MLRF Lecture 01

J. Chazalon, LRDE/EPITA, 2021

Scope of this course

Apply Machine Learning (ML) techniques
to solve some practical Computer Vision (CV) problems.

⇒ It is a course about **Computer Vision** (CV).
⇒ It should be called CV-ML, ML4CV or so, but…

We need some definitions:

- What is Computer Vision? What is Pattern Recognition? Shape Recognition?
- What is Machine Learning?
- How do those concepts relate together?

Agenda for lecture 1

1. Some definitions and basic notions
2. Course outline
3. Introduction to *Twin it!*
4. Pattern Matching

Some definitions

Lecture 01 part 01

Computer Vision

Computer Vision: a definition

Computer Vision: *the automation of visual tasks with the goal of producing results directly or indirectly usable by humans.* ⇐ *Engineer definition*

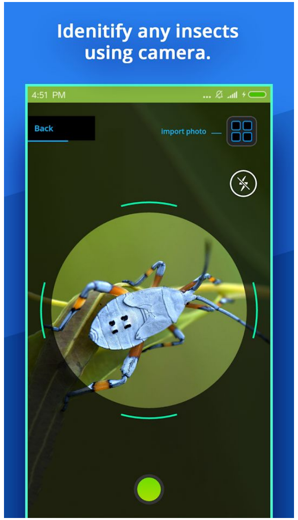
Input: image(s) in machine format (image acquisition is a subpart of CV)

Output: some pieces of information about the image, new image(s)…

Computer Vision: some examples (1/4)

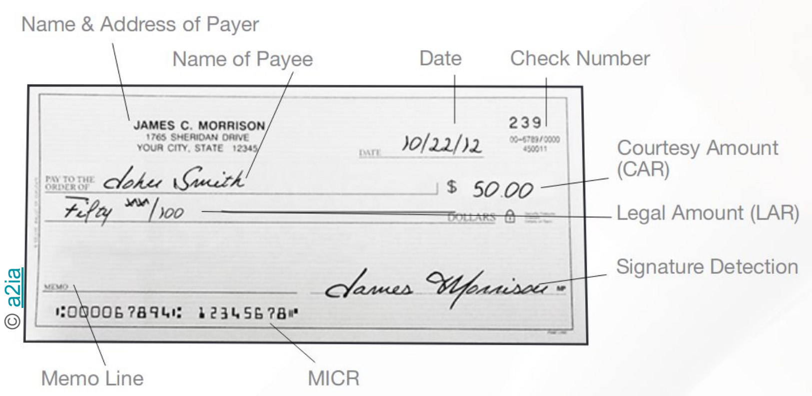
How would you process image pixels to get those results?

© Happimoji



© NASA

© a2ia



Input: still picture of insect
Output: insect name
Application: farming, …

Classification

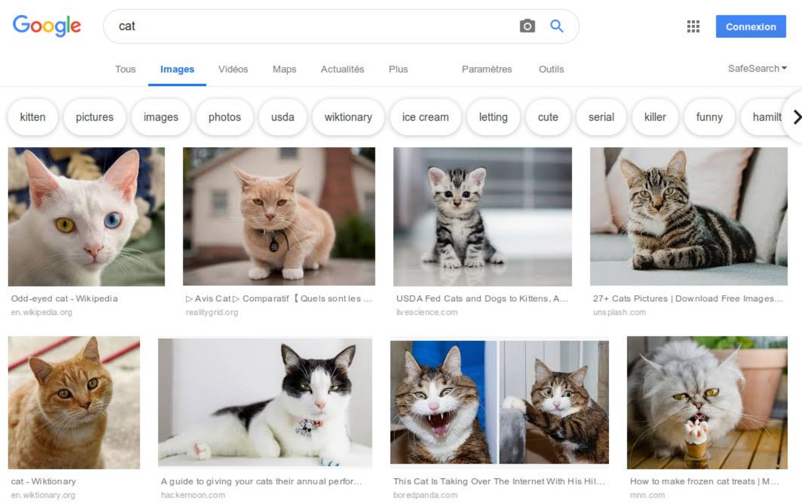
Input: satellite image (near visible range)
Output: crop maturity
Application: farming, trading…

