

## Multimedia Indexing and Retrieval

Georges Quénot

Multimedia Information Modeling and Retrieval Group



Laboratory of Informatics of Grenoble



Georges Quénot

EARIA

17 October 2014

1

## (Indicative) outline

- Introduction
- Descriptors
- QBE, search, classification, fusion, post-processing ...
- Deep learning
- Conclusion

Georges Quénot

EARIA

17 October 2014

2

## Multimedia Retrieval

- User need → retrieved documents
- Images, audio, video
- Retrieval of full documents or passages (e.g. shots)
- Search paradigms:
  - Surrounding text → may be missing, inaccurate or incomplete
  - Query by example → need for what you are precisely looking for
  - Content based search (using keywords or concepts)
    - need for *content-based indexing* → “semantic gap problem”
  - Combinations including feedback
- Need for specific interfaces

Georges Quénot

EARIA

17 October 2014

3

## The “semantic gap”

“... the lack of coincidence between the information that one can extract from the visual data and the interpretation that the same data have for a user in a given situation” [Smeulders et al., 2002].

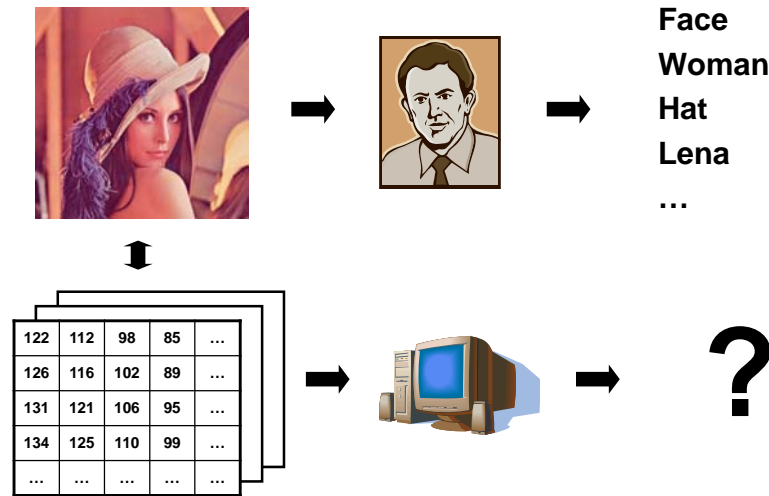
Georges Quénot

EARIA

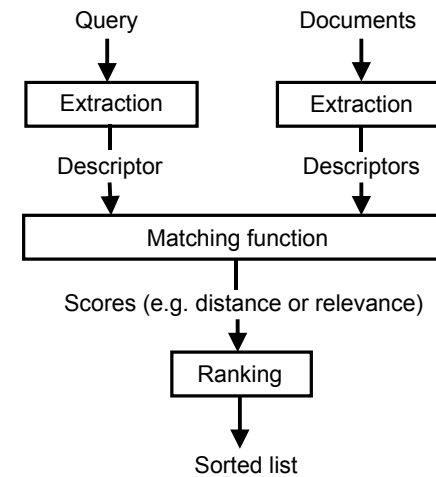
17 October 2014

4

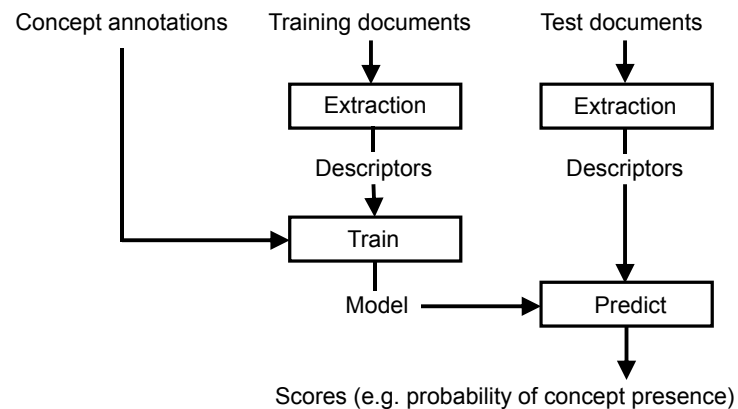
## The “semantic gap” problem



## Query BY Example (QBE)



## Content based indexing by supervised learning



## Example : the QBIC system

- Query By Image Content, IBM (stopped demo)  
<http://www.qbic.almaden.ibm.com/cgi-bin/photo-demo>



