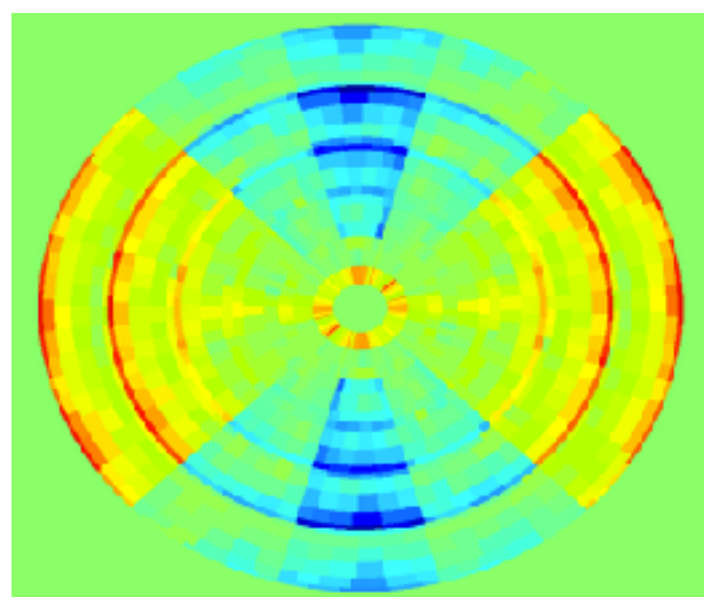
Deep Learning and (Image) Classification

Edouard Oyallon



advisor: Stéphane Mallat

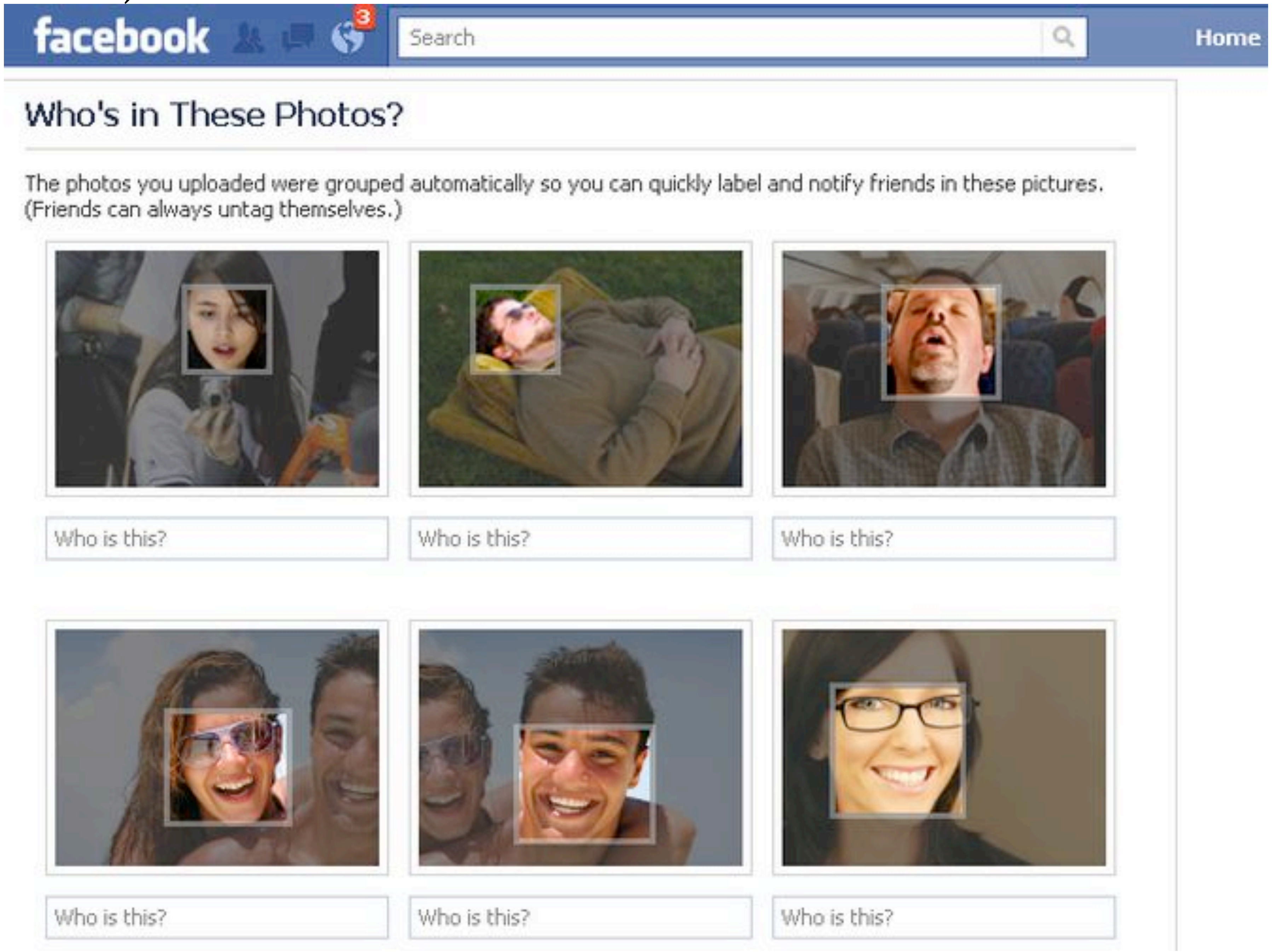
following the works of Laurent Sifre, Joan Bruna, …

Deep learning: Technical breakthrough

- Deep learning has permitted to <u>**solve**</u> a large number of task that were considered as extremely challenging for a computer.
- The technique that is used is **generic** and **scalable.** It simply requires a **large** amount of data.
- Pretty much hype and engineers with deep learning profiles are **highly** demanded.

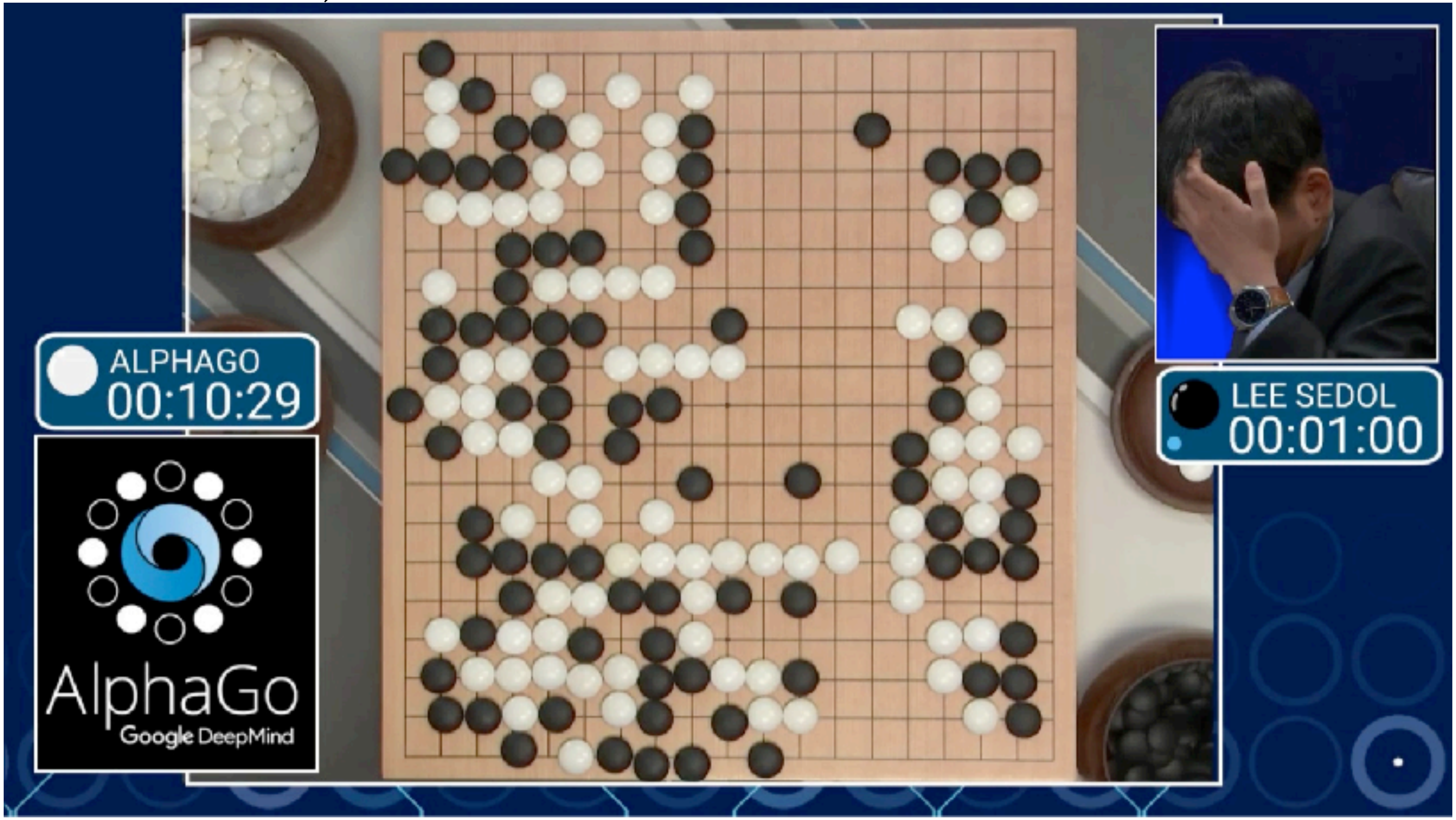
Face recognition

- Face recognition tasks almost solved(**three** years of research):



Strategy Games

- Game of GO: completely impossible to solve with Monte Carlo tree search, and solved (**two** years of research):



Natural Language Processing

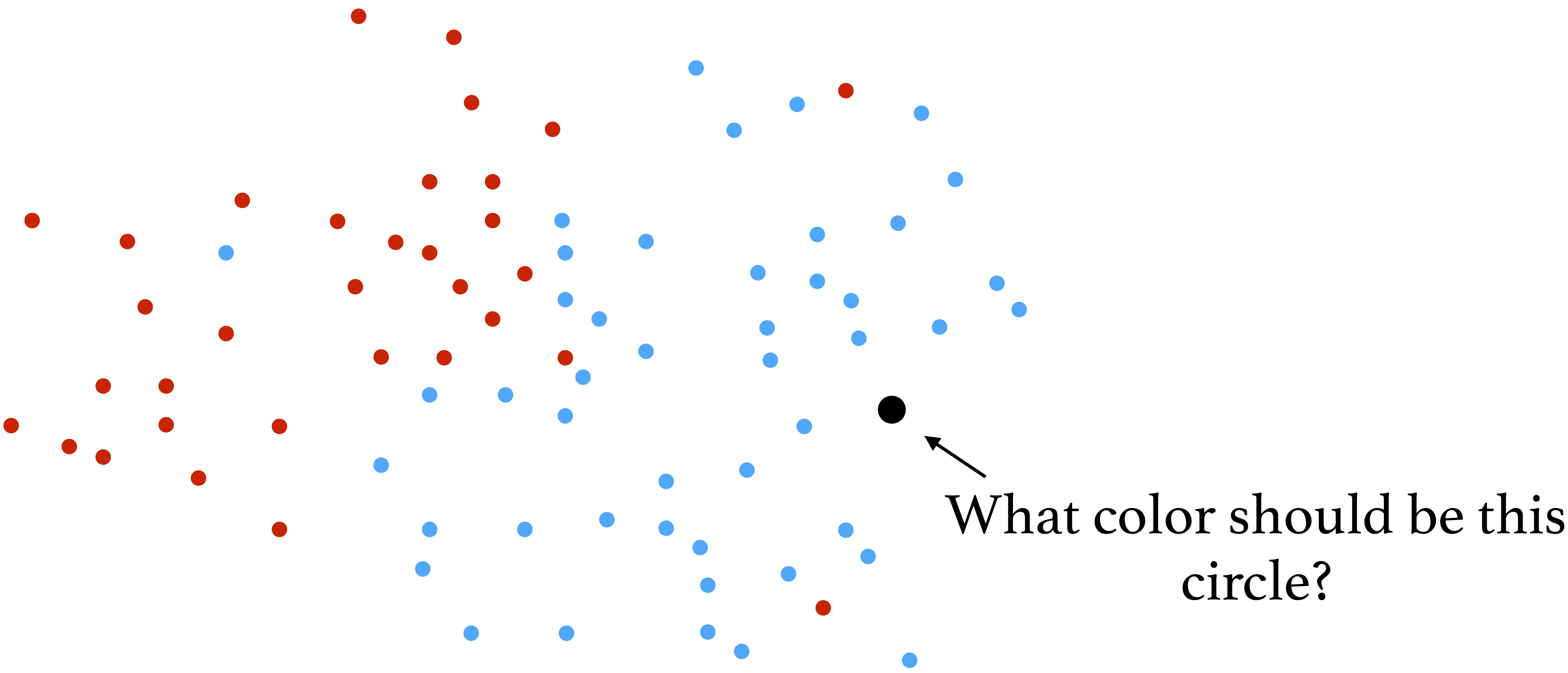
- Translation (Google just updated its traduction system with Recurrent Neural Network):

Pure blackbox

- However, nobody has no idea how it works and **few** research works have clues.
- People claim "**AI**" **is raising** and that we are simulating the "brain", while mathematicians **avoid those techniques like a prawn.**
- Can we do maths in deep learning?

