TRAITEMENT D'IMAGES

Partie Introductive

Frédéric Cointault
Institut Agro Dijon
Responsable Equipe ATIP
UMR Agroécologie
26 Bd Dr Petitjean
21000 Dijon
+33 3 80 77 27 54
frederic.cointault@agrosupdijon.fr

L'INSTITUT NATIONAL D'ENSEIGNEMENT SUPÉRIEUR POUR L'AGRICULTURE, L'ALIMENTATION ET L'ENVIRONNEMENT

0 - Préambule

I - Introduction

II - Définitions

III - Pré-traitement des images

IV - Segmentation image et contours

V - Hough et morphologie mathématique

VI – Analyse et Reconnaissance de formes

VII – Détection de mouvement

VIII – Introduction au Deep Learning

VIII – Introduction au Deep Learning

Systèmes intelligents

Artificial Intelligence (AI ou IA)

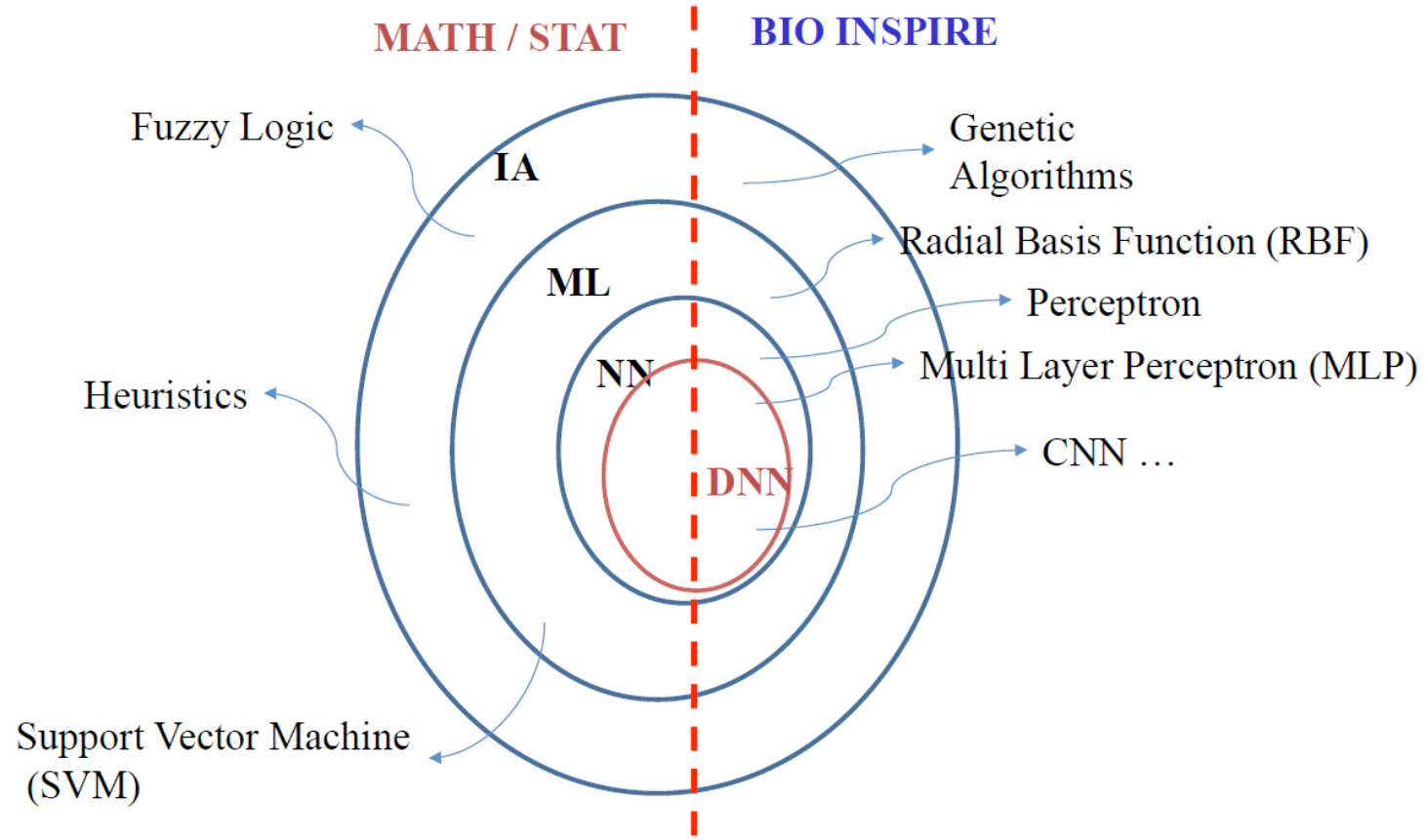
Machine Learning (ML)

