Lecture: Deep Learning and Differential Programming

3.2 Visualization of learned knowledge

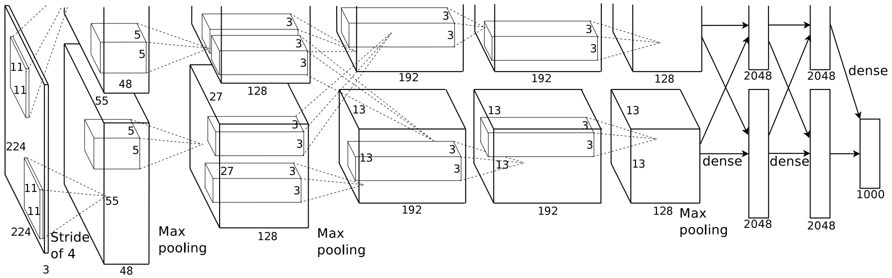
https://liris.cnrs.fr/christian.wolf/teaching

INSA LYON Christian Wolf

Our models are getting more complex.

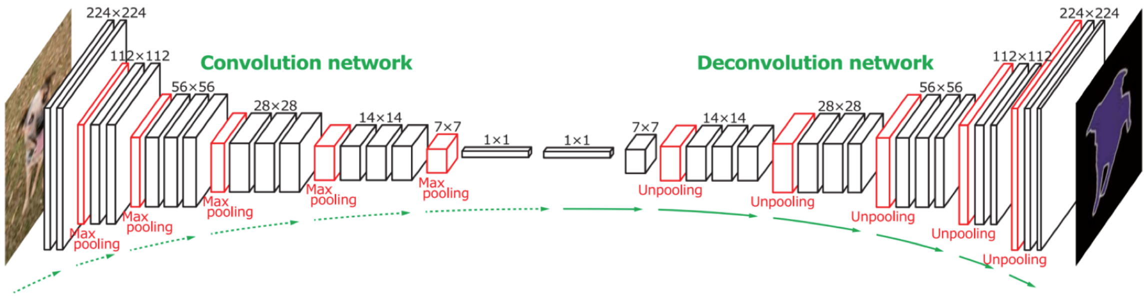
How can we visualize the knowledge acquired from data?

