AI4Life

AI4Life Workshop
June 10, 2024
Campus Puerta de Toledo
Universidad Carlos III de Madrid

Deep Learning for Microscopy

Or How to do BioImage Analysis in the era of Deep Learning

Ignacio Arganda-Carreras, PhD.
Ikerbasque Research Associate
Computer Science and Artificial Intelligence Department
University of the Basque Country

ikerbasque
Basque Foundation for Science

Donostia International Physics Center

eman ta zabal zazu
Universidad del País Vasco
Euskal Herriko Unibertsitatea

BIOFISIKA
Basque Centre for Biophysics

neubias
network of european bioimage analysts

Outline

- What is Bioimage Analysis?
- Image Segmentation.
  - Machine learning-based segmentation.
    - What is Machine Learning?
      - Important concepts and definitions.
    - Shallow Learning.
      - Trainable Weka Segmentation plugin.
    - Deep Learning.
      - Historical view of Artificial Neural Networks.
      - Available tools.

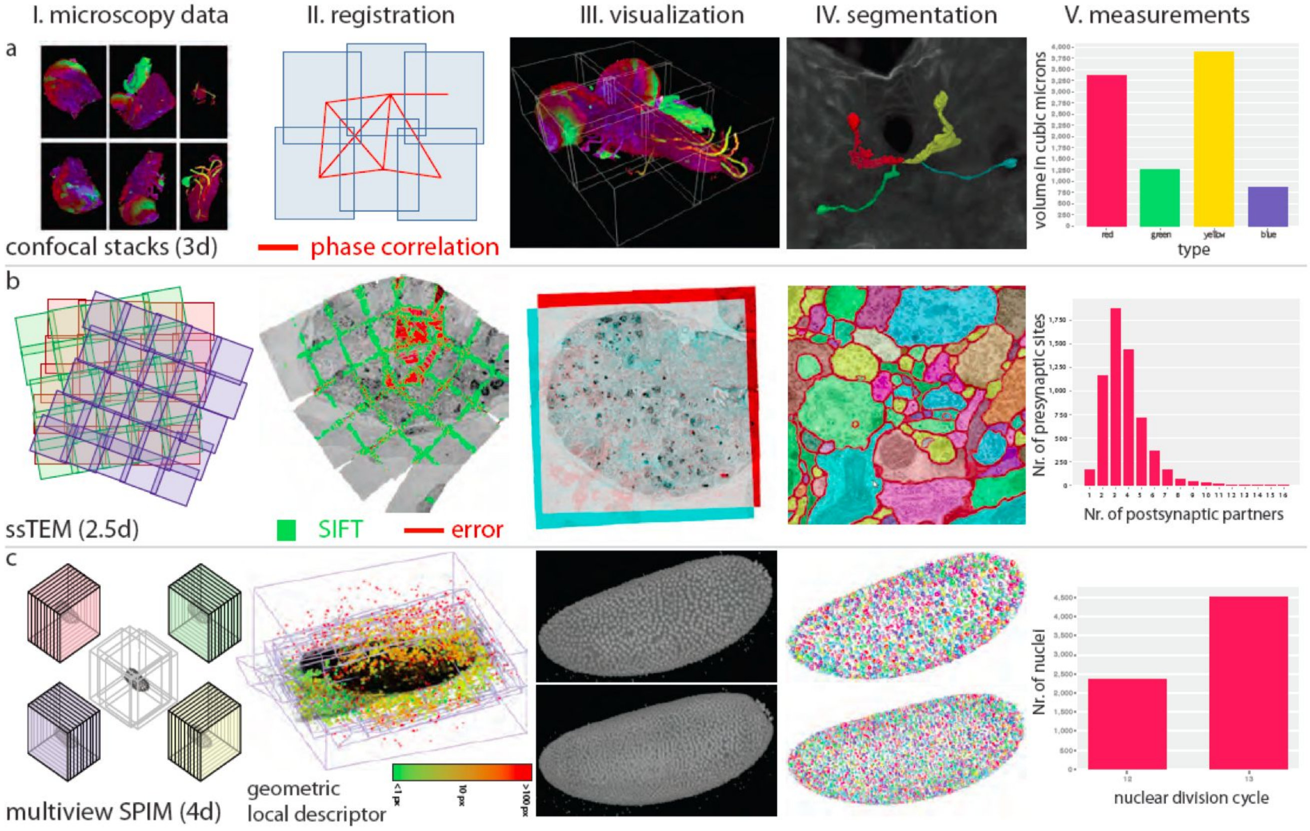
What is bioimage analysis?

Automatically extract biophysically meaningful information from microscopy images of biological samples.

EU funded action: https://www.cost.eu/actions/CA15124
Network of European Bioimage Analysts: www.neubias.org

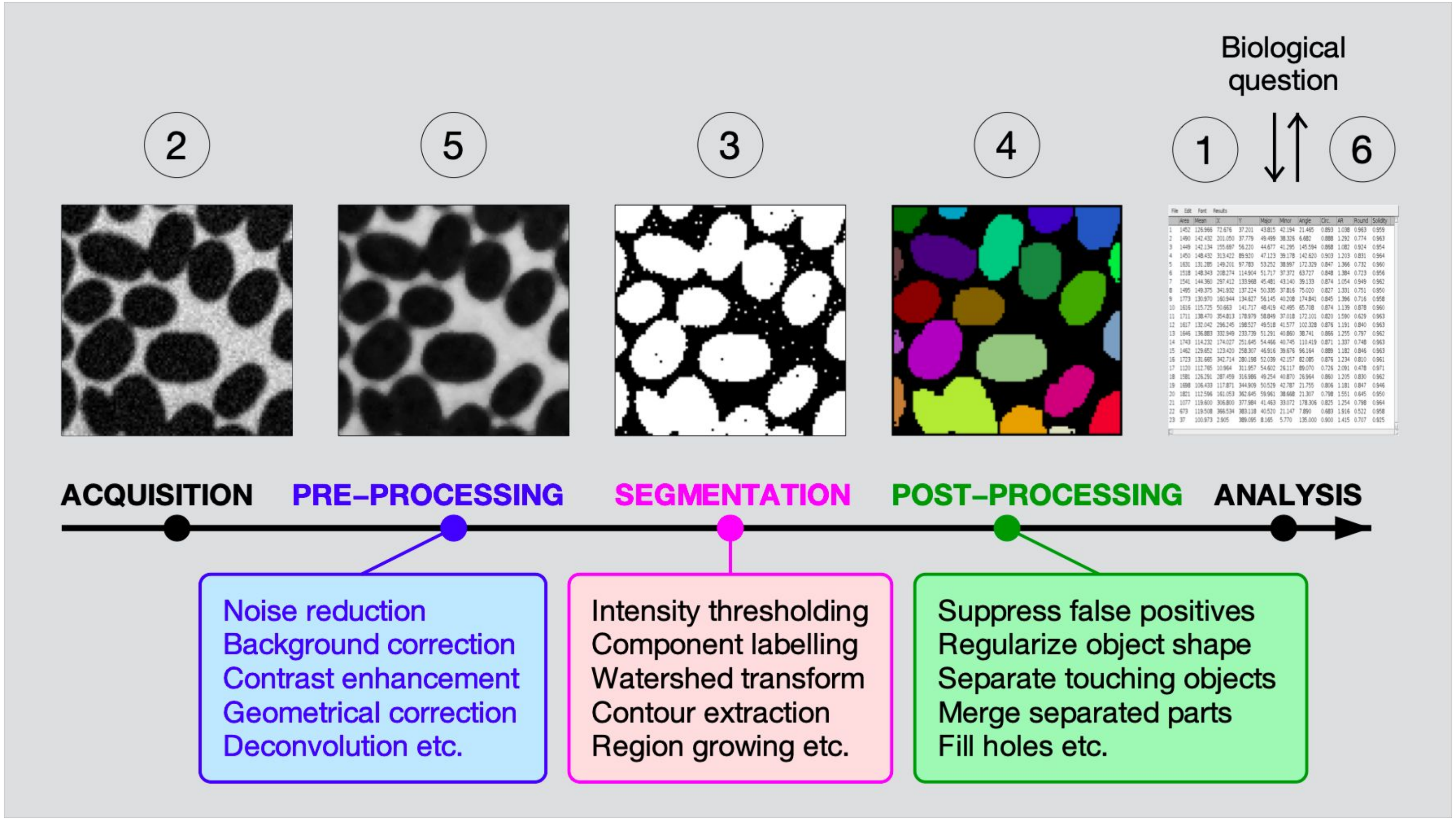
Slide adapted from Christian Tischer

Fiji: our open-source solution



Schindelin,
Nature Methods,
2012

Typical analysis pipeline



Noise reduction
Background correction
Contrast enhancement
Geometrical correction
Deconvolution etc.

Intensity thresholding
Component labelling
Watershed transform
Contour extraction
Region growing etc.

Suppress false positives
Regularize object shape
Separate touching objects
Merge separated parts
Fill holes etc.

Arganda-Carreras, I. & Andrey, P., Light Microscopy, 2017

Image Segmentation

- “Process of partitioning a digital image into multiple segments”.
- Typically used to locate **objects** and **boundaries**.

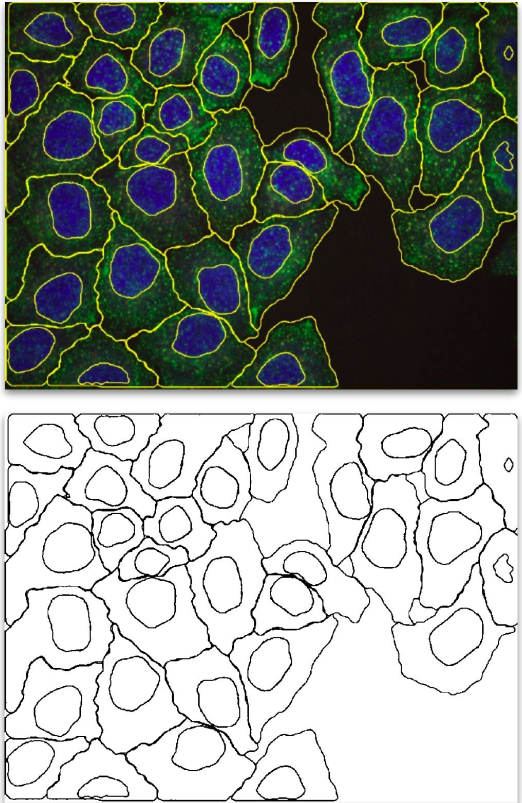


Image Segmentation

- "Process of partitioning a digital image into multiple segments".
- Typically used to locate **objects** and **boundaries**.
- More precisely, image segmentation is the process of **assigning a label** to every pixel in an image such that pixels with the same label share certain visual characteristics.