Artificial Neural Network (ANN ou NN)

Deep Neural Network (DNN)

VIII – Introduction au Deep Learning

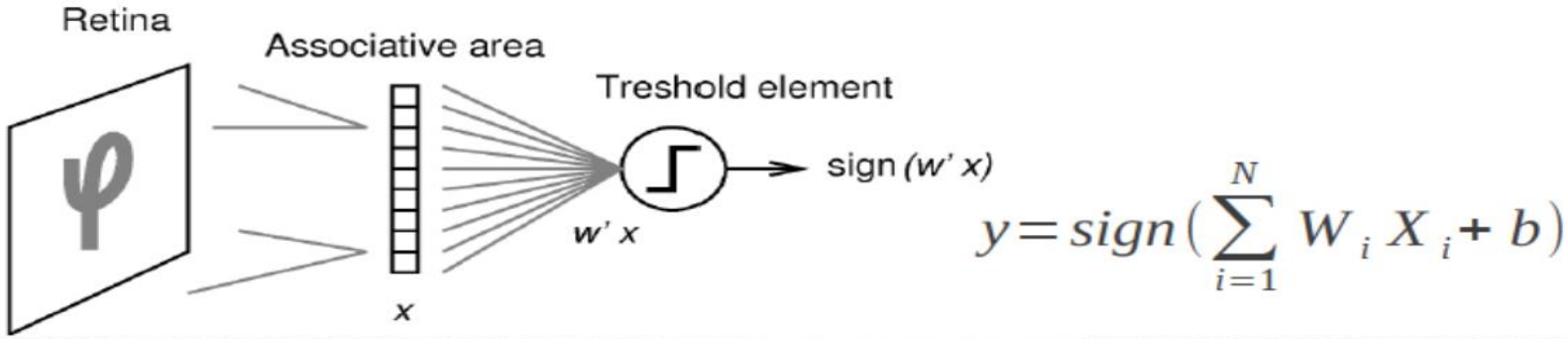
IA > ML > ANN > DL



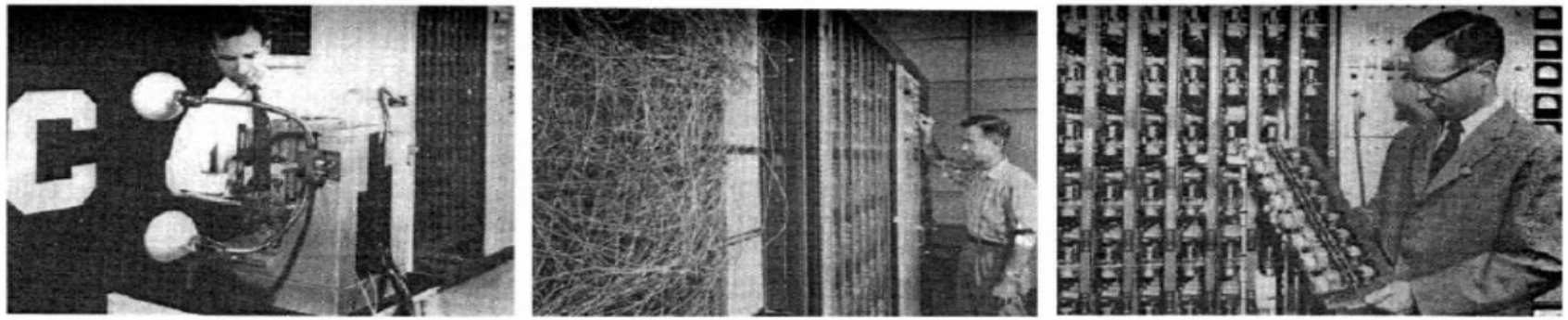
VIII – Introduction au Deep Learning

Perceptron (Rosenblatt 1957)

- A simple simulated neuron with adaptive "synaptic weights"
  - Computes a weighted sum of inputs
  - Output is +1 if the weighted sum is above a thresold, -1 otherwise.