## Descriptors

- Engineered descriptors
  - Color
  - Texture
  - Shape
  - Points of interest
  - Motion
  - Semantic
  - Local versus global
  - ...
- Learned descriptors
  - Deep learning
  - Auto encoders
  - ...

## Histograms - general form

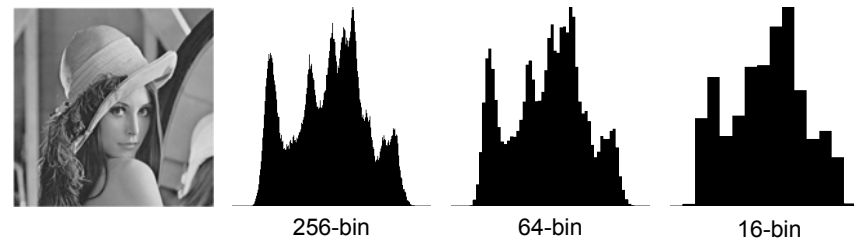
- A fixed set of *disjoint categories* (or *bins*), numbered from 1 to  $K$ .
- A set of *observations* that fall into these categories
- The histogram is the vector of  $K$  values  $h[k]$  with  $h[k]$  corresponding to the number of observations that fell into the category  $k$ .
- By default, the  $h[k]$  are integer values but they can also be turned into real numbers and normalized so that the  $h$  vector length is equal to 1 considering either the  $L_1$  or  $L_2$  norm
- Histograms can be computed for several sets of observations using the same set of categories producing one vector of values for each input set

## Histograms – text example

- A vector of term frequencies (tf) is an histogram
- The categories are the index terms
- The observations are the terms in the documents that are also in the index
- A tf.idf representation corresponds to a weighting of the bins, less relevant in multimedia since histograms bins are more symmetrical by construction (e.g. built by K-means partitioning)

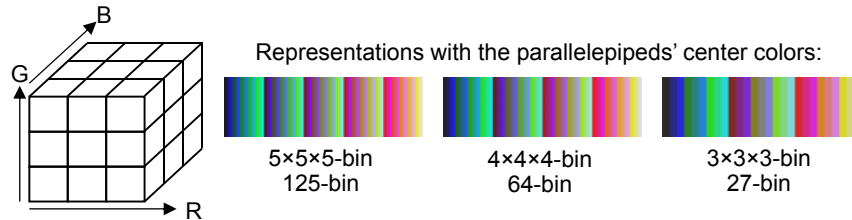
## Image intensity histogram

- The set of categories are the possible intensity values with 8-bit coding, ranging from 0 (black) to 255 (white) or ranges of these intensity values