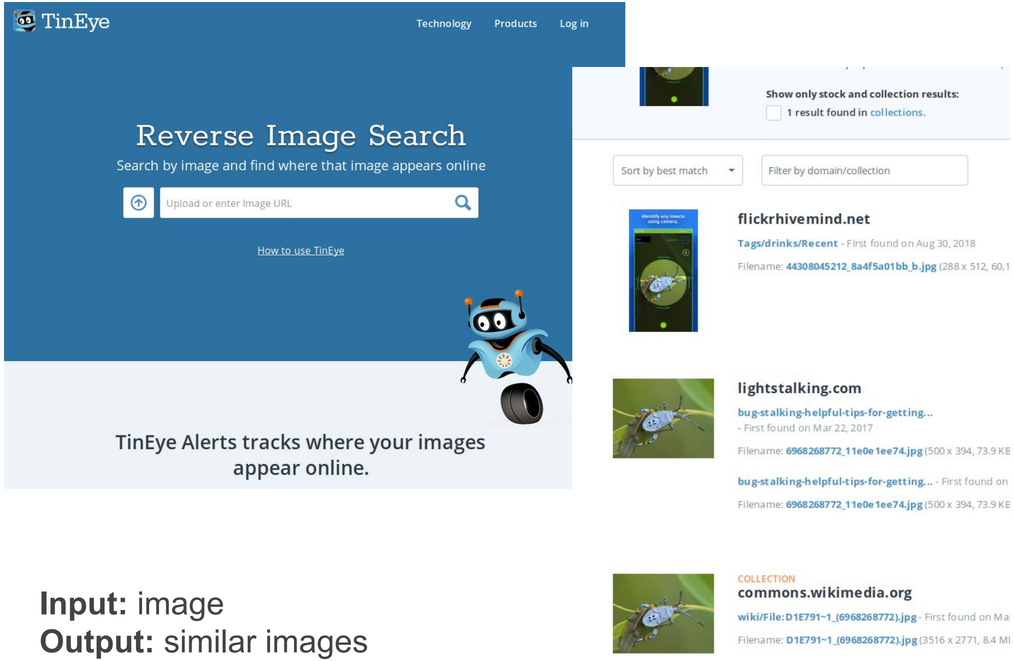
Regression

Input: bank check image (greylevel)
Output: account number, amount…
Application: banks

Segmentation, Classification, …

Computer Vision: some examples (2/4)





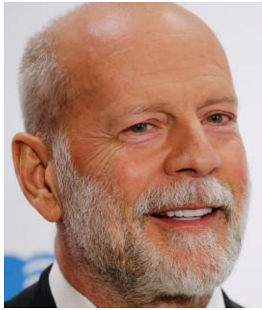
Input: text
Output: list of relevant images
Application: look for cats, …

Indexing, …

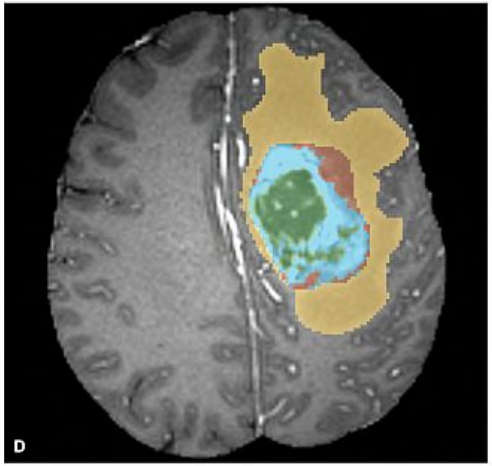
Input: image
Output: similar images
Application: duplicate detection, copyright, …

Content-based Image retrieval, …

Computer Vision: some examples (3/4)



vs



Input: two images of face
Output: same/different person
Application: authentication, …

