: *k*-NN

- Pros
  - Very intuitive and easy to implement
  - Local and possibly nonlinear

- Cons
  - Euclidean distances are used, sensitive to differences in variance between the bands
  - Mid running time: N distances for each pixel (N = # of labeled)
  - *k* must be found heuristically

K-NN
a super-inefficient (Matlab) code

```
% xtr are the training data, (ntr x ndim)
% ytr are the training labels
% xts are the data to be predicted (nts x ndim)

predKnn = zeros(size(xts,1),1);ntr=length(xtr);nts=length(xts);%initialize answers matrix
K = 5;% K is a parameter to be set.
for i = 1:nts
    dist = zeros(1,ntr);%re-initialize distance matrix every time
    for j = 1: ntr
        dist(j) = pdist([xtr(j,:);xts(i,:)]); %calculate distances
    end
  [sDist,sID] = sort(dist); % sort by distance, keep the indices sID
  sID = sID(1:K); % keep only the k shortest distances
  m = ytr(sID); % find the class of the samples among the knn-s
  m = mode(m); % the most occurring class is the mode
  predKnn(i) = m; % that is the winner!
end
```

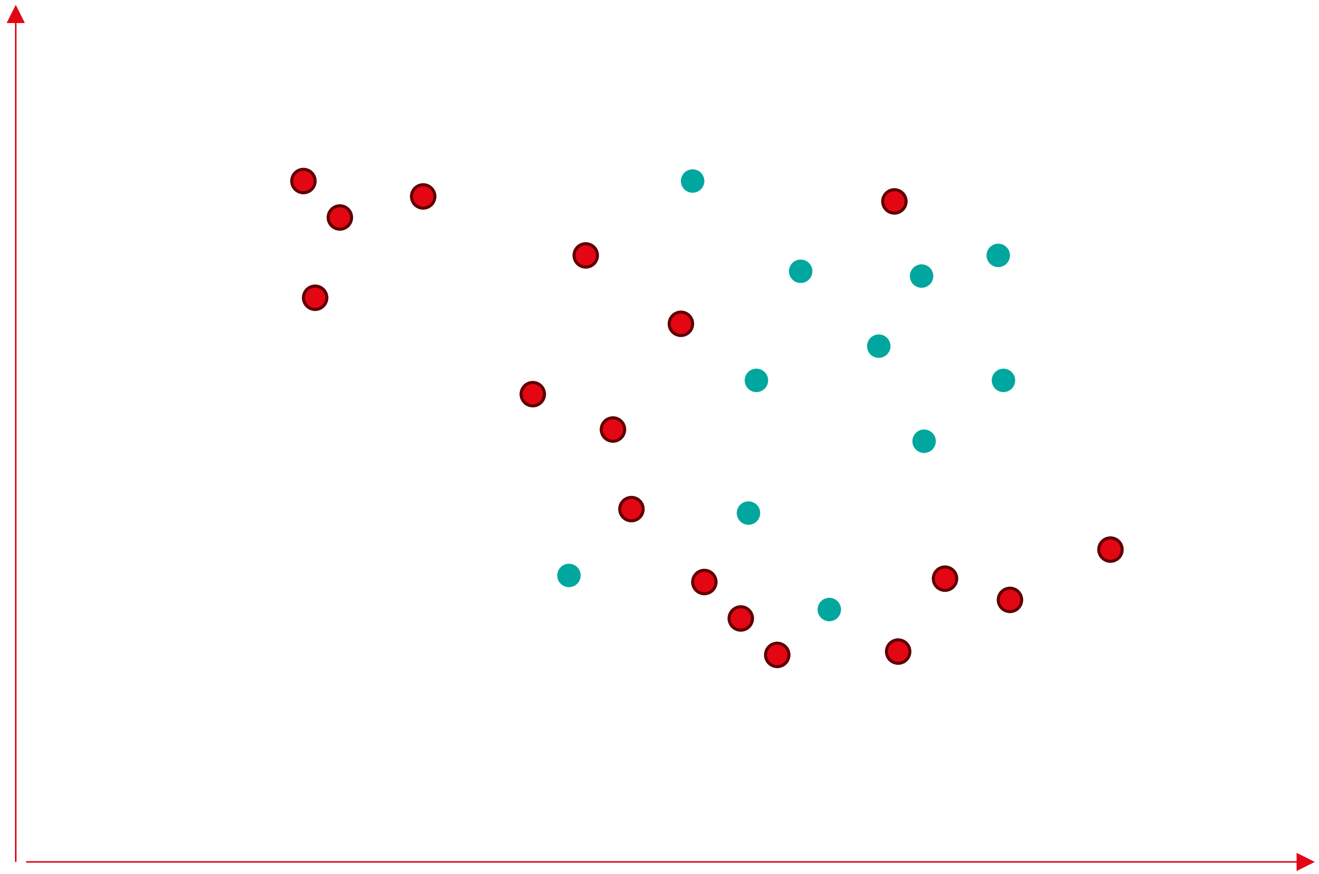
kNN – pros and cons

- *k*-NN does not make any assumption about classes
- It is simple but very nonlinear!

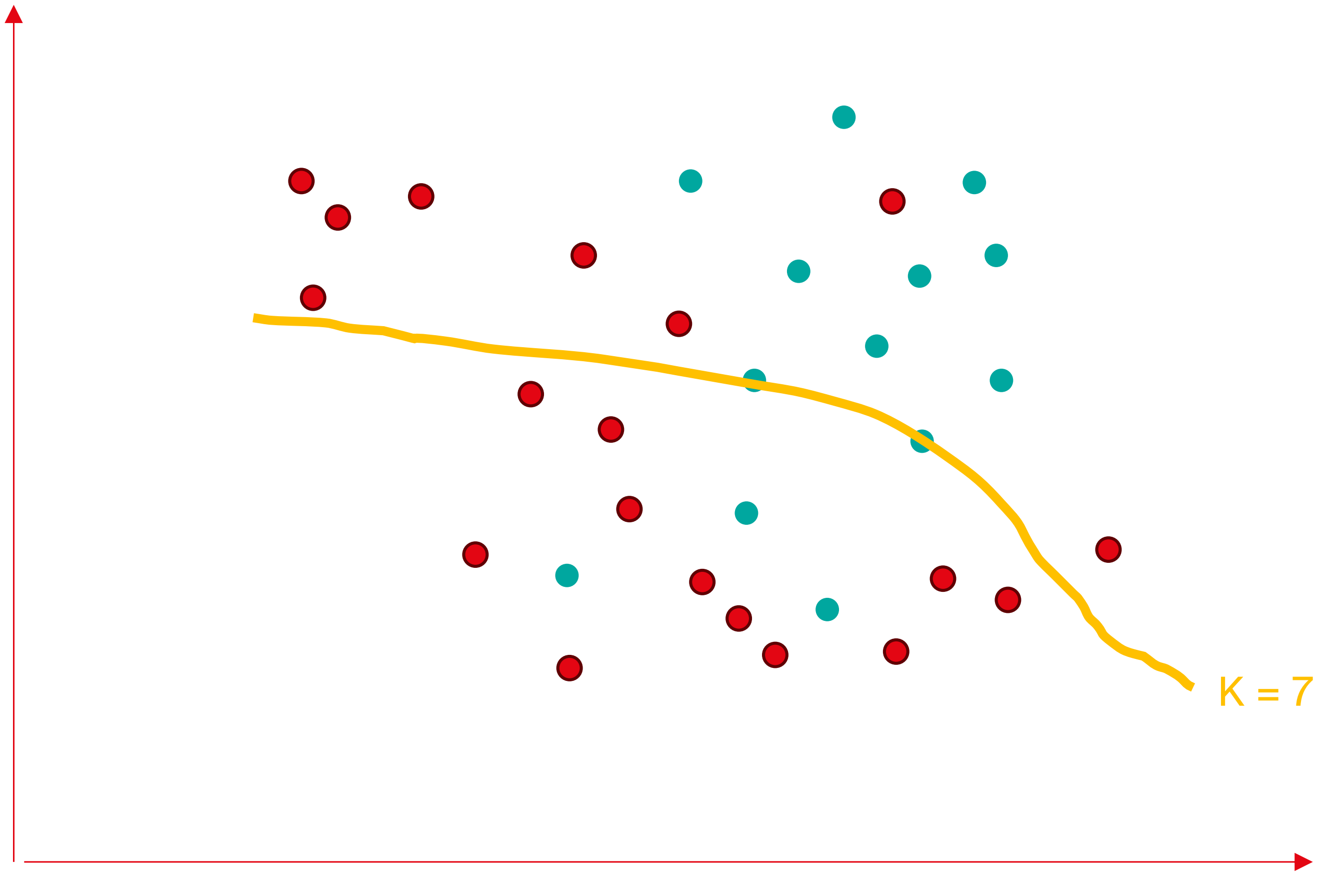
- It does not even model class distributions! (it's lazy)