## Image color histogram

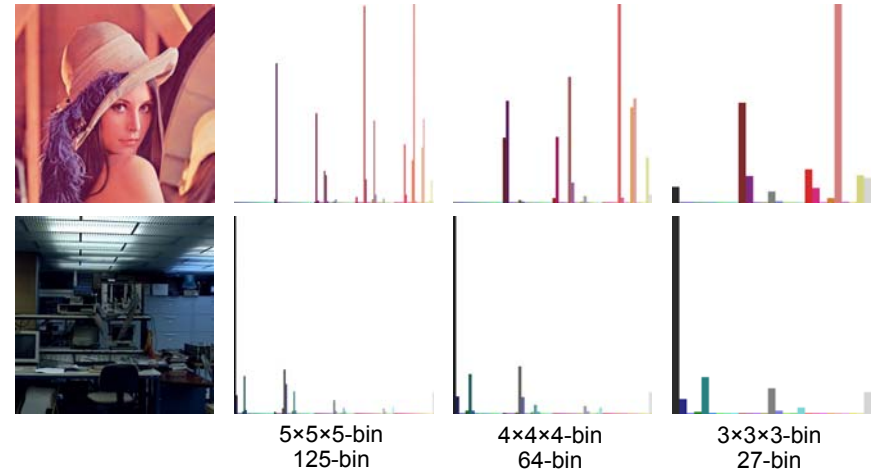
- The set of categories are ranges of possible color values
- A common choice is a per component decomposition resulting in a set of parallelepipeds



- Any color space can be chosen (YUV, HSV, LAB ...)
- Any number of bins can be chosen for each dimension
- The partition does not need to be in parallelepipeds

## Image color histogram

- The set of categories are ranges of possible color values



## Image histograms

- Rather invariant to image size if normalized to unit vector length with  $L_1$  or  $L_2$  norm
- Rather invariant to content displacements or symmetries
- NOT invariant to illuminations changes, gain and offset normalization may be needed
- Histograms are distributions, better compared using a  $\chi^2$  distance than Euclidean one:

$$d(x, y) = \sum_i \frac{(x_i - y_i)^2}{x_i + y_i}$$

- Earth Mover Distance (EMD) can be even better
- Alternatively, taking the square root of the histogram elements can make the Euclidean distance suitable

## Image histograms

- Can be computed on the whole image,
- Can be computed by blocks:
  - One (mono or multidimensional) histogram per image block,
  - The descriptor is the concatenation of the histograms of the different blocks.
  - Typically : 4 x 4 complementary blocks but non symmetrical and/or non complementary choices are also possible. For instance: 2 x 2 + full image center
- Size problem → only a few bins per dimension or a lot of bins in total

## Fuzzy histograms

- Objective: smooth the quantization effect associated to the large size of bins (typically  $4 \times 4 \times 4$  for RGB).
- Principle: split the accumulated value into two adjacent bins according to the distance to the bin centers.

## Correlograms

- Parallelepipeds/bins are taken in the Cartesian product of the color space by itself : six components  $H(r1,g1,b1,r2,g2,b2)$  (or only four components if the color space is projected on only two dimensions:  $H(u1,v1,u2,v2)$ ).
- Bi-color values are taken according to a distribution of the image point couples:
  - At a given distance one from the other,
  - And/or in one or more given direction.
- Allows for representing *relative spatial relationships between colors*,
- Large data volumes and computations

## Color moments

- Moments (color distribution global statistics)
  - Means
  - Covariances
  - Third order moments
  - Can be combined with image coordinates
  - Fast and easy to compute and compact representation but not very accurate

## Image normalization

- Objective : to become more robust again illumination changes before extracting the descriptors.
- Gain and offset normalization: enforce a mean and a variance value by applying the same affine transform to all the color components, non-linear variants.
- Histogram equalization: enforce an as flat as possible histogram for the luminance component by applying the same increasing and continuous function to all the color components.
- Color normalization: enforce a normalization which is similar to the one performed by the human visual: “global” and highly non linear.

## Texture descriptors

- Computed on the luminance component only
- Frequential composition or local variability
- Fourier transforms
- Gabor filters
- Neuronal filters
- Cooccurrence matrices
- Normalization possible.