Distance/similarity learning, …

Input: brain 3D scan
Output: regions with high tumor probability
Application: assisted diagnosis, …

Segmentation, classification, …

Computer Vision: some examples (4/4)

Some applications are direct (like the insect recognition app):
a human reads and uses the output

Some applications are indirect (like bank check reading):
the output is fed to a business system

Some applications extend what humans can naturally do:
either by extending our range of visible colors (satellite example)
or by simply being more efficient (face verification)

And there are many many many more examples…

Pattern Recognition

Pattern Recognition: a definition

Pattern Recognition: *The field of pattern recognition is concerned with the automatic discovery of regularities in data through the use of computer algorithms and with the use of these regularities to take actions such as* ***classifying*** *the data into different categories.* *— Bishop 2006*

The International Association for Pattern Recognition (IAPR) is an international association [...] concerned with **pattern recognition**, **computer vision**, and **image processing** in a broad sense.

Pattern Recognition: examples

OCR Computer Vision

Pedestrian detection Computer Vision

Credit card fraud detection NOT Computer Vision

…

⇒ CV ∩ PR ≠ ∅, “recognition” often mean “classification”

⇒ We will focus on actual tasks rather than trying to classify them

Pattern Recognition is an <u>inverse</u> problem

OCR example – Why Pattern Recognition is hard

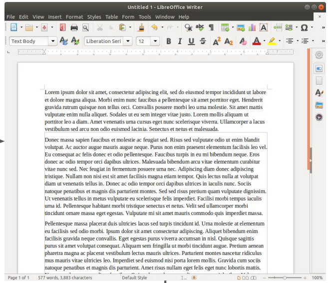
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus nulla ...

Forward

Reverse

Loren gypsum dolor sit mate, consecrate disciplining lit. Phallus nulls ...

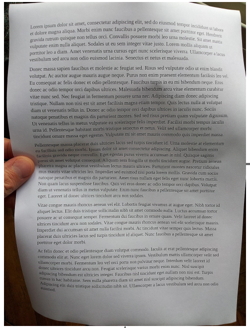
Type some text



Print it

...

“Scan” it



Try to guess was the original text was

“Shapes”

“Shapes”

Sometimes used to describe “visual percepts” (image patterns) which exhibit a “large [deviation] from randomness”. See *Cao et al. 2008*.

A way to designate meaningful visual patterns.

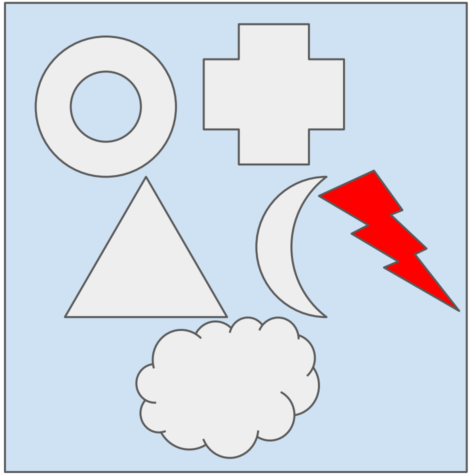
An interesting mathematical foundation to compare them:

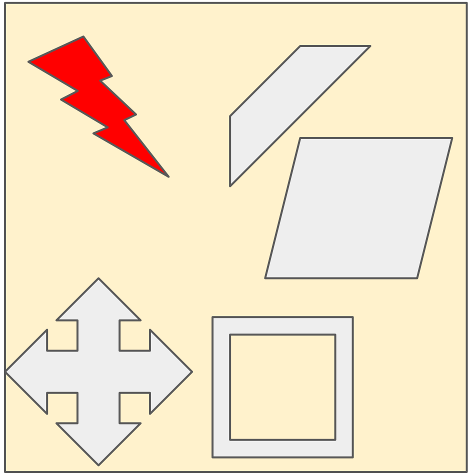
Let S and S’ be two shapes observed in two different images and which happen to be similar. Denote their (small) Hausdorff distance after registration by $\delta = d(S, S')$. Assume we know enough of the background model to compute the probability $Pr(S, \delta) = Pr(d(S, \Sigma) \leq \delta)$ that some shape in the background, Σ be as similar to S as S ' is. If this probability is very small one can deduce that S ' does not look like S just by chance. Then S and S ' will be identified as the same shape.