AlexNet

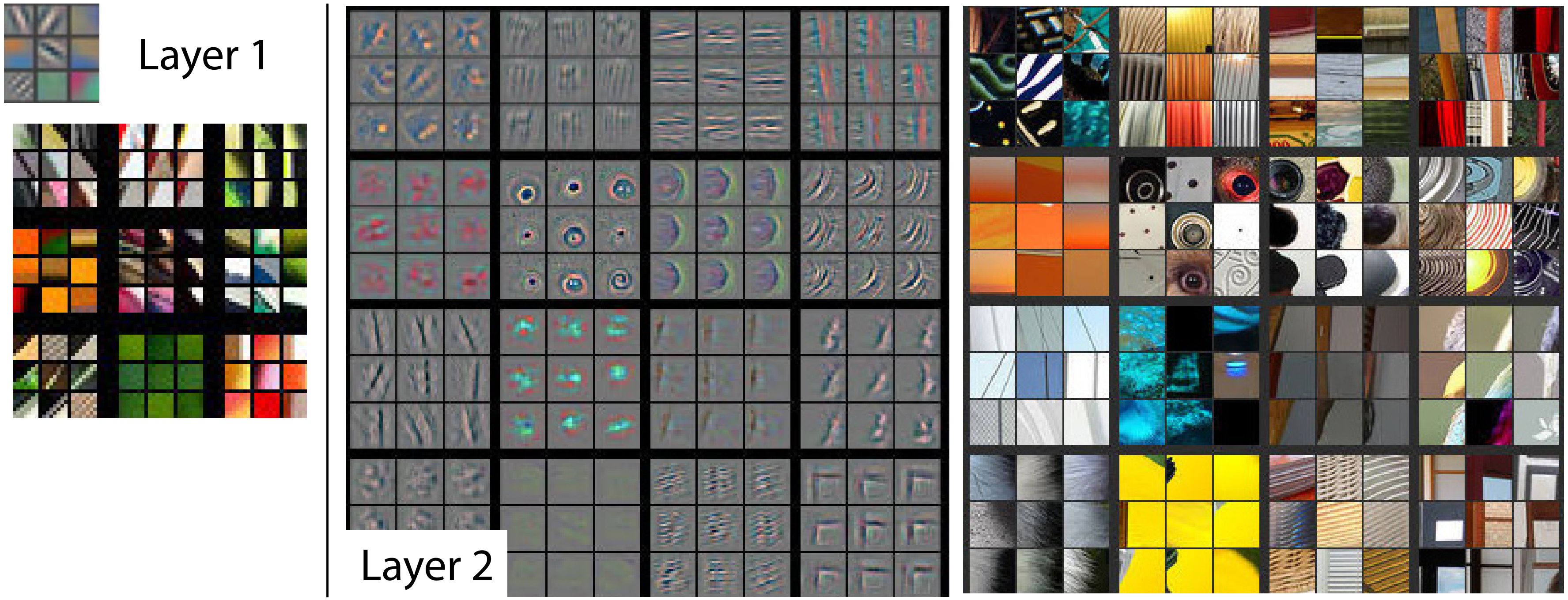


ResNet

Conv-Deconv

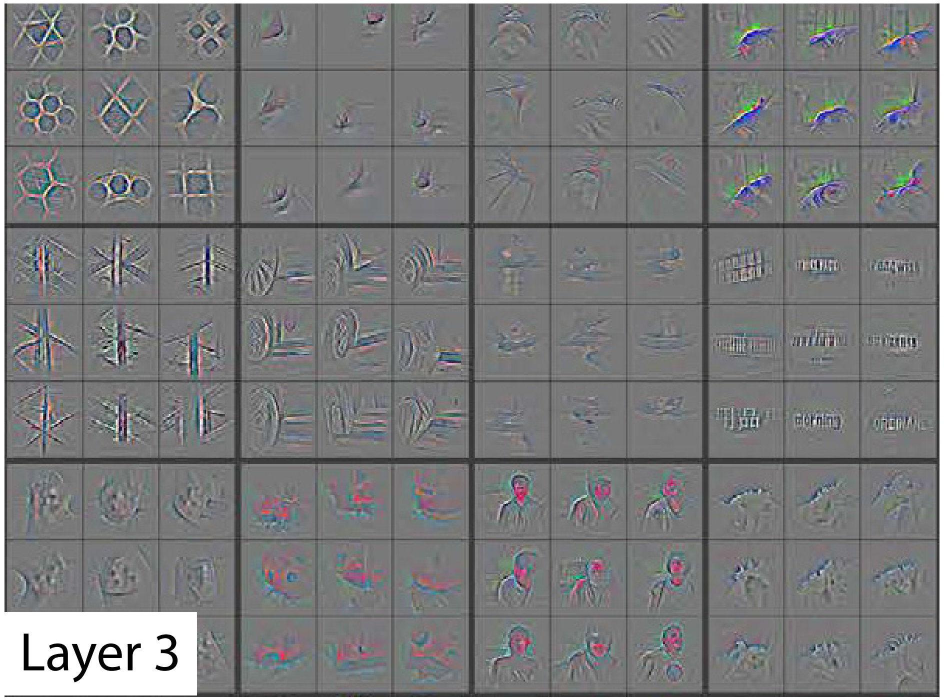


Visualizing feature map activations



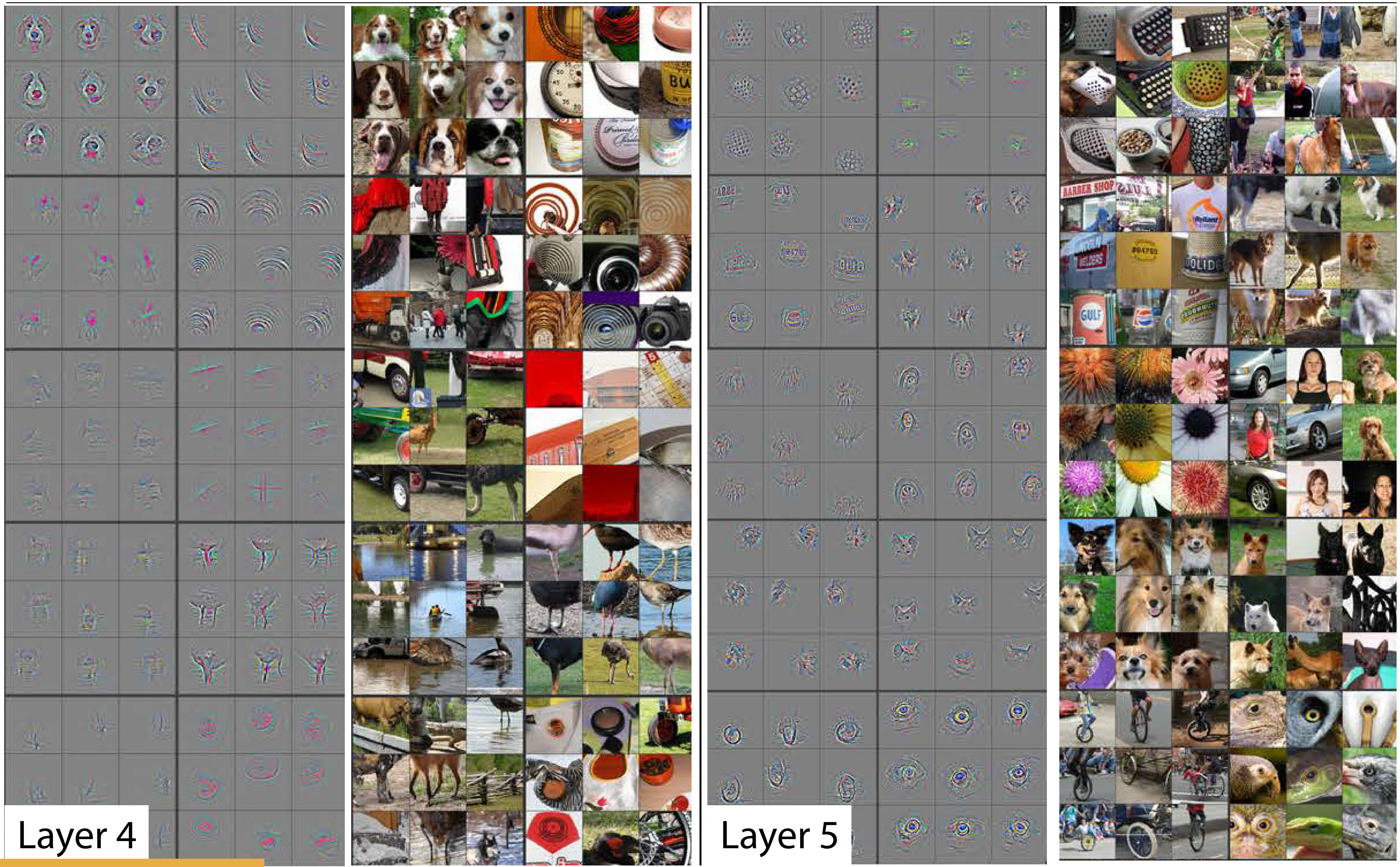
[Zeiler and Fergus, ECCV 2014]

