INTRODUCTION TO DATA ANALYSIS, PATTERN RECOGNITION & ARTIFICIAL INTELLIGENCE

Raúl Benítez
raul.benitez@upc.edu

CREB UPC
UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Centre de Recerca en Enginyeria Biomèdica

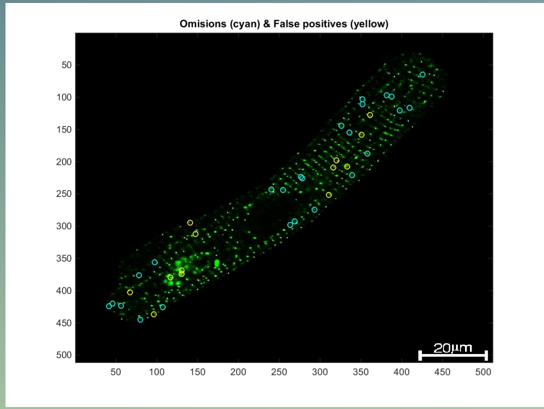
A. What is AI?

B. How does it work?

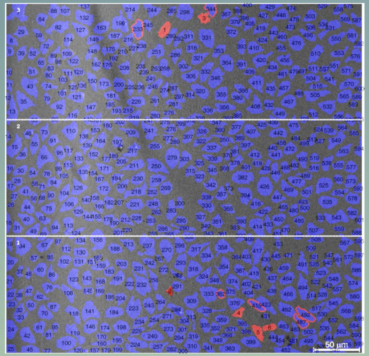
C. Risks

ANALYSIS OF MEDICAL & BIOLOGICAL IMAGES

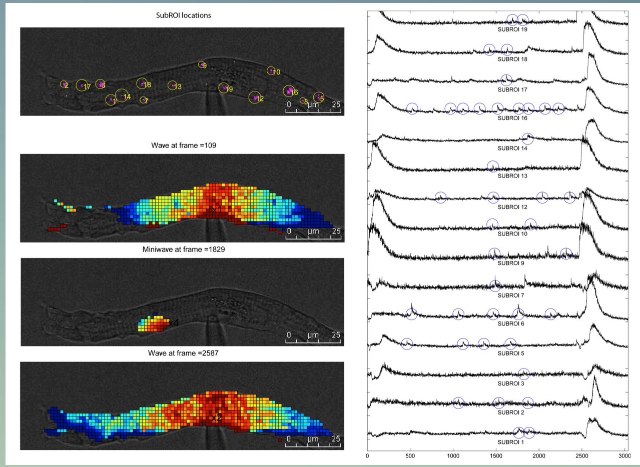
Molecular receptors



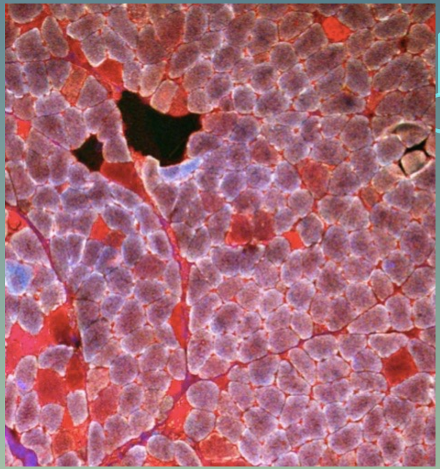
Cell cultures



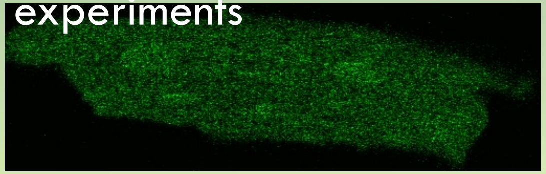
Calcium dynamics



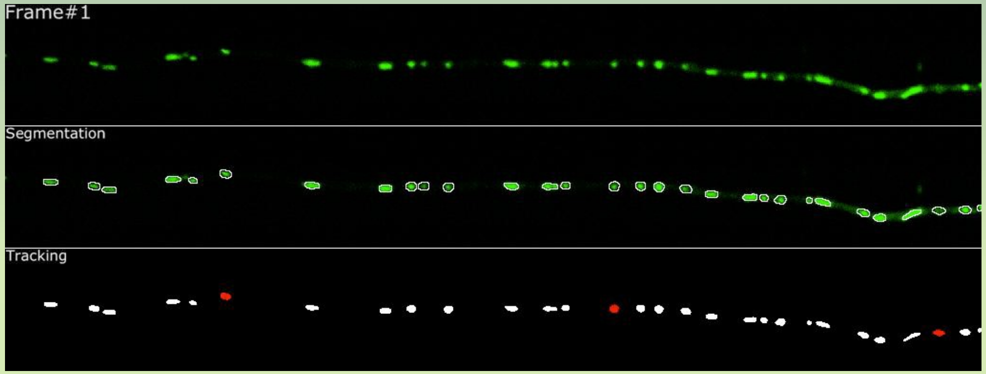
Pathology



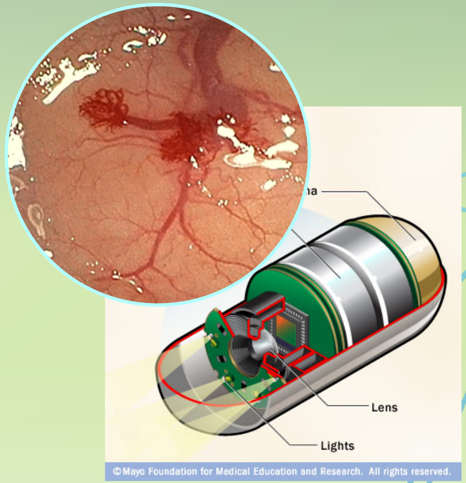
Single-cell experiments



Mitochondrial transport



Endoscopy

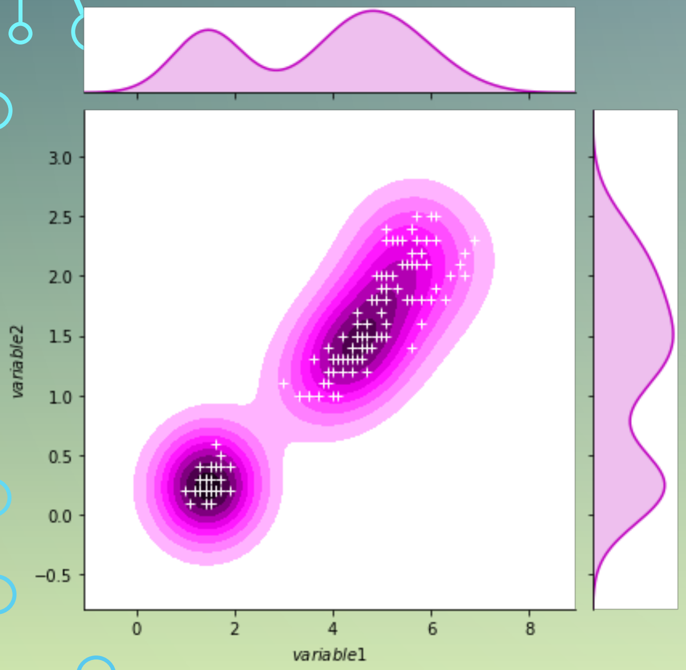


Extraction & analysis of multiscale biomarkers

A. WHAT IS AI?


artificial Intelligence
data analysis
machine learning
unsupervised learning
pattern recognition
data modeling
expert systems
statistical inference
deep learning
big data
business intelligence
knowledge retrieval
cybernetics
data analytics

Data



Text
Images
Multivariate numerical data
Genetics
Audio, video
HETEROGENEOUS

Mathematics

$$
\begin{aligned}
p(\mathcal{D}|\theta) &= p(x_1, x_2, \ldots, x_n|\mu, \sigma^2) \\
&= \prod_{i=1}^{n} p(x_i|\theta) \\
&= \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x_i-\mu)^2}{2\sigma^2}\right) \\
&= \left(\frac{1}{2\pi\sigma^2}\right)^{\frac{n}{2}} \exp\left(-\frac{\sum_{i=1}^{n}(x_i-\mu)^2 + n(\frac{1}{n}\sum_{i=1}^{n} x_i - \mu)^2}{2\sigma^2}\right)
\end{aligned}
$$

Statistics
Geometry
Optimization
Stochastic processes

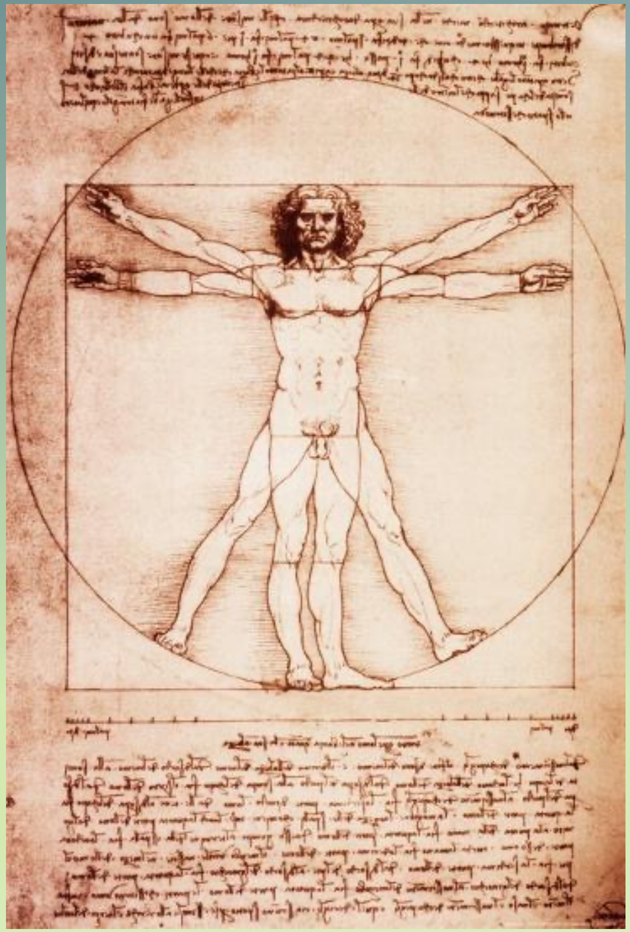
Computer Science

```python
import numpy as np
from sklearn import decomposition
from sklearn import datasets

iris = datasets.load_iris()
X = iris.data
y = iris.target

pca = decomposition.PCA(n_components=2)
pca.fit(X)
Xproj = pca.transform(X)
```

Algorithms
Computational complexity
Information theory
Network analysis

We emulate human body

Biomechanics



Humanoid robotics

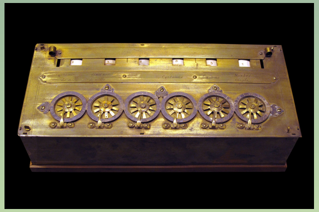
Robotics

WE EMULATE HUMAN MIND

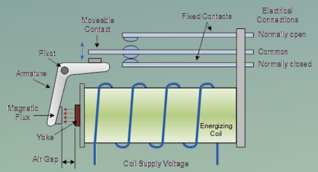
Cajal & Golgi Nobel 1906
Hebbian learning1949
Hodkin & Huxlery 1952 Nobel 1963
Hubel & Wiesel 1959 Visual Cortex Nobel 1981