Plot

- Classification?
- Understanding variabilities in high dimension
- Deepnets
- Wavelets



Classification ?

Classification of signals

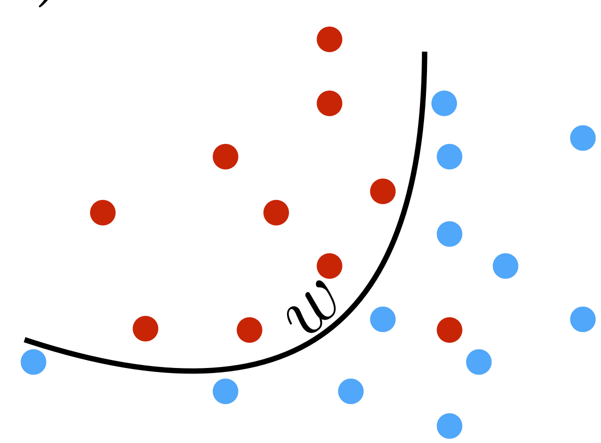
- Let $n > 0$, $(X, Y) \in \mathbb{R}^n \times \mathcal{Y}$ random variables
- **Problem**: Estimate $\hat{y}$ such that $\hat{y} = \arg\inf_{\tilde{y}} \mathbb{E}(|\tilde{y}(X) - Y|)$
- We are given a training set $(x_i, y_i) \in \mathbb{R}^n \times \mathcal{Y}$ to build $\hat{y}$
- Say one can write $\hat{y} = \text{Classifier}(\Phi x)$, Classifier being built with $(\Phi x_i, y_i)$
- 3 ways to build Φ:

Supervised	Unsupervised	Predefined
$(x_i, y_i)_i$	$(x_i)_i$	Geometric priors

$\mathcal{Y} = \{\bullet, \bullet\}$

$n = 2$

Classifier w

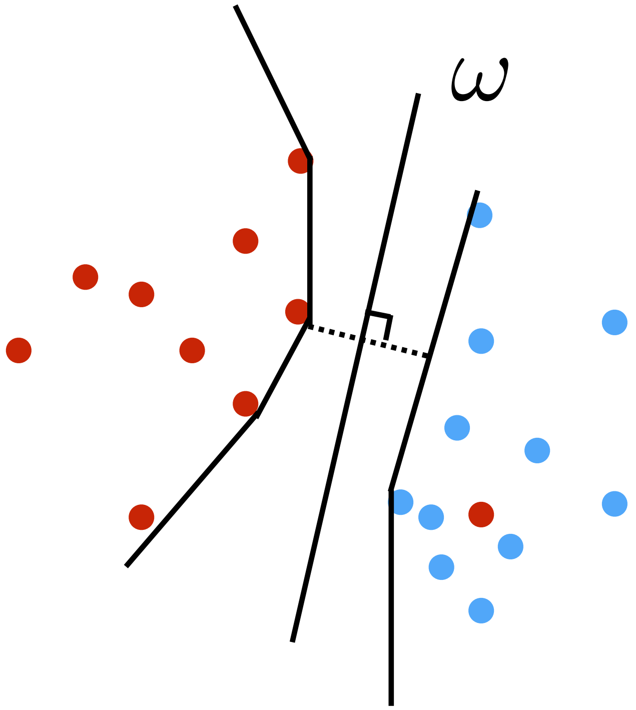


Classifier

- A classifier is an **algorithm** that outputs the probability distribution for a given sample x_i to belong to a class y_i.

- A classical example is given by the Support Vector Machine (SVM):

Ref.: Vapnik, Chervonenkis, 63



$$\omega^T x + b \geq 0?$$

Linear classifier

Minimizing the distance between the convex-hull and taking the associated hyperplan ω

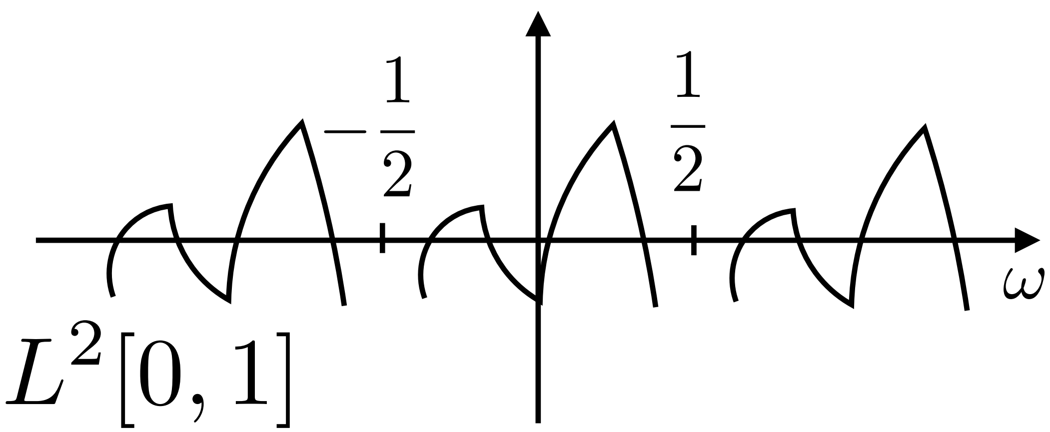
Discrete image to continuous image

- An image x corresponds to the discretisation of a **physical** anagogic signal (light!)

- An array of numbers: $x[n_1, n_2] \in \mathbb{R}, n_1, n_2 \leq N$

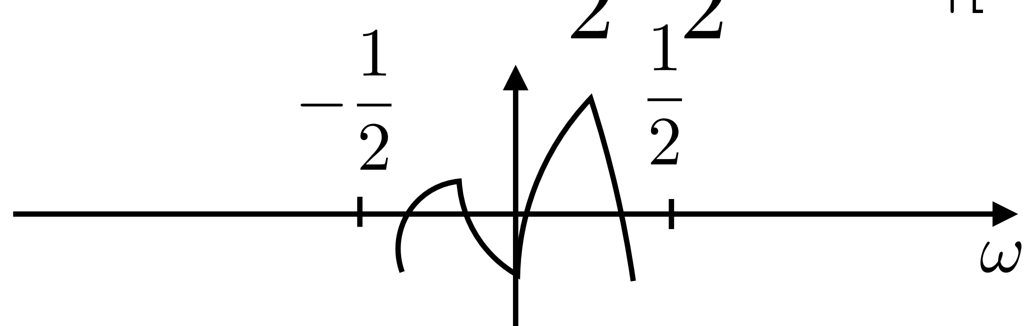
- One can set $x(u) = \sum_{n \in \mathbb{Z}^2} x[n]\delta_n(u)$

  then, $\mathcal{F}x(\omega) = \sum_{n \in \mathbb{Z}^2} x[n]e^{-in\omega}, \mathcal{F}x \in L^2[0,1]$



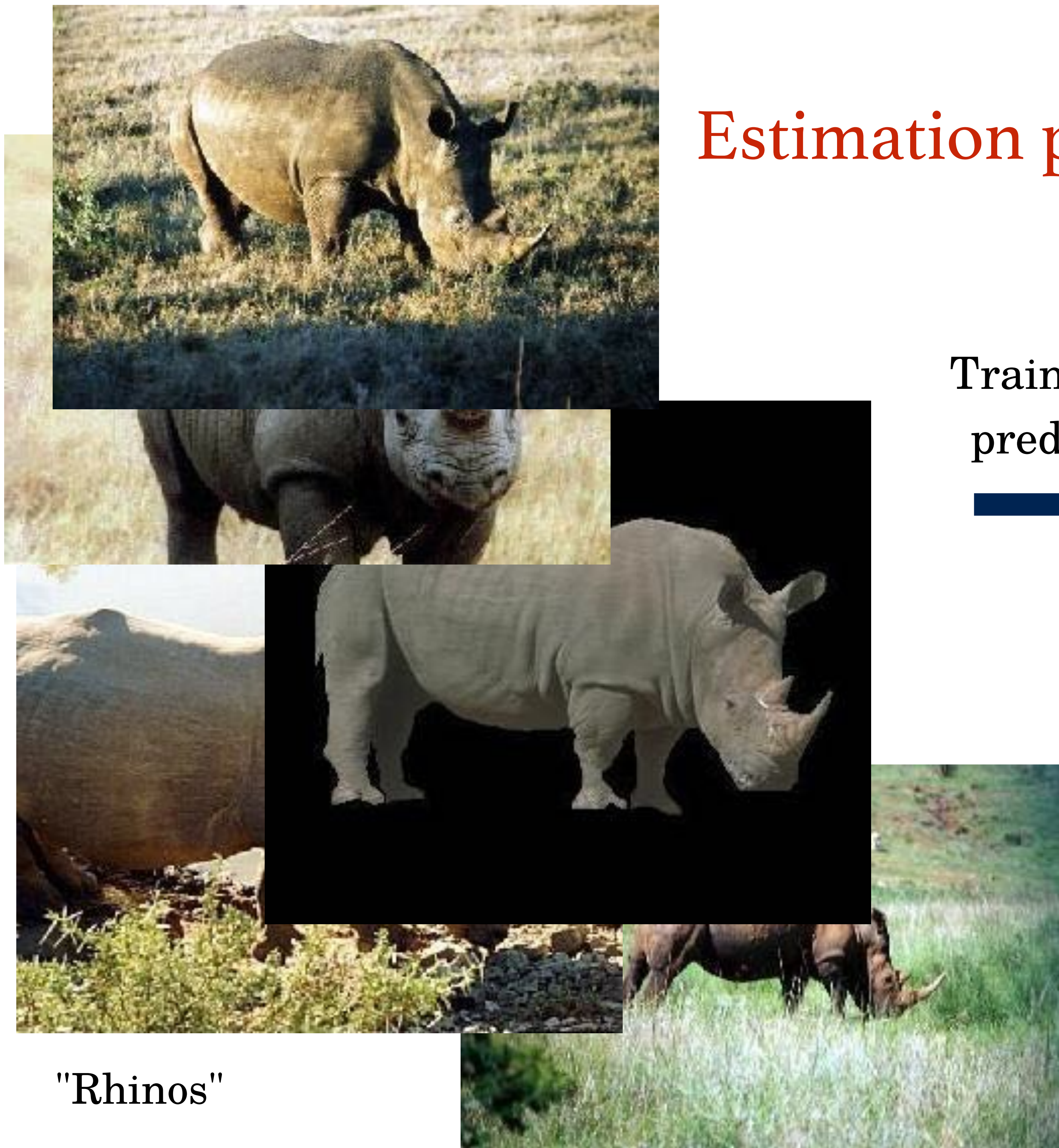
- Nyquiest-Shannon sampling property:

  $\exists! \tilde{x} \in \mathbb{L}^2(\mathbb{R})$, support$(\mathcal{F}\tilde{x}) \subset [-\frac{1}{2}, \frac{1}{2}], \mathcal{F}\tilde{x}_{|[-\frac{1}{2},\frac{1}{2}]} = \mathcal{F}x$



High Dimensional classificatio

$$(x_i, y_i) \in \mathbb{R}^{224^2} \times \{1, ..., 1000\}, i < 10^6 \longrightarrow \hat{y}(x)?$$



"Rhinos"