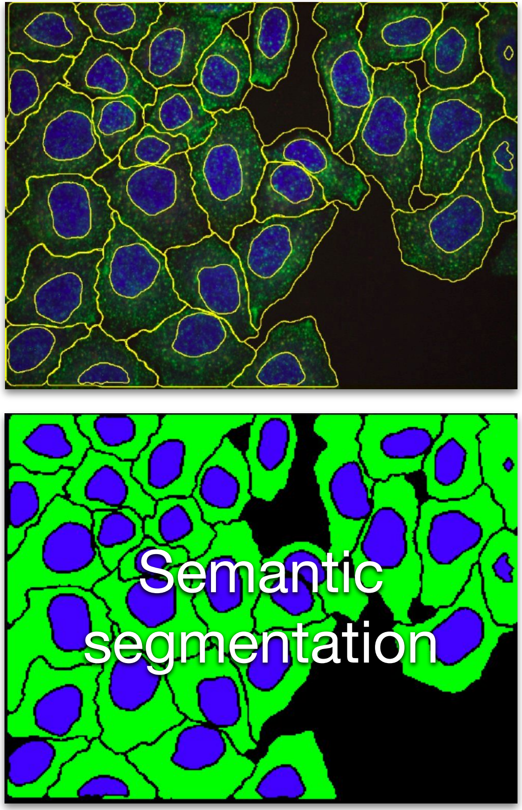
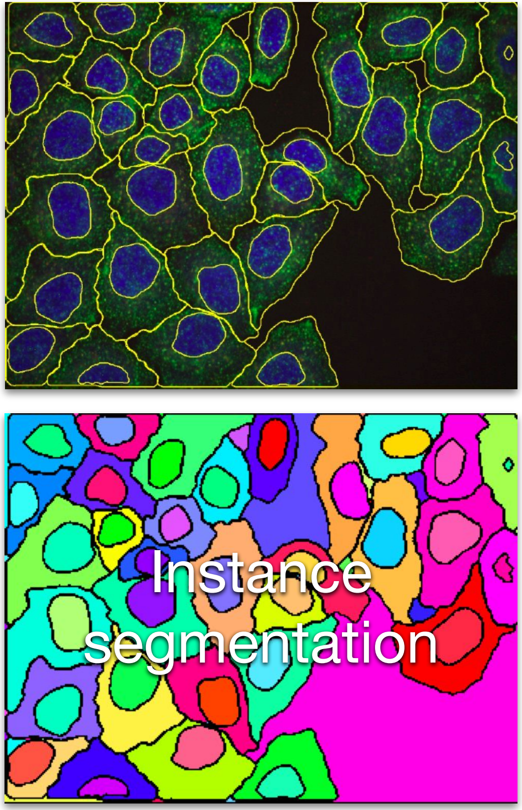


Image Segmentation

- “Process of partitioning a digital image into multiple segments”.
- Typically used to locate **objects** and **boundaries**.
- More precisely, image segmentation is the process of **assigning a label** to every pixel in an image such that pixels with the same label share certain visual characteristics.



Segmentation methods

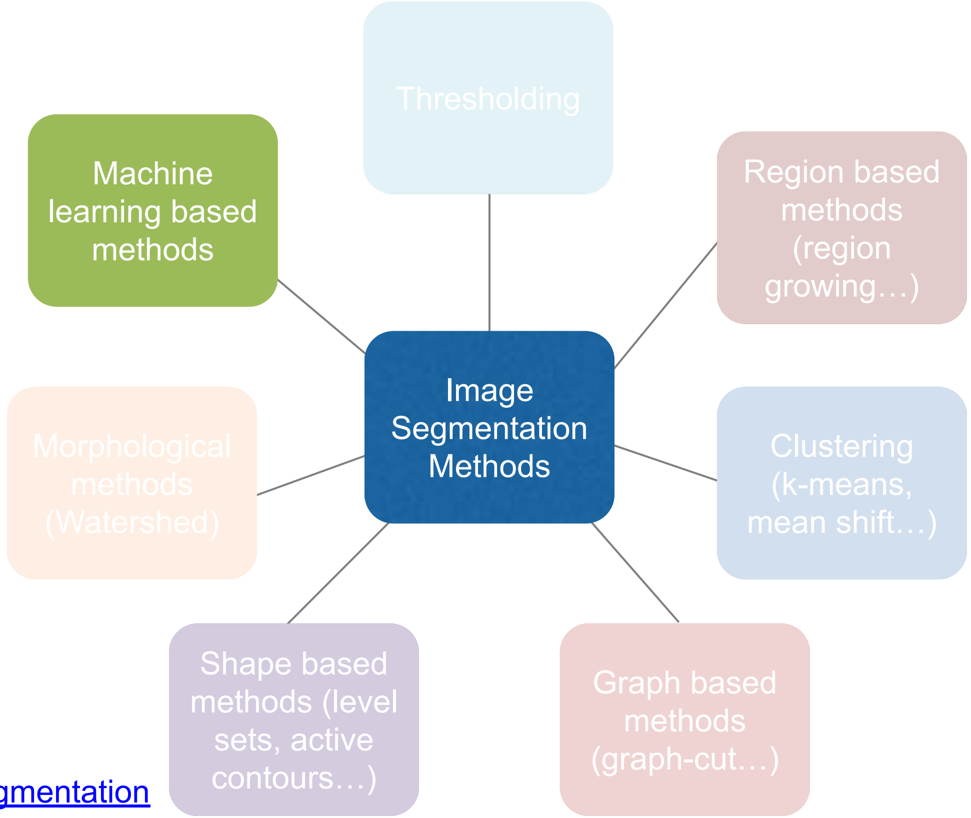


Image Segmentation Methods

Thresholding

Machine learning based methods

Region based methods (region growing…)

Morphological methods (Watershed)

Clustering (k-means, mean shift…)

Shape based methods (level sets, active contours…)

Graph based methods (graph-cut…)

http://imagej.net/Category:Segmentation

Machine learning based segmentation

What is machine learning?

- Technological advances in the last decades have made possible to automate many tasks that required a significant amount of time and repetitive manual work.
- Now technology allows us to work with big data and automate tasks that are not simply mechanical but require a certain degree of intelligence.
- Some tasks are easy for humans but difficult for machines. For example, face recognition, has many challenges:
    - Position, illumination, haircut…
- Some tasks are hard for humans due to the large amount of data to handle.
- Data mining and machine learning techniques have achieved great results in this direction, making intelligent systems an important part of research and business models.

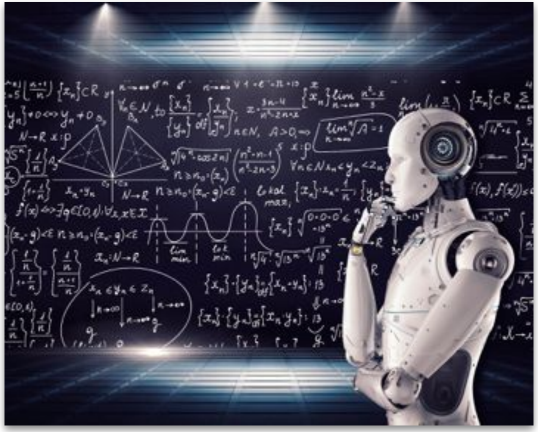


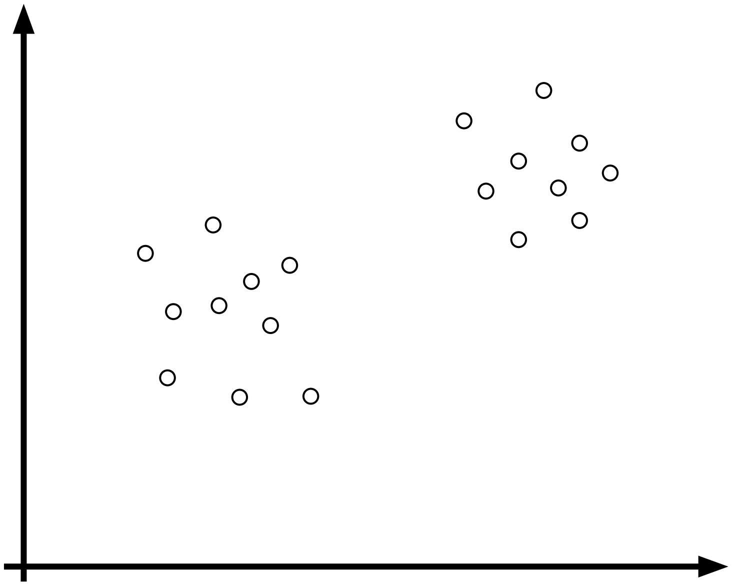
Image via www.vpnsrus.com

Machine learning

- Subfield of computer science that "gives computers the ability to learn without being explicitly programmed" (Arthur Samuel, 1959).

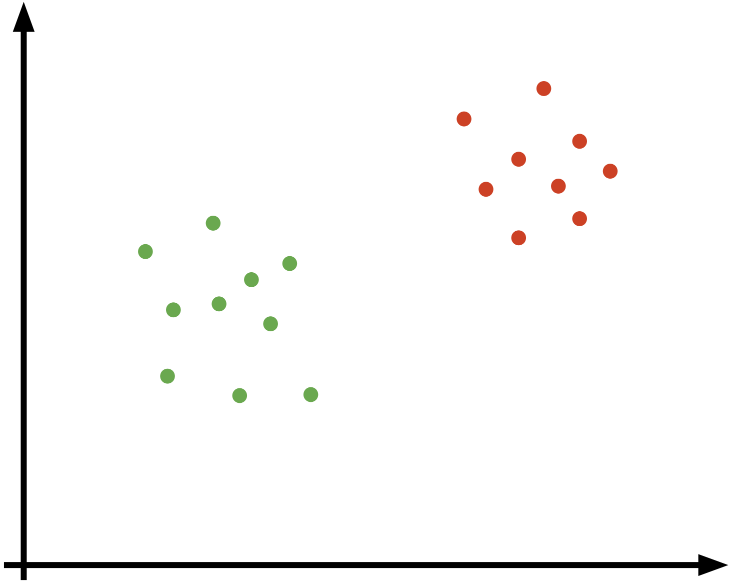
Machine learning

- Subfield of computer science that "gives computers the ability to learn without being explicitly programmed" (Arthur Samuel, 1959).



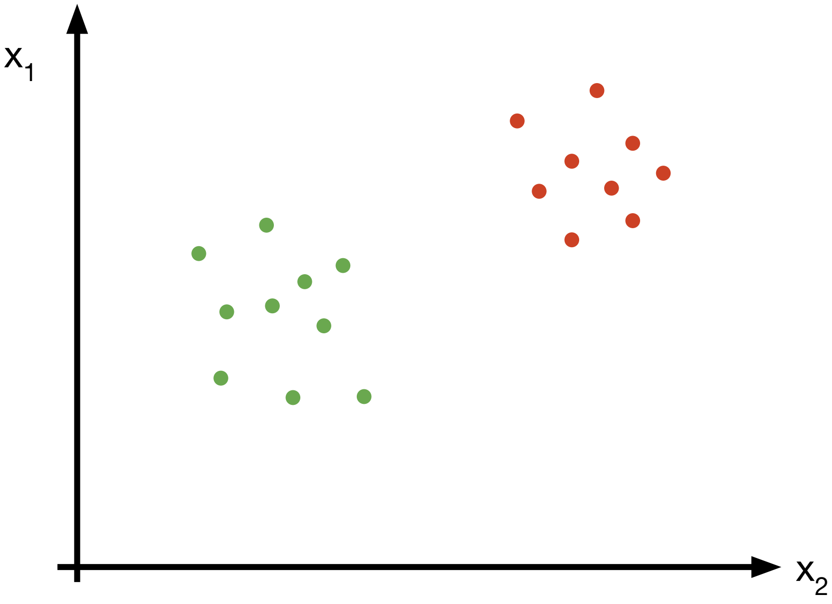
Machine learning

- Subfield of computer science that "gives computers the ability to learn without being explicitly programmed" (Arthur Samuel, 1959).
- Assign labels to objects indicating their class.

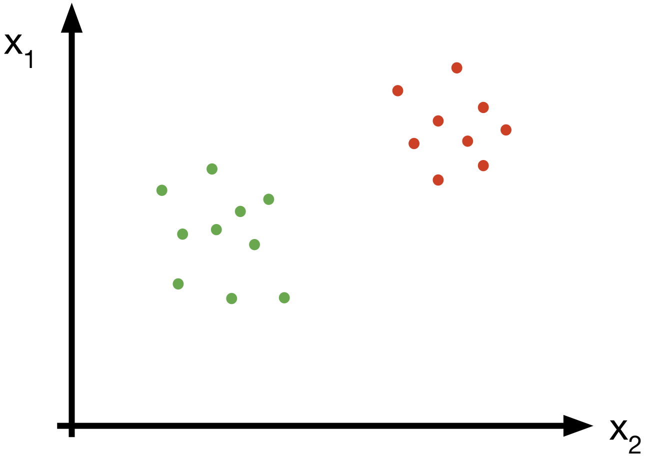


Machine learning

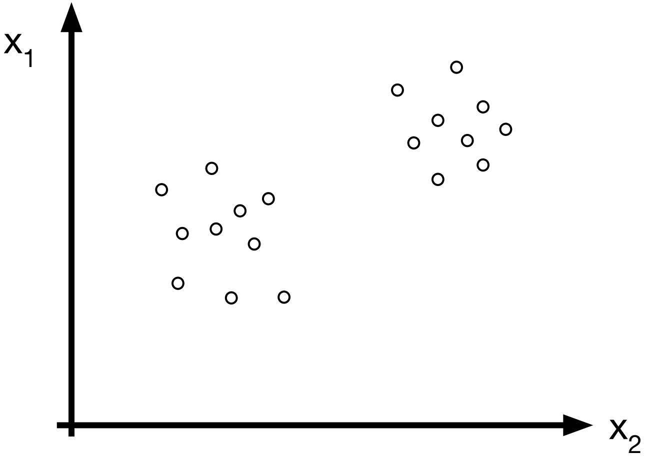
- Subfield of computer science that "gives computers the ability to learn without being explicitly programmed" (Arthur Samuel, 1959).
- Assign labels to objects indicating their class.
- Objects represented by a set of measurements or features.