Techology & IA

Mechanical calculators (s.XVII)



Electromechanical relay (s. XIX)

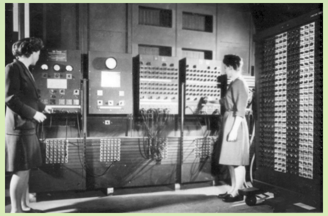


Telegraph 1830

Enigma machine 1918

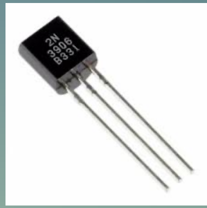
Vacuum tube(1904)

ENIAC 1946



The transistor (1947)

Nobel 1956



IBM 608 1957

GPUs (2018)

IA today

RESEARCH

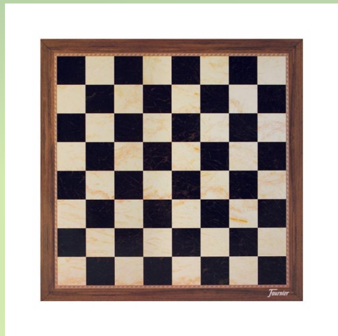
COMPUTER SCIENCE

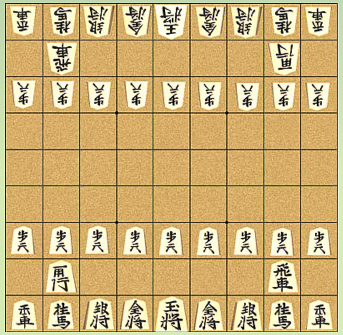
A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play

David Silver[1,2]*†, Thomas Hubert[1]*, Julian Schrittwieser[1]*, Ioannis Antonoglou[1], Matthew Lai[1], Arthur Guez[1], Marc Lanctot[1], Laurent Sifre[1], Dharshan Kumaran[1], Thore Graepel[1], Timothy Lillicrap[1], Karen Simonyan[1], Demis Hassabis[1]†

The game of chess is the longest-studied domain in the history of artificial intelligence. The strongest programs are based on a combination of sophisticated search techniques, domain-specific adaptations, and handcrafted evaluation functions that have been refined by human experts over several decades. By contrast, the AlphaGo Zero program recently achieved superhuman performance in the game of Go by reinforcement learning from self-play. In this paper, we generalize this approach into a single AlphaZero algorithm that can achieve superhuman performance in many challenging games. Starting from random play and given no domain knowledge except the game rules, AlphaZero convincingly defeated a world champion program in the games of chess and shogi (Japanese chess), as well as Go.

Science, *362* (6419), 1140-1144 (2018)



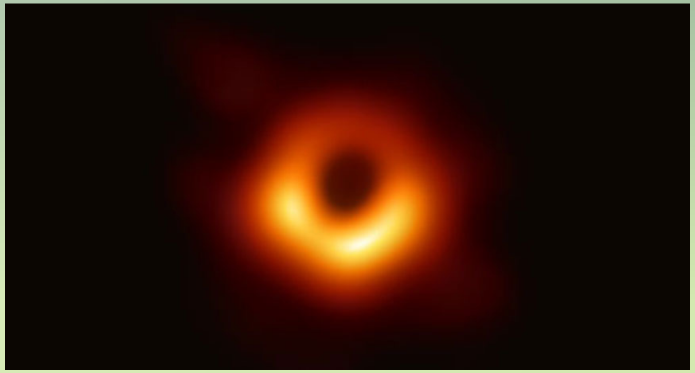


COMPUTATIONAL COMPLEXITY

First blach hole image 2019

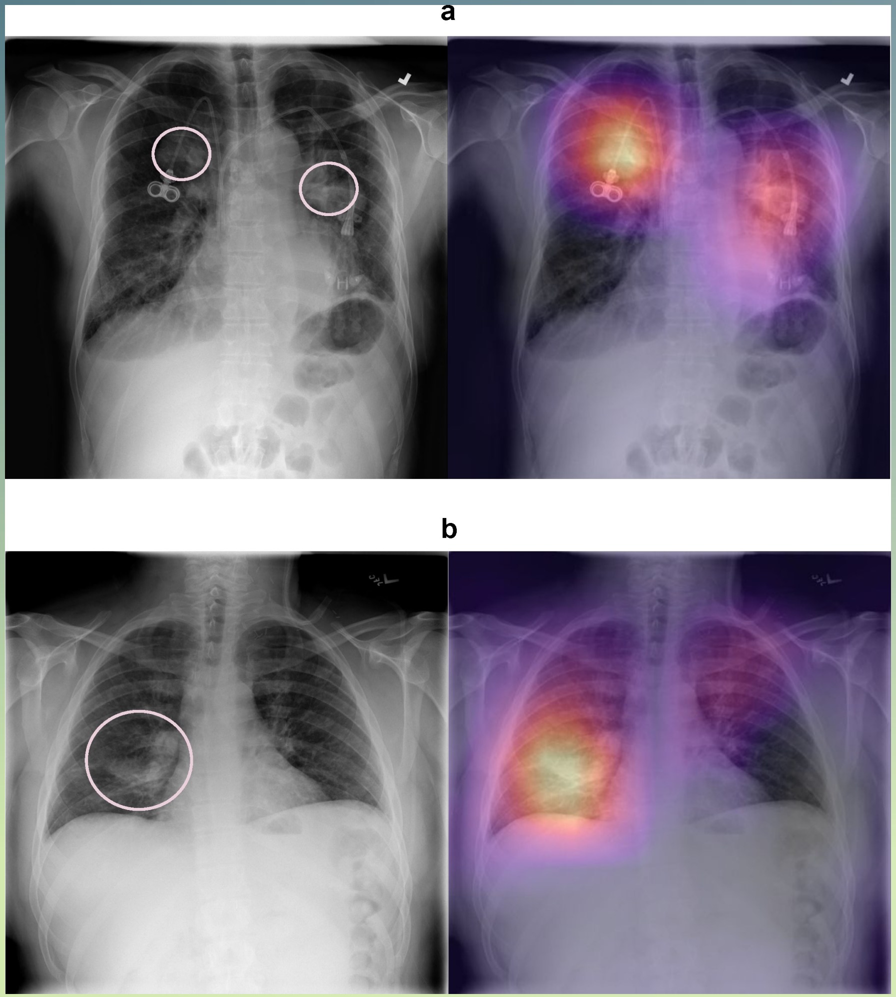
Event Horizon Telescope (EHT)

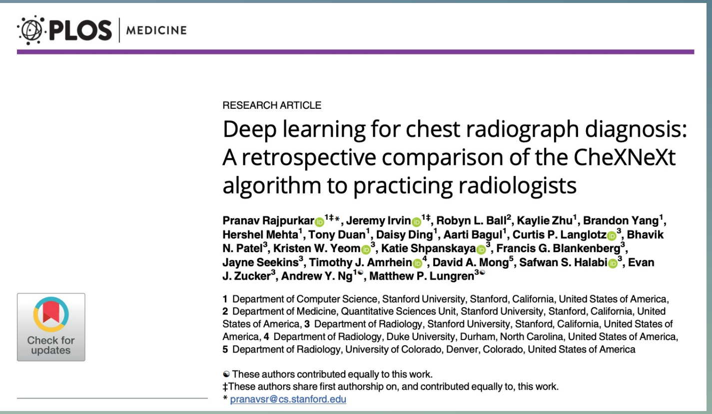
The Astrophysical Journal Letters



Applications

PLOS Medicine 15(11): e1002686 (2018)




PLOS | MEDICINE

RESEARCH ARTICLE

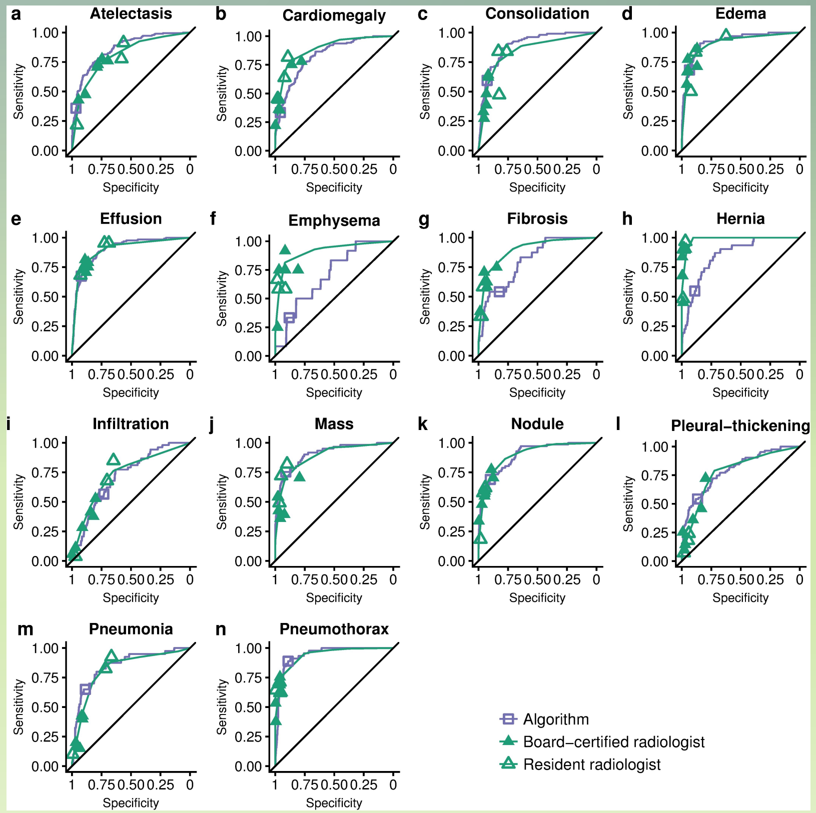
Deep learning for chest radiograph diagnosis: A retrospective comparison of the CheXNeXt algorithm to practicing radiologists

Pranav Rajpurkar[1‡*], Jeremy Irvin[1‡], Robyn L. Ball[2], Kaylie Zhu[1], Brandon Yang[1], Hershel Mehta[1], Tony Duan[1], Daisy Ding[1], Aarti Bagul[1], Curtis P. Langlotz[3], Bhavik N. Patel[3], Kristen W. Yeom[3], Katie Shpanskaya[3], Francis G. Blankenberg[3], Jayne Seekins[3], Timothy J. Amrhein[4], David A. Mong[5], Safwan S. Halabi[3], Evan J. Zucker[3], Andrew Y. Ng[1☯], Matthew P. Lungren[3☯]

1 Department of Computer Science, Stanford University, Stanford, California, United States of America, 2 Department of Medicine, Quantitative Sciences Unit, Stanford University, Stanford, California, United States of America, 3 Department of Radiology, Stanford University, Stanford, California, United States of America, 4 Department of Radiology, Duke University, Durham, North Carolina, United States of America, 5 Department of Radiology, University of Colorado, Denver, Colorado, United States of America