- It only assumes that
  - close elements are of the same class
  - classes are separated by a low-density region

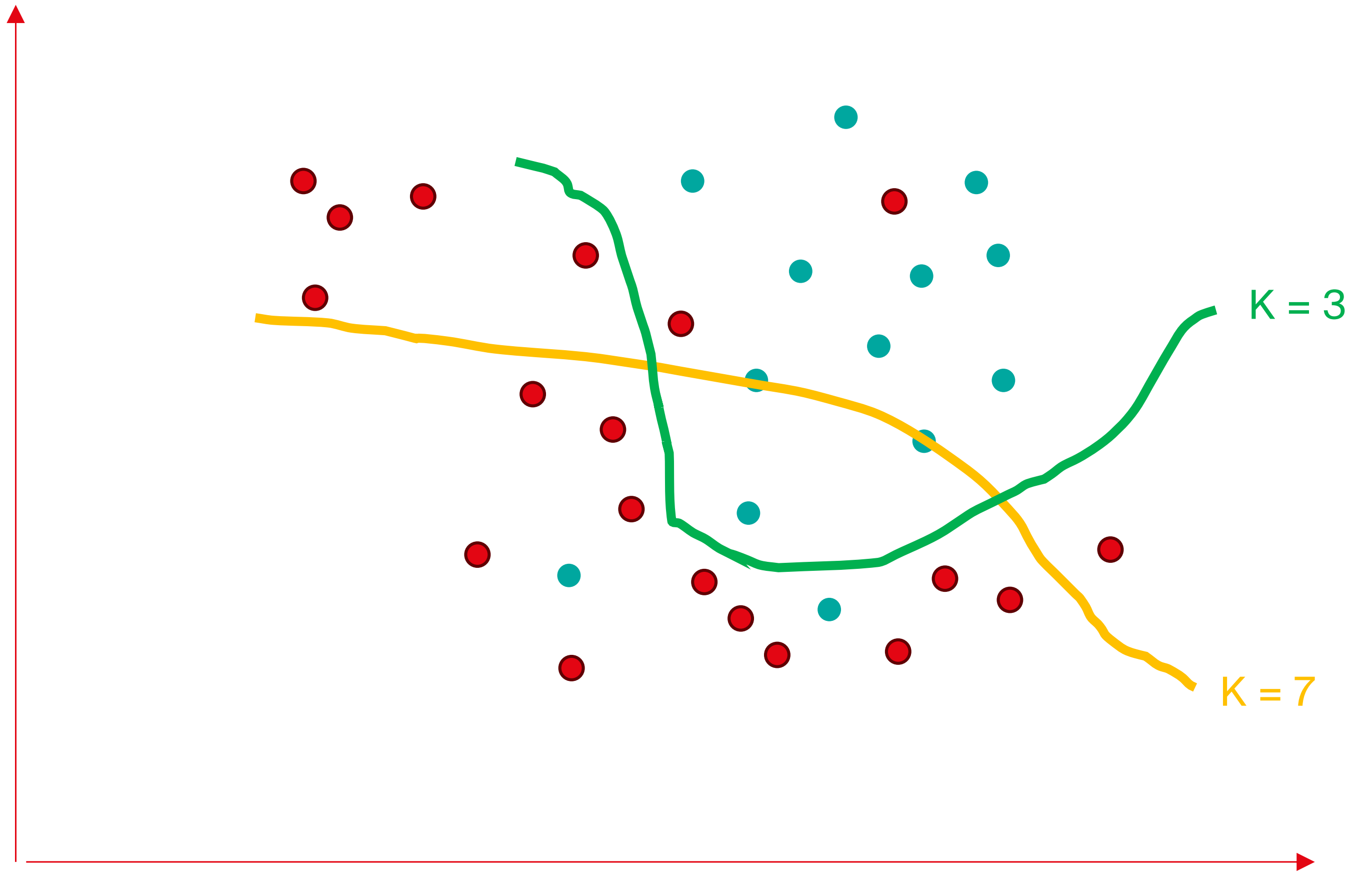
The lower the k, the more you fit data



The lower the k, the more you fit data



The lower the k, the more you fit data



EPFL

The lower the k, the more you fit data