Supervised vs unsupervised learning



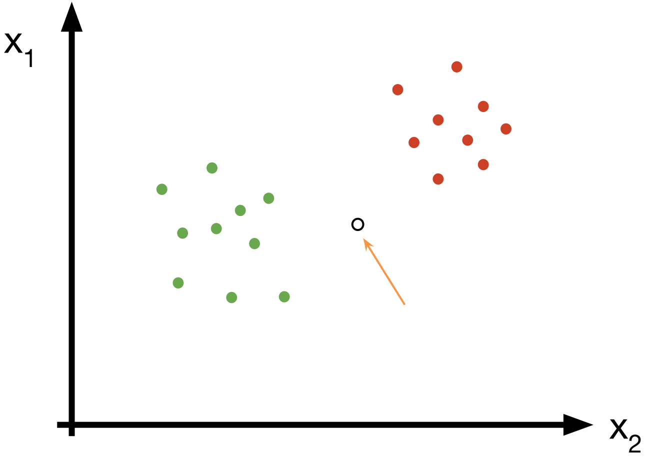
Supervised



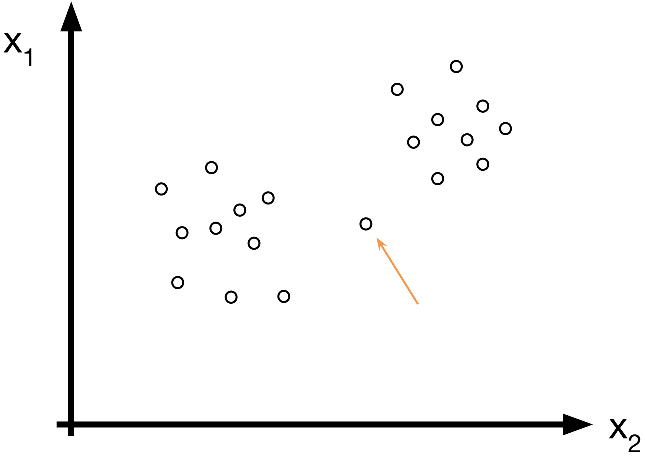
Unsupervised

Supervised vs unsupervised learning

- To which class belongs each new point?



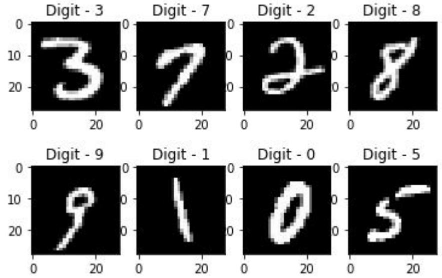
Supervised



Unsupervised

Supervised vs unsupervised learning

- Supervised learning:
  - Data is labeled.
  - Target: build a model or classifier to automatically label novel data ⇒ training.
  - If labels are not discrete ⇒ regression.
  - Examples:
    - Character or digit recognition (classification).
    - Age estimation (regression).
- Unsupervised learning:
  - Data without labels.
  - Target: model data to discover groups ⇒ clustering.



Age: 26.7

Class, features and dataset

- In machine learning, each sample is represented by a feature vector:

$$x = [x_1, ...x_n]$$

- Features can be quantitative or qualitative.
- Supervised learning ⇒ predefined classes (labels). $U_L = \{u_0, ..., u_{L-1}\}$
  - A sample belongs (usually) to one class only.
  - Each class contains similar samples different to the samples of other classes.
  - Ex.: biopsy classification ⇒ 2 classes (tumorous - non-tumorous).
- Dataset: $S = \{(x^k, d^k), k = 1, ..., N\}$
  - Set of pairs sample-class (supervised)
  - Set of samples without class (unsupervised)

Classification

- Classifier: function that relates samples to labels:

$$f_\theta : x \to d$$

- Some classifiers have two steps:
  - "Soft" decision $y^k = f_\theta(x^k)$
  - "Hard" decision $\hat{d}$
- Training algorithm adjusts classifier parameters θ to minimize a cost or error function.

Classifier evaluation

- How do we evaluate the performance of a classifier?
- How do we compare classifiers?
- Confusion matrix:

		Prediction	
		u_0	u_1
Real class	u_0	TP	FN
	u_1	FP	TN

⇐ Predicted labels

Groundtruth ⇒

TP: True Positive, **TN**: True Negative, **FP**: False Positive, **FN**: False Negative.

Total number of positive samples: P=TP+FN

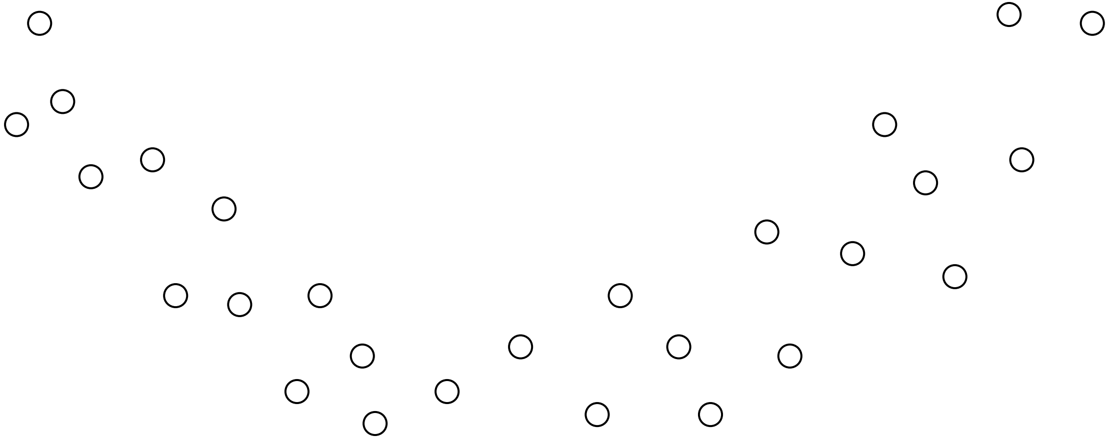
Total number of negative samples: N=TN+FP

Performance metrics

True Positive Rate (hit rate, recall, sensitivity)	TP/P	Proportion of positive samples correctly classified
False Positive Rate (False Alarm Rate)	FP/N	Proportion of negative samples incorrectly classified as positive
False Negative Rate	FN/P	Proportion of positive samples incorrectly classified as negative
True Negative Rate (specificity)	TN/N	Proportion of negative samples correctly classified
Precision (Positive Predictive Value)	TP/(TP+FP)	Proportion of samples classified as positive that are really positive
F1 Score	2 x precision x recall / (precision+recall)	Harmonic mean of precision and recall
Error Rate	(FP+FN) / (P+N)	Proportion of samples incorrectly classified
Accuracy	(TP+TN) / (P+N)	Proportion of samples correctly classified

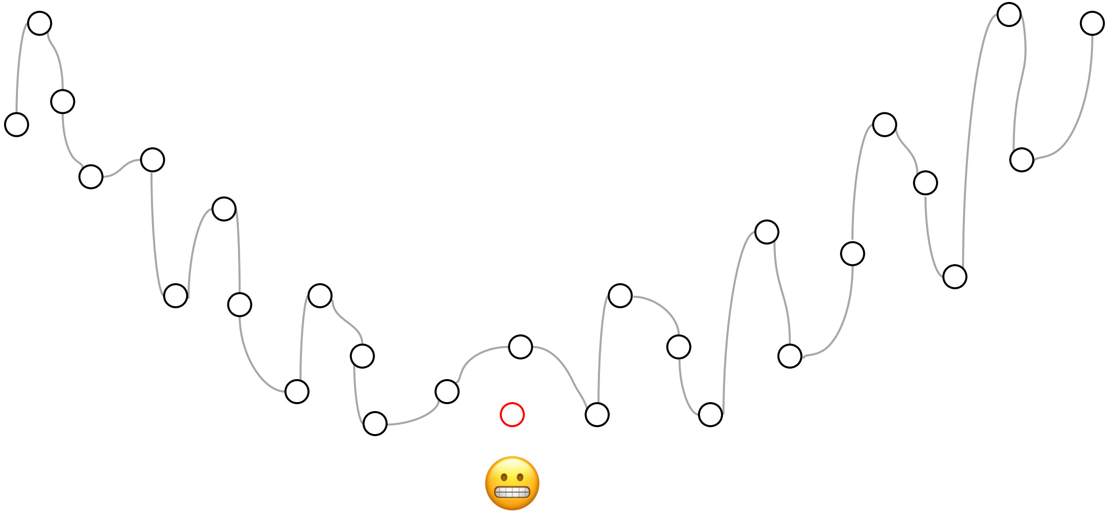
Classifier design

- How many samples we use to build the classifier?



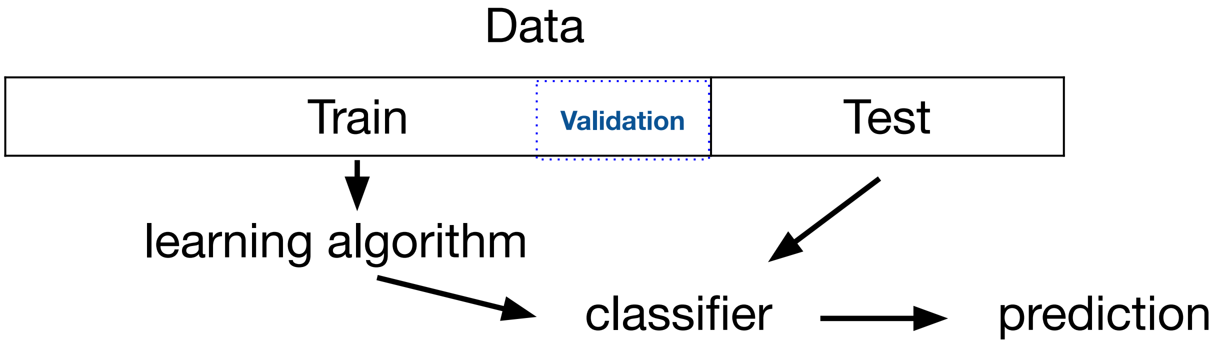
Classifier design

- How many samples we use to build the classifier?
- If we use all samples ⇒ overfitting



Train and test

- Divide dataset in two sets:
  - Train: to build the model (~66% of samples)
    - Some models need a validation subset (~10% of training samples).
  - Test: to evaluate the model (~33% of samples)