Estimation problem

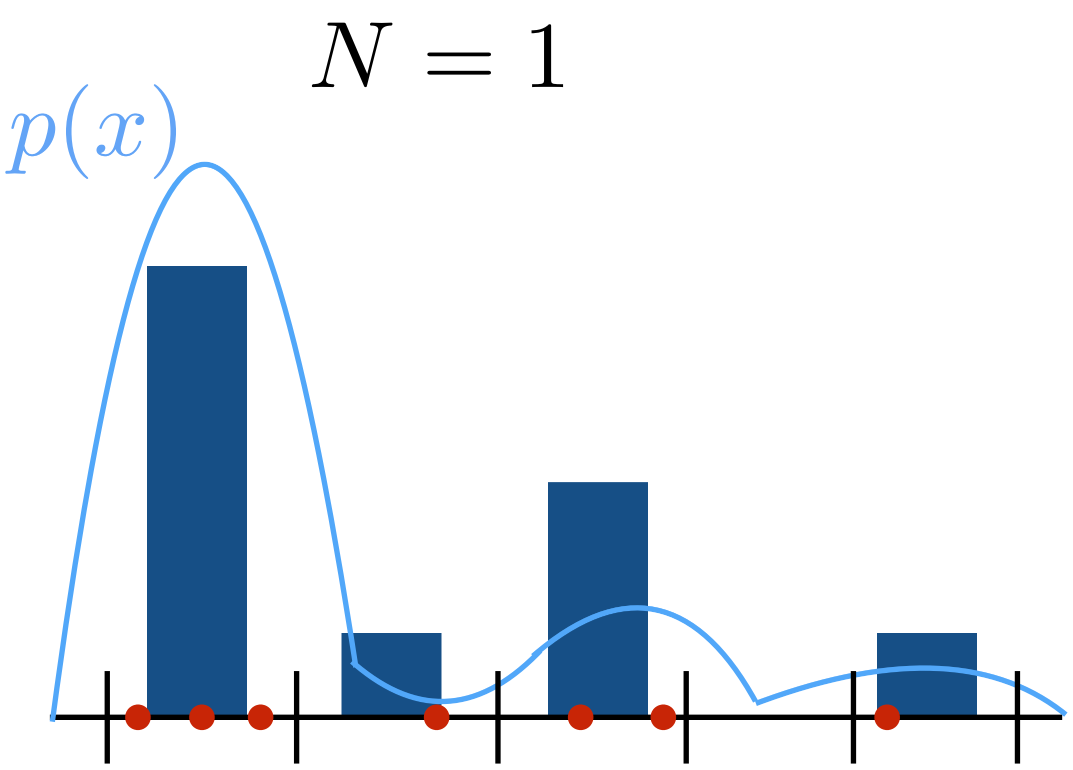
Training set to predict labels

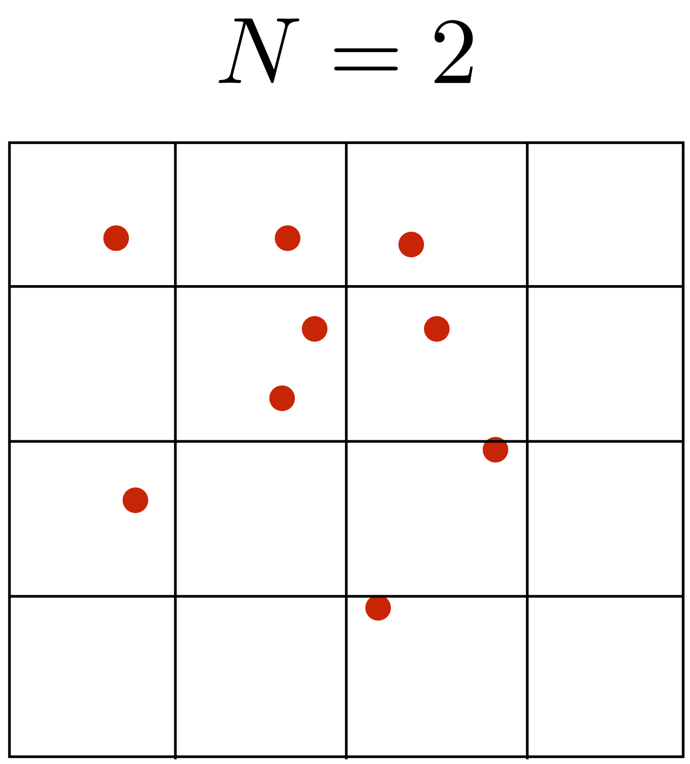
"Rhino"

Not a "rhino"

High-dimensionality issues

- Density functions are difficult to estimate in high dimension.





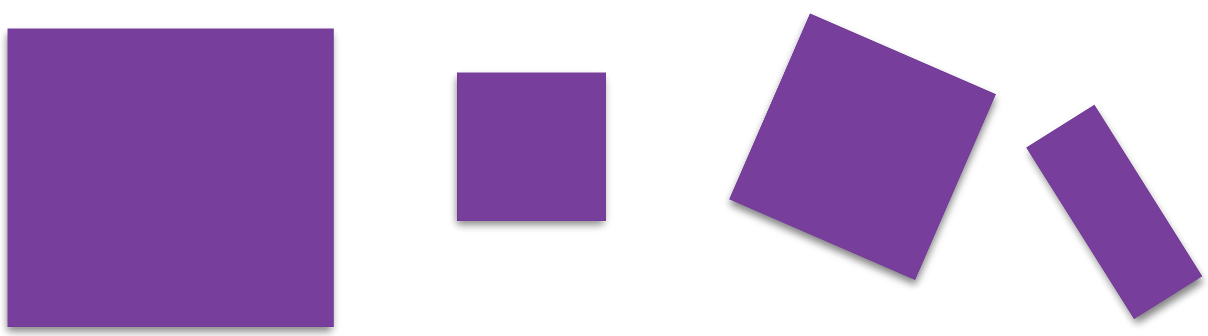
- For a fixed number of points and bin size, as N increases, the bins will be likely to be empty.

Curse of dimensionality

Image variabilities

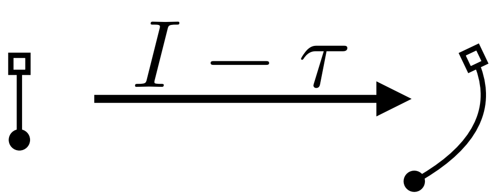
Geometric variability

Groups acting on images:
translation, rotation, scaling



Other sources : luminosity, occlusion,
small deformations

$$x_\tau(u) = x(u - \tau(u)), \tau \in \mathcal{C}^\infty$$



Class variability

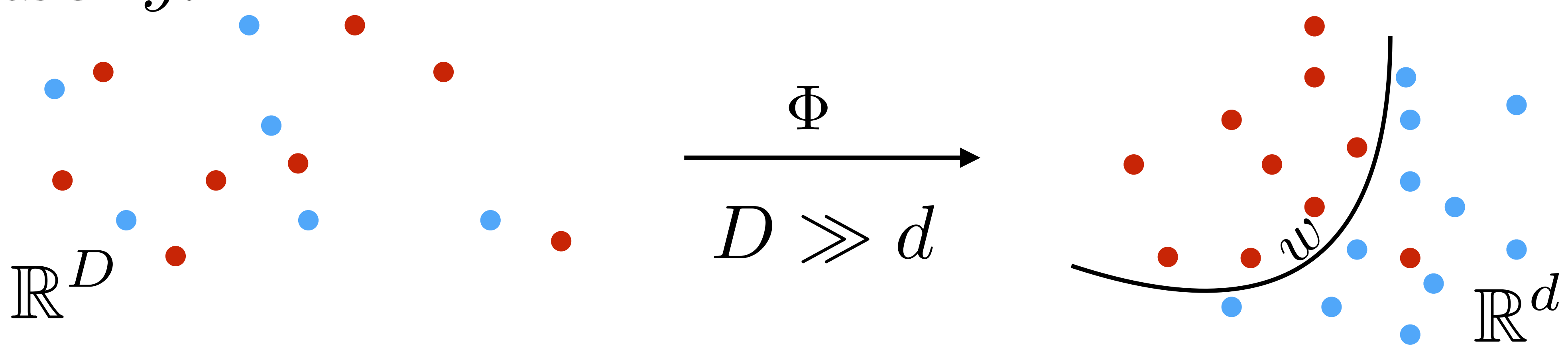
Intraclass variability
Not informative

Extraclass variability

High variance: must be reduced

Fighting the curse of dimensionality

- **Objective:** building a representation Φx of x such that a simple (say euclidean) classifier $\hat{y}$ can estimate the label y:



- Designing Φ consist of building an approximation of a low dimensional space which is regular with respect to the class:

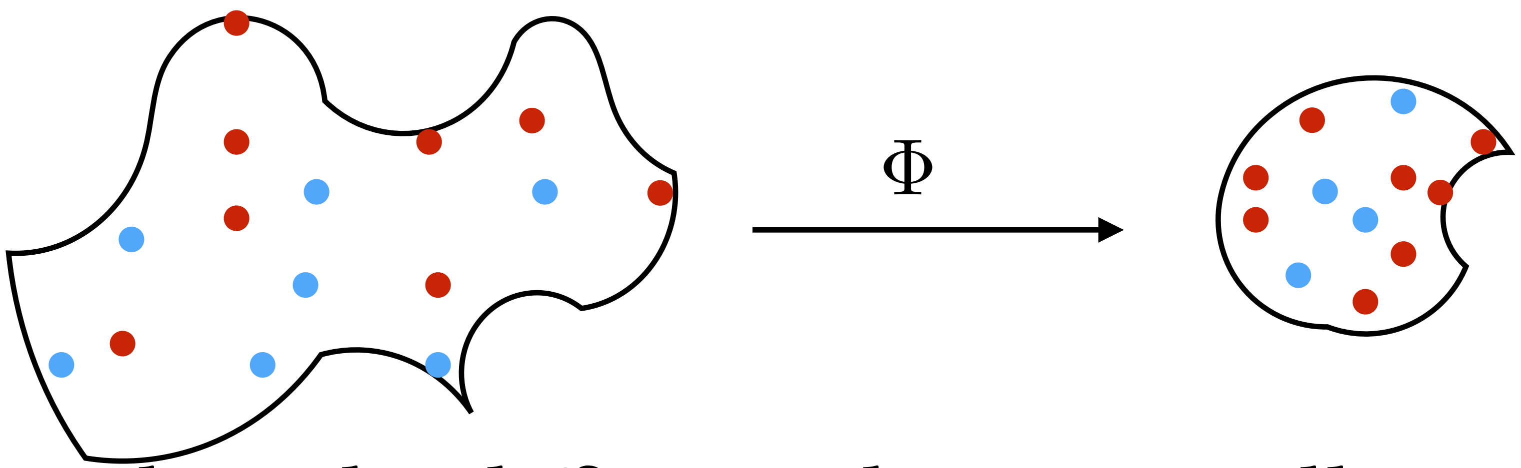
$$\|\Phi x - \Phi x'\| \lll 1 \Rightarrow \hat{y}(x) = \hat{y}(x')$$

- How can we do that?