☯ These authors contributed equally to this work.
‡These authors share first authorship on, and contributed equally to, this work.
* pranavsr@cs.stanford.edu

Check for updates



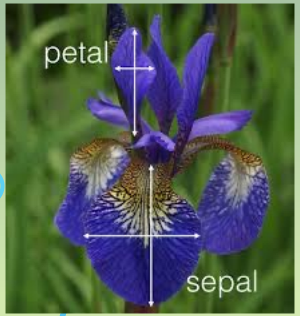
14 pathologies 69,682,060 parameters

B. HOW DOES IT WORK?

MATHEMATICAL REPRESENTATION OF DATA

$$A\colon\ m \text{ observations} \times n \text{ features}$$

$$A_{m\times n} = \begin{pmatrix} x_1^1 & x_2^1 & \cdots & x_n^1 \\ x_1^2 & x_2^2 & \cdots & x_n^2 \\ x_1^3 & x_2^3 & \cdots & x_n^3 \\ \cdots & & & \\ x_1^m & x_2^m & \cdots & x_n^m \end{pmatrix}$$



	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

Class labels $w(i)$

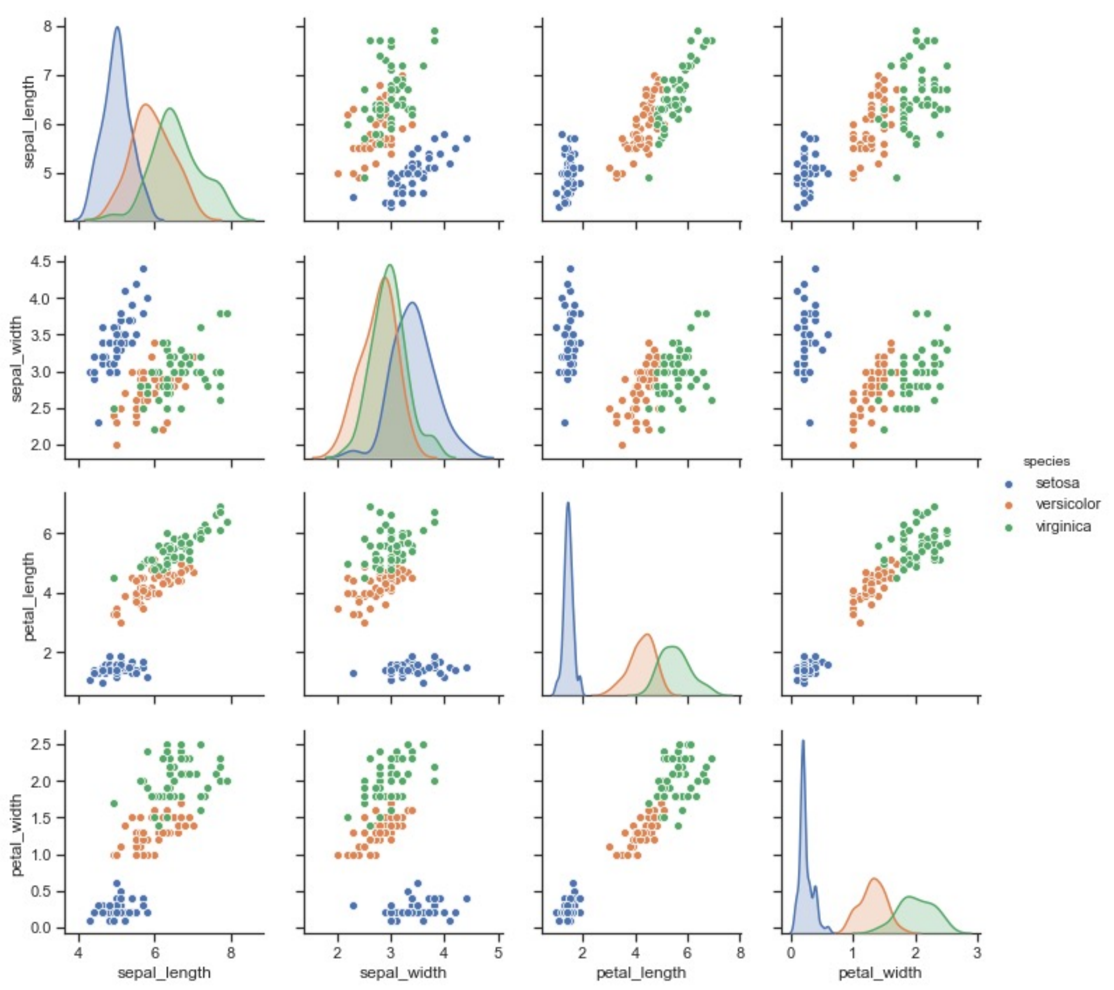
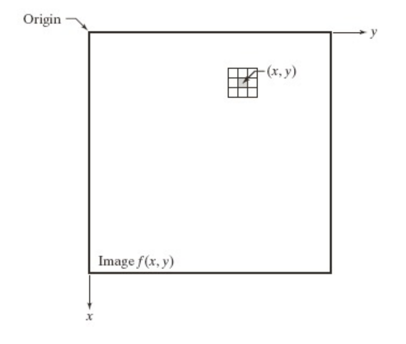
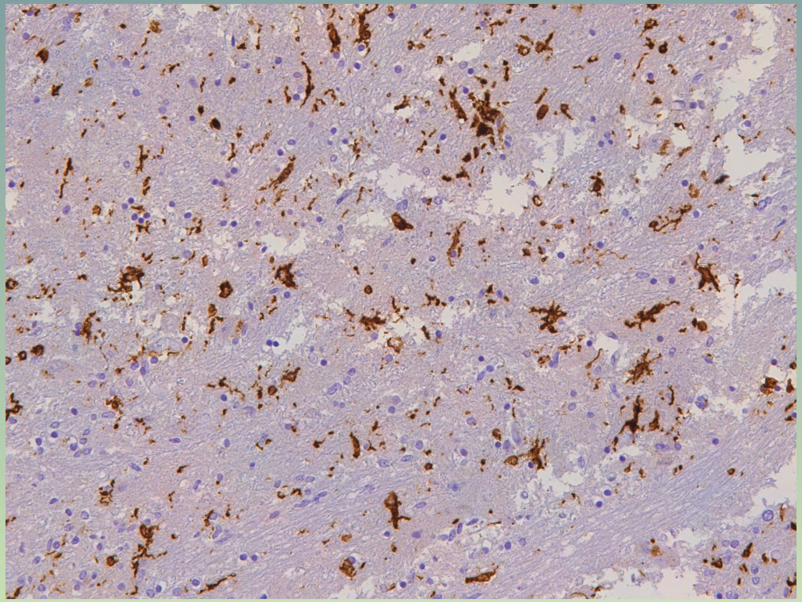


IMAGE DATA

grayscale



color images



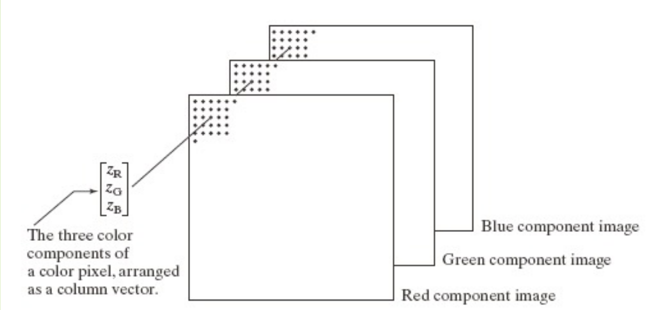
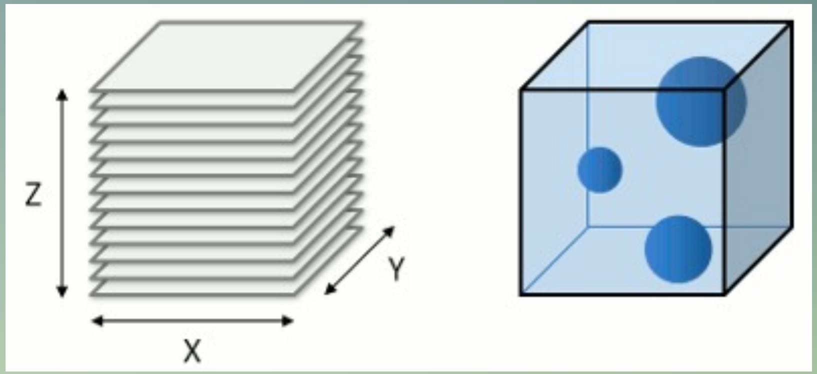


FIGURE 7.1 Schematic showing how pixels of an RGB color image are formed from the corresponding pixels of the three component images.

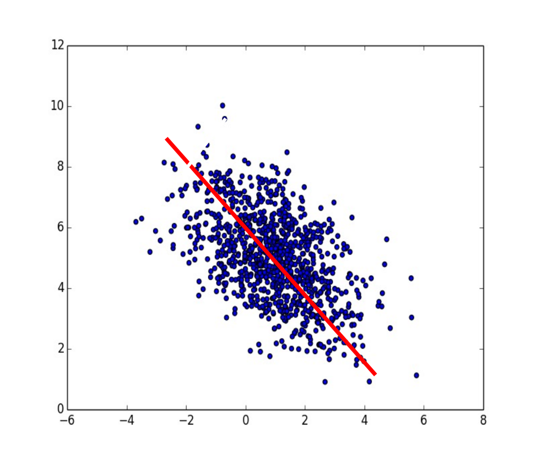
3D stacks



video sequences

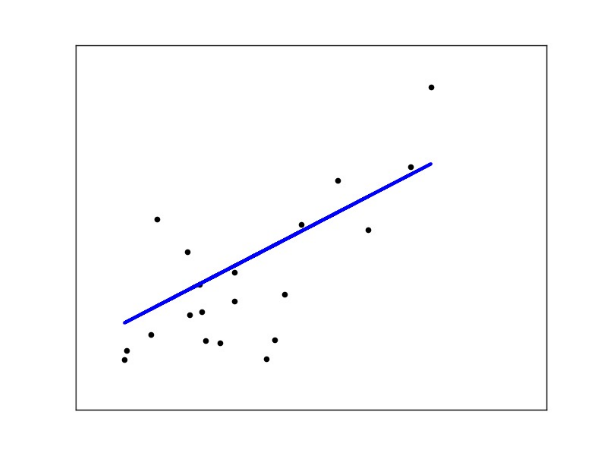
PATTERN RECOGNITION

DIMENSIONALITY REDUCTION



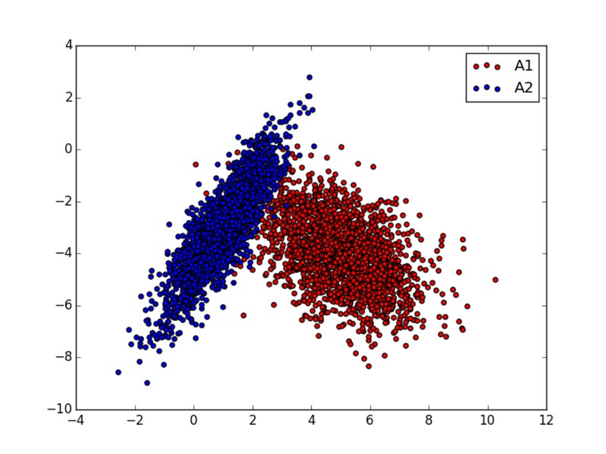
Principal Component Analysis
Singular Value Decomposition
Independent Component Analysis