We can check whether two shapes are <u>significantly</u> close.

“Shapes”

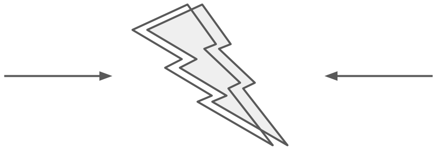
Let S and S’ be two shapes observed in two different images and which happen to be similar.





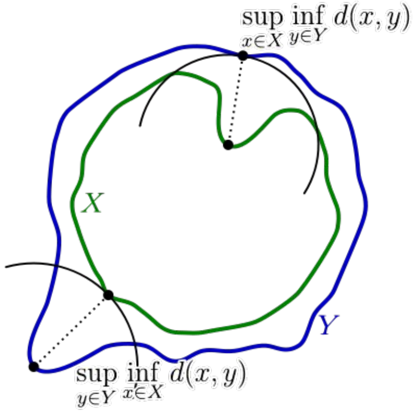
“Shapes”

Let S and S’ be two shapes observed in two different images and which happen to be similar. Denote their (small) Hausdorff distance after registration by δ = d(S, S').



Hausdorff distance = max of min distances between points on the contours of two shapes.

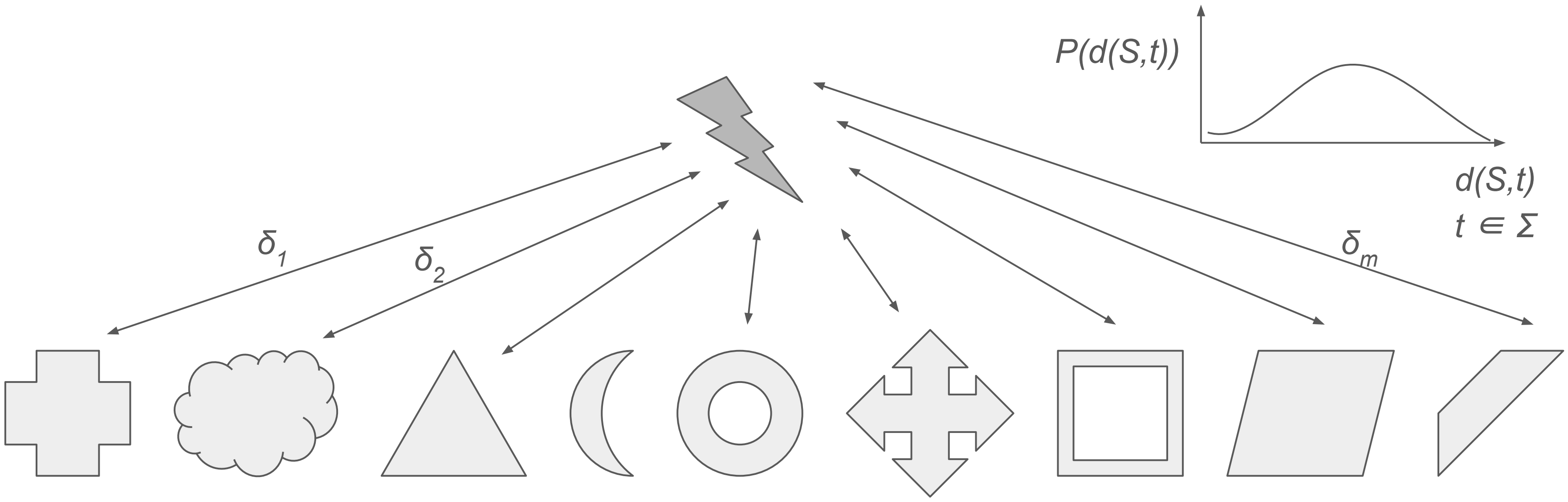
$$d_{\mathrm{H}}(X,Y) = \max\{\, \sup_{x\in X} \inf_{y\in Y} d(x,y),\ \sup_{y\in Y} \inf_{x\in X} d(x,y)\,\}.$$



Cao et al. 2008.