Separation - Contraction

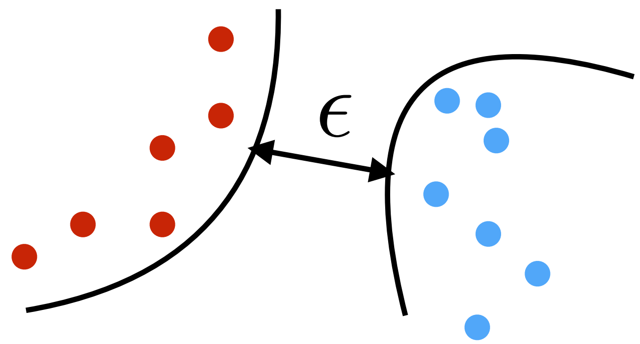
- In high dimension, typical distances are huge, thus an appropriate representation must contract the space:

$$\|\Phi x - \Phi x'\| \leq \|x - x'\|$$



- While avoiding the different classes to collapse:

$$\exists \epsilon > 0, y(x) \neq y(x') \Rightarrow \|\Phi x - \Phi x'\| \geq \epsilon$$



Nature of the variabilities

- A classification problem can be written as a loss minimisation:

$$\inf_{\text{Classifier},\Phi} \sum_i \text{loss}(x_i, y_i)$$

$$\text{loss}(x,y) = \|\text{Classifier}(\Phi x) - y\|$$

- A symmetry L corresponds to a transformation that preserves the class:

$$(x,y) \text{ in the training set} \iff (Lx, y) \text{ in the training set}$$

- That should preserve also the representation:

$$\Phi L x = \Phi x \Rightarrow \text{loss}(Lx, y) = \text{loss}(x,y)$$

An example: translation

- Translation is a linear action:

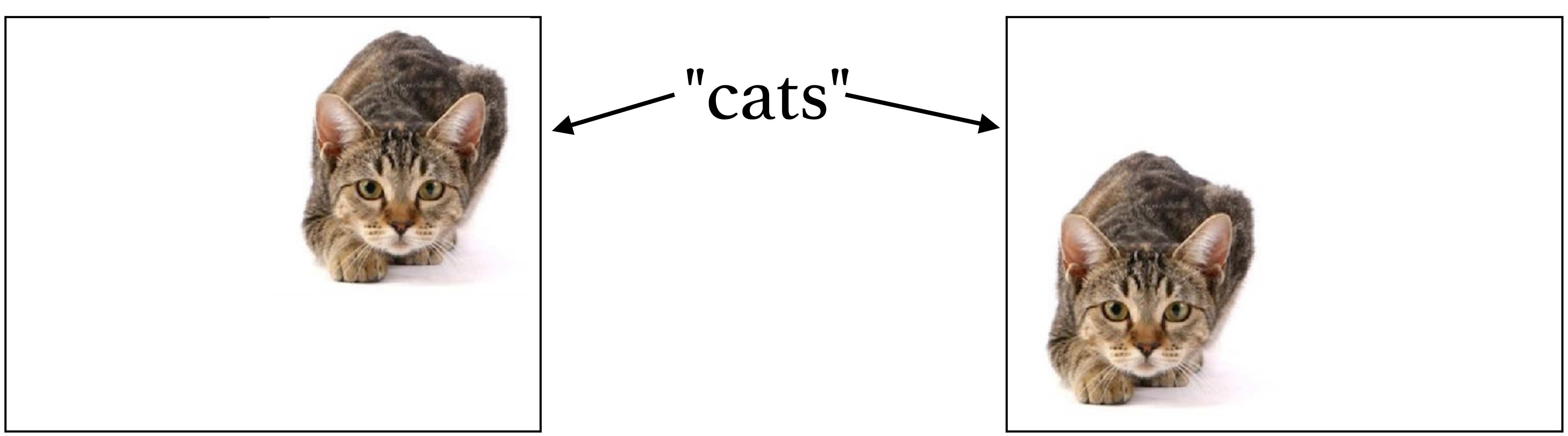
$$\forall u \in \mathbb{R}^2, L_a x(u) = x(u-a)$$

- (Culture) The set of translations is a (Lie) group with an exponential map:

$$L_{a+b} = L_a \circ L_b \text{ and } L_a x(u) = \sum_{n \geq 0} (\frac{d^n x}{du^n})_u \frac{(-a)^n}{n!} = e^{-a(\frac{d}{du})_u} x$$

Similar to: $\theta \to e^{i\theta}$

- In many case, it is a variability to reduce:



Convolution: covariance to translation

- A linear (bounded) operator W of L^2 is a convolution iff it is covariant with the action of translations:

$$\forall a, L_a W = W L_a \Rightarrow W x(u-a) = W x_a(u), x_a(u) = x(u-a)$$

- In this case,

$$\exists w, W x(u) = \int x(t) w(u-t) dt$$

- And it is diagonalised by its Fourier basis:

$$W e^{i\omega^T u} = \mathcal{F} w(\omega) e^{i\omega^T u}$$

Invariance to translation

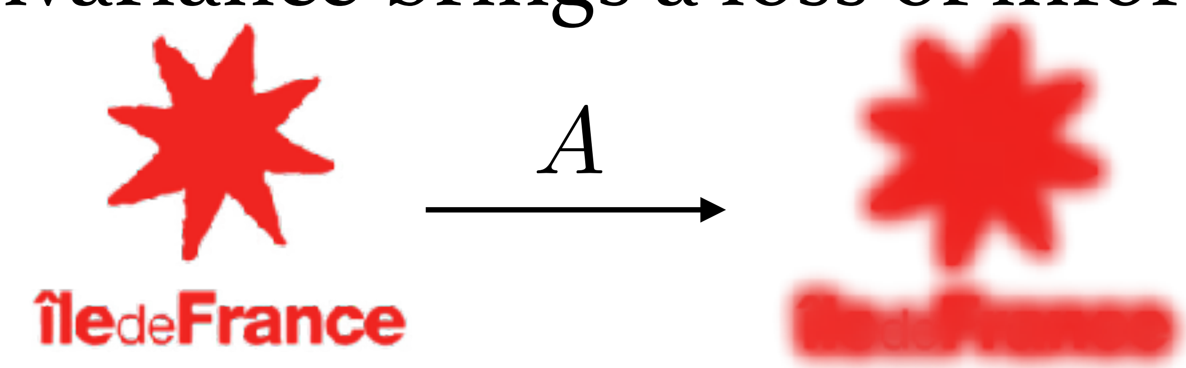
- In many cases, one wish to be invariant globally to translation, a simple way is to perform an averaging:

$$Ax = \int L_a x da = \int x(u)du$$

It's the 0 frequency!

$$AL_a = A$$

- Even if it can be localized, the averaging keeps the low frequency structures: the invariance brings a loss of information!



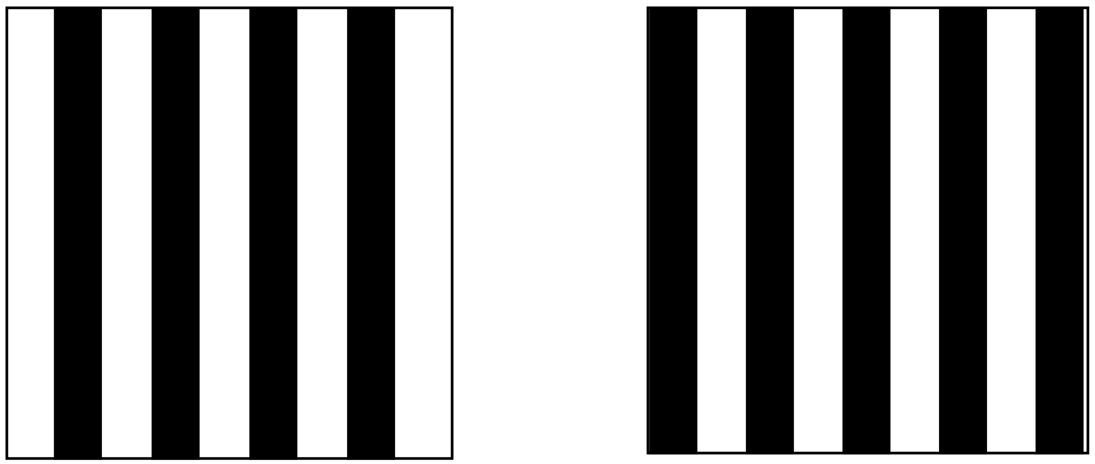
- Covariance (even non linear) and averaging imply invariance:

$$WL_a = L_a W \Rightarrow \quad AWL_a x = AL_a W x = AWx$$

An invariant is created!

DATA

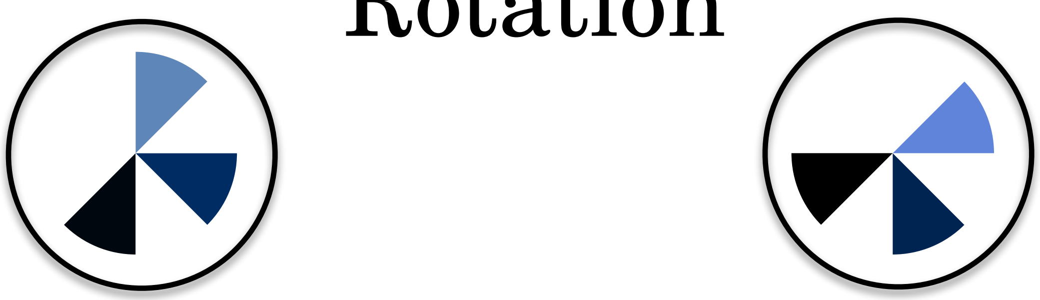
Translation



x y

$$\|x - y\|_2 = 2$$

Rotation



x y

Averaging is the key to get invariants

Averaging makes euclidean distance meaningful in high dimension

How to tackle the curse of dimensionality?

- Cascade of covariant operators with translation to build an invariant to translations:

$$AW_J ... W_1 L_a x = AW_J ... W_1 x$$

- Linear and non-linear contraction to reduce the volume:

$$\|\rho(x) - \rho(y)\| \leq \|x - y\|$$

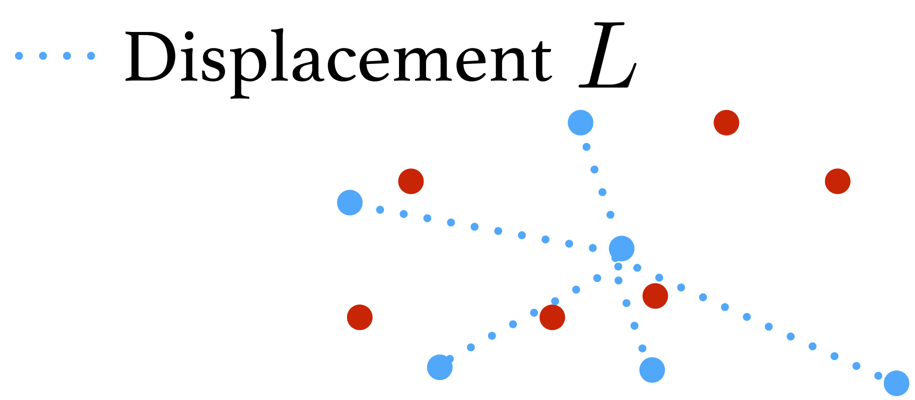
- An interesting object: $\Phi x = A\rho W_J ... \rho W_1 x$

How to tackle the curse of dimensionality? (2)

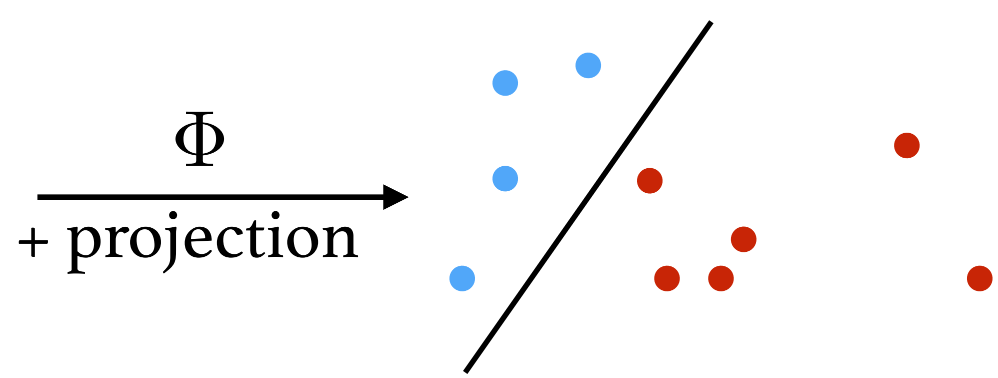
- Weak differentiability property:

$$\sup_L \frac{\|\Phi L x - \Phi x\|}{\|Lx - x\|} < \infty \Rightarrow \exists \text{ "weak" } \partial_x \Phi$$

$$\Rightarrow \Phi L x \approx \Phi x + \underbrace{\partial_x \Phi L}_{\text{A linear operator}} + o(\|L\|)$$



- A linear projection (to kill L) build an invariant



How can we build Φ ?