Visualizing feature map activations



[Zeiler and Fergus, ECCV 2014]

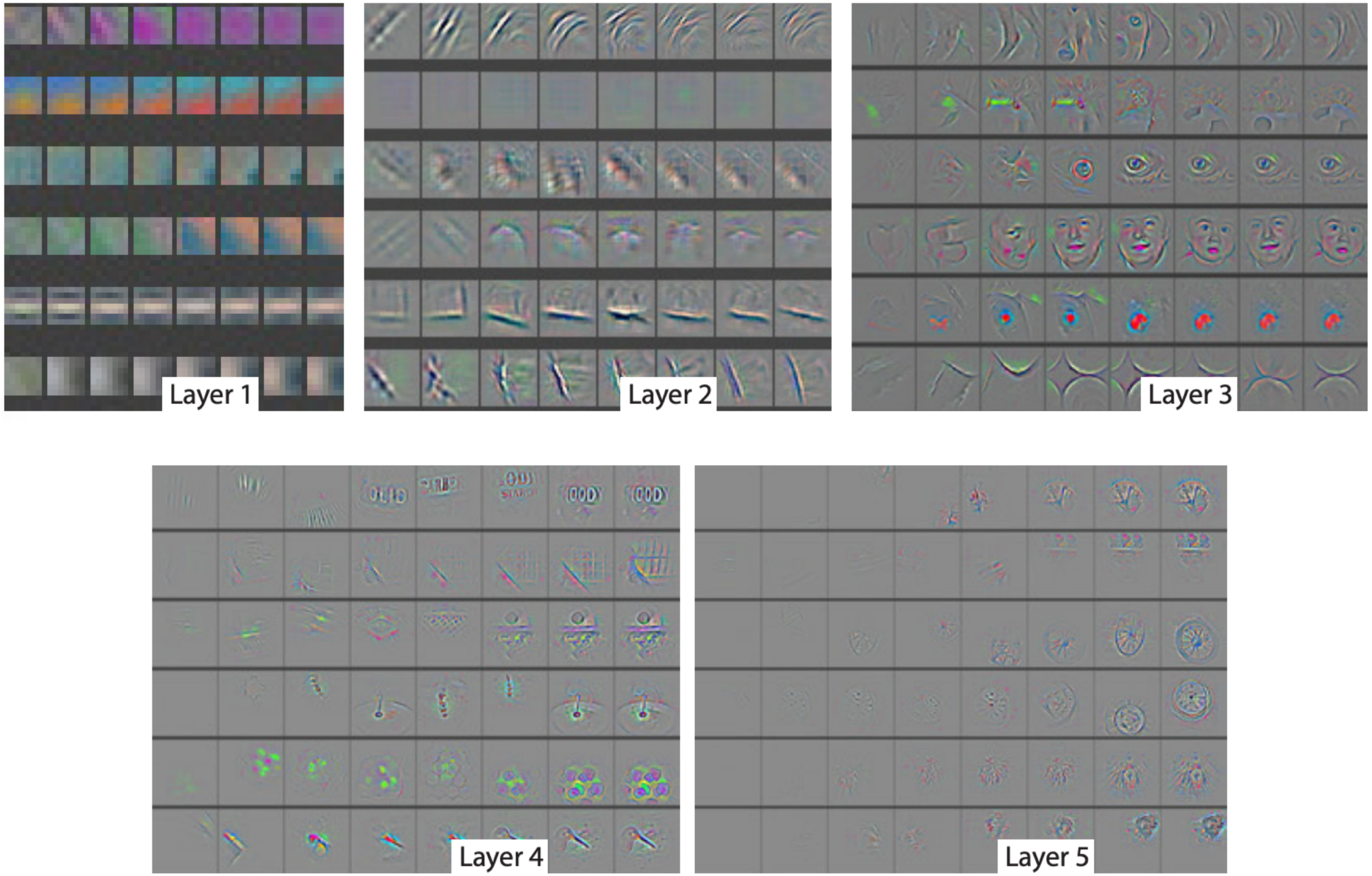
Visualizing feature map activations



[Zeiler and Fergus, ECCV 2014]