MULTIVARIATE REGRESSION



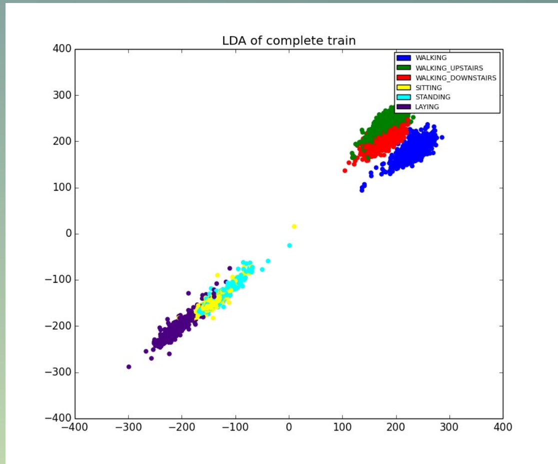
Multivariate linear regression
Regression using decision trees
Support Vector Machines

SUPERVISED CLASSIFICATION



Discriminant Analysis
Artificial Neural Networks
Support Vector Machines
Bayesian methods
Decision Trees
Ensemble classifiers
Deep Learning

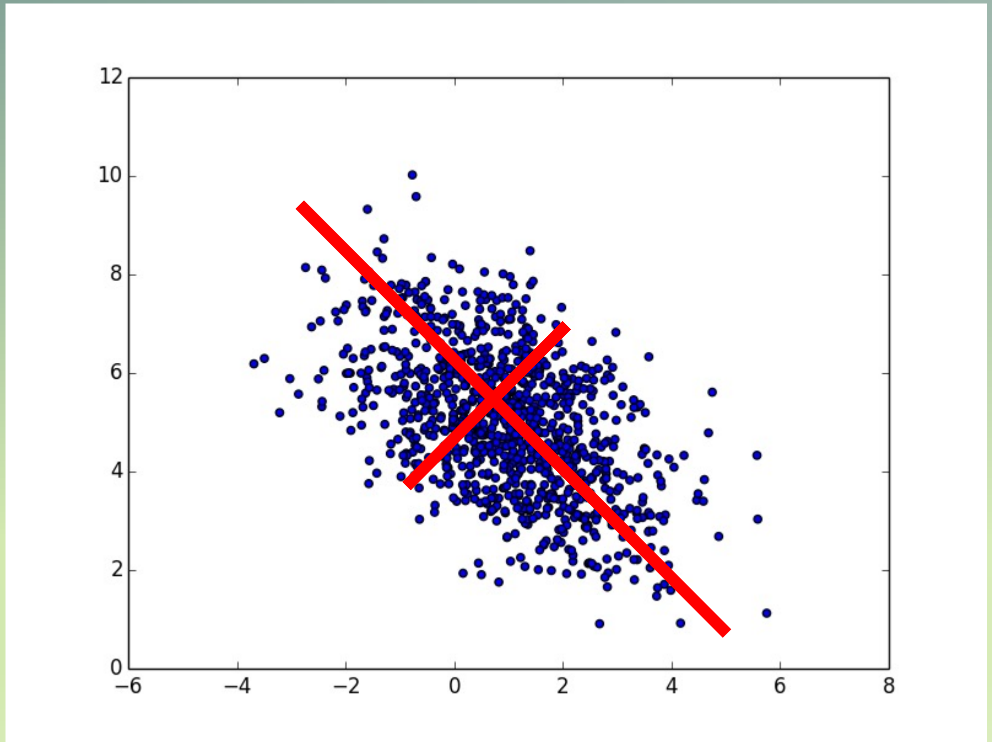
CLUSTER ANALYSIS



K-means
Hierarchical clustering
Gaussian Mixture Methods

DIMENSIONALY REDUCTION
PRINCIPAL COMPONENT ANALYSIS (PCA)

Reduce dimensionality of data



Mathematics

$$C_{n\times n} = (A - \bar{A})^T (A - \bar{A})$$

$$C \cdot \vec{v}_i = \lambda_i \cdot \vec{v}_i, \; i = 1 \ldots n$$

Computer Science

python™

```
# Obtain covariance matrix:
A1 = A - A.mean(0)
matcov = dot(A1.transpose(),A1)

# Diagonalization of covariance matrix:
valp,vecp = linalg.eig(matcov)

ind_creciente = argsort(valp) # sort eigenvalues
```

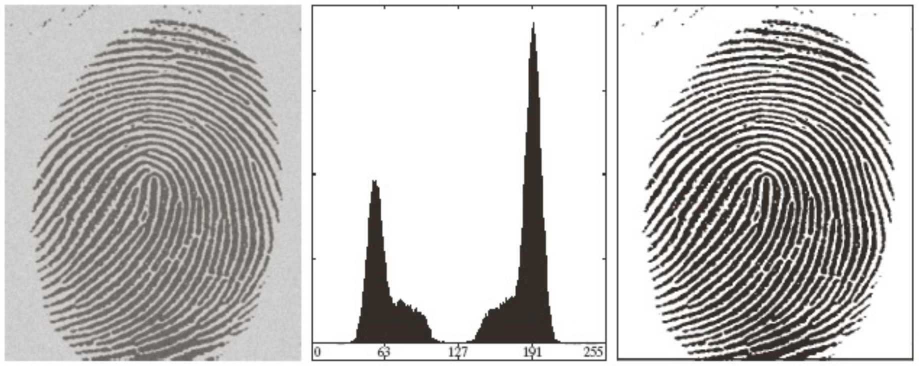
TRADITIONAL MACHINE LEARNING

Processament d'imatges

FIltering
Artefact removal

Segmentation

Feature extraction



a b c

FIGURE 11.13 (a) Noisy fingerprint. (b) Histogram. (c) Segmented result using a global threshold (the border was added manually for clarity). (Original courtesy of the National Institute of Standards and Technology.)

features

- Shape
- Size
- Color
- Texture
- Orientation
- Contours
- Intensity

Pattern Recognition

Supervised classification

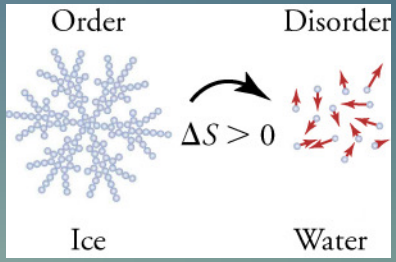
$$A_{m\times n} = \begin{pmatrix} x_1^1 & x_2^1 & \cdots & x_n^1 \\ x_1^2 & x_2^2 & \cdots & x_n^2 \\ x_1^3 & x_2^3 & \cdots & x_n^3 \\ \cdots & & & \\ x_1^m & x_2^m & \cdots & x_n^m \end{pmatrix} \qquad w(i)$$

Fase 1: TRAINING
Fase 2: TEST

SHANNON'S ENTROPY

Physics: Measure of the disorder of a system

$$S = k \log W$$



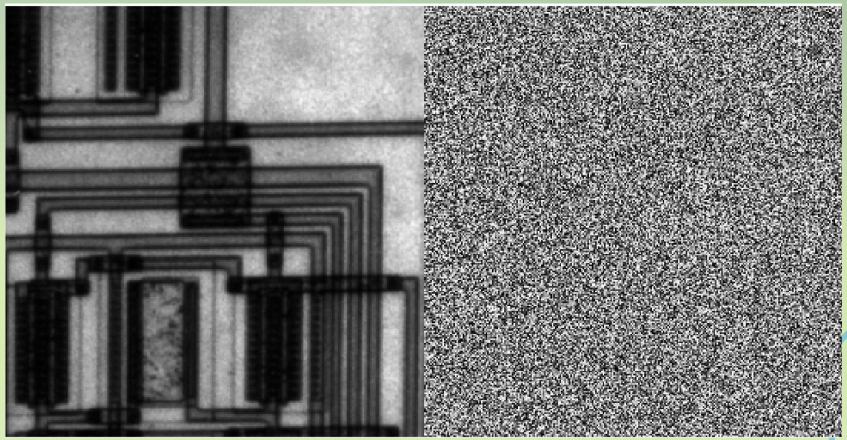
Information theory: Information content

Set of symbols (alphabet): $a_1, a_2, a_3, \ldots, a_n$

Probability of occurrence of each symbol in a message $p_1, p_2, p_3, \ldots, p_n$