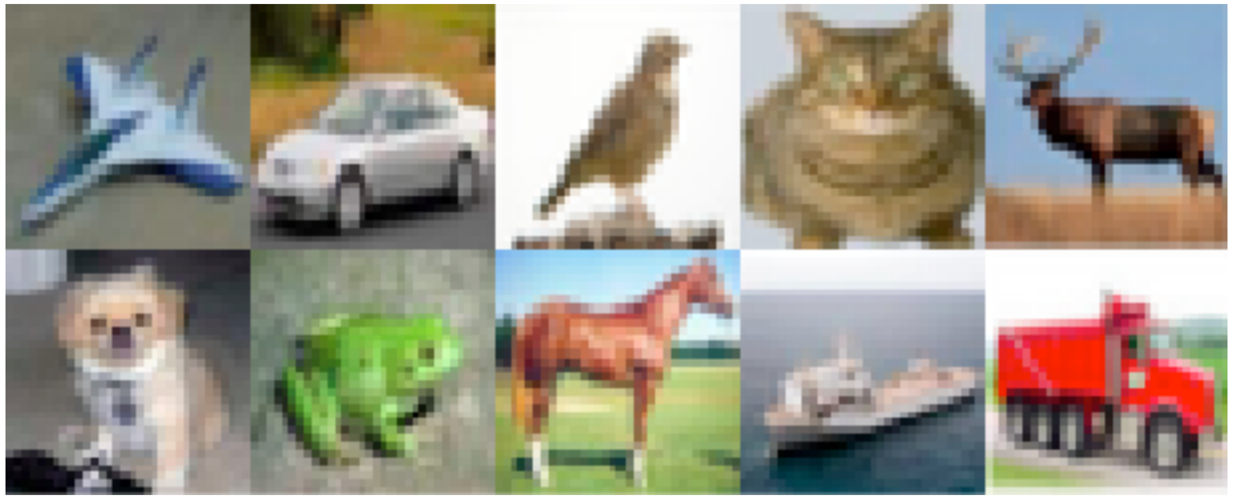
- Enumerating the different variabilities is **hard.**
- Since Deep neural networks solve the vision classification task, it is **necessary** they build invariance to a large set of intra-class variabilities.
- So, what is a Deep network?

Delving into the technique

- Building a Deep network is **challenging.**
- It requires a **large** amount of data and **GPUs**
- … and there are many more details.

Dataset: CIFAR

- 50 000 images for training, 10 000 images for testing, of size 32x32 (small), 10/100 classes

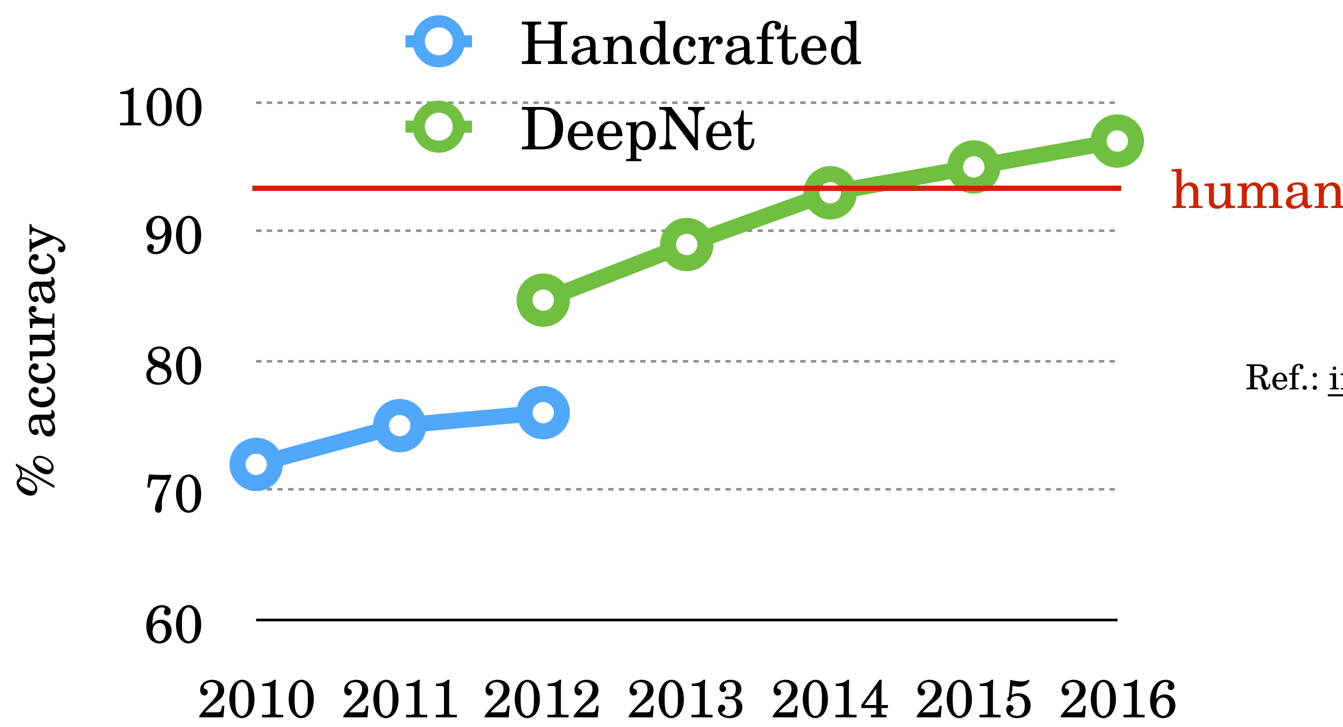


Dataset: Imagenet

- 1,2M labeled images for training, 1000 classes (car, dog, ...) of various sizes, 400k for testing
- Natural images with large variability

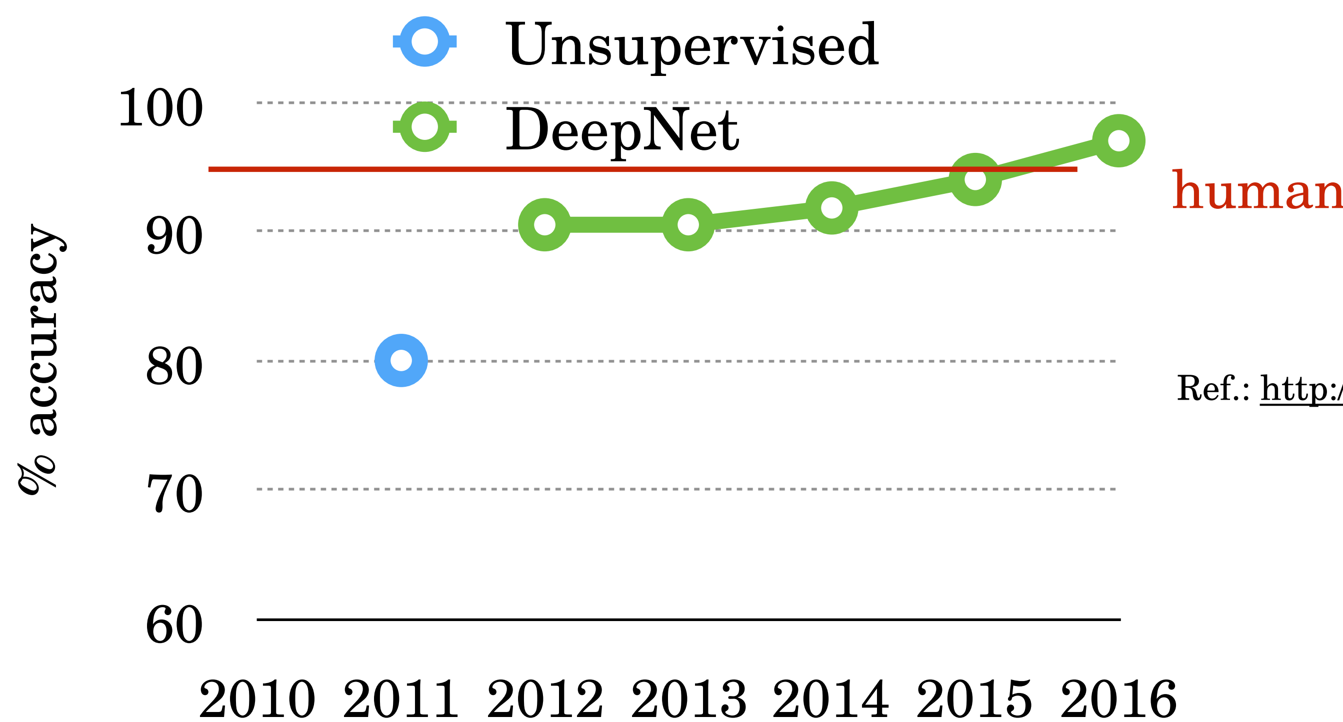
Benchmarks

ImageNet



Ref.: image-net.org

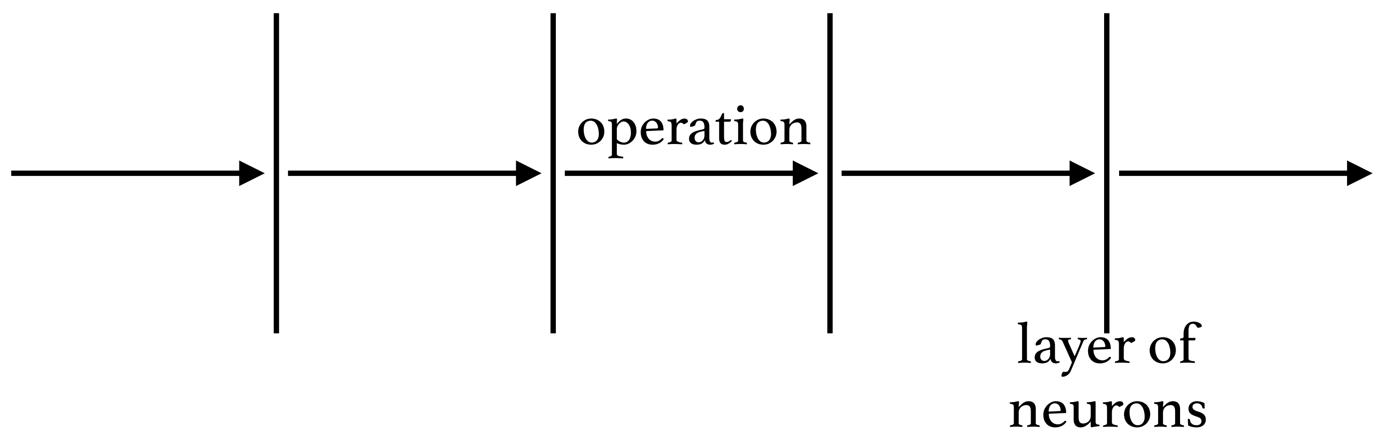
CIFAR



Ref.: http://rodrigob.github.io/are_we_there_yet/build/

DeepNet?

- A DeepNet is a **cascade** of linear operators with a point-wise non-linearity.



Ref.: Rich feature hierarchies for accurate object detection and semantic segmentation. Girshick et al.

Convolutional network and applications in vision. Y. LeCun et al.

- Each operators is **supervisedly** learned

- Formal way to write it:

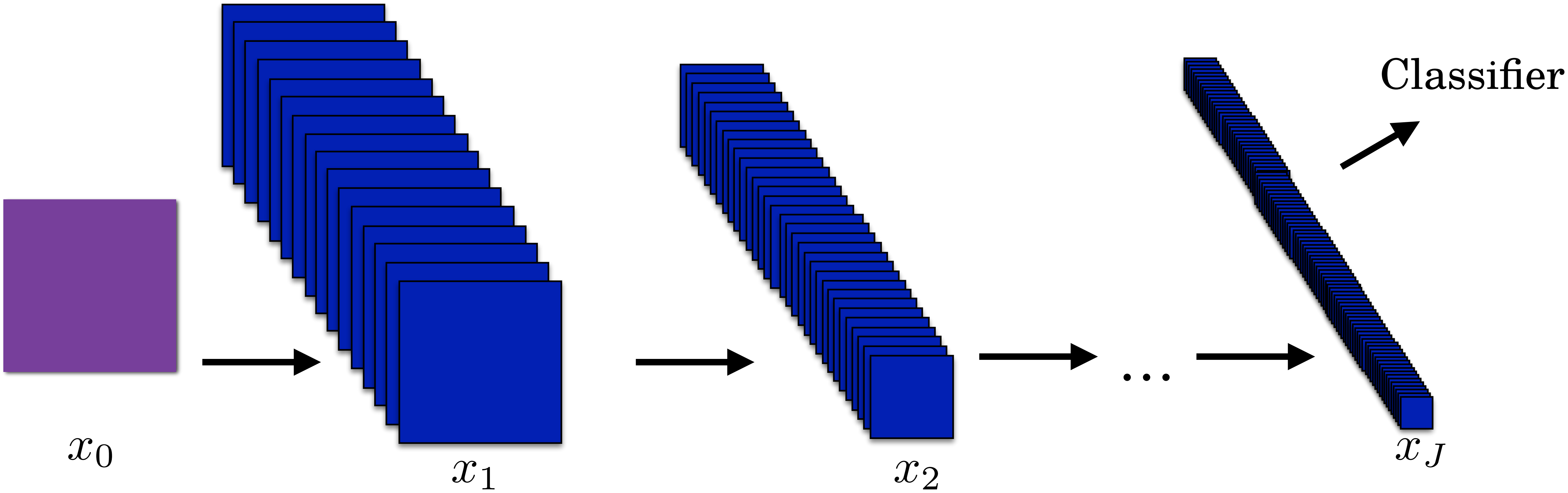
Linear

$$x_{j+1} = \rho W_j x_j$$

Pointwise non-linearity

Architecture of a CNN

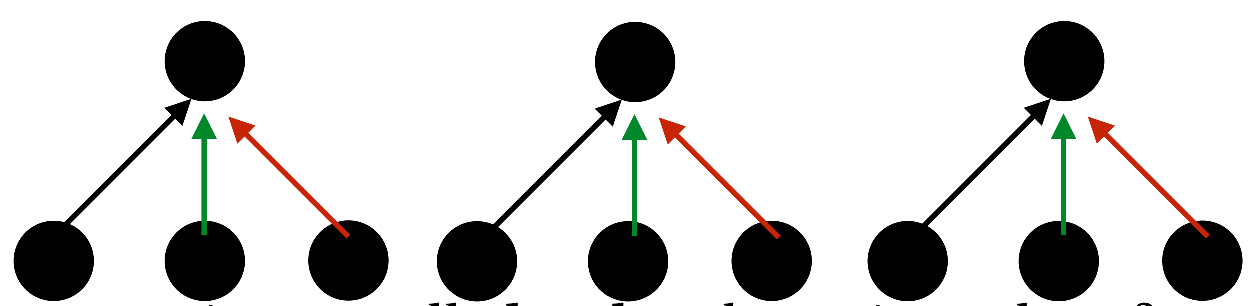
- Cascade of convolutional operator and non-linear operator:



$$x_{j+1}(u,\lambda) = \rho\Big(\sum_{\tilde{\lambda}} x_j(.,\tilde{\lambda}) \star w_{j,\lambda,\tilde{\lambda}}(u)\Big)$$

The kernel is learned

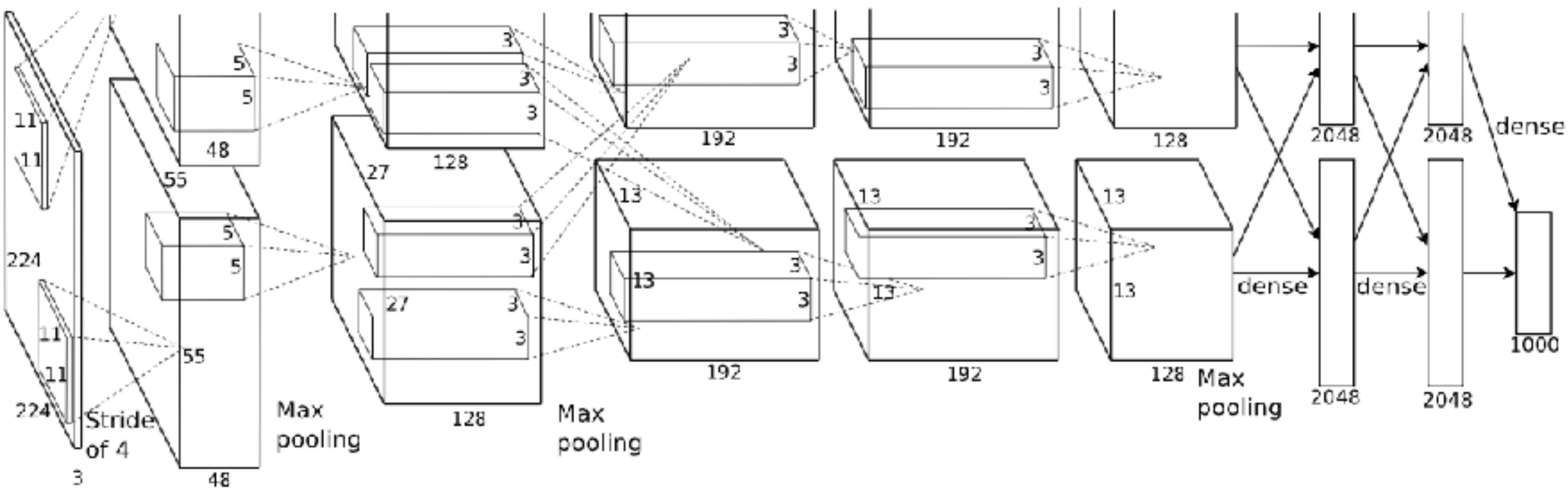
- Can be interpreted as neurons sharing weights:



- Designing a state-of-the-art deep net is generally **hard** and requires a lot of **engineering**

Typical CNN architecture

Ref.: ImageNet Classification with Deep Convolutional Network, A Krizhevsky et al.



"AlexNet"

60M parameters, 8 layers

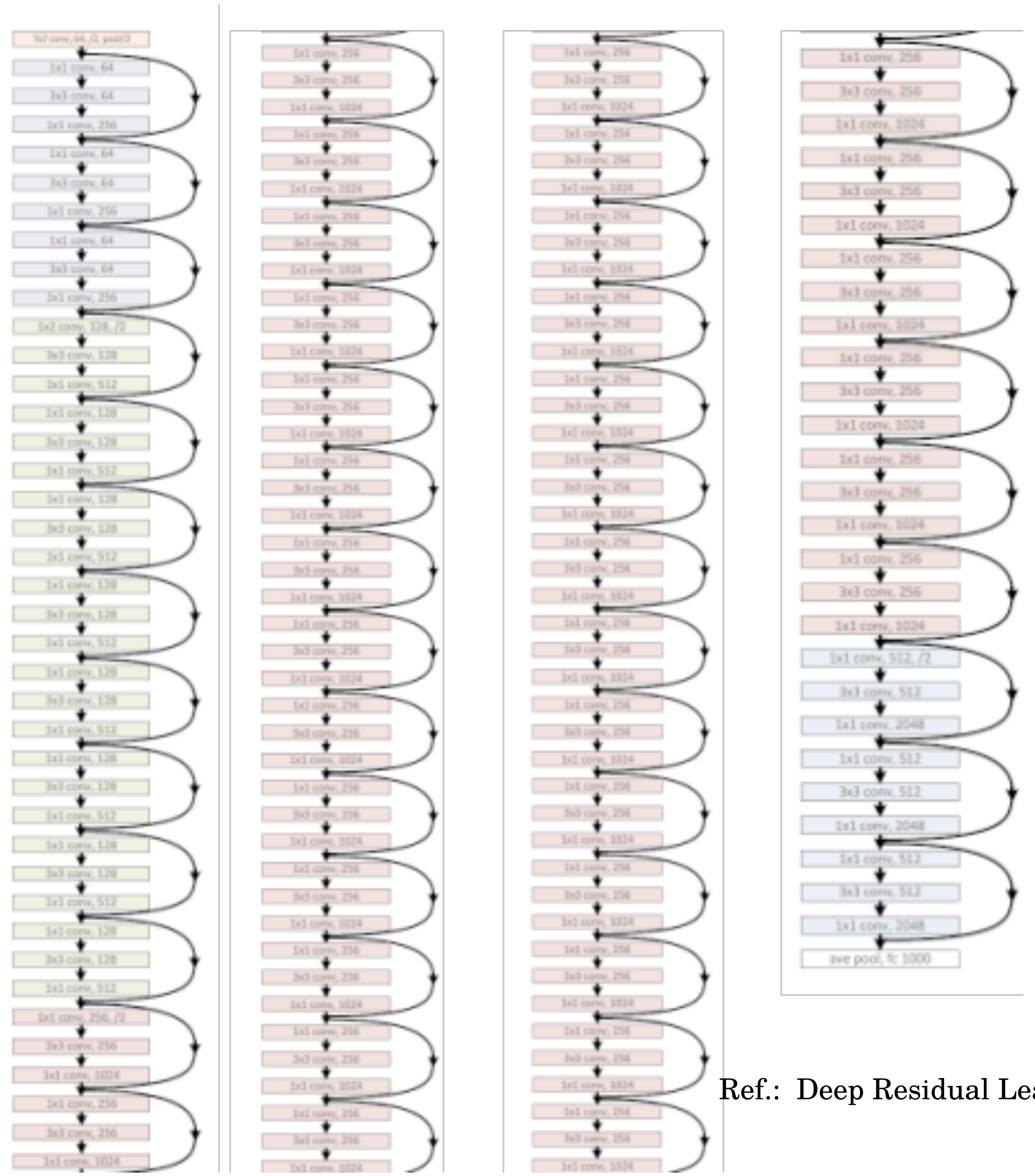
ENS

Inception Net

Ref.: Going Deeper with Convolutions, C Szegedy et al.

5M parameters, 38 layers

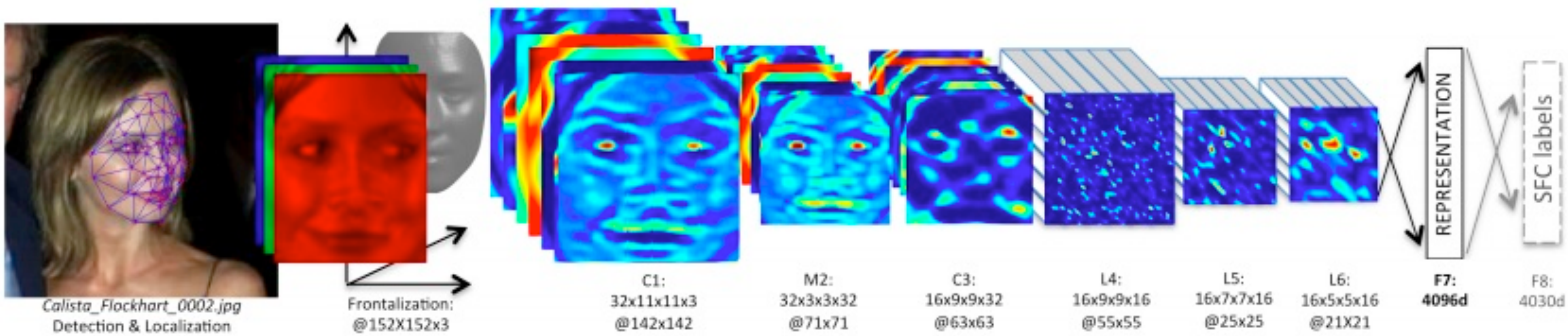
ResNet



Ref.: Deep Residual Learning for Image Recognition, K He et al.

4M parameters, 152 layers

Deep Face



120M parameters, 7 layers

Optimizing a DeepNet

- The output Φx has the dimension of the number of classes. The DeepNet operators are optimised via the neg cross entropy and a stochastic gradient descent:

$$-\sum_{n}\sum_{\text{class}} 1_{y_n=\text{class}} \log(\Phi x_n)_{\text{class}}$$

Ref.: Convolutional network and applications in vision. Y. LeCun et al.

- All the functions are differentiable: back propagation algorithm+ stochastic gradient:

learning rate

randomly selected sample

$$w_j^{i+1} = w_j^i - \alpha_i \nabla w_j (w_j^i, X_j)$$

- It is **absolutely non-convex! No guarantee** to converge.