Earlier layers train earlier

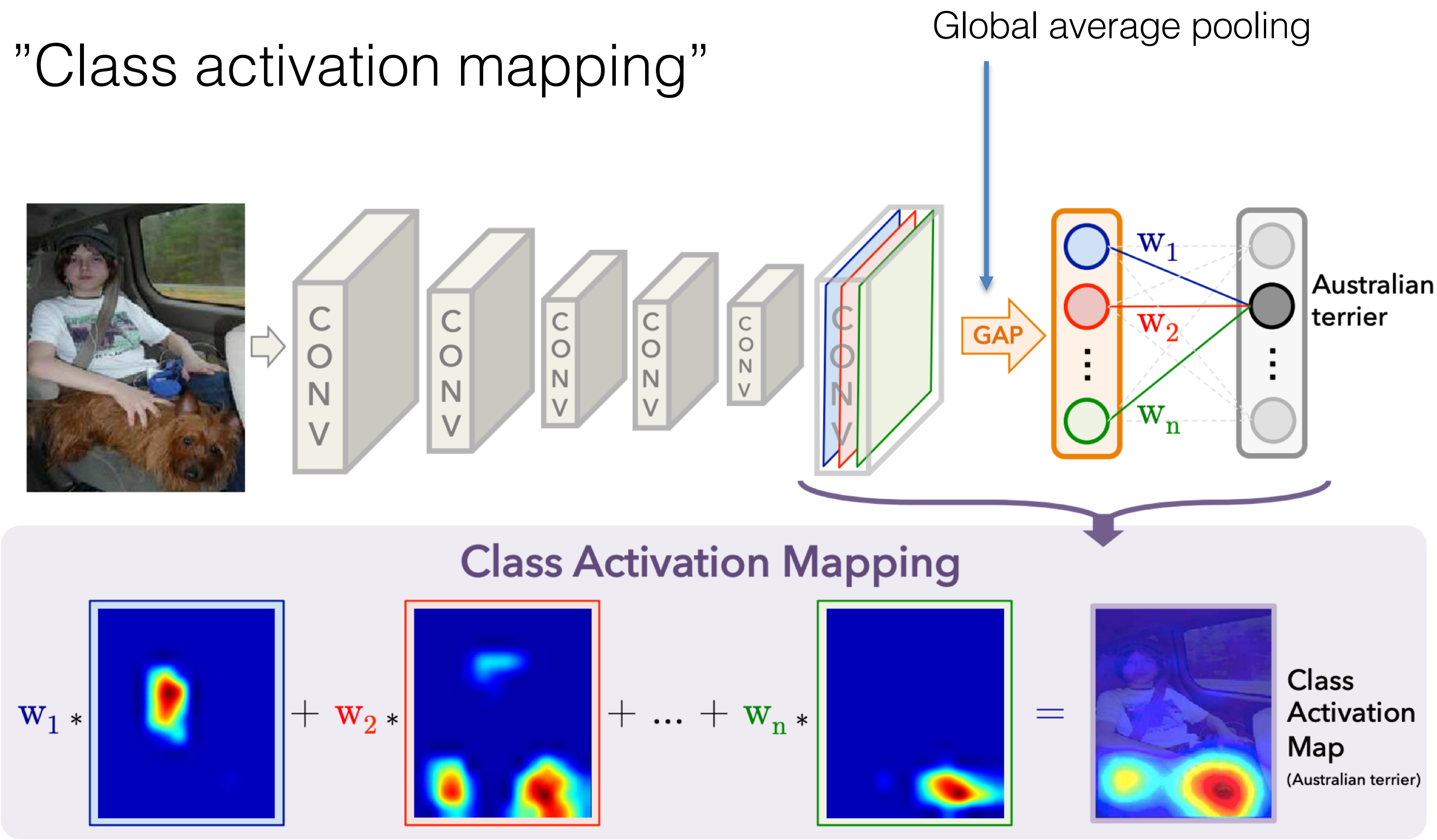
Evolution of features during training.



[Zeiler and Fergus, ECCV 2014]