Data

Train

Validation

Test

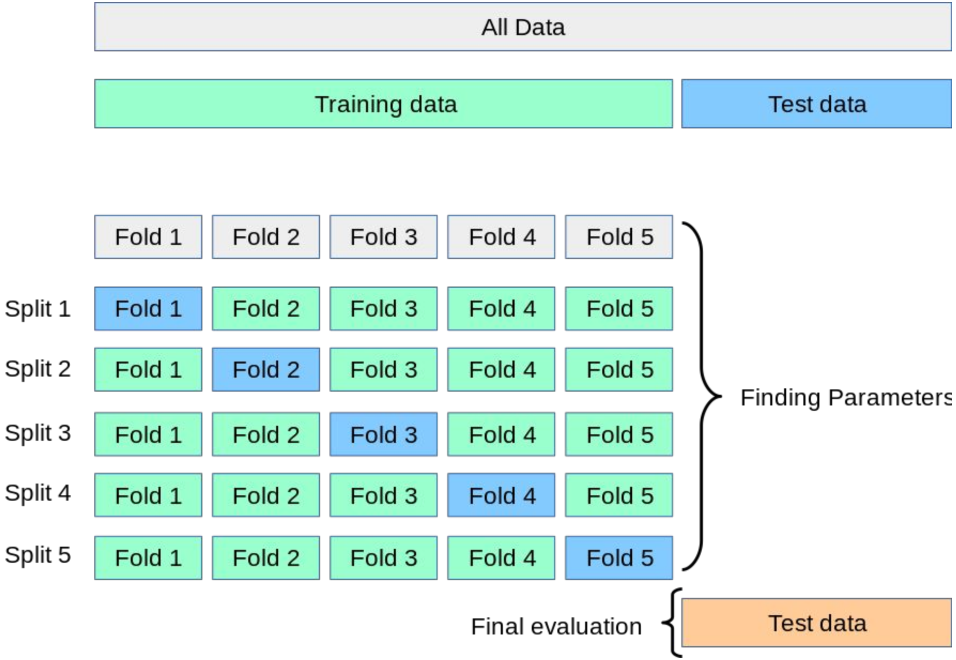
learning algorithm

classifier

prediction

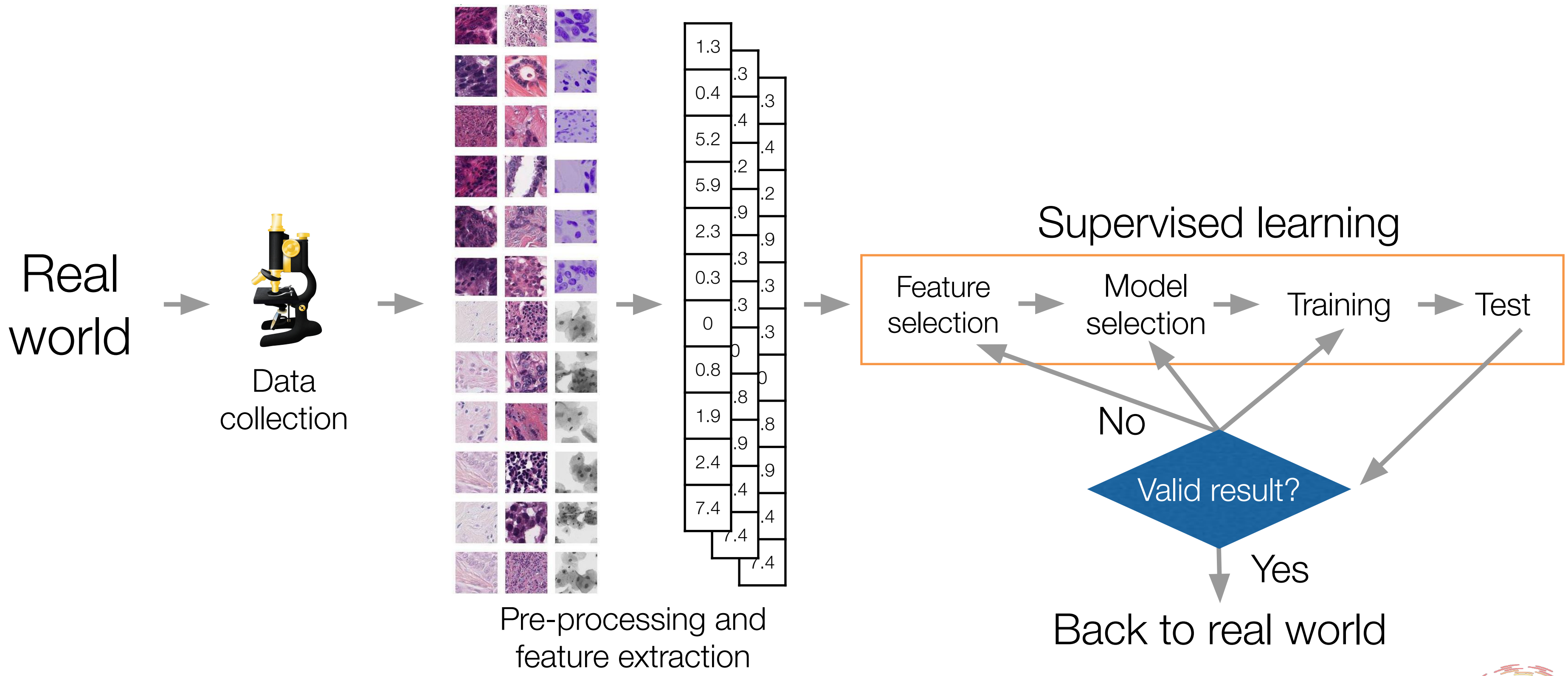
k-Fold cross-validation

- Very common alternative.
- Divide dataset of ***N*** samples in ***k*** “equal” parts: 1 for test and ***k***-1 for train.
- Usual values of ***k***: 5 or 10.
- Train/test ***k*** times and evaluate based on test labels
- Particular case: *leave-one-out* (***N***-1 for training and 1 for testing).



Source: https://scikit-learn.org/stable/modules/cross_validation.html

Classic classifier design cycle



Real world

Data collection

Pre-processing and feature extraction

Supervised learning

Feature selection

Model selection

Training

Test

No

Valid result?

Yes

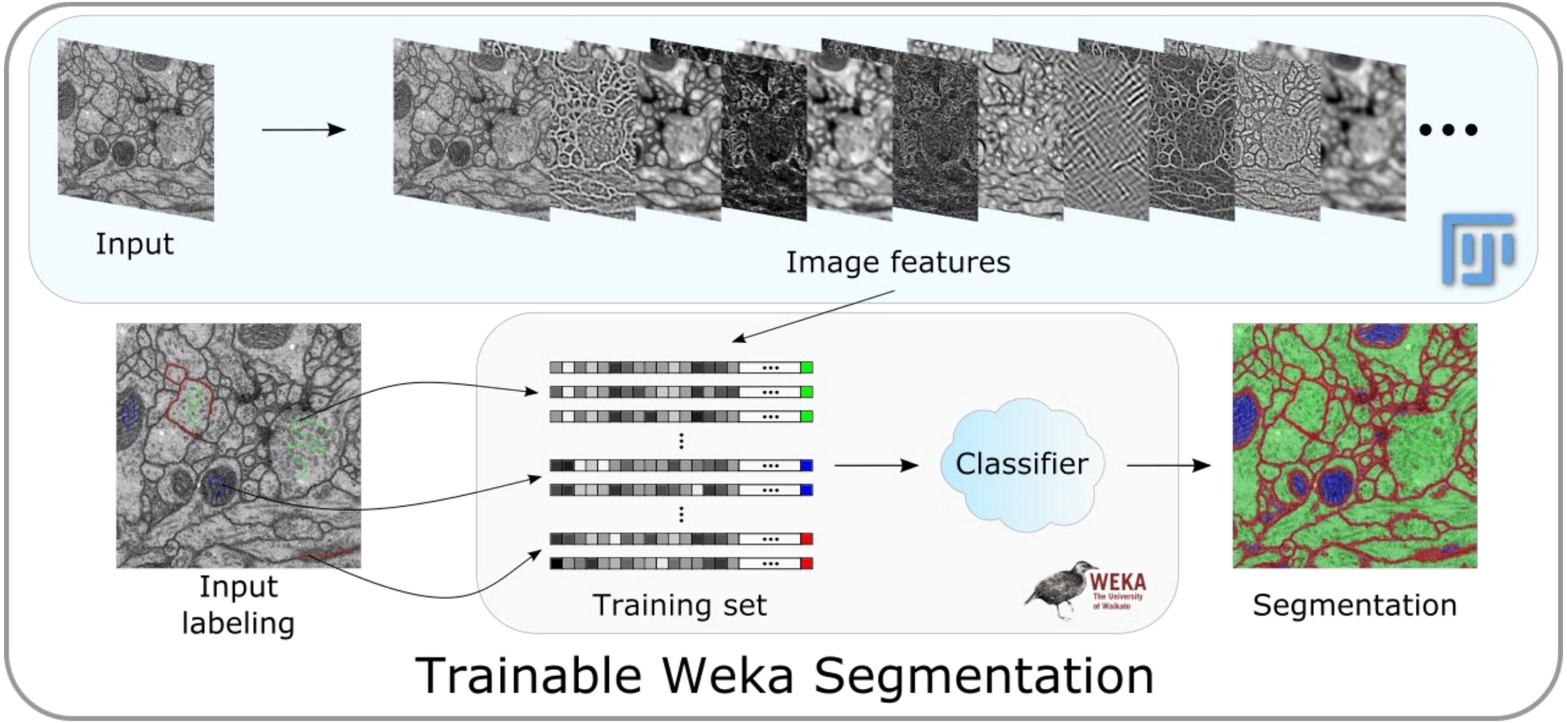
Back to real world

Shallow Machine Learning in ImageJ/Fiji

Trainable Weka Segmentation

Trainable segmentation basics

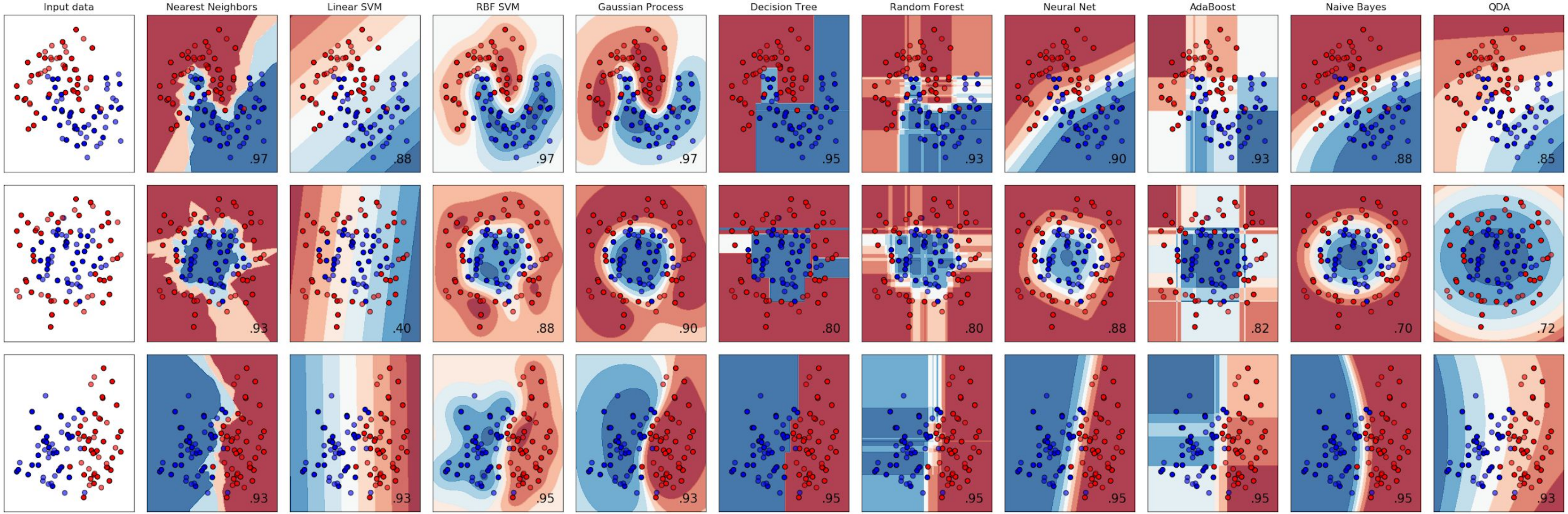
- Transform segmentation problem into pixel classification.