## Gabor transforms

(Circular) Gabor filter of direction  $\theta$ , of wavelength  $\lambda$  and of extension  $\sigma$ :

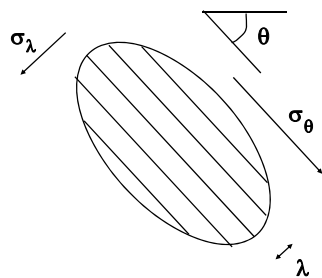
$$g(\sigma, \theta, \lambda, I, i, j) = \frac{1}{2\pi\sigma^2} \sum_{k,l} e^{-\left(\frac{k^2+l^2}{2\sigma^2}\right)} \cdot e^{2\pi i \left(\frac{k \cdot \cos\theta + l \cdot \sin\theta}{\lambda}\right)} \cdot I(i+k, j+l)$$

Energy of the image through this filter:

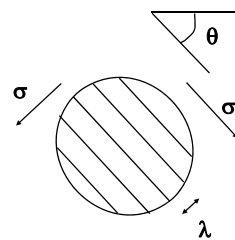
$$E_g(\sigma, \theta, \lambda, I)^2 = \frac{1}{N} \sum_{i,j} |g(\sigma, \theta, \lambda, I, i, j)|^2$$

## Gabor transforms

Elliptic:



Circular:



## Gabor transforms

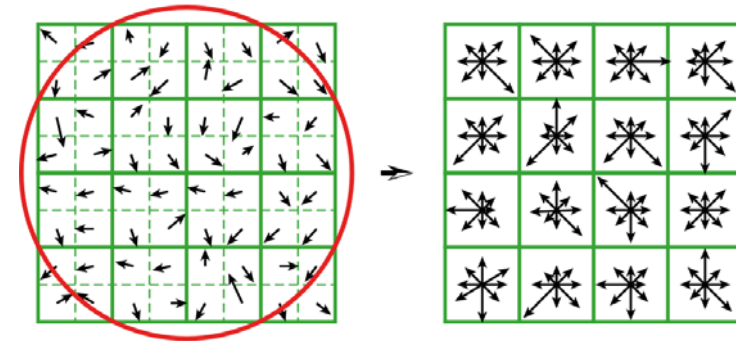
- **Circular:**
  - scale  $\lambda$ , angle  $\theta$ , variance  $\sigma$ ,
  - $\sigma$  multiple of  $\lambda$ , typically :  $\sigma = 1.25 \lambda$ ,
  - (“same number” of wavelength whatever the  $\lambda$  value)
- **Elliptic:**
  - scale  $\lambda$ , angle  $\theta$ , variances  $\sigma_\lambda$  and  $\sigma_\theta$ ,
  - $\sigma_\lambda$  and  $\sigma_\theta$  multiples of  $\lambda$ , typically :  $\sigma_\lambda = 0.8 \lambda$  et  $\sigma_\theta = 1.6 \lambda$ ,
- **2 independent variables:**
  - scale  $\lambda$  :  $N$  values (typically 4 to 8) on a logarithmic scale (typical ratio of  $\sqrt{2}$  to 2)
  - angle  $\theta$  :  $P$  values (typically 8),
  - $N \cdot P$  elements in the descriptor,

## Selection of points of interest

- “High curvature” points or “corners”,
- “Singular” points of the  $I[i][j]$  surface,
- Extracted using various filters:
  - Computation of the spatial derivatives at several scales,
  - Convolution with derivatives of Gaussians,
  - Harris-Laplace detector.
- Interest points are selected, filtered and described
- 2D (image): Scale Invariant Feature Transform (SIFT) [Lowe, 2004]
- 3D (video): Space-Time Interest Points (STIP) [Laptev, 2005]
- Variable number of points per image or per video shot → need for aggregation

## Descriptors of points of interest

- SIFT descriptor: Histogram of gradient direction: 8 bins times 4 x 4 blocks in a neighborhood of the point.