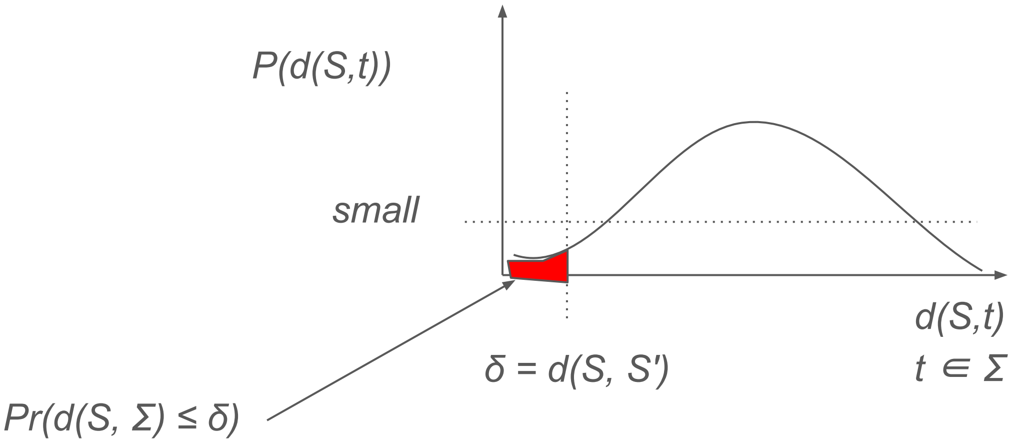
“Shapes”

Let S and S’ be two shapes observed in two different images and which happen to be similar. Denote their (small) Hausdorff distance after registration by δ = d(S, S'). <u>Assume we know enough of the background model to compute the probability Pr(S, δ) = Pr(d(S, Σ) ≤ δ) that some shape in the background, Σ be as similar to S as S ' is.</u>



"Shapes"

Let S and S' be two shapes observed in two different images and which happen to be similar. Denote their (small) Hausdorff distance after registration by δ = d(S, S'). Assume we know enough of the background model to compute the probability Pr(S, δ) = Pr(d(S, Σ) ≤ δ) that some shape in the background, Σ be as similar to S as S ' is. <u>If this probability is very small one can deduce that S ' does not look like S just by chance. Then S and S ' will be identified as the same shape.</u>



So, some **statistics** can help us making better decisions…

Idea: **learn** the distance threshold under which shapes can be deemed identical.

Machine Learning

Many forms of Machine Learning

We will focus on **inductive learning** (generalize from examples)
vs deductive learning (explain, revise knowledge…)

We will consider both **supervised** (a “teacher” provides labels for examples) and **unsupervised** (only samples) cases (but not *reinforcement*).

We will focus on **optimization-based learning techniques** (examples are represented as numerical vectors – *more about it in the next slides*)
vs exploration-based techniques (grammatical inference, etc.)
vs …

Another example is **regular expression inference**:
Given a set of strings to match (and eventually a set of string not to match), generate a regular expression which matches them (only the right ones).

What would be a good regular expression according to you?

Examples of optimization-based learning techniques

Linear classifiers, SVMs

Neural networks

Decision trees

Bayesian networks

Hidden Markov models

Genetic algorithms

…

(“Statistical”) Machine Learning

People working on CV/PR problem often distinguish **statistical** from **structural pattern recognition**.