Classification algorithms

- Support Vector Machines (SVM).
- Decision Trees / Random Forests.
- Artificial Neural Networks.

scikit learn https://scikit-learn.org/

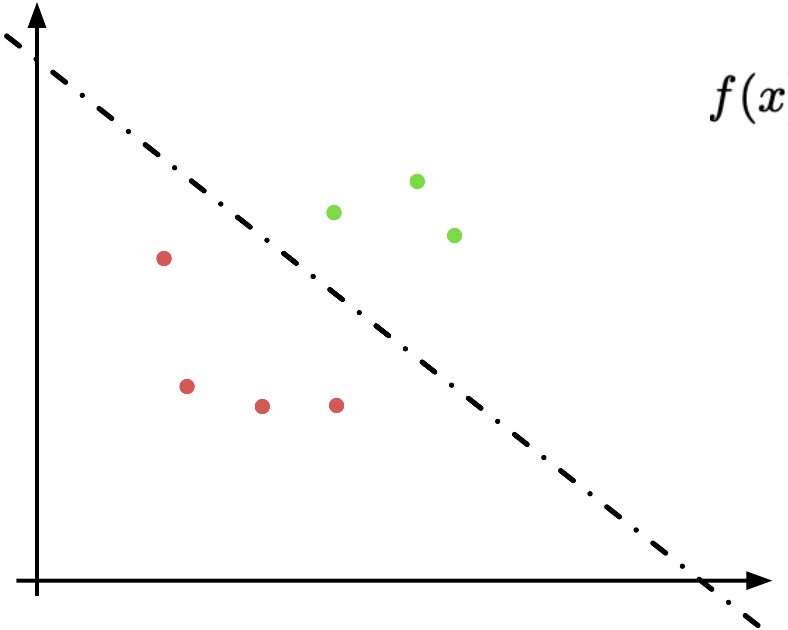


Slide courtesy: Martin Weigert, EPFL Lausanne

What about deep learning?

Linear classifiers: perceptron

- Perceptron (1956): Find a linear classifier f(x) such that



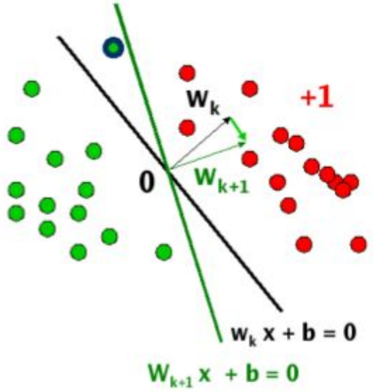
$$f(x) = \begin{cases} 1 & if \quad w \cdot x + b > 0 \\ 0 & otherwise \end{cases}$$

w: weights

b: bias

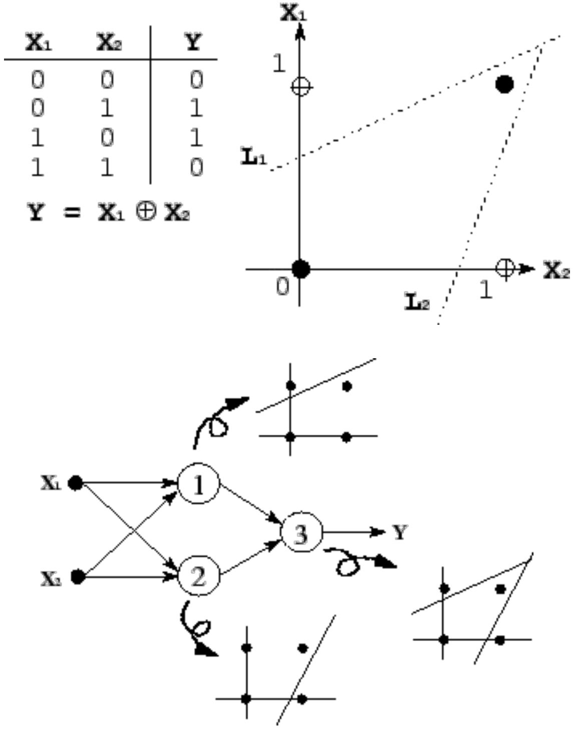
Linear classifiers: perceptron

- Separating two classes A and B
  - y = 1 if the example belongs to A
  - y = 0 if the example belongs to B
- Perceptron algorithm:
  - At iteration k, we consider example x:
    - w(k) = w(k-1) if x is correctly classified
    - w(k) = w(k-1) + (d-y) x otherwise (d: desired output)
- The algorithm stops once it finds a linear separation.
- It converges if and only if the examples are linearly separable.



Failures, promises…

- The perceptron (PCP) is a linear machine, so it can learn the AND and OR predicates… but it cannot learn XOR
- Linearly separable problems are very unlikely when sample size >> sample dimension
- But a PCP with **one hidden layer** can solve XOR
- And a PCP with **two hidden layers** can (essentially) solve any classification or regression problem
- However, there were no algorithms to build (learn) them from a given sample
- This led to the first NN winter (1969) and a concentration of AI efforts in symbolic systems...



X_1	X_2	Y
0	0	0
0	1	1
1	0	1
1	1	0

$Y = X_1 \oplus X_2$

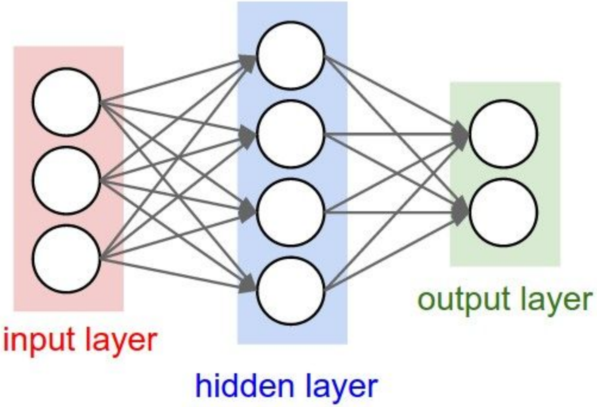
Source: K. Kawaguchi

Back in business

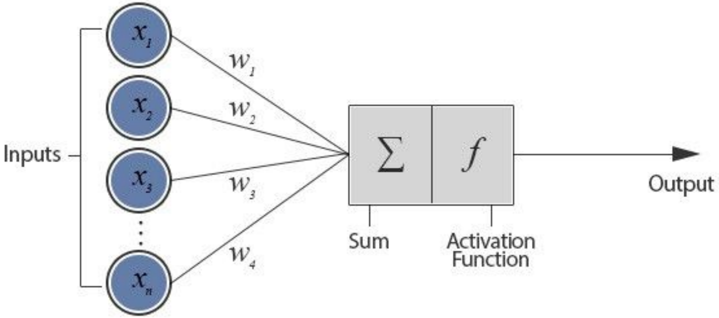
- We can summarize early work on PCPs by saying that
  - The architecture was right
  - But the training approaches were wrong
- Things change in 1986 with the book: *Parallel Distributed Processing. Explorations in the Microstructure of Cognition: Psychological and Biological Models*, J.L. McClelland, D.E. Rumelhart and the PDP Research Group (G. Hinton among them)
  - Grand title, but learning becomes error minimization (i.e., moves from AI to optimization)
- Multilayer PCPs (MLPs) became highly flexible and very efficient non–linear regression and classification machines.

Multi-layer perceptron architecture

- General organization:
    - 1 input layer, 1 or more hidden layers, 1 output layer
    - Each fully connected with feedforwad processing
- Many layered MLPs define a highly non–linear, weight–depending, transformation
- Learning's goal: minimization of a suitable error function
- Gradient can be easily computed by backpropagation
- This went on strongly until the late 90's when
    - New relevant contributions decrease
    - New competitors appear: Boosting, SVMs, Random Forest ⇒ Second Winter of NN.

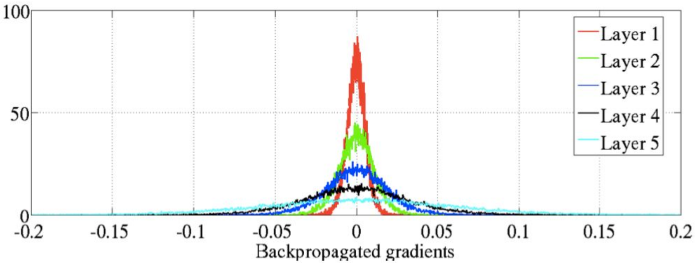


http://cs231n.github.io/neural-networks-1/



Neural Networks decline

- A nagging issue are deeper MLPs
- One hidden layer MLPs are enough for most applications
- But nobody knew how to train MLPs with three or more hidden layers
- One main obstacle was vanishing gradients: in a 5 layer MLP



From Glorot & Bengio, AISTATS 2010

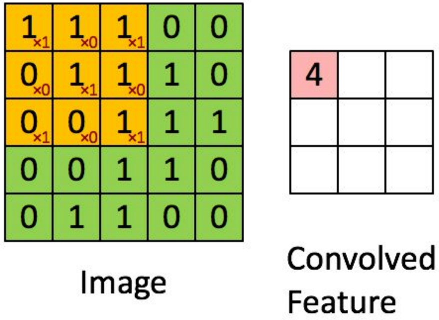
- Gradients in the last (5–th) layer are nonzero but vanish as we go back towards the first layer
- Inputs cease to have any effect and training stalls

Golden era: Deep Networks

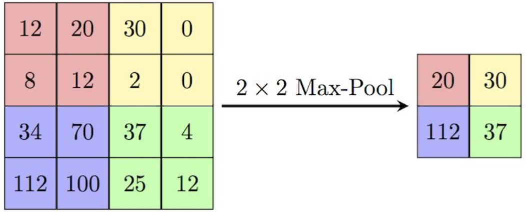
- First breakthrough around 2007: deep MLP unsupervised pretraining using stacked RBMs (Hinton) or autoencoders (Bengio)
  - Easier fine-tuning afterwards by standard back-propagation
- Floodgates opened: large nets with huge number of weights and new convolutional layers, regularizations, initializations or activations
  - New mood: what was impossible before is now much easier and leads to better results and even major breakthroughs in significant problems
  - New techniques appear ... that are not that different from the old ones
- 2012: the convolutional network AlexNet (Hinton group) wins the ImageNet challenge → the Deep Learning revolution starts.
- 2013: Google hires Geoff Hinton.
- 2014: Facebook hires Yann LeCun (father of the first convolutional network).
- 2018: Turing Award for Hinton, LeCun and Bengio.

Convolutional and pooling layers

- Starting assumption: patterns organized in features having a one, two (or multi) intrinsic dimensional structure
- Basic processing: to apply a $K \times K$ convolutional filter w over an image patch x_j as $y_j = f(w * x_j + b)$
- An $M_1 \times M_2$ input "image" X becomes an $(M1 - K + 1) \times (M2 - K + 1)$ output $X' = C(X)$
- This is followed by a pooling transformation P over L x L patches of X'
  - Possible transforms: averages, max