## Local versus global descriptors

- Global descriptors: single vector for a whole image
- Local descriptors: one vector for each pixel, image patch, image block shot 3D patch ... e.g. SIFT or STIP
- Need for a single vector of fixed length for any image and with comparable components across images
- *Aggregation* of local descriptors → global descriptor
- Homogeneous with the local descriptor:
  - max or average pooling
- Heterogeneous with the local descriptor:
  - Histogramming according to clusters in the local descriptor space [Sivic, 2003][Csurka, 2004]
  - Gaussian Mixture Models (GMM)
  - Fisher Vectors (FV) [Perronnin, 2006], Vectors of Locally Aggregated Descriptors (VLAD) [Jégou, 2010] or Tensors (VLAT) [Gosselin, 2011], Supervectors

## Semantic or intermediate descriptors

- Use of classifiers trained on other data and for other target concepts [Ayache, 2007]
- Vectors of scores of the other target concepts can be used as intermediate or high level descriptors (opposed to low-level ones that are “close to the signal”)
- Semantic descriptors can be either global or local (e.g. on pixels or patches)
- Semantic descriptors carry different information than low-level one and of higher semantic value
- The target concepts composing the semantic descriptors does not need to be related to the final target ones
- They do not need either to be recognized very accurately
- Semantic descriptors are often as good as or better than state of the art low-level ones and boost performance when combined with them



## Query by example

- Single query sample:
  - $\chi^2$ , EMD or histogram intersection for histograms
  - Euclidian Distance : searching for identities
  - Angle between vectors : searching for similarities robust to illumination changes (for some other descriptors, e.g. Gabor transforms)
- Multiple queries or relevance feedback:
  - Linear combination of distances with different weights for positively and negatively marked samples [Rocchio, 1971]
  - Supervised learning from the marked samples (active learning)
  - Rely also on the choice of a distance between global descriptions
- Direct matching and scoring between sets of local descriptors:
  - Costly but good for searching specific instances rather than general categories

## Content-based indexing

- Training from annotated collections:
  - LSCOM-TRECVID for videos
  - Pascal VOC or ImageNet for still images
  - Many others, e.g. Hollywood2 for actions in movies
- Use of supervised learning methods:
  - Support Vector Machines (SVM), linear or RBF
  - K nearest neighbors (KNN)
  - Neural Networks (NN), Multi-Layer Perceptrons (MLP)
  - Many others again
  - Adaptations for highly imbalanced data sets
- Fusion if several descriptors and/or several learning methods are simultaneously used.

## Fusion for concept classification

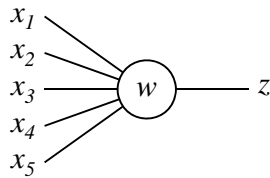
- Several possible descriptors
- Several possible classifiers
- Early versus late fusion [Snoek, 2005]
  - Early: concatenation of normalized descriptors
  - Late: combination of classification scores
- Kernel fusion [Ayache, 2007]
  - Fusion of kernels in RBF-based (e.g. SVM) learning methods

## Re-ranking for concept classification

- Re-ranking (or re-scoring): use of detections scores for other concepts or for other samples for improving the detection of a given concept for a given sample
- Temporal re-scoring [Safadi, 2010]
  - Re-score shots in a video with the hypothesis of a global or a local homogeneity of the contents
- Conceptual re-scoring [Hamadi, 2013]
  - Re-score an image or video sample for several concepts using implicit (co-occurrences) or explicit (ontologies) between them
- Combination of both