CUDA

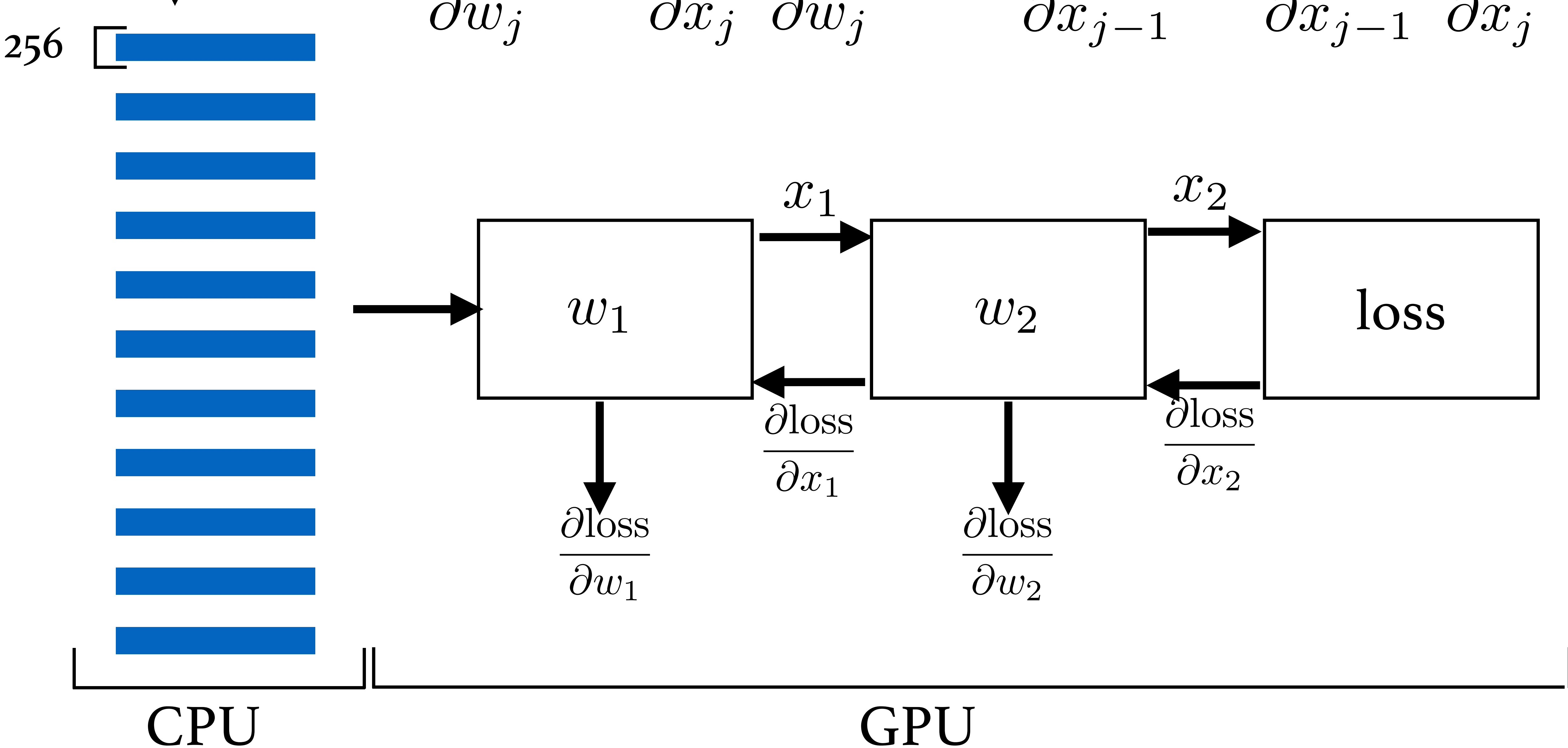
- Deep learning algorithms rely a lot on **linear** operations.
- CUDA routines permit to implement efficiently linear algebra routines: speed up of **<u>80.</u>**
- What costs a lot of with a GPUs are the I/O

Implementation of a CNN

Typical training time on imagenet: 100 epochs
2 hours per epoch

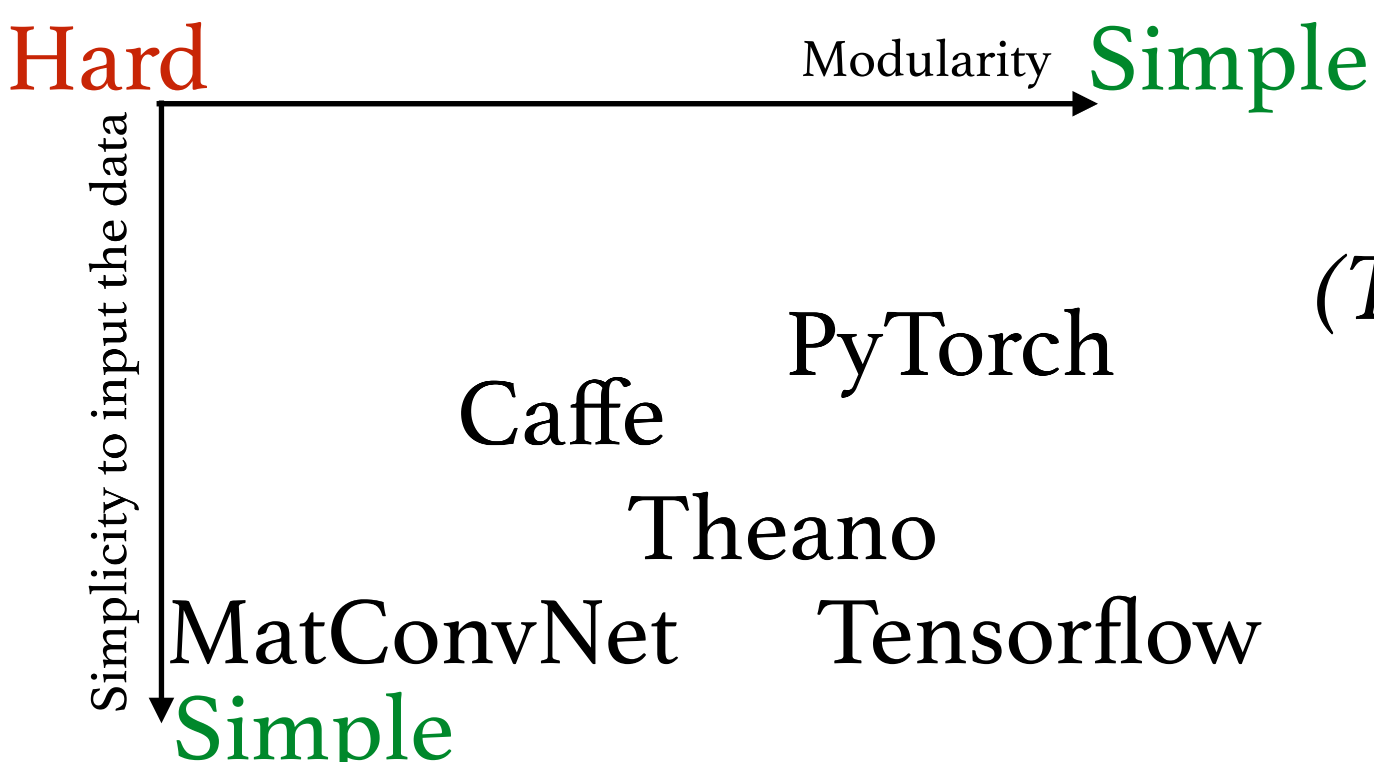
Splitting dataset
into batches of size
256

$$\frac{\partial \text{loss}}{\partial w_j} = \frac{\partial \text{loss}}{\partial x_j} \frac{\partial x_j}{\partial w_j} \qquad \frac{\partial \text{loss}}{\partial x_{j-1}} = \frac{\partial x_j}{\partial x_{j-1}} \frac{\partial \text{loss}}{\partial x_j}$$



Softwares…

- All the packages are based on GPUs, select your favorite via: simplicity of benchmarking, data input…
- All available in python or C++ ; developed by FB, Google, … there is a war!



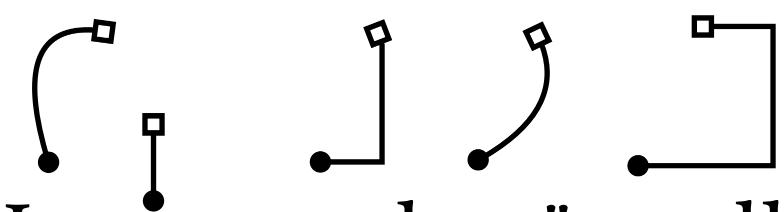
(This is subjective)

Training your own CNN

- Again, the optimisation is no convex: a lot of hyper parameters (learning rate, l2 regularization…) to tune:
- Demo!

Why is deep learning dangerous?

- **Pure black box.** Few mathematical results are available. Many rely on a "manifold hypothesis". Clearly wrong:
  Ex: stability to diffeomorphisms



- No **stability results.** It means that "small" variations of the inputs might have a large impact on the system. And this happens.
- **Small** data?
- Shall we learn **each layer** from scratch? (geometric priors?)
- Thanks to the cascade, features are hard to **interpret**

Identifying the variabilities?

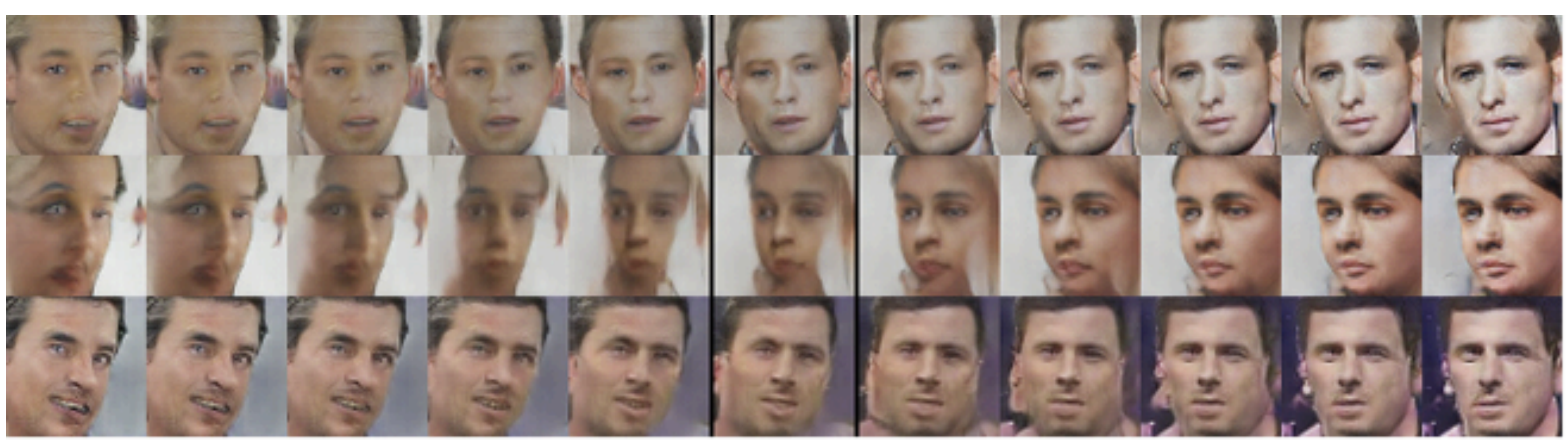
- Several works showed a deepnet exhibits some covariance:

(c) Object color (d) Background color

(a) Lighting (b) Scale

Ref.: Understanding deep features with computer-generated imagery, M Aubry, B Russel

- Manifold of faces at a certain depth:

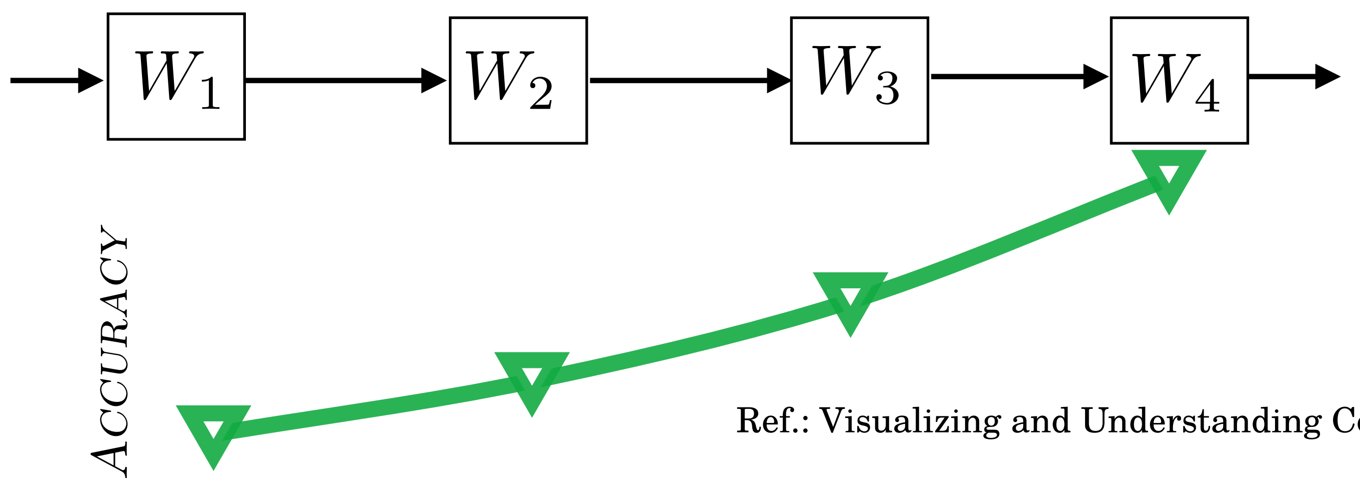


Ref.: Unsupervised Representation Learning with Deep Convolutional GAN, Radford, Metz & Chintalah

- Can we generalise these?