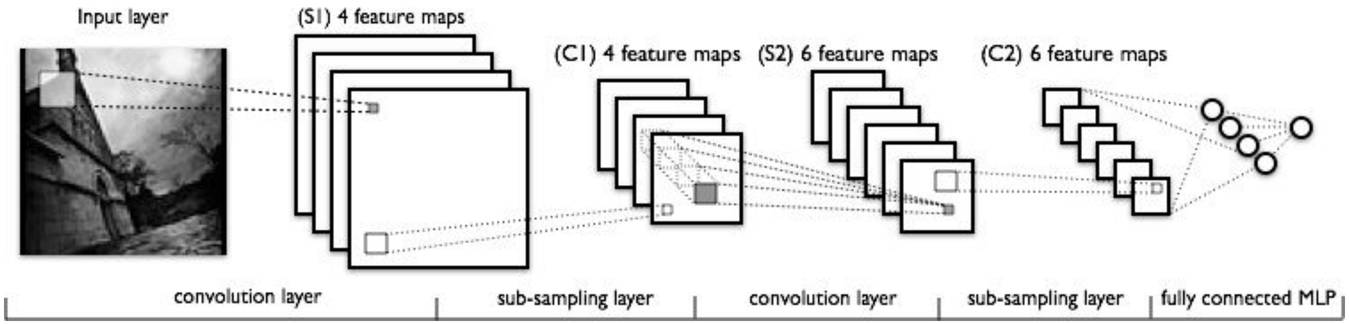
From UFLDL Stanford tutorial



From computersciencewiki.org

Deep Convolutional Networks

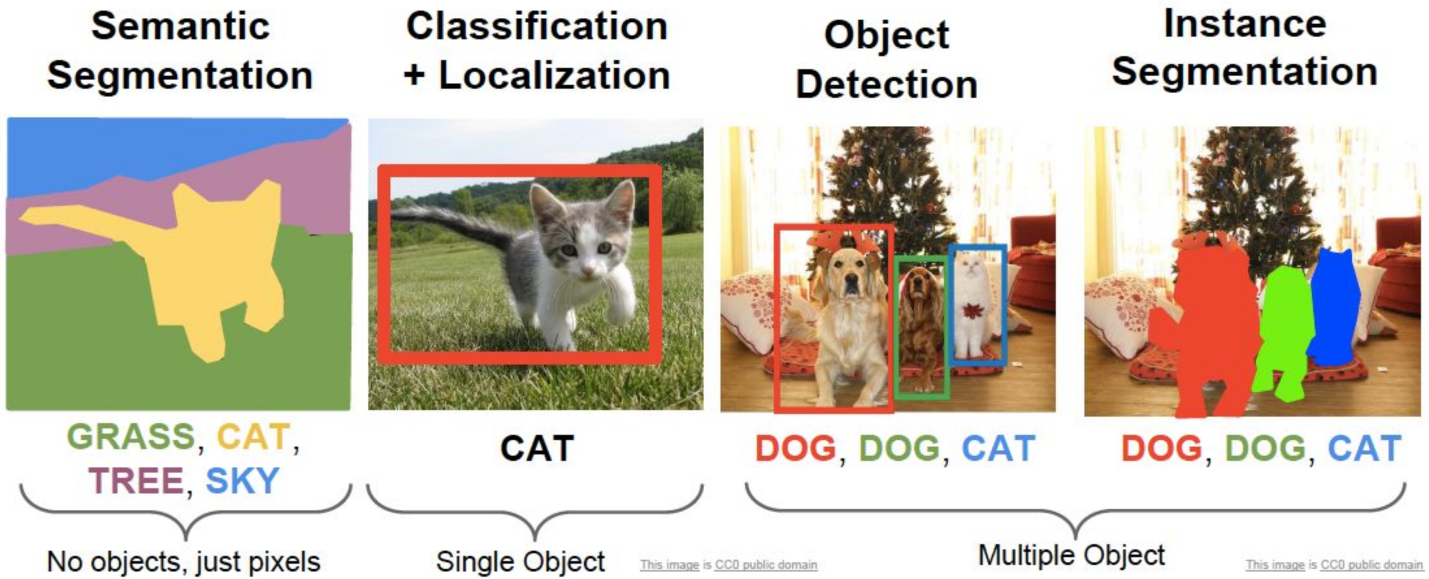
- The previous steps are often combined in a Deep Convolutional NN with
  - An initial number of convolutional layers, followed by
  - A number of fully connected inner product layers and, finally
  - A readout layer that yields the NN's response
- A typical architecture for image processing can be



From [Convolutional Neural Networks (LeNet) tutorial]

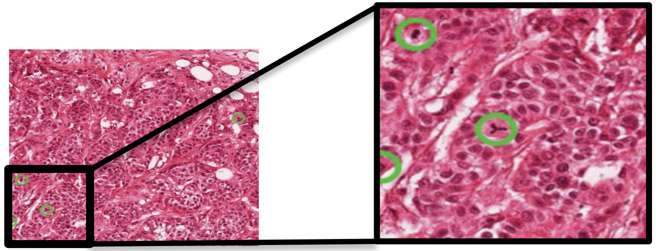
- Possibly with connections and weights in the millions

Many computer vision applications

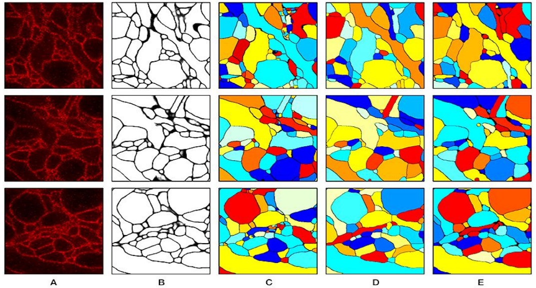


Current situation in Bioimage analysis

Mitosis Detection [1]

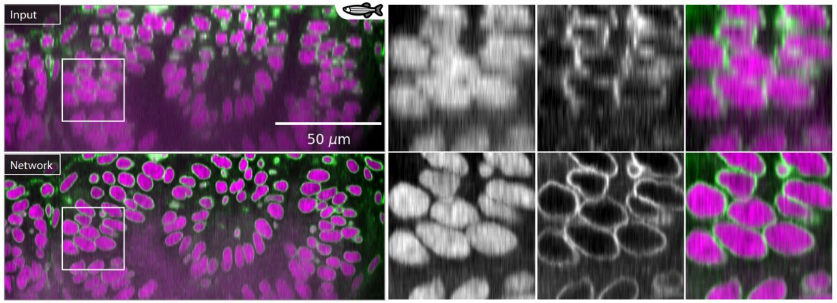


3D Segmentation [2]

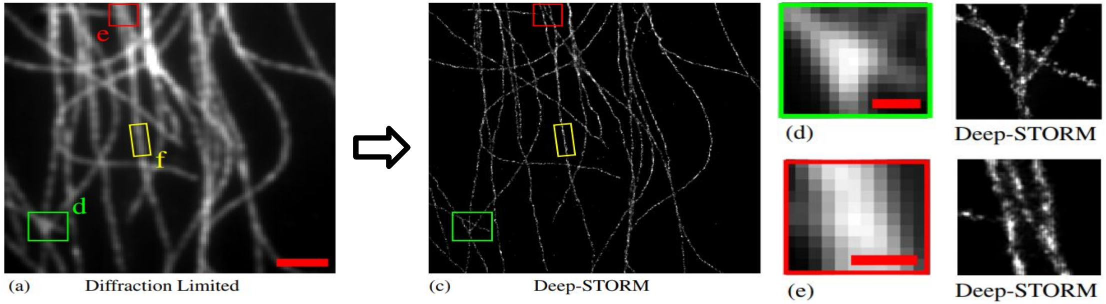


[1] Chen et al., *ISBI,* 2016.
[2] Yoon et al. Front. in Comp. Neuro., 2017.
[3] Weigert et al., Nat. Methods, 2018.
[4] Nehme et al., Optica, 2018

Context-aware image restoration [3]



Super-resolution [4]

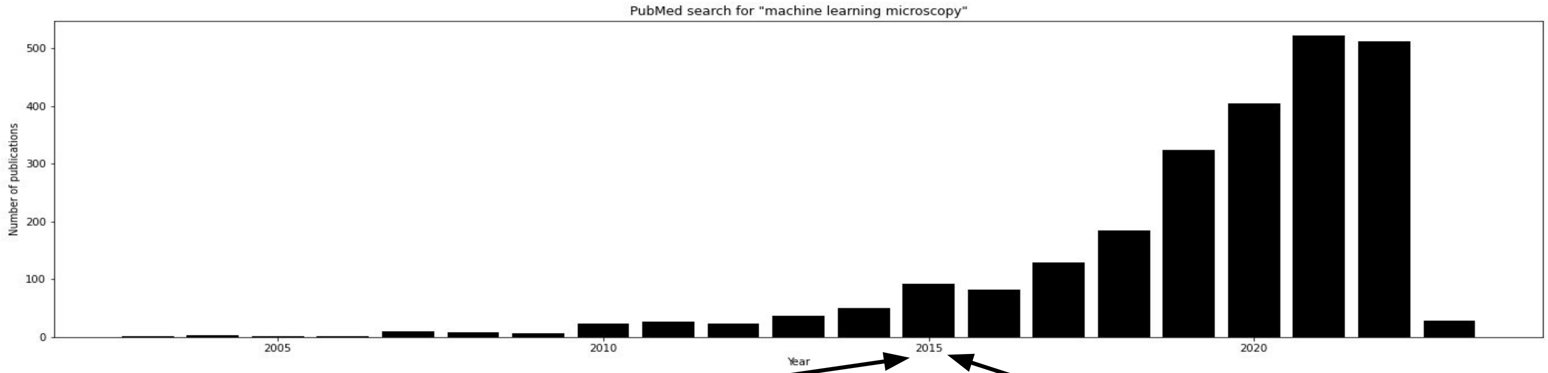


When did it start in the field?

Effective deep learning architectures

O. Çiçek, *et al.*, *3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation*, MICCAI 2016

O. Ronneberger, *et al.*, *U-Net: Convolutional Networks for Biomedical Image Segmentation*, MICCAI 2015



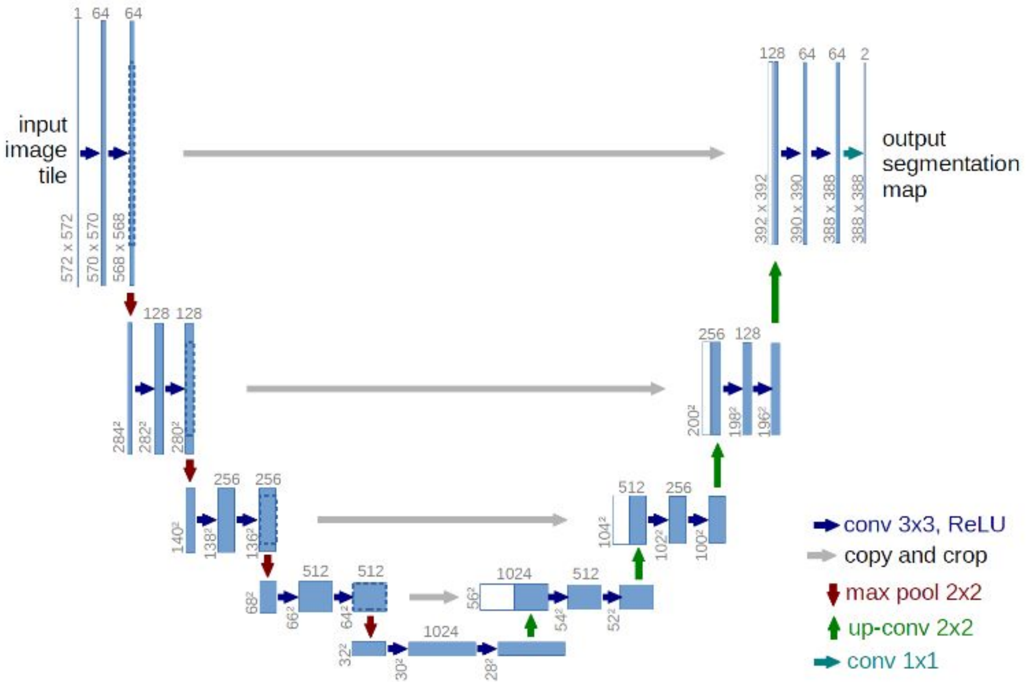
F. Chollet, *et al., https://keras.io* 2015

Deep Learning toolbox, Matlab 2015

User friendly libraries

Advantages of U-Net

- Image processing:
  - Contracting path extracts high dimension features ⇒ abstract analysis.
  - Expanding path refines the processing.
- Data augmentation applied to medical image processing.



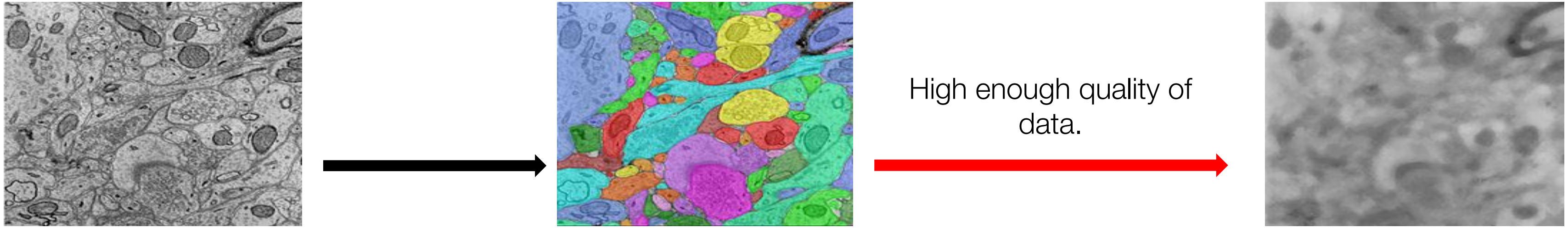
O. Ronneberger, et al., U-Net: Convolutional Networks for Biomedical Image Segmentation, MICCAI 2015.

What do you need?

The problem to solve by machine learning techniques has to be well defined.

Classification: the number of classes has to be determined and their description cannot be ambiguous

Segmentation: the result of any manual annotation when performed twice by an expert, should always coincide.