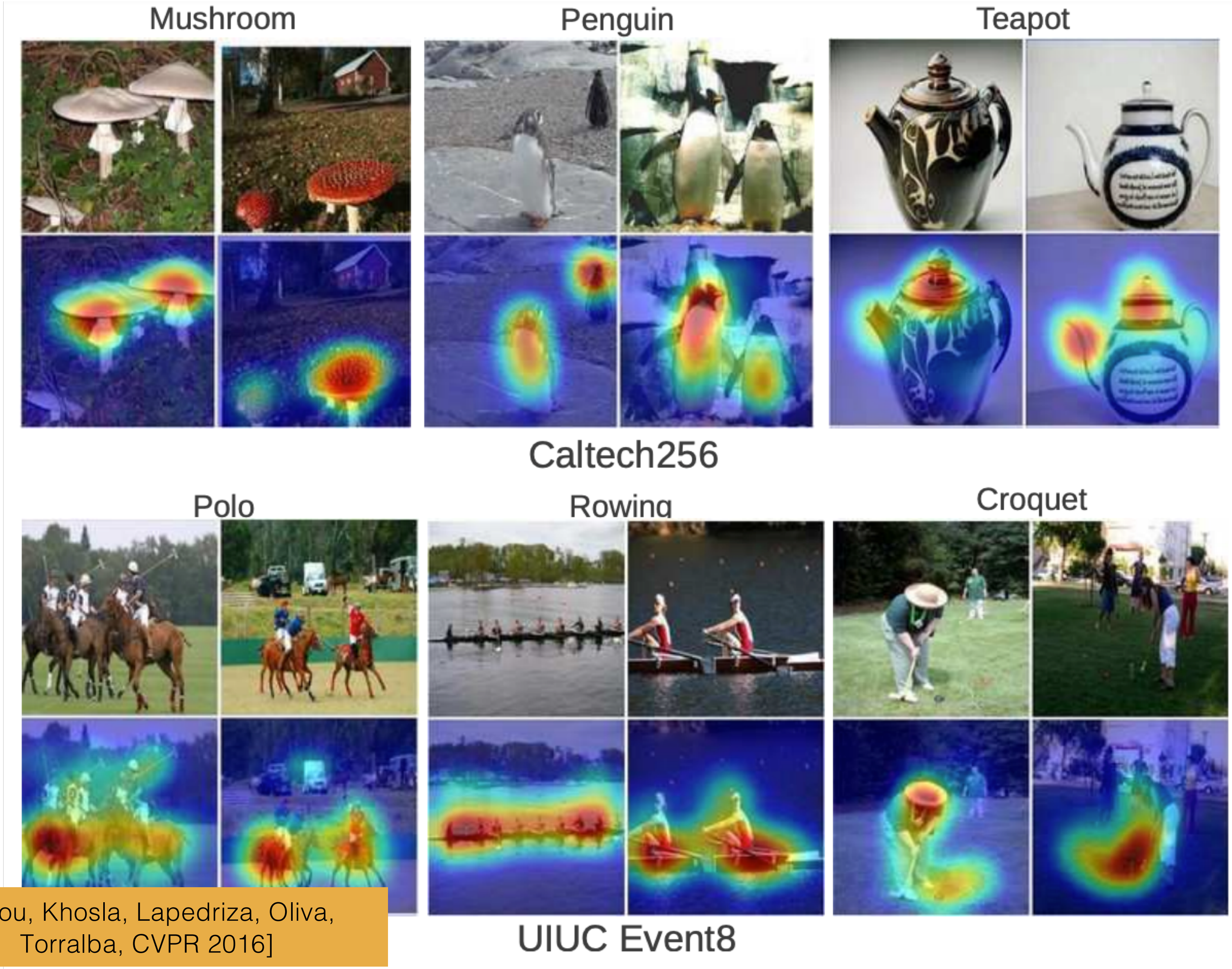
CAM

"Class activation mapping"



[Zhou, Khosla, Lapedriza, Oliva, Torralba, CVPR 2016]

CAM: examples



[Zhou, Khosla, Lapedriza, Oliva, Torralba, CVPR 2016]

Guided Backpropagation

We can directly calculate the influence of each individual input pixel to a given feature map layer by calculating the gradient of this layer w.r.t. to the input signal:

$$\frac{\partial A^k_{i,j}}{\partial x_{m,n}}$$

$A^k_{i,j} \ldots$ feature cell (i, j) of a given map k.
$x_{m,n} \ldots$ input pixel (i, j).

[Springenberg, Dosovitskiy, Brox, Riedmiller, ICLR-Workshop 2015]

Guided Backpropagation: effects on class

We can derive the output for class k w.r.t. the input pixels:

$$\frac{\partial y^c}{\partial x_{m,n}}$$

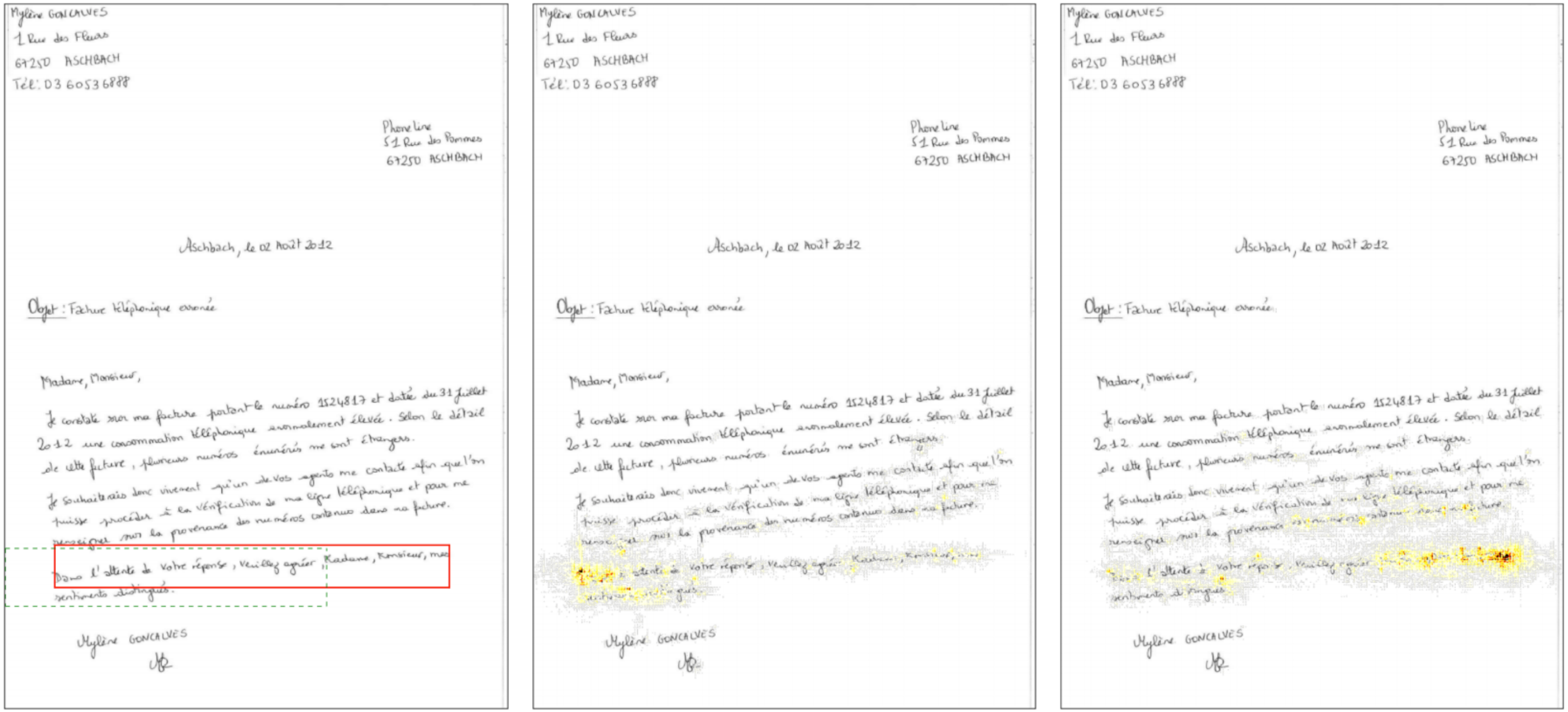
$u^c \ldots$ network output for class c given input image x
$x_{m,n} \ldots$ input pixel (m, n).