$$S = -\sum_{i=1}^{n} p_i \log p_i$$

$S = 6.9439$ bits/pixel $\quad$ $S = 8$ bits/pixel



Example: Grayscale image

Alphabet: grayscale intensity levels $0, 1, 2, \ldots, 255$

Probability of occurence: Histogram $p_1, p_2, p_3, \ldots, p_n$

Low entropy ➔ less bits needed to encode image ➔ image compression

IMAGE RECOGNITION

Database of labelled images

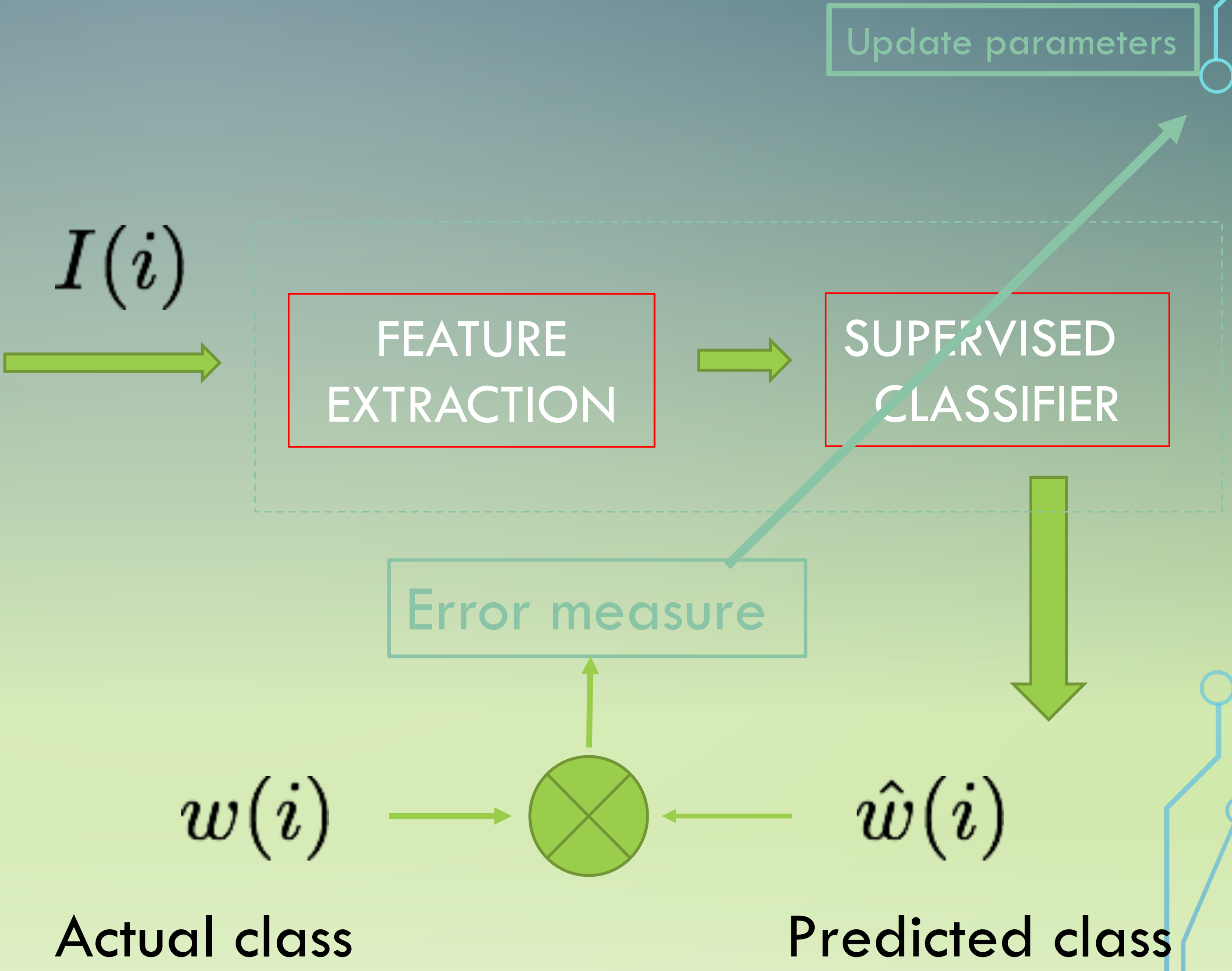
- **Images** $I(i),\, i = 1, \ldots N_{im}$

- **Class labels** (name, happy/sad, healthy/ii, etc.) $w(i)$

Represent intra and inter-class variability (100-1000 images)
70% training set/ 30% test set

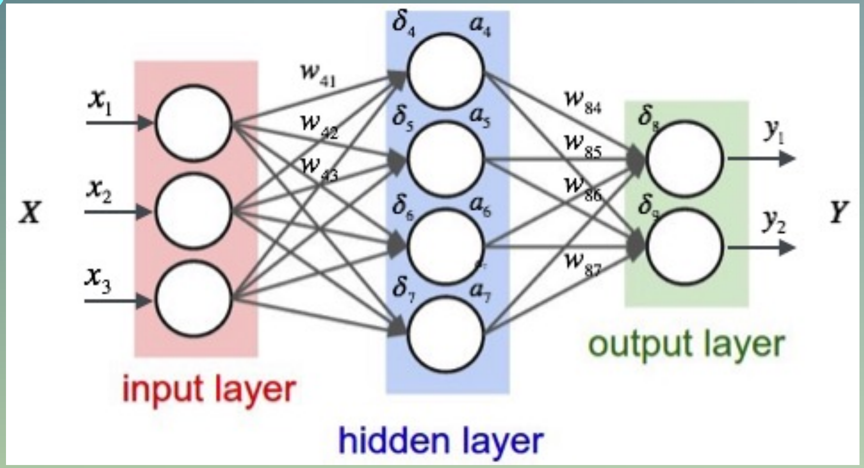
THE TRAINING PROCESS

Training set

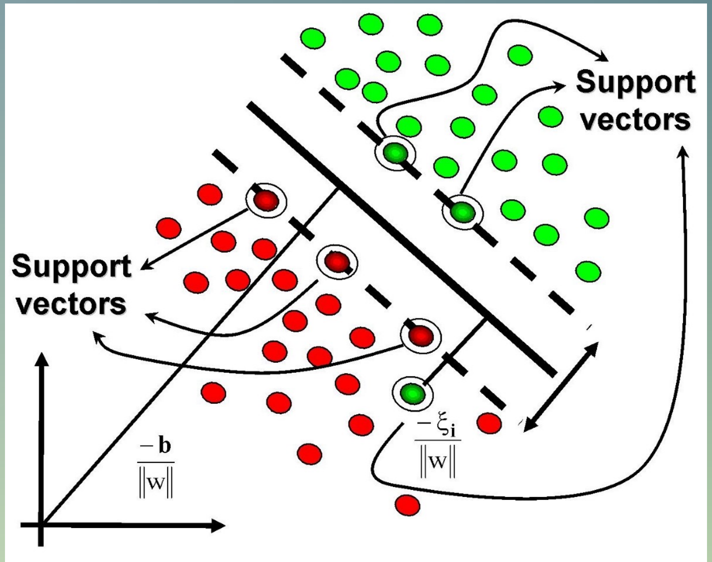


DIFFERENT SUPERVISED ALGORITHMS

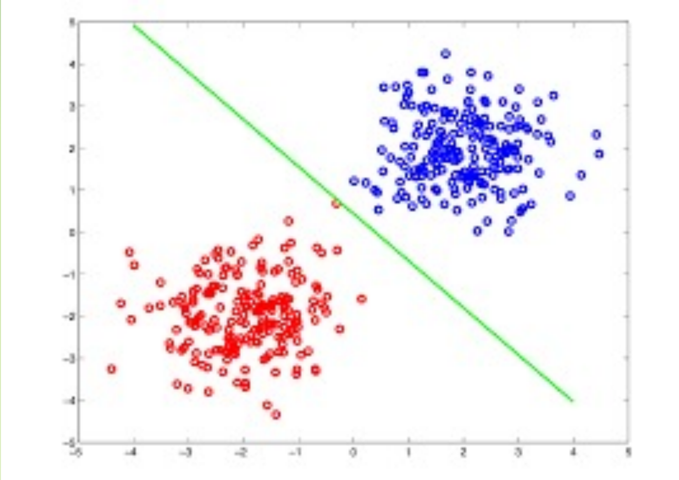
Neural Networks



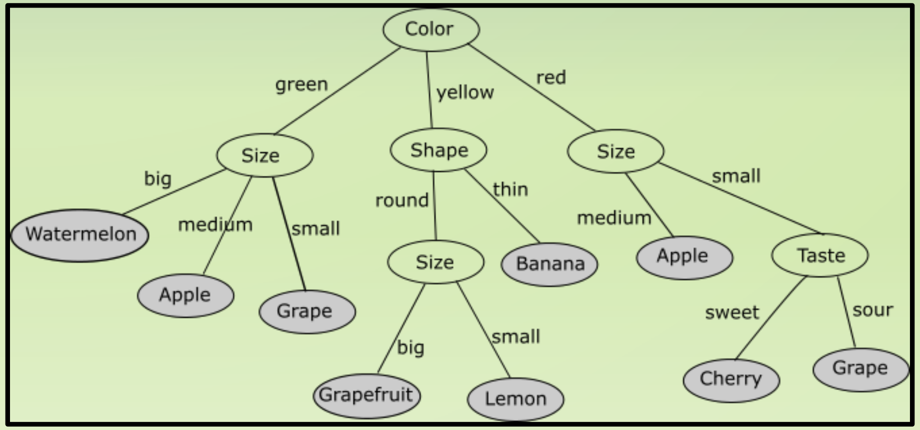
Support Vector Machines



Probabilistic discriminants



Decision trees



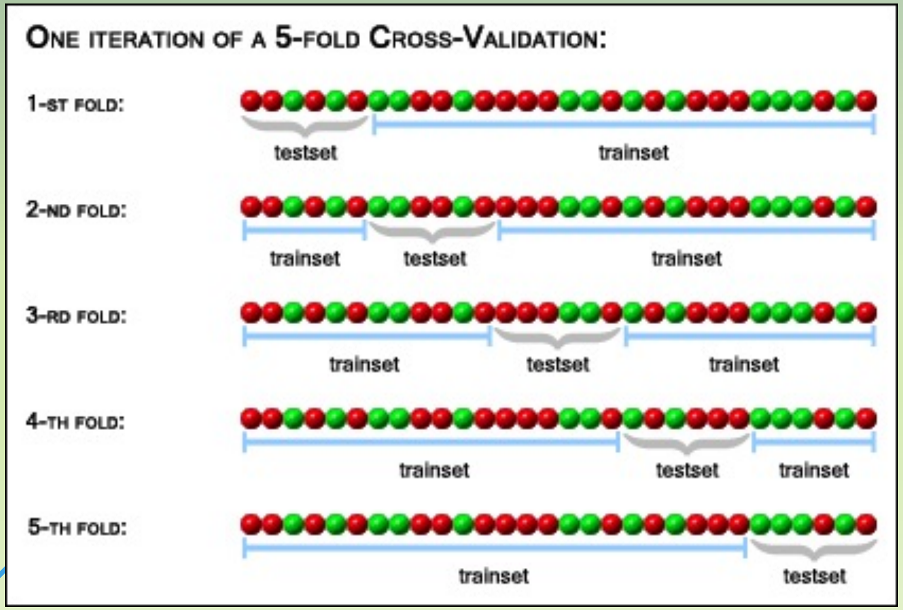
TEST: PERFORMANCE EVALUATION

$I(i)$

FEATURE EXTRACTION

SUPERVISED CLASSIFIER

test set



$w(i)$

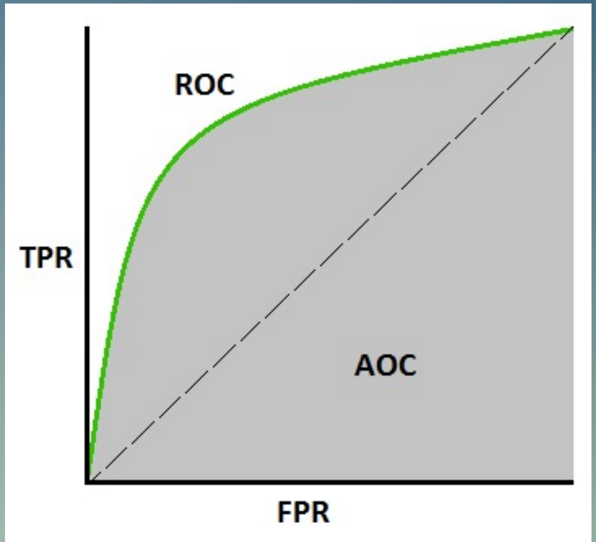
Actual class

$\hat{w}(i)$

Predicted class

Error measures

PERFORMANCE MEASURES



Confusion matrix

		True condition				
	Total population	Condition positive	Condition negative	Prevalence = $\frac{\Sigma \text{ Condition positive}}{\Sigma \text{ Total population}}$	Accuracy (ACC) = $\frac{\Sigma \text{ True positive} + \Sigma \text{ True negative}}{\Sigma \text{ Total population}}$	
Predicted condition	Predicted condition positive	**True positive**	**False positive**, Type I error False alarm!	Positive predictive value (PPV), Precision = $\frac{\Sigma \text{ True positive}}{\Sigma \text{ Predicted condition positive}}$	False discovery rate (FDR) = $\frac{\Sigma \text{ False positive}}{\Sigma \text{ Predicted condition positive}}$	
	Predicted condition negative	**False negative**, Type II error Omission!	**True negative**	False omission rate (FOR) = $\frac{\Sigma \text{ False negative}}{\Sigma \text{ Predicted condition negative}}$	Negative predictive value (NPV) = $\frac{\Sigma \text{ True negative}}{\Sigma \text{ Predicted condition negative}}$	
		True positive rate (TPR), Recall, Sensitivity, probability of detection, Power $= \frac{\Sigma \text{ True positive}}{\Sigma \text{ Condition positive}}$	False positive rate (FPR), Fall-out, probability of false alarm $= \frac{\Sigma \text{ False positive}}{\Sigma \text{ Condition negative}}$	Positive likelihood ratio (LR+) = $\frac{\text{TPR}}{\text{FPR}}$	Diagnostic odds ratio (DOR) = $\frac{\text{LR+}}{\text{LR}-}$	F_1 score = $2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$
		False negative rate (FNR), Miss rate $= \frac{\Sigma \text{ False negative}}{\Sigma \text{ Condition positive}}$	Specificity (SPC), Selectivity, True negative rate (TNR) = $\frac{\Sigma \text{ True negative}}{\Sigma \text{ Condition negative}}$	Negative likelihood ratio (LR–) = $\frac{\text{FNR}}{\text{TNR}}$		