Statistical: patterns are vectors
Structural: patterns are graphs, trees…

Quoting S. Bengio:

Learning means **changing** in order to be **better** (according to a given **criterion**) when a similar situation arrives

Learning **IS NOT** learning by heart

Any computer can learn by heart, the difficulty is to **generalize** a behavior to a novel situation

From an engineer’s point of view

Quoting scikit-learn documentation:

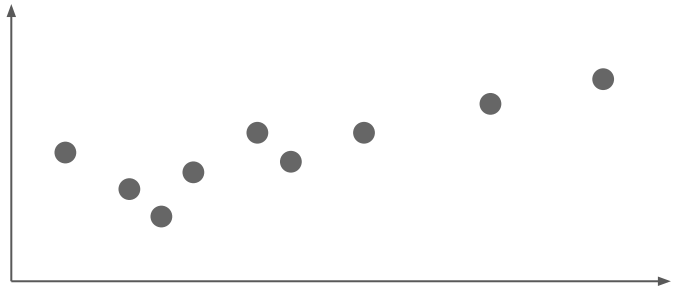
Machine Learning is about building programs with **tunable parameters** (typically an array of floating point values) that are **adjusted automatically** so as to improve their behavior by **adapting to previously seen data.**

Machine Learning can be considered **a subfield of Artificial Intelligence** since those algorithms can be seen as building blocks to make computers learn to behave more intelligently by somehow **generalizing** rather that just storing and retrieving data items like a database system would do.

Why Learning is Difficult?

Given a **finite amount of training data**, you have to derive a **relation for an infinite domain.**

In fact, there is an **infinite** number of such relations.

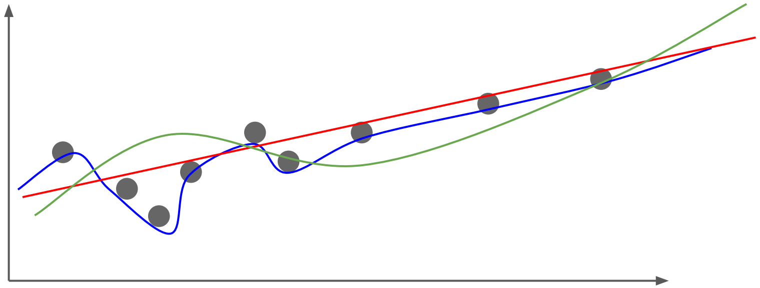


How should we draw the relation?

Why Learning is Difficult?

Given a **finite amount of training data**, you have to derive a **relation for an infinite domain.**

In fact, there is an **infinite** number of such relations.

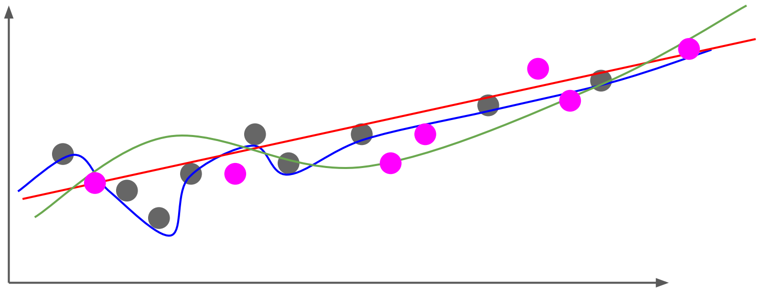


Which relation is the most appropriate?

Why Learning is Difficult?

Given a **finite amount of training data**, you have to derive a **relation for an infinite domain.**

In fact, there is an **infinite** number of such relations



... the hidden test points...

The **learning bias** can also be called **“inductive bias”**.

Learning bias

It is **always** possible to find a model **complex enough** to fit **all** the examples.

Example: polynom with very high degree

But how would this help us with **new samples**?

It should not **generalize** well.

We need to define a **family of acceptable solutions to search from**.

It forces to learn a “smoothed” representation.

… but it should not smooth the representation too much!

Occam’s Principle of Parsimony:

One should not increase, beyond what is necessary, the number of entities required to explain anything.

So, in practice, we need

Examples (data!)