[Springenberg, Dosovitskiy, Brox, Riedmiller, ICLR-Workshop 2015]

Example: document analysis

Task: detection of text line bounding boxes.

Derivatives w.r.t. to the 4 outputs (left, right, top, bottom)

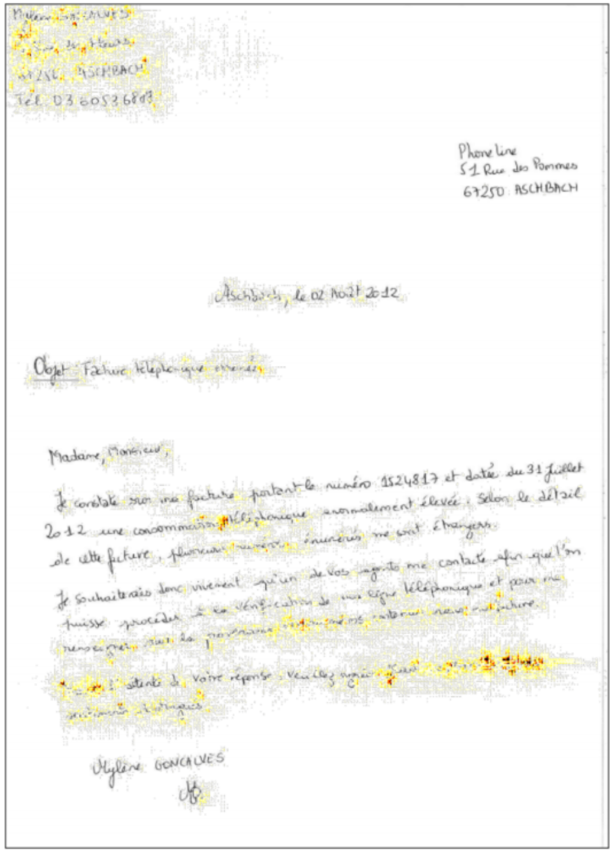


Left

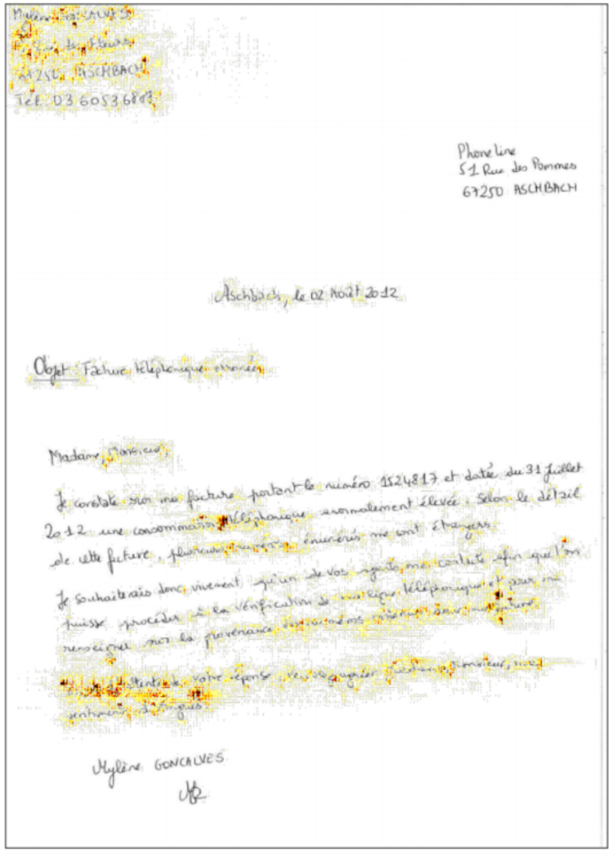
Right

[Bastien Moysset, PhD Thesis, 2018]

Example: document analysis



Top



Bottom

The network looks at the right side to regress the top coordinate: it figured out line slant!