Why does it work?

- Progressively, there is a linear separation that occurs



Ref.: Visualizing and Understanding Convolutional Networks, M Zeiler, R Fergus

- In fact, euclidean distances become more meaningful with depth and symmetry groups seem to appear.

Ref.: Building a Regular Decision Boundary with Deep Networks, CVPR 2017, EO
Mutiscale Hiearchical Convolutional Network, Jacobsen, O, Mallat, Smeulders

<u>**Indicates a *progressive* dimensionality reduction!**</u>

Wavelets: avoiding learning?

- Wavelets help to describe signal structures. ψ is a wavelet iff

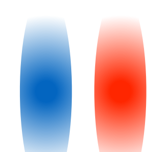
$$\psi \in \mathcal{L}^2(\mathbb{R}^2, \mathbb{C}) \text{ and } \int_{\mathbb{R}^2} \psi(u)du = 0$$

- They are chosen localised in space and frequency.

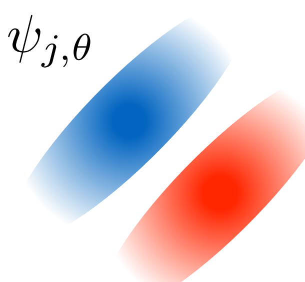
- Wavelets can be dilated in order to be a **multi-scale** representation of signals, **rotated** to describe rotations.

$$\psi_{j,\theta} = \frac{1}{2^{2j}} \psi(\frac{r_\theta(u)}{2^j})$$

ψ



$\psi_{j,\theta}$



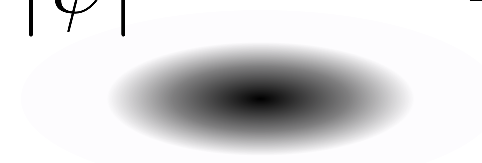
- Design wavelets selective to an **informative** variability.

Isotropic

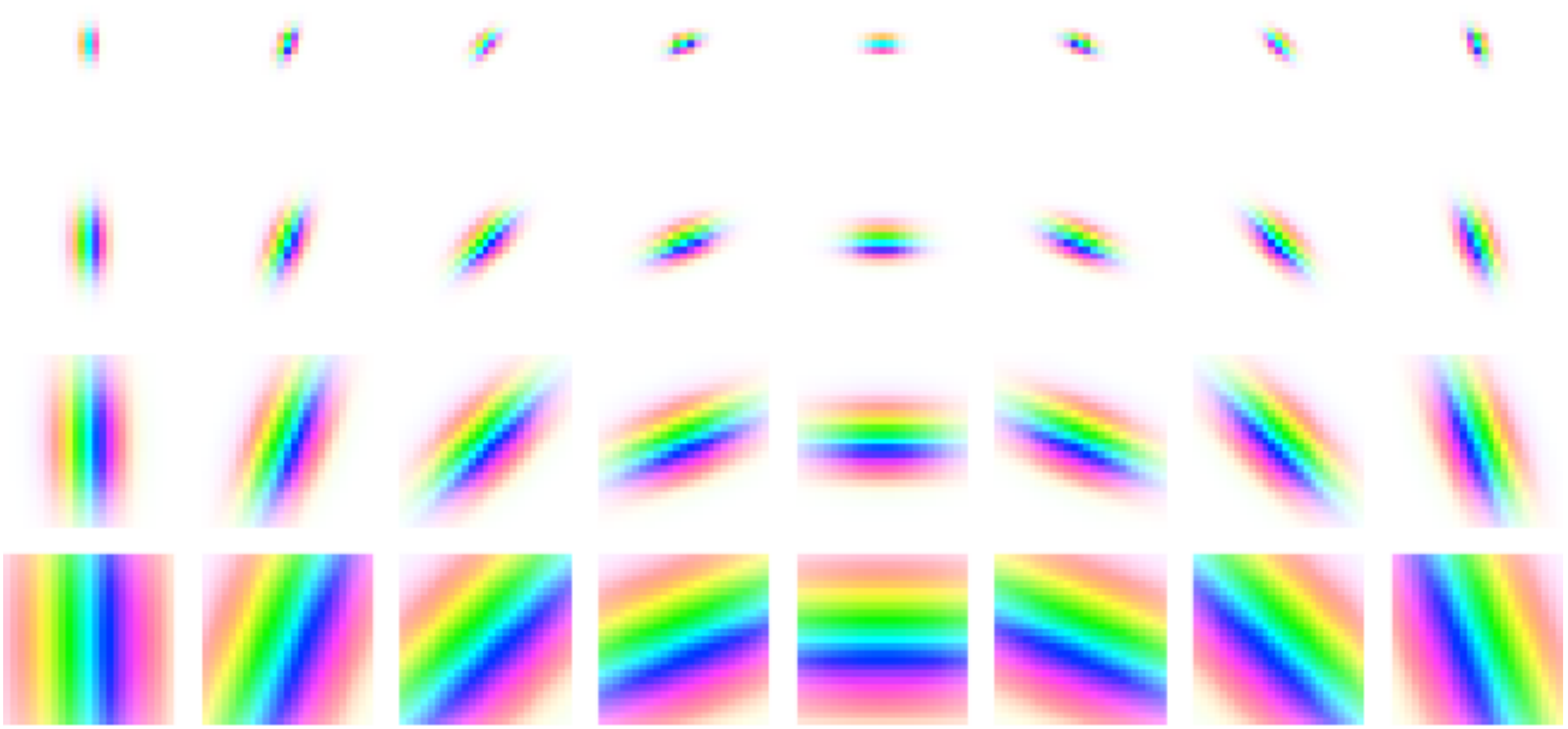
$|\hat{\psi}|$

VS

$|\hat{\psi}|$



Non-Isotropic



$$\psi(u) = \frac{1}{2\pi\sigma} e^{-\frac{\|u\|^2}{2\sigma}} (e^{i\xi . u} - \kappa)$$

$$\phi(u) = \frac{1}{2\pi\sigma} e^{-\frac{\|u\|^2}{2\sigma}}$$

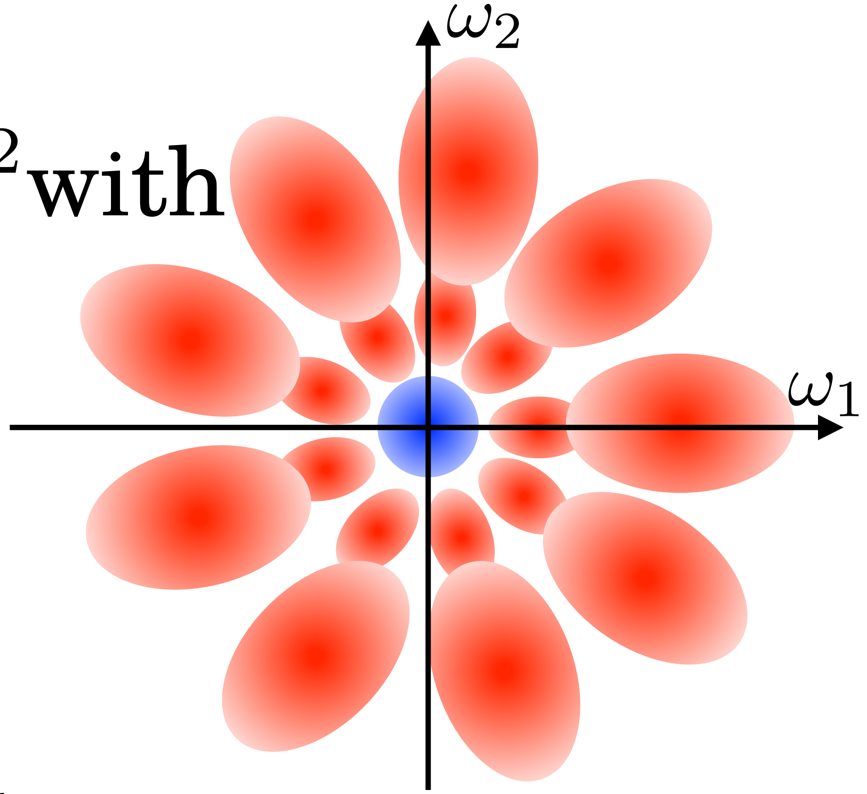
Heisenberg principle!
Good localisation in space and Fourier

(for sake of simplicity, formula are given in the isotropic case)

The Gabor wavelet

Wavelet Transform

- Wavelet transform : $Wx = \{x \star \psi_{j,\theta}, x \star \phi_J\}_{\theta, j \leq J}$



- Isometric and linear operator of L^2, with

$$\|Wx\|^2 = \sum_{\theta, j \leq J} \int |x \star \psi_{j,\theta}|^2 + \int x \star \phi_J^2$$

- Covariant with translation

$$W(x_{\tau=c}) = (Wx)_{\tau=c}$$

- Nearly commutes with diffeomorphisms

$$\|[W, ._\tau]\| \leq C \|\nabla \tau\|$$

Ref.: Group Invariant Scattering, Mallat S

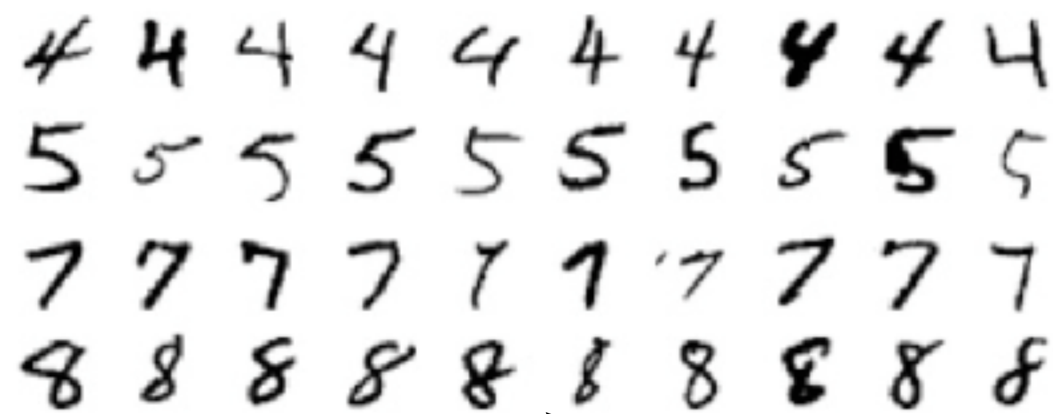
- A good baseline to describe an image!

Success story
Wavelets for Textures&Digits

- Non-learned representation have been successively used on:

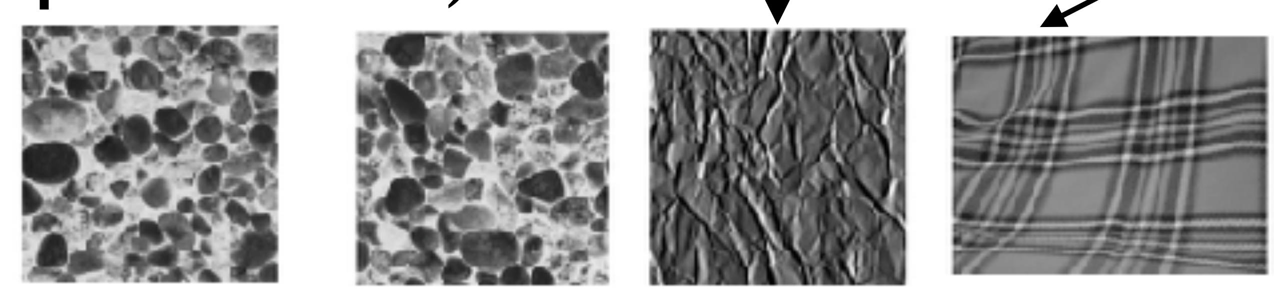
Ref.: Invariant Convolutional Scattering Network, J. Bruna and S Mallat

- Digits (patterns):



- Textures (stationary processes):

Ref.: Rotation, Scaling and Deformation Invariant Scattering for texture discrimination, Sifre L and Mallat S.



- However all the variabilities (groups) here are **perfectly** understood. (not with natural images)

Conclusion

- Deep Learning architectures are of interest thanks to their **outstanding numerical results.**
- Black boxes must be opened via maths.
- Check my website for softwares and papers: http://www.di.ens.fr/~oyallon/

Thank you!