A tunable algorithm (model)

A evaluation of the model fitness to examples (risk, loss)

A definition of the model search space (not too big, not too small)

An optimization strategy

The bias / variance compromise:
Small search space:

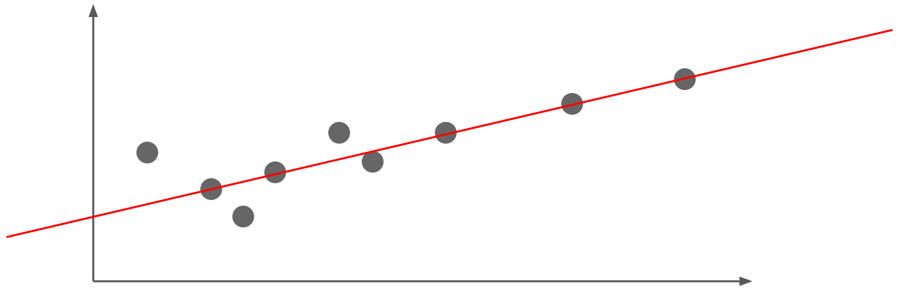
- Easier to find the best (available) solution
- But it may be far from the ideal one

Large search space:

- It is hard to find the best (available) solution
- Too bad because it should be closer to the ideal one

There are 3 kinds of problems

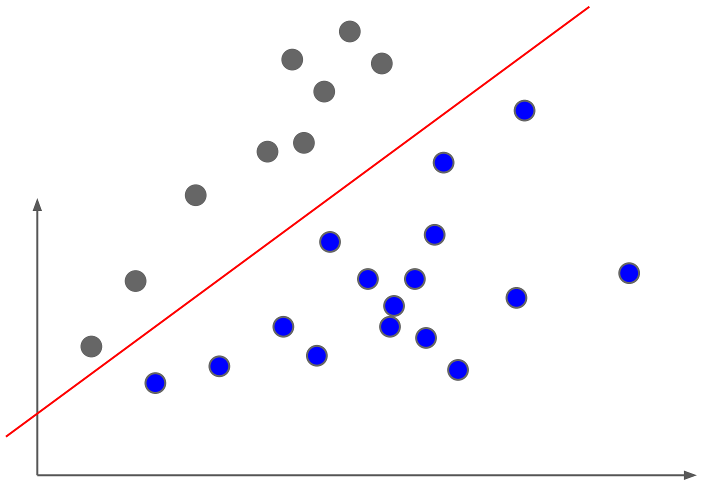
Regression



from S. Bengio 2005

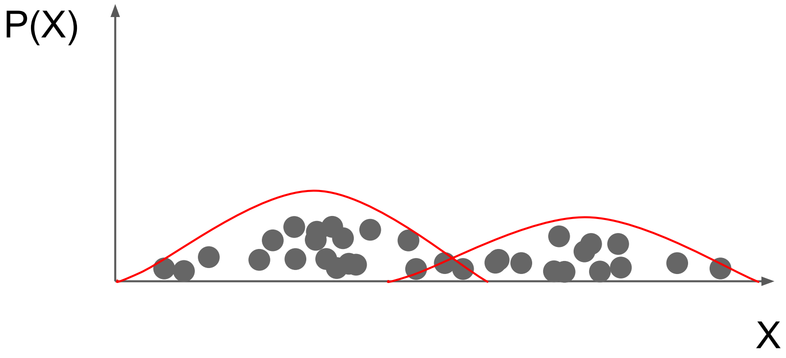
There are 3 kinds of problems

Regression, **Classification**



There are 3 kinds of problems

Regression, Classification, **Density estimation**



from S. Bengio 2005

And 3 types of Learning

Richer information

Supervised learning

The training data contains the desired behavior
(desired class, outcome, etc.)

Reinforcement learning

The training data contains partial targets
(for instance, simply whether the machine did well or not)

Reinforcement learning vs **weakly-supervised** learning?
Sample availability?

Unsupervised learning

The training data is raw, no class or target is given
There is often a hidden goal in that task
(compression, maximum likelihood, etc.)

Now, experts say **self-supervised.**

More data

Model validation

More on that later.

For now, just remember that:

1. You need to **test the generalization** power of your approach.
2. So you need data not seen during the training: **a test set.**
3. For which you know the **expected output** (“ground-truth”, “gold standard”, “target”…).