[Bastien Moysset, PhD Thesis, 2018]

Visualizing high-dimensional spaces

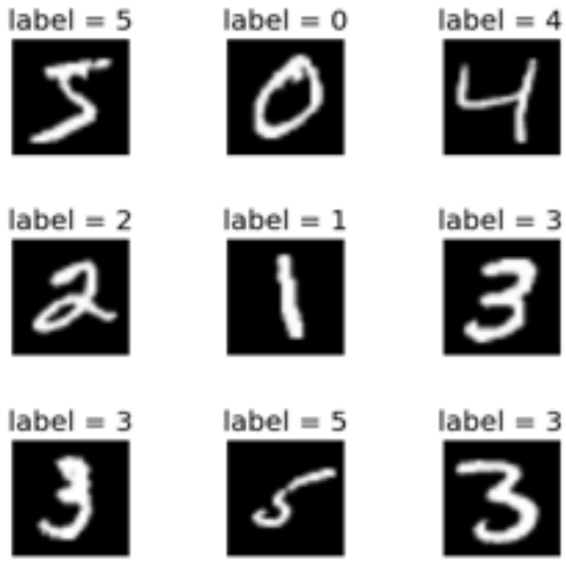
Problem: many tensors (input images or signals, intermediate feature maps etc.) are embedded in high-dimensional spaces.

Humans cannot imagine or visualize more than 3D easily.

Can we find a mapping to a lower dimensional space which approximates the structure of the original space?

Close points should stay close.

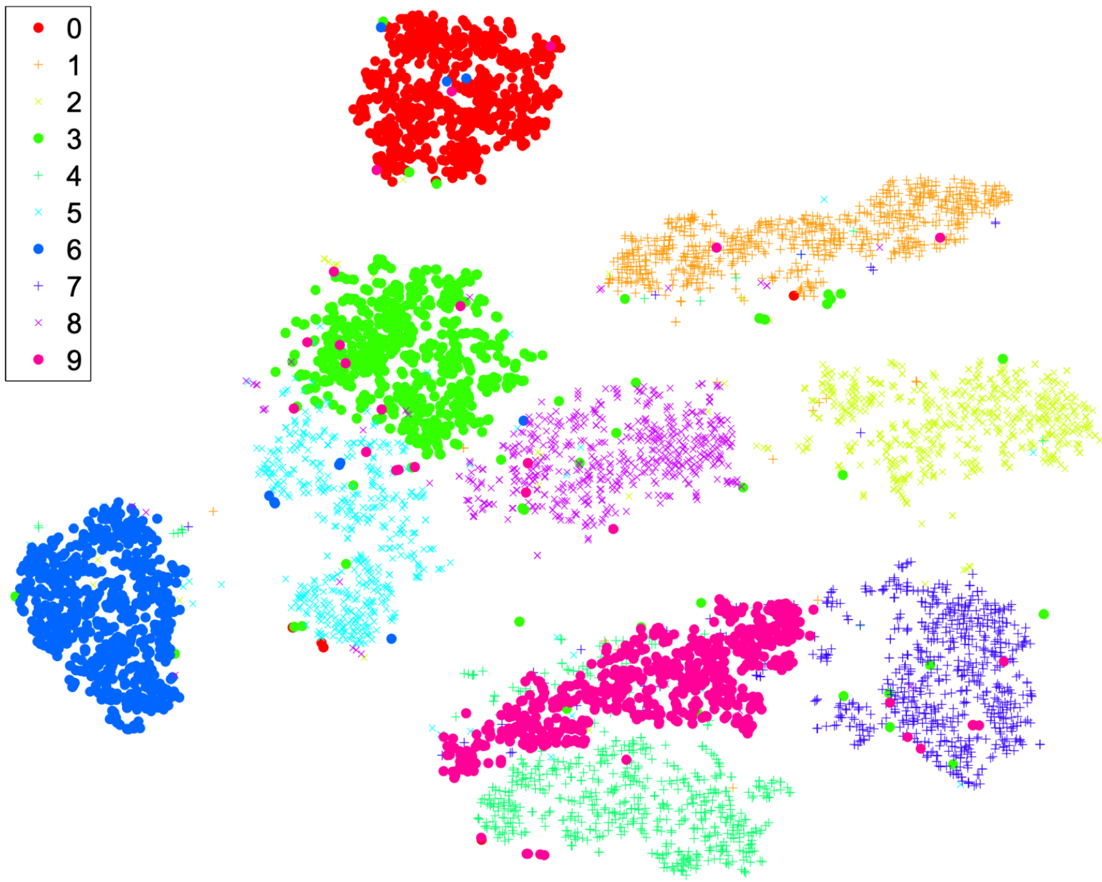
Far points should stay far.



Example: MNIST images = 28x28 pixels = 784dim input space

t-SNE

t-distributed stochastic neighbor embedding.



[Van der Maaten and Hinton, JMLR 2008]

t-SNE: model

Data points in input space: x_i.
Data points in low-dim space: y_i.

We assign a conditional probability to the (assymetric) pair of **high-dim** datapoint (x_i, x_j): Given x_i, will x_j be selected as neighbor if neighbors are selected according to distance (using a Gaussian kernel with variance σ_i):

$$p_{j|i} = \frac{\exp\left(-\|x_i - x_j\|^2 / 2\sigma_i^2\right)}{\sum_{k \neq i} \exp\left(-\|x_i - x_k\|^2 / 2\sigma_i^2\right)}$$

A symmetric version is given as:

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n}.$$

Van der Maaten, Hinton, JMLR 2008

t-SNE: model

For points (x_i, x_j) in the **lows-dim** map, we use a Cauchy distribution, which is heavier tailed:

$$q_{ij} = \frac{\left(1 + \|y_i - y_j\|^2\right)^{-1}}{\sum_{k \neq l} \left(1 + \|y_k - y_l\|^2\right)^{-1}}$$

Reason:

- high-dim spaces are less crowded (the volume of a sphere of radius r and dim d grows with r^d).
- Heaver tails better model datapoints which are not too close away from each other.