$$y = sign\left(\sum_{i=1}^{N} W_i X_i + b\right)$$

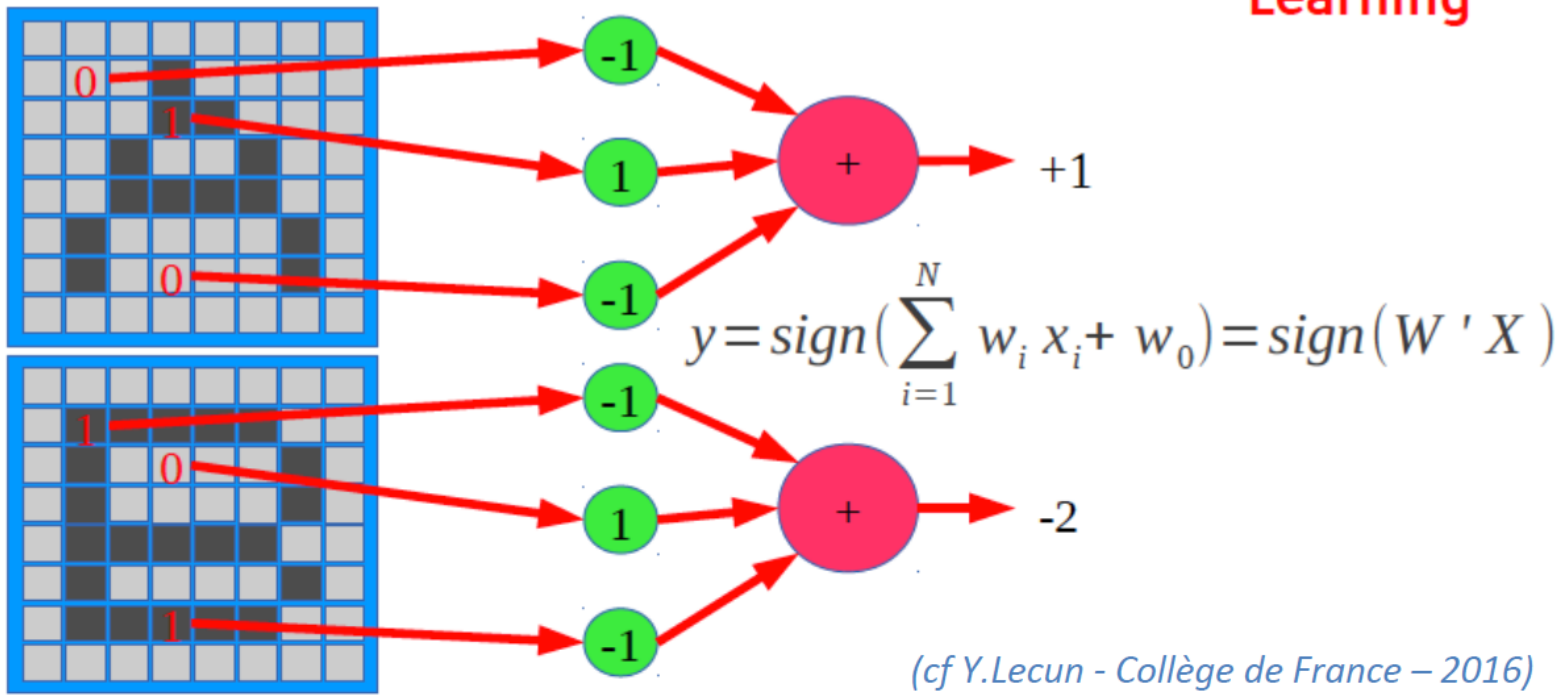


(cf Y.Lecun - Collège de France – 2016)

VIII – Introduction au Deep Learning

- Example: classifying letters "A" from "B"
- Learning: find the weight values that produce +1 for A and -1 for B
- Training set: $(X^1,Y^1),(X^2,Y^2),.....,(X^P,Y^P)$
- Example: (**A**,+1),(**B**,-1),(**A**,+1),(**B**,-1),(**A**,+1),(**B**,-1),.....