Benefits of ML for CV/PR

A duck example

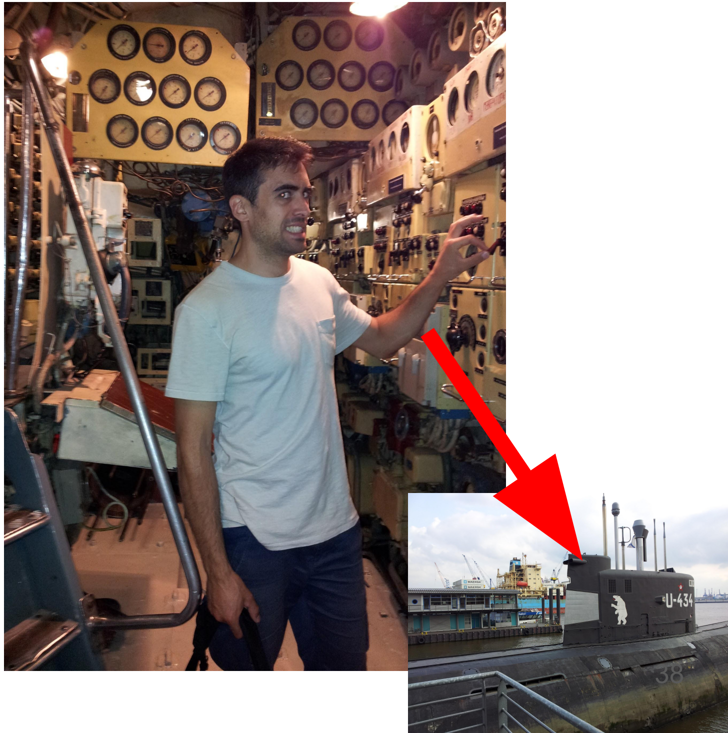
How to filter the grass to keep only the duck shape, using thresholds in the color domain?

Try it during practice session!

Why using Machine Learning in Computer Vision?

To avoid knob tuning. It’s complex. It’s unsafe.

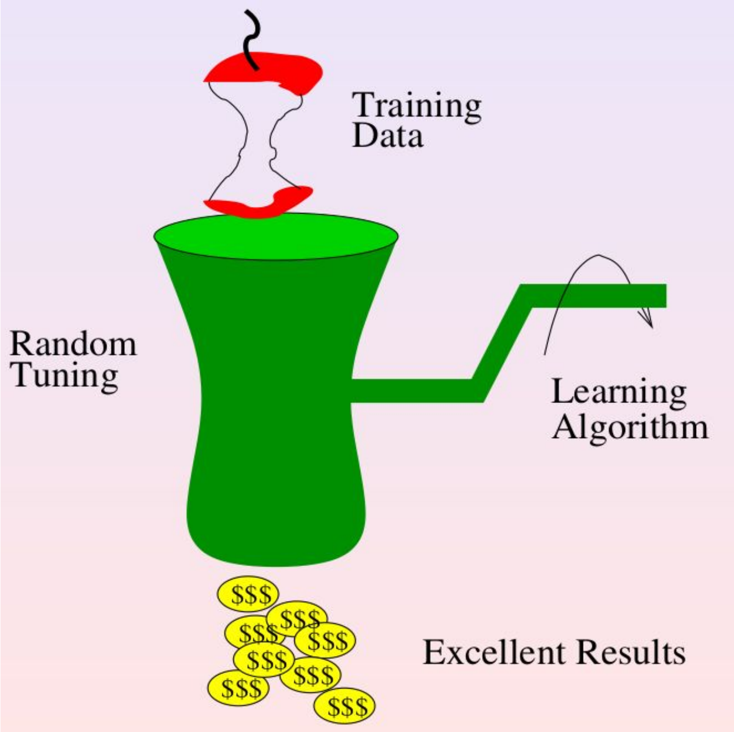
Photo by jc.winkler



Computer Vision / Pattern Recognition

But Beware of the Machine Learning Magic

What they sell you…



But most often…