Van der Maaten, Hinton, JMLR 2008

t-SNE: training

We solve for the points $y_i, \forall i$ using SGD and optimizing Kullback-Leibler divergence (KL):

$$C = KL(P \| Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

Van der Maaten, Hinton, JMLR 2008

t-SNE: Hyper-parameter perplexity

We need to set the σ_i parameter for the distribution $P(i)$ each datapoint i.
A global hyper-parameter $Perp$ (=*Perplexity*) is set by the user ($\sim$ number of neighbors of each point).

Then se solve for

$$\mathrm{Perp}\left(P_i\right) = 2^{H\left(P_i\right)}$$

where

$$H\left(P_i\right) = -\sum_j p_{j|i} \log_2 p_{j|i}$$

Van der Maaten, Hinton, JMLR 2008

Example: the breakfast action data

4 labels per video clip:

- The recipe (e.g. cesar salade)
- The short term action (e.g. cut chicken)
- The person performing the action
- The camera viewpoint

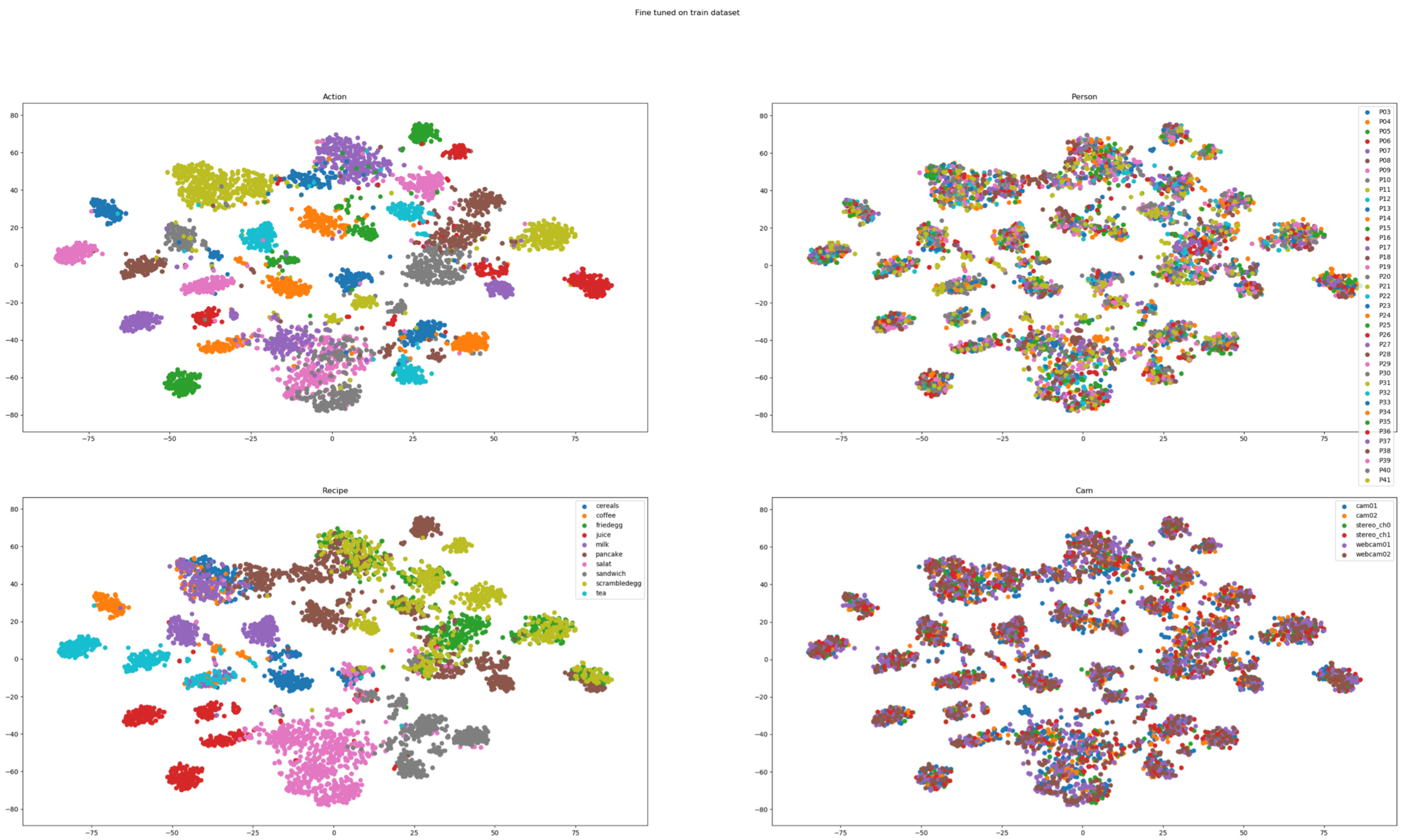
[H. Kuehne, A. B. Arslan and T. Serre. The Language of Actions: Recovering the Syntax and Semantics of Goal-Directed Human Activities. CVPR, 2014.]

t-SNE: activity recognition

Work of Tom Gillooly

Breakfast Dataset (train split), before fine-tuning

t-SNE: activity recognition



Work of Tom Gillooly

Breakfast Dataset (train split), after fine-tuning