High enough quality of data.

What do you need?

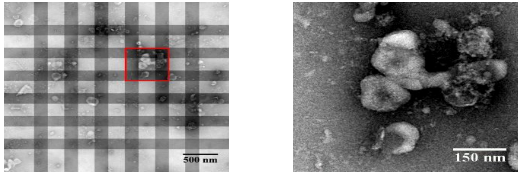
- To train your own model you need:
  - Technological infrastructure.
    - Graphics processing units (GPU).
    - Cloud computing (Google Colab, Kaggle notebooks, Amazon).
  - Data: Ground Truth (GT) ⇒ manual annotations supervised by experts.
    - GT has to represent the real scenario of the problem.
    - Large enough to train the model and evaluate it.

NVIDIA Quadro P5000

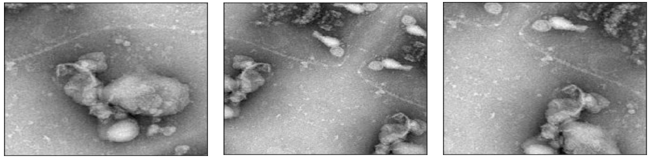
What do you need?

- Data augmentation.
- Patching
  - Geometrical transformations
    - Linear transformations (preserve shape)
      - Rotation
      - Translation
- (!) Non-linear (elastic) transformations (shape changes)
  - Zooming
  - Shearing
- (!) Add artifacts: noise

Patching

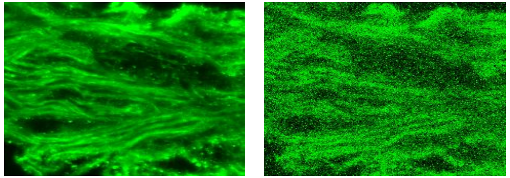


Linear transformations



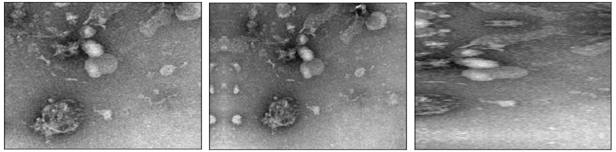
Original patch Rotation + Shift Rotation

Noising



Original image Noisy

Non-linear transformations



Original patch Zoom Shearing

Resources

Data annotation

- Slicer
- Polygon-RNN++
- ImageJ/Fiji
- Napari

Data repositories

- Kaggle
- Zenodo
- Cell tracking challenge

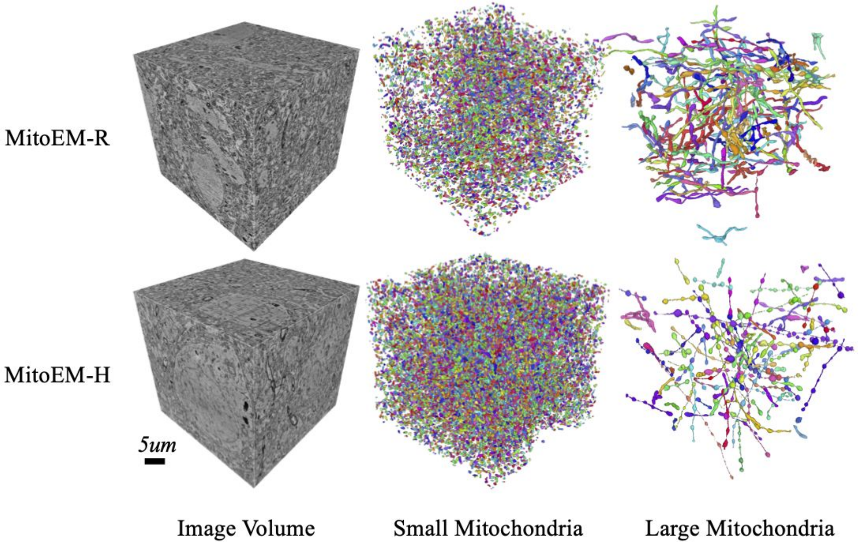
Deep learning software

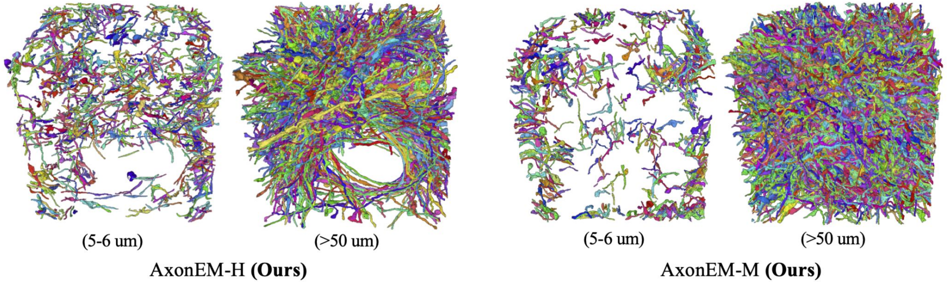
- Python (Tensorflow, Keras, PyTorch)
- Matlab
- C++ (Caffe)

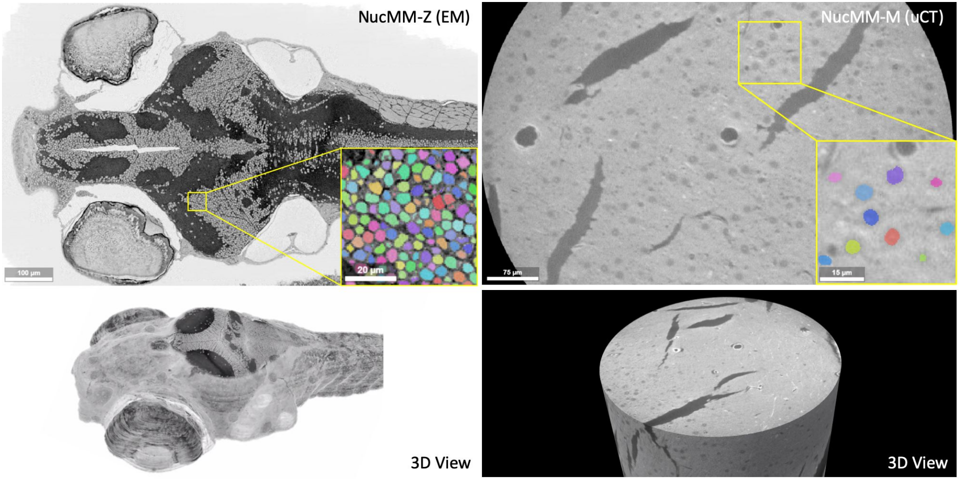
User friendly software

- ImageJ (U-Net, CARE, **deep**ImageJ)
- Cell profiler
- ZeroCostDL4Mic / DL4MicEverywhere
- Ilastik
- ImJoy
- BiaPy **New!**

Instance Segmentation Challenges!



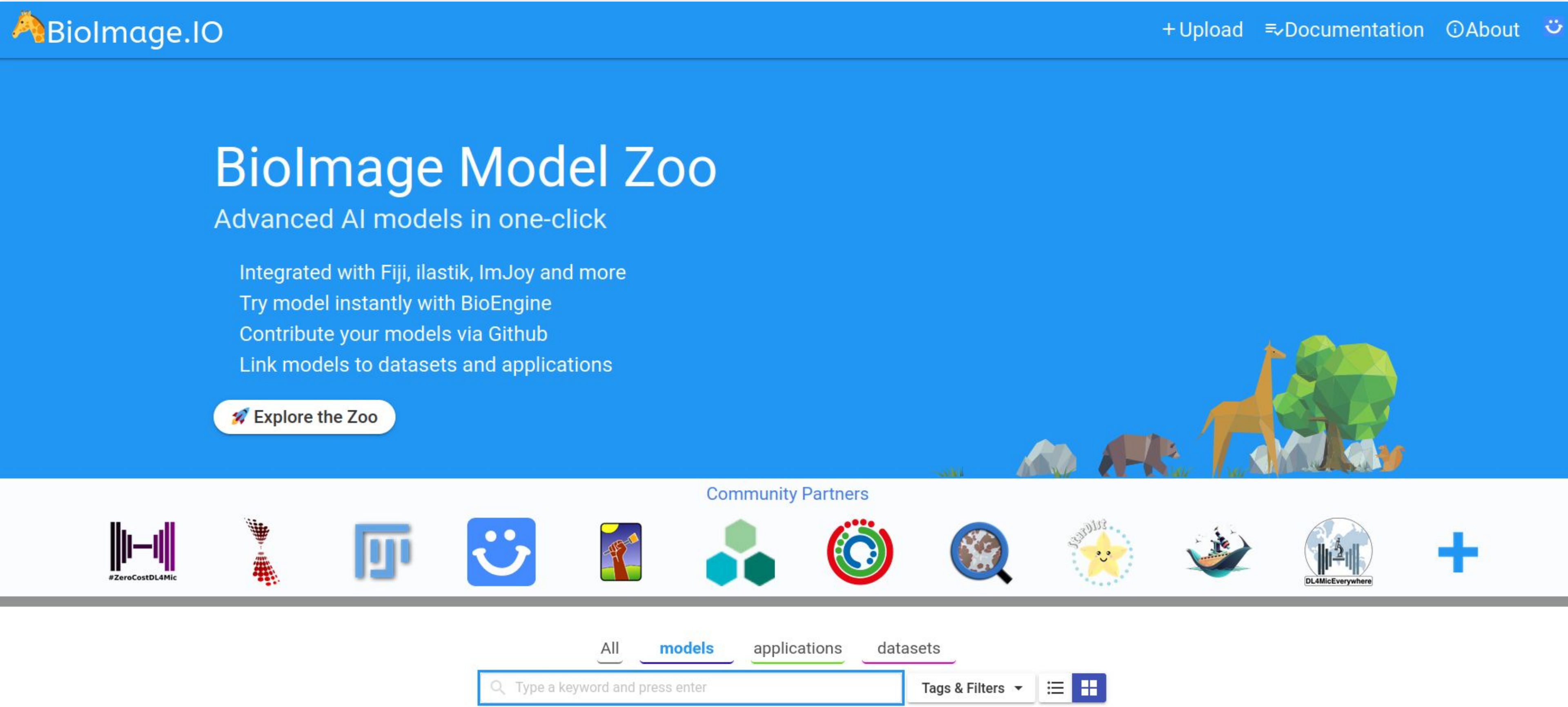




Wei et al., "[MitoEM](#) dataset: Large-scale 3D mitochondria instance segmentation from EM images", MICCAI 2020

Lin et al., "[NucMM](#) Dataset: 3D Neuronal Nuclei Instance Segmentation at Sub-Cubic Millimeter Scale", MICCAI 2021

Wei et al., "[AxonEM](#) Dataset: 3D Axon Instance Segmentation of Brain Cortical Regions", MICCAI 2021



BioImage Model Zoo

Advanced AI models in one-click

Integrated with Fiji, ilastik, ImJoy and more
Try model instantly with BioEngine
Contribute your models via Github
Link models to datasets and applications

Current related lab projects

BiaPy: BioImage Analysis Pipelines in Python

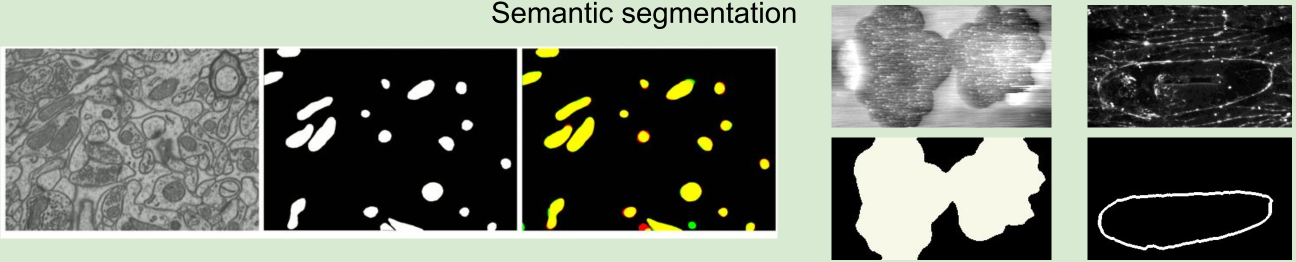
Daniel Franco-Barranco
(PhD student)

Donostia International Physics Center

eman ta zabal zazu

Universidad del País Vasco
Euskal Herriko Unibertsitatea

Semantic segmentation

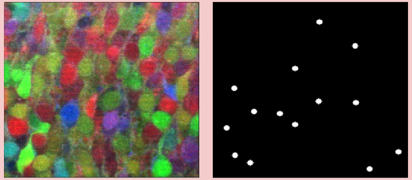


Instance segmentation

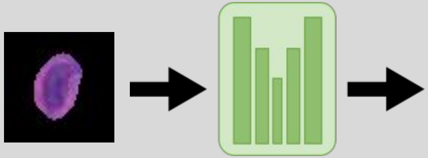
Do not miss Dani's workshop this afternoon at 2:30pm!