Example: Fetal plane recognition

Data:

Fetal ultrasound 1st trimester
506 RGB Images of size 960x720 pixels / class
3 class labels (fetal planes BPD, CLR, NT)

SJD Sant Joan de Déu Barcelona · Hospital

BPD - Biparietal Diameter

CLR - Crown-Rump-Length

NT - Nuchal translucency

TRAINING & TEST SETS

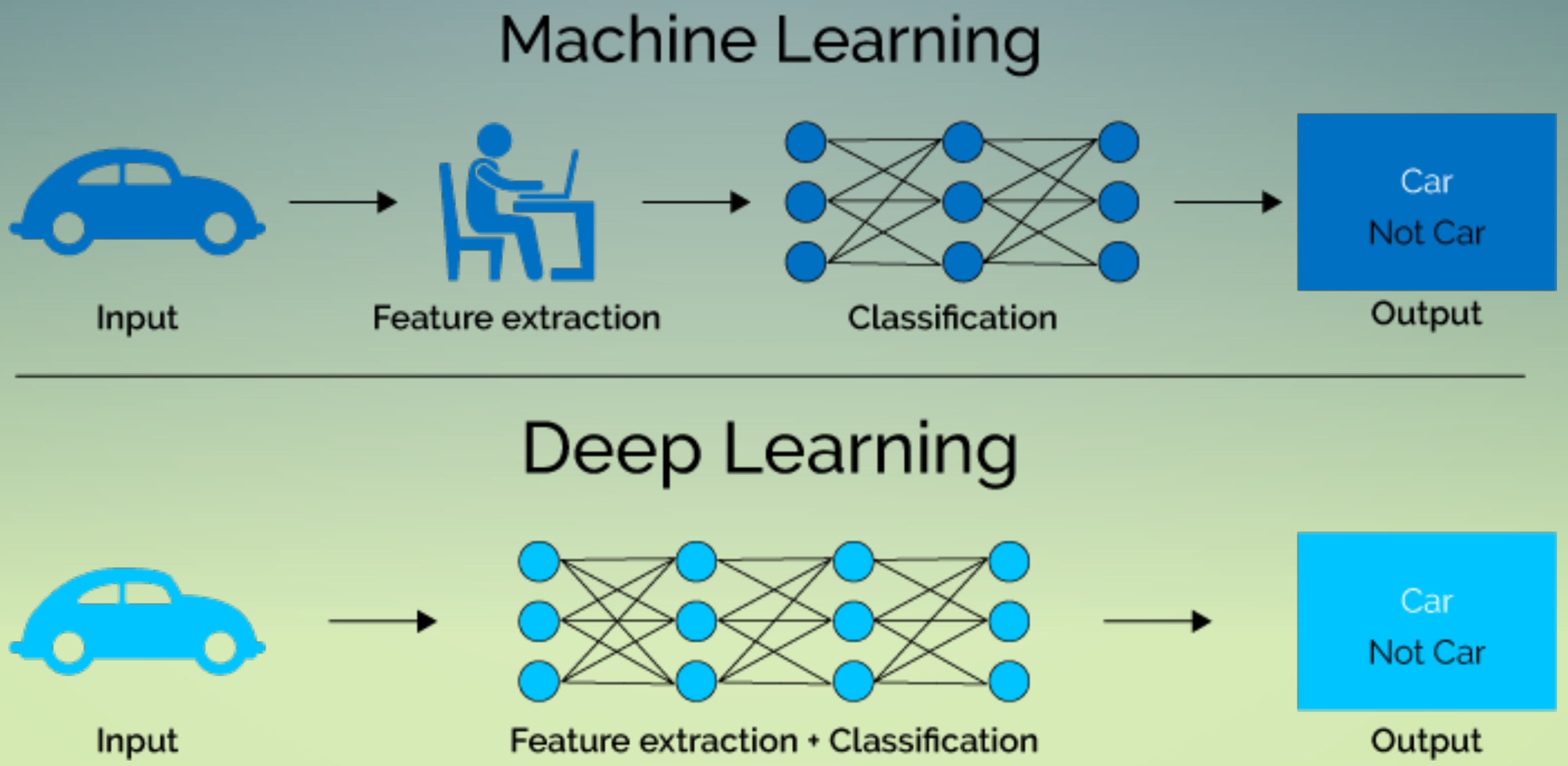
	BPD	CLR	NT
Training	406	406	406
Test	100	100	100
Total	**506**	**506**	**506**

PREDICTED

ACTUAL

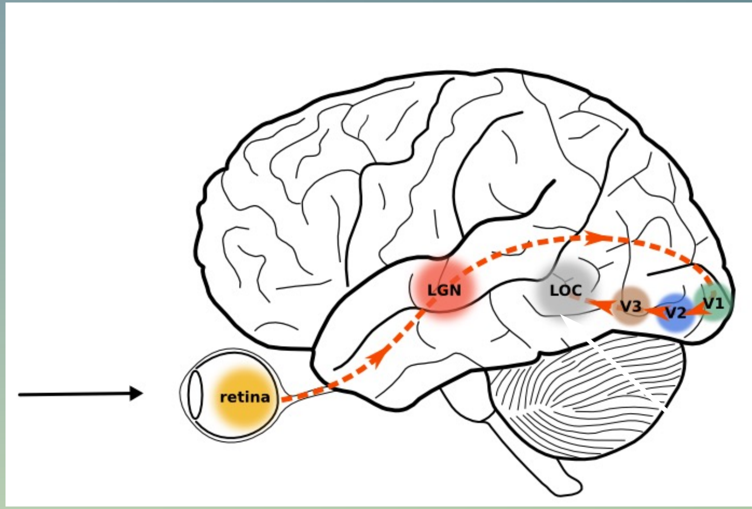
	BPD	CLR	NT
BPD	95	4	1
CLR	1	97	2
NT	2	6	92

CHANGE IN PARADIGM

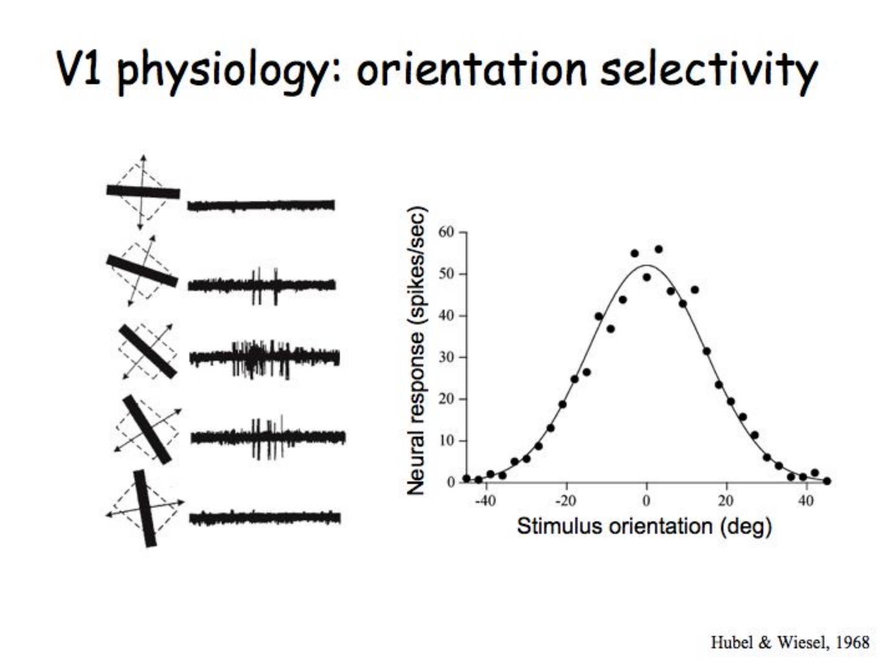


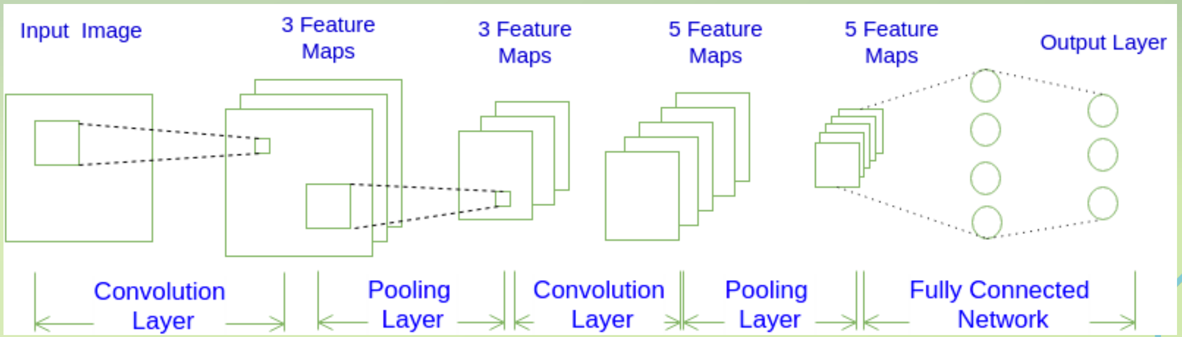
DEEP LEARNING – CONVOLUTIONAL NEURAL NETWORKS

Hubel & Wiesel: Neural basis of visual perception
LGN: Lateral Geniculate Nucleus
V1: Primary Visual cortex