Supervised Learning



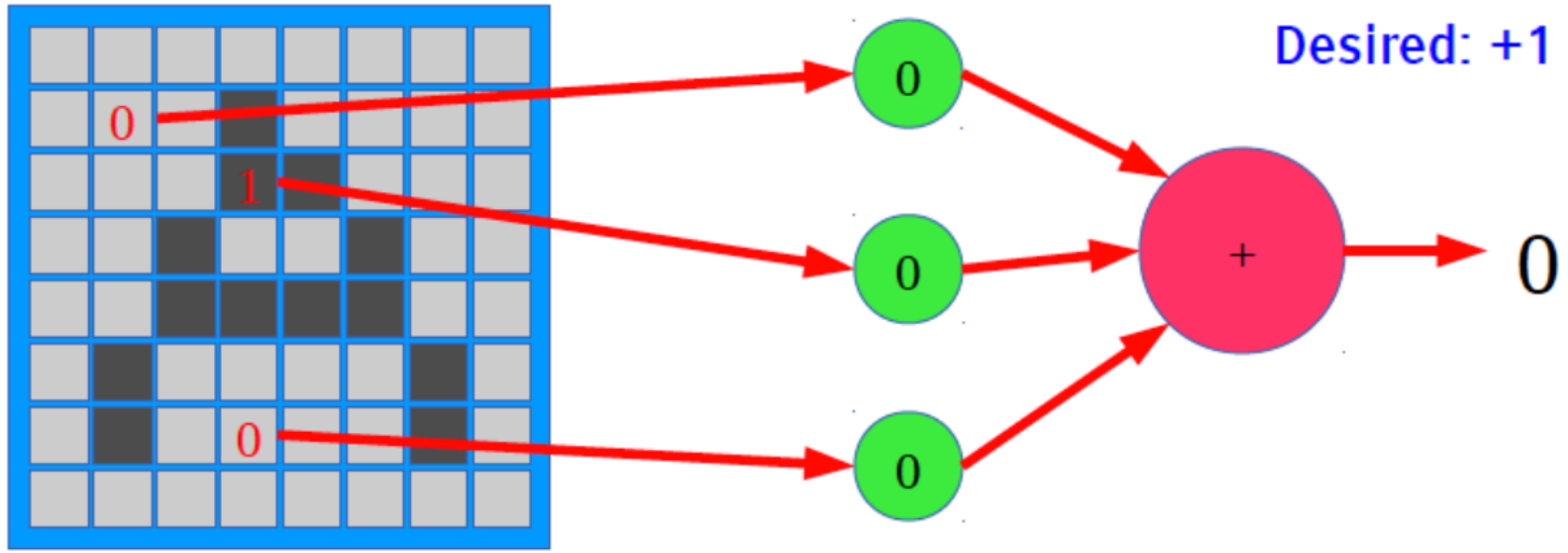
$$y = sign\left(\sum_{i=1}^{N} w_i x_i + w_0\right) = sign(W' X)$$

(cf Y.Lecun - Collège de France – 2016)

VIII – Introduction au Deep Learning

Apprentissage des poids synaptiques

- Learning: adjusting the weights so as to obtain the desired result
  - Initially, the weights are 0.