## Formal neural or unit



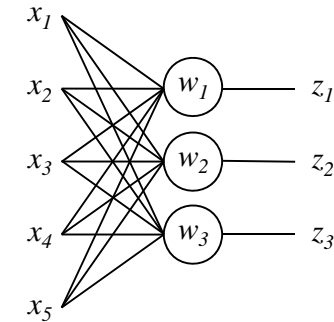
$$y = \sum_j w_j x_j$$

linear combination

$$z = \frac{1}{1 + e^y}$$

sigmoid function

## Neural layer (all to all)



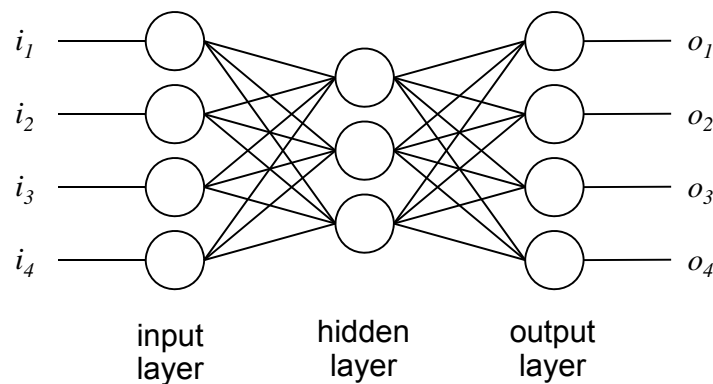
$$y_i = \sum_j w_{ij} x_j$$

matrix-vector multiplication

$$z_i = \frac{1}{1 + e^{y_i}}$$

per component operation

## Multilayer perceptron



## Feed forward

- Global network definition:  $O = F(W, I)$
- Layer values:  $(X_0, X_1 \dots X_N)$   
with  $X_0 = I$  and  $X_N = O$
- Vector of all unit parameters:  
 $W = (W_1, W_2 \dots W_N)$   
(weights by layer concatenated)
- Feed forward:  $X_{n+1} = F_{n+1}(W_{n+1}, X_n)$

## Error back-propagation

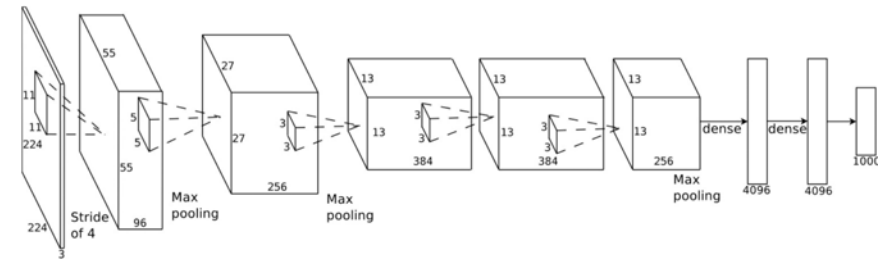
- Training set:  $(I_p, O_p)_{(1 \leq p \leq P)}$  input-output samples
- $X_{p,0} = I_p$  and  $X_{p,n+1} = F_{n+1}(W_{n+1}, X_{p,n})$
- Error on the training set:  

$$E(W) = \sum_p (F(W, I_p) - O_p)^2 = \sum_p (X_{p,N} - O_p)^2$$
- Minimization of  $E(W)$  by gradient descent:
  - Randomly initialize  $W(0)$
  - Iterate  $W(t+1) = W(t) - \eta \frac{\partial E}{\partial W}(t)$      $\eta = f(t)$  or  $\eta = \left(\frac{\partial^2 E}{\partial W^2}(t)\right)^{-1}$
  - Back-propagation:  $\frac{\partial E}{\partial W_n}$  is computed by backward recurrence from  $\frac{\partial F_n}{\partial W_n}$  and  $\frac{\partial F_n}{\partial X_{n-1}}$  applying iteratively  $(g \circ f)' = (g' \circ f) \cdot f'$

## ImageNet Challenge 2012

[Krizhevsky et al., 2012]

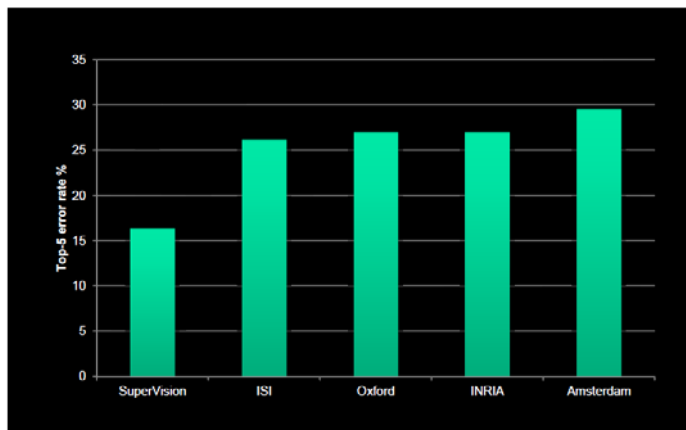
- 7 hidden layers, 650K units, 60M parameters (  $W$  )
- GPU implementation (50× speed-up over CPU)
- Trained on two GPUs for a week