Mug



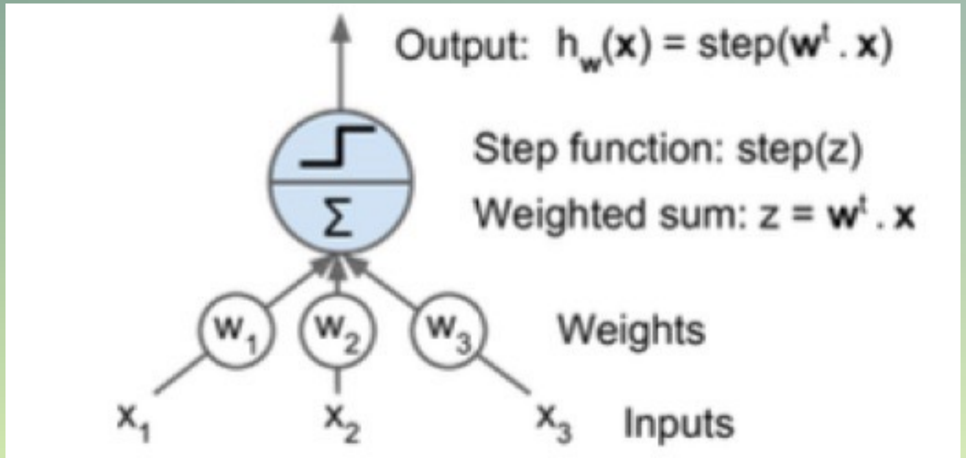


CNN's: 1982 – Now become hot topic due to GPU's (gamers)
SUPERVISED LEARNING OF WEIGHTS

NEURAL NETWORK

Artificial neuron:

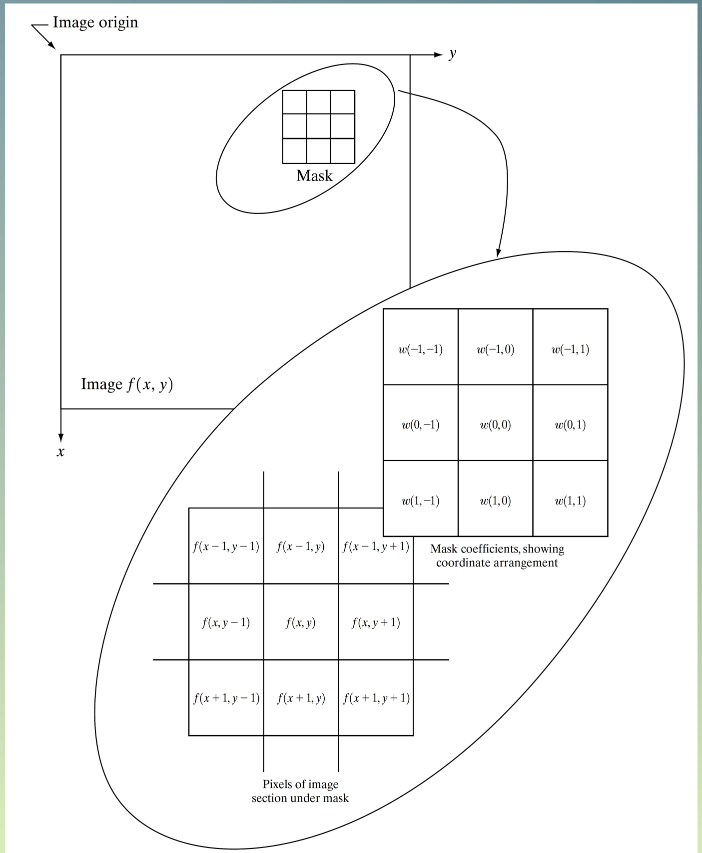
Soma, post-synaptic inputs, activation function



Training / learning weights:

Hebbian rule "wiring by firing"

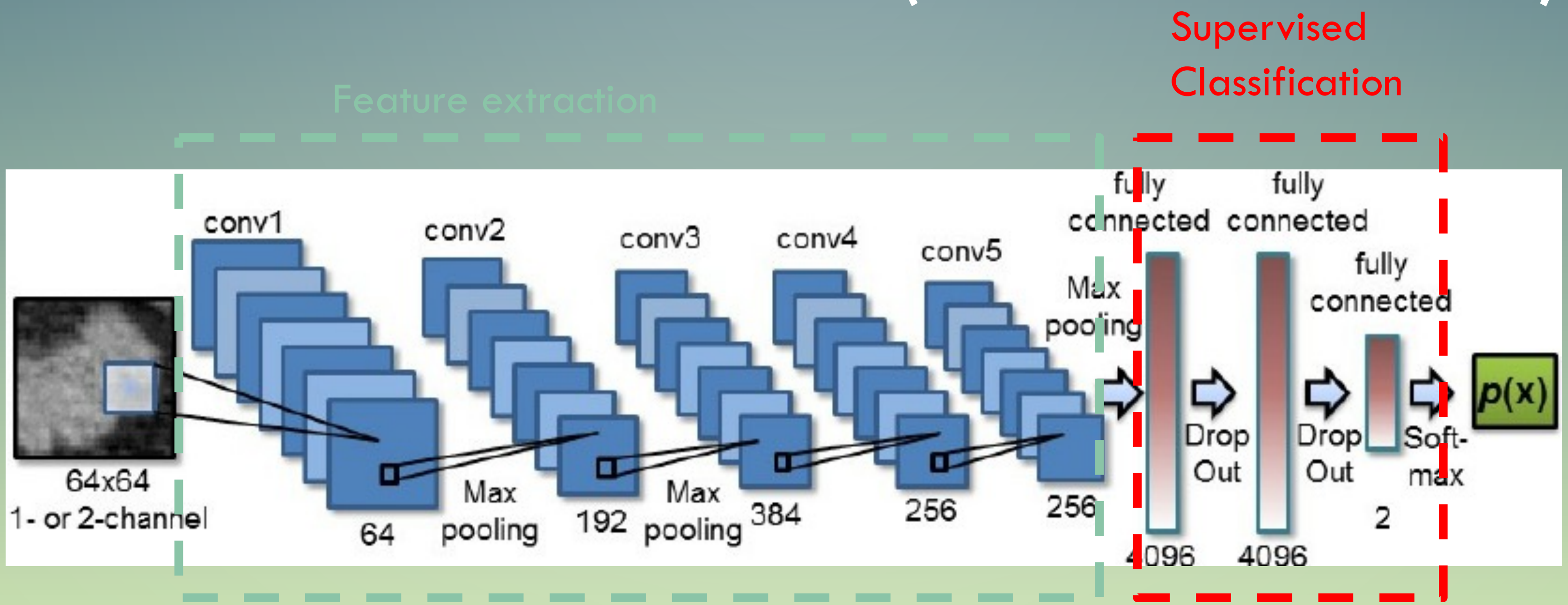
Convolutional filters



$$\boxed{g(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) \cdot f(x+s, y+t)}$$

Mask Image

DL: BLACK-BOX LEARNING ! (DARK-LEARNING)



Learns both:

- Which features are more relevant
- How to classify the images

TRANSFER LEARNING: PRE-TRAINED MODELS

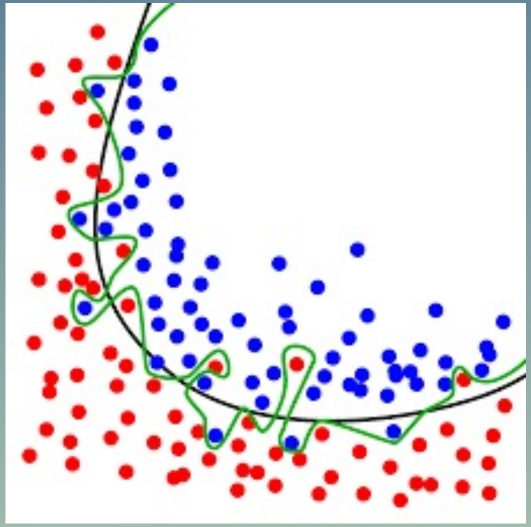
```
from keras.applications.inception_v3 import InceptionV3
from keras.layers import Input

# this could also be the output a different Keras model or layer
input_tensor = Input(shape=(224, 224, 3))  # this assumes K.image_data_format() == 'channels_last'

model = InceptionV3(input_tensor=input_tensor, weights='imagenet', include_top=True)
```

Documentation for individual models

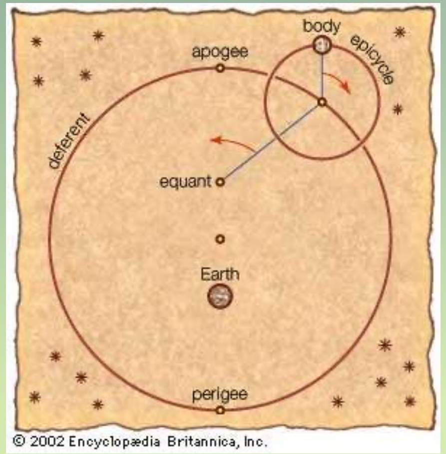
Model	Size	Top-1 Accuracy	Top-5 Accuracy	Parameters	Depth
Xception	88 MB	0.790	0.945	22,910,480	126
VGG16	528 MB	0.715	0.901	138,357,544	23
VGG19	549 MB	0.727	0.910	143,667,240	26
ResNet50	99 MB	0.759	0.929	25,636,712	168
InceptionV3	92 MB	0.788	0.944	23,851,784	159
InceptionResNetV2	215 MB	0.804	0.953	55,873,736	572
MobileNet	17 MB	0.665	0.871	4,253,864	88
DenseNet121	33 MB	0.745	0.918	8,062,504	121
DenseNet169	57 MB	0.759	0.928	14,307,880	169
DenseNet201	80 MB	0.770	0.933	20,242,984	201