BiaPy

Detection



Classification



Class A
Class B
. . .
Class F

Wound healing modeling by video prediction

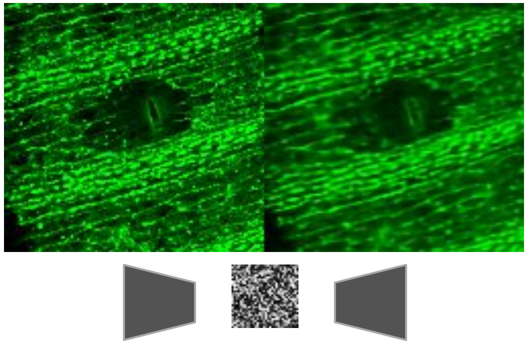
Given a set of initial frames, predict the next frames of a video

Lenka Backovà
(PhD student, Biofisika)

BIOFISIKA
Basque Centre for Biophysics

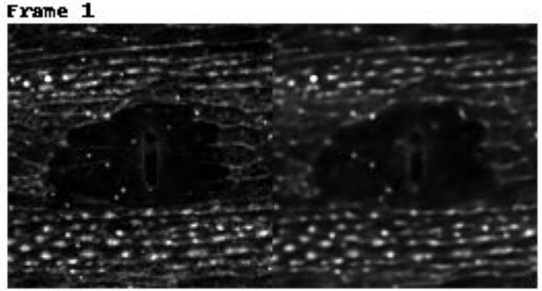
eman ta zabal zazu
Universidad del País Vasco
Euskal Herriko Unibertsitatea

(1) Frame encoding → (2) Prediction based on 8 initial frames:



Encoder

Decoder



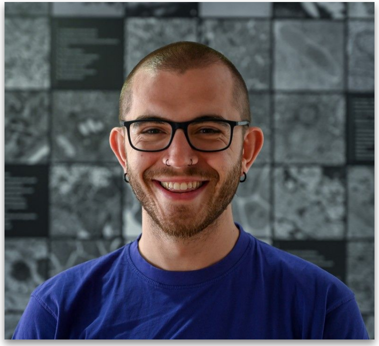
Ground truth

Prediction

neubias
network of european bioimage analysts

Super-resolution deep learning for microscopy

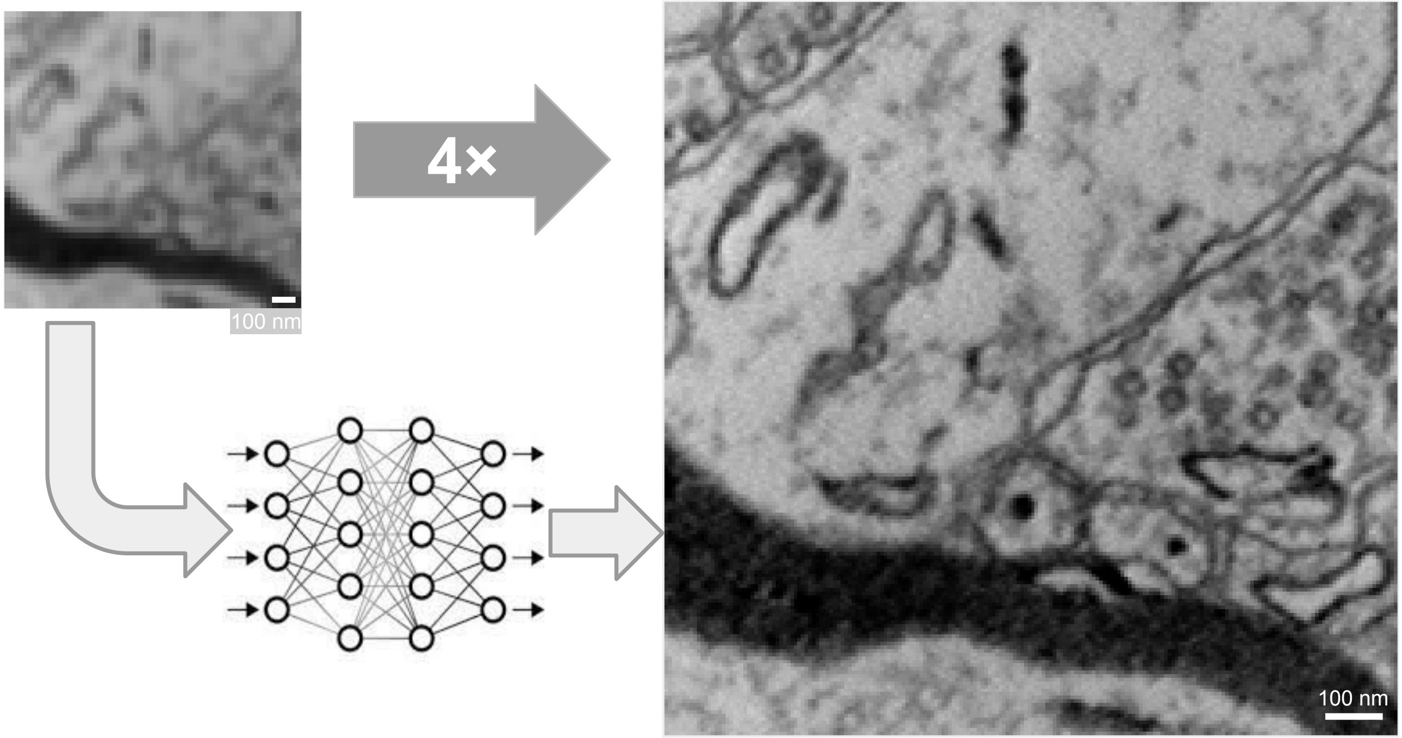
Ivan Hidalgo
(former Master student,
EHU)



Donostia International Physics Center

eman ta zabal zazu

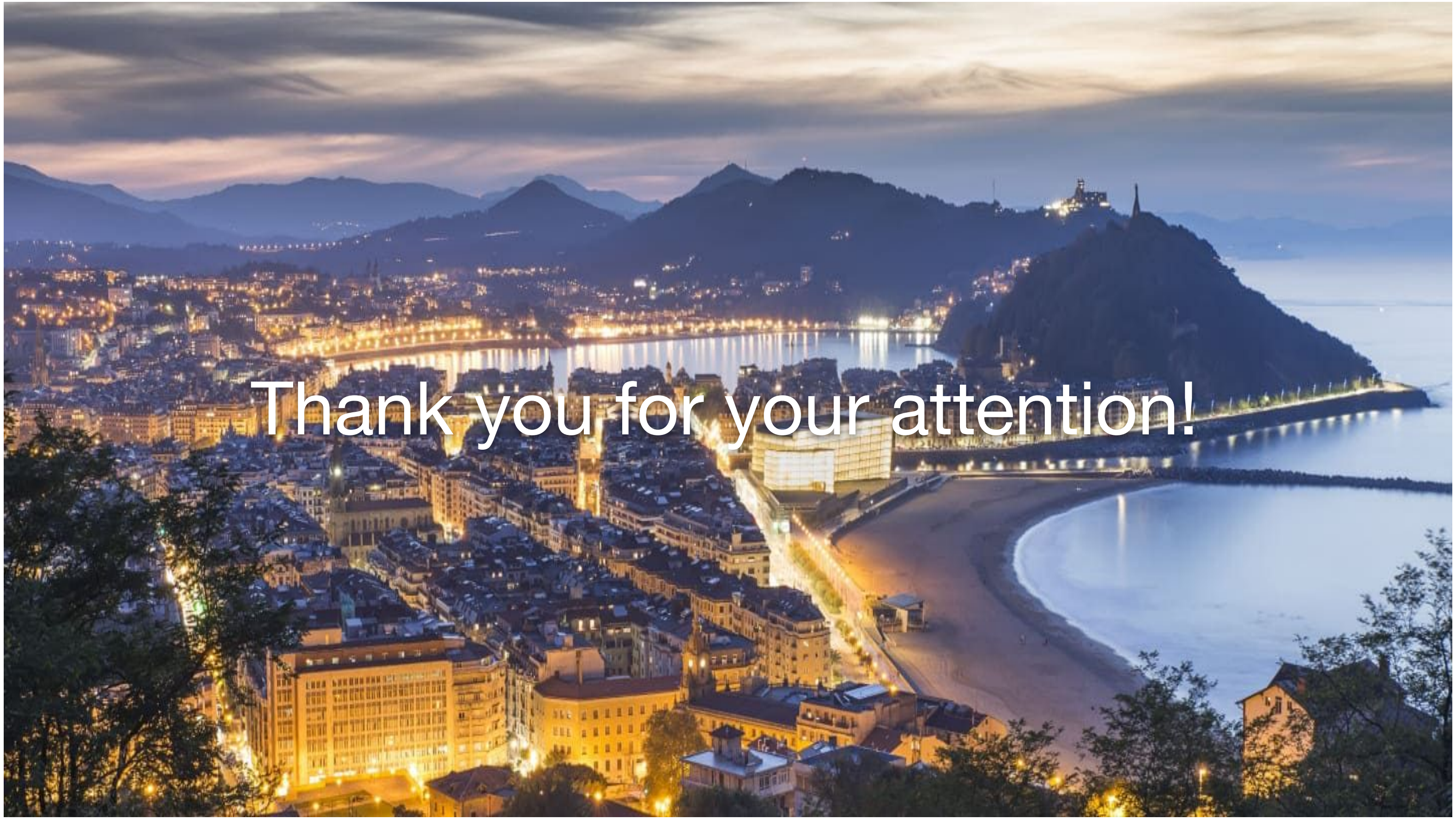
Universidad del País Vasco
Euskal Herriko Unibertsitatea



Take home messages

- Machine / Deep Learning is spreading out in the field of Bioimage Analysis.
- Tons of applications if you have:
  - 2D / 3D / ND images, videos…
- and need to do:
  - Classification,
  - detection / segmentation / tracking,
  - super-resolution…
- Drawbacks:
  - Generalization.
  - Interpretability.
  - Computing resources.

https://xkcd.com/1838/



Thank you for your attention!

neubias
network of european bioimage analysts

Slides credits and references

- Dr. Ulas Bagci, UCF, CAP5415-Computer Vision.
- Dr. Mubarak Shah, UCF, CAP5415-Computer Vision.
- Dr. Fadi Dornaika. Pattern Recognition master class, 2009.
- Conceptos y Métodos en Visión por Computador. Enrique Alegre, Gonzalo Pajares, Arturo de la Escalera. Capítulos 9-10.
- Erik Learned-Miller, University of Massachusetts, Amherst, CS370, Introduction to Computer Vision, "UNIT 3: Pattern Recognition and Classification.".
- Selim Aksoy (Bilkent University), "Introduction to Pattern Recognition", CS 551, Fall 2016.
- Deep Networks, J. Dorronsoro, EPS-IIC, UAM.
- Fei-Fei Li & Justin Johnson & Serena Yeung, Stanford, cs231, lecture 11.
- Estibaliz Gómez-de-Mariscal, "Machine learning - Deep learning, Applications to BioImage analysis", SPAOM2018.
- Arrate Muñoz-Barrutia, "deepImageJ, A user-friendly plugin to run deep learning models in ImageJ", SPAOM2019.
- Estibaliz Gómez-de-Mariscal, "deepImageJ: bridging deep learning to ImageJ", ISBI 2020.