A. Krizhevsky, I. Sutskever, and G. Hinton, ImageNet Classification with Deep Convolutional Neural Networks, NIPS 2012

## ImageNet Classification 2012 Results

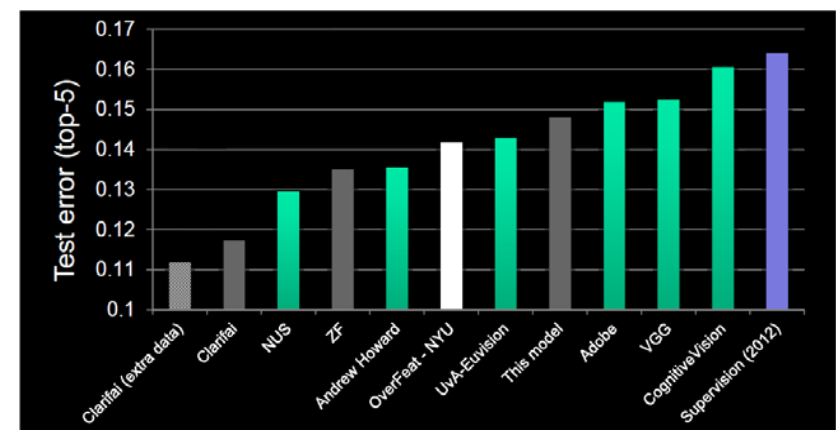
Krizhevsky et al. -- **16.4% error** (top-5)  
 Next best (non-convnet) – **26.2% error**



## ImageNet Classification 2013 Results

<http://www.image-net.org/challenges/LSVRC/2013/results.php>

Demo: <http://www.clarifai.com/>



## Engineered versus learned descriptors

- Classical “classification pipeline”
  - Extraction(s) – [aggregation] – optimization(s) – classifier(s) – one or more levels of fusion – re-scoring (non exhaustive example)
  - Most of the stages are explicitly engineered: the form of descriptors or processing steps has been thought and designed by a skilled engineer or researcher
  - *Lots* of experience and acquired expertise by thousands of smart people over tens of years
  - Learning concerns only the classifier(s) stages and a few hyper-parameters controlling the other ones
  - Almost everything has been tried
  - The more you incorporate, the more you get (at a cost)

## Engineered versus learned descriptors

- Deep learning pipeline: MLP with about 8 layers
  - Advances in computing power (Tflops): large networks possible
  - Algorithmic advance: combination of convolutional layers for the lower stages with all-to-all layers; the topology of the image is preserved in the lower layers with weights shared between the units within a layer
  - Algorithmic advances: NN researchers finally find out how to have back-propagation working for MLP with more than three layers
  - Image pixels are entered *directly* into the first layer
  - The first (resp. intermediate, last) layers practically compute low-level (resp. intermediate level, semantic) descriptors
  - Everything is made using a unique and homogeneous architecture
  - A single network can be used for detecting many target concepts
  - All the level are jointly optimized at once
  - Requires *huge* amounts of training data

## Engineered versus learned descriptors

- Deep learning (learned descriptors) outperform almost everything else in more and more domains
- The only observed weakness is that it does so only when a lot of training data is available
- Some trials show that combining deep learning and classical approaches outperforms both [Snoek 2013]
- Many hybrid approaches are being studied and appear promising