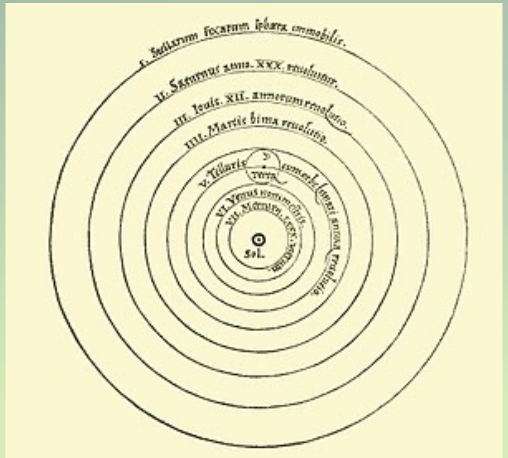
C. RISKS

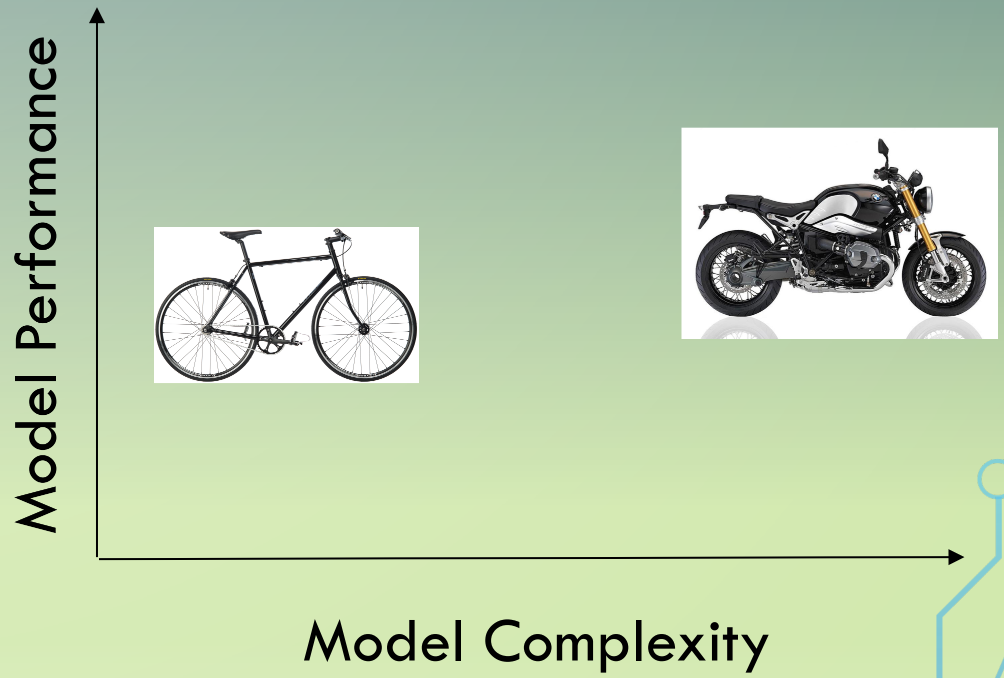
OVERFITTING



Parsimony principle





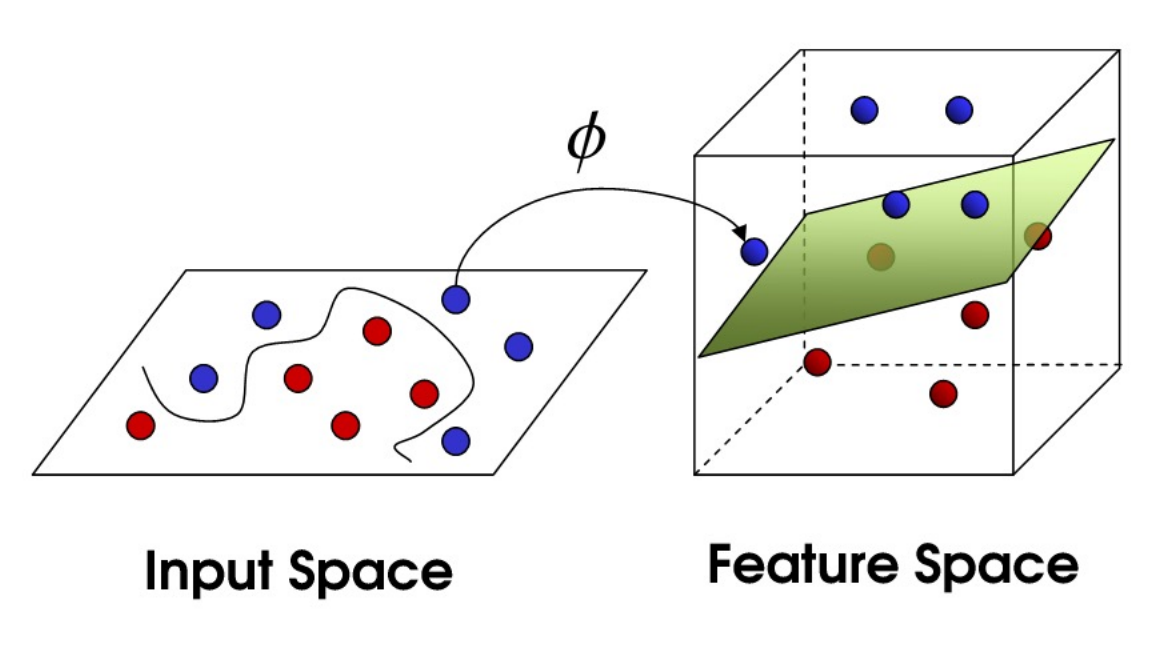


data-driven models

knowledge-driven models

Errors due to excess of complexity

HUMAN INTERPRETABILITY



Algorithms should be intepretable by humans

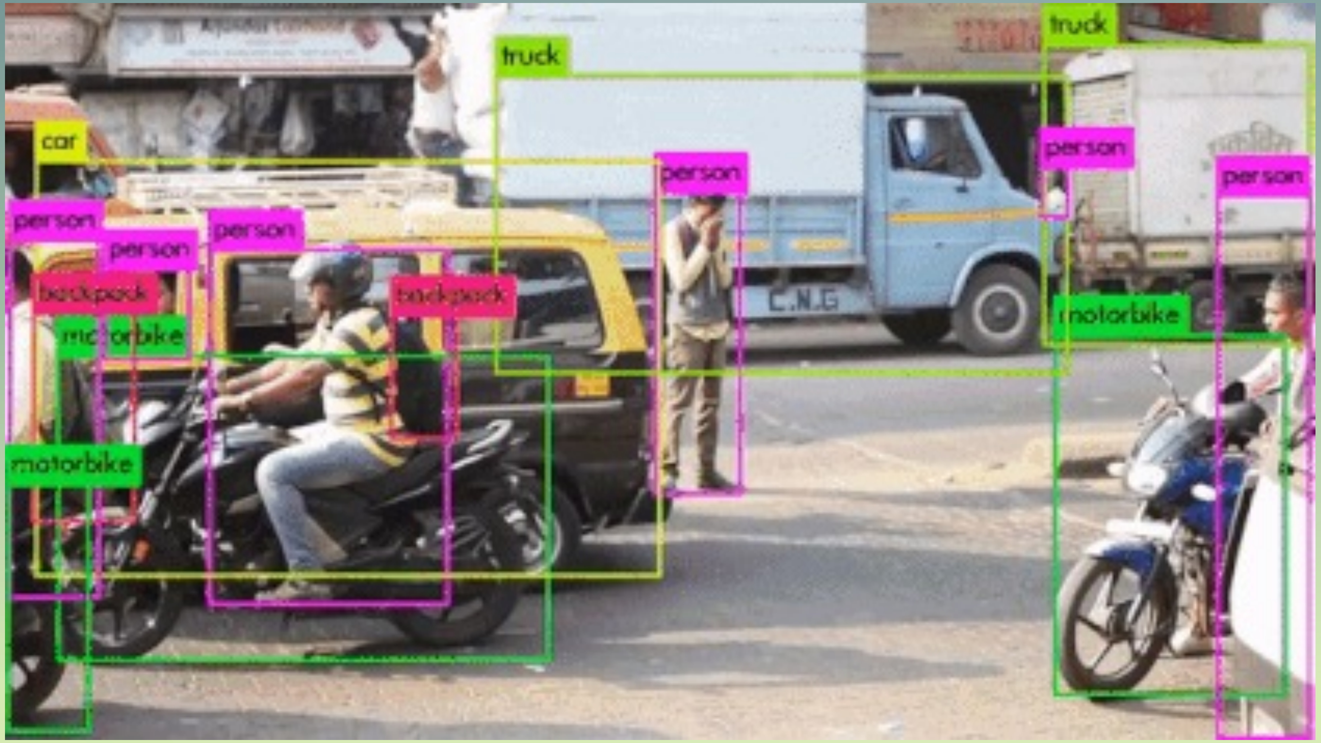
OPEN AND CLOSED DOMAIN PROBLEMS

Include open, contextual and dynamic information

WHO IS RESPONSIBLE OF ERRORS?

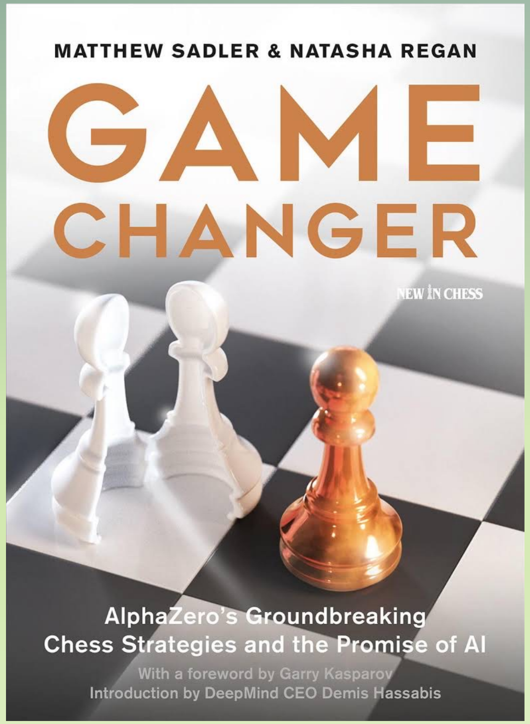
Ethical considerations

YOU ONLY LOOK ONCE (YOLO)



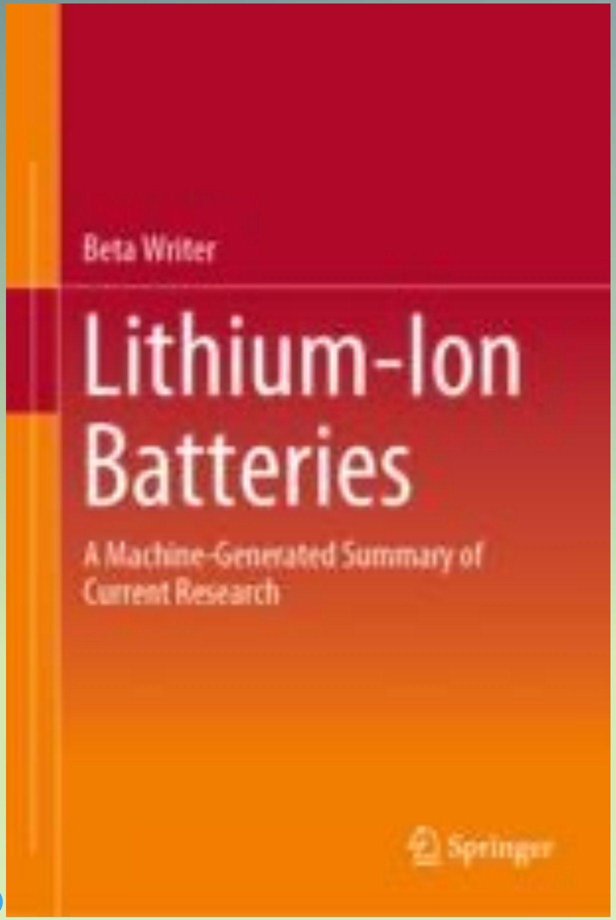
Reduce the costs of computational resources and data

HUMANS WILL BE REPROGRAMMED BY MACHINES



WHAT ELSE?

Creativity?



Lithium-Ion Batteries | © Springer Nature

Empathy?
Pity?
Consciousness?
Respect?
Enthusiasm?

Thanks! Let's get started