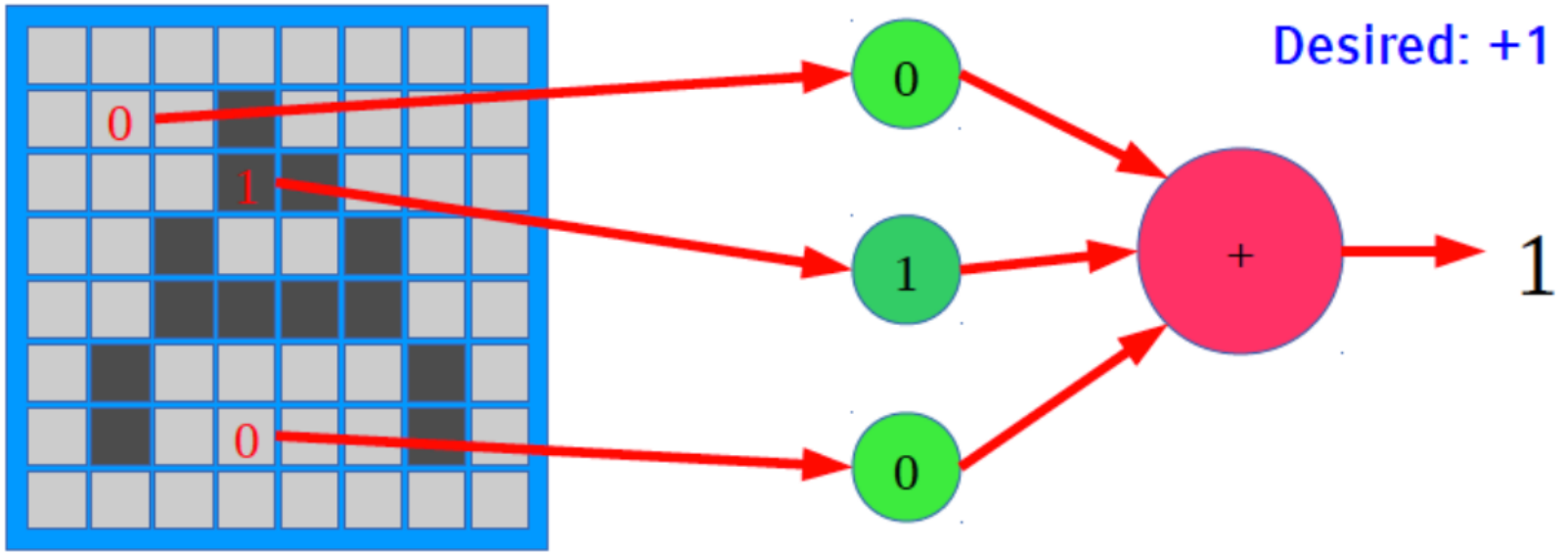


(cf Y.Lecun - Collège de France – 2016)

VIII – Introduction au Deep Learning

- Adjusting the weights when the the output is incorrect
  - If the desired output is +1, add pixel values to the weights (Hebbian learning)

Apprentissage des poids synaptiques

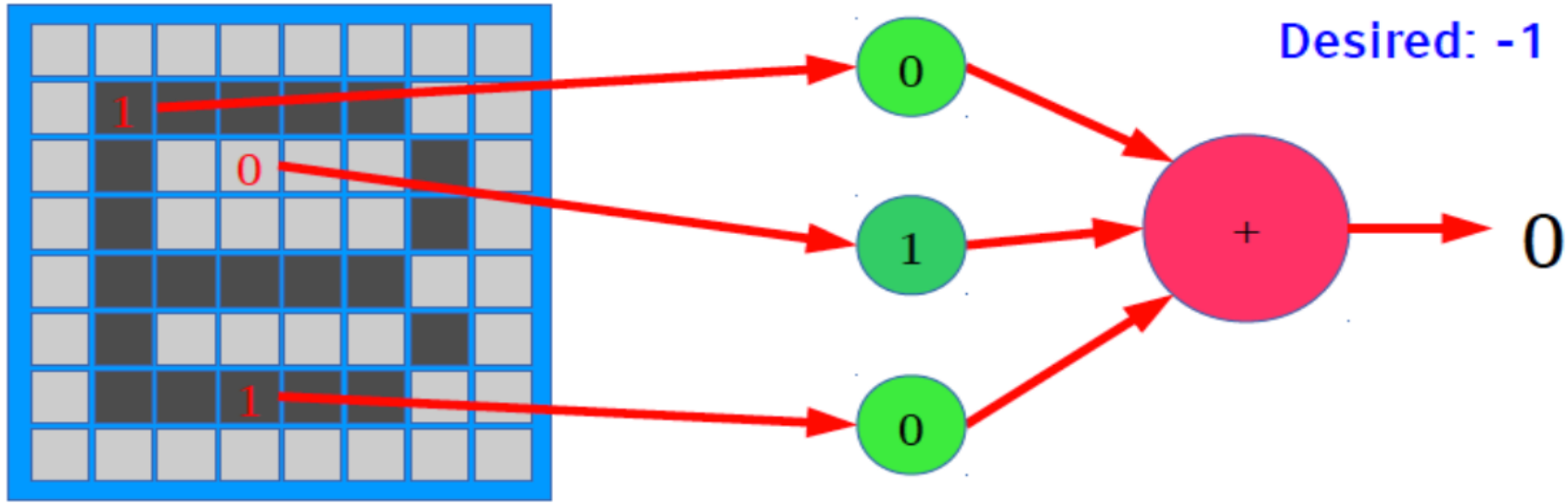


(cf Y.Lecun - Collège de France – 2016)

VIII – Introduction au Deep Learning

Apprentissage des poids synaptiques

- Adjusting the weights when the the output is incorrect
  - If the desired output is -1, subtract pixel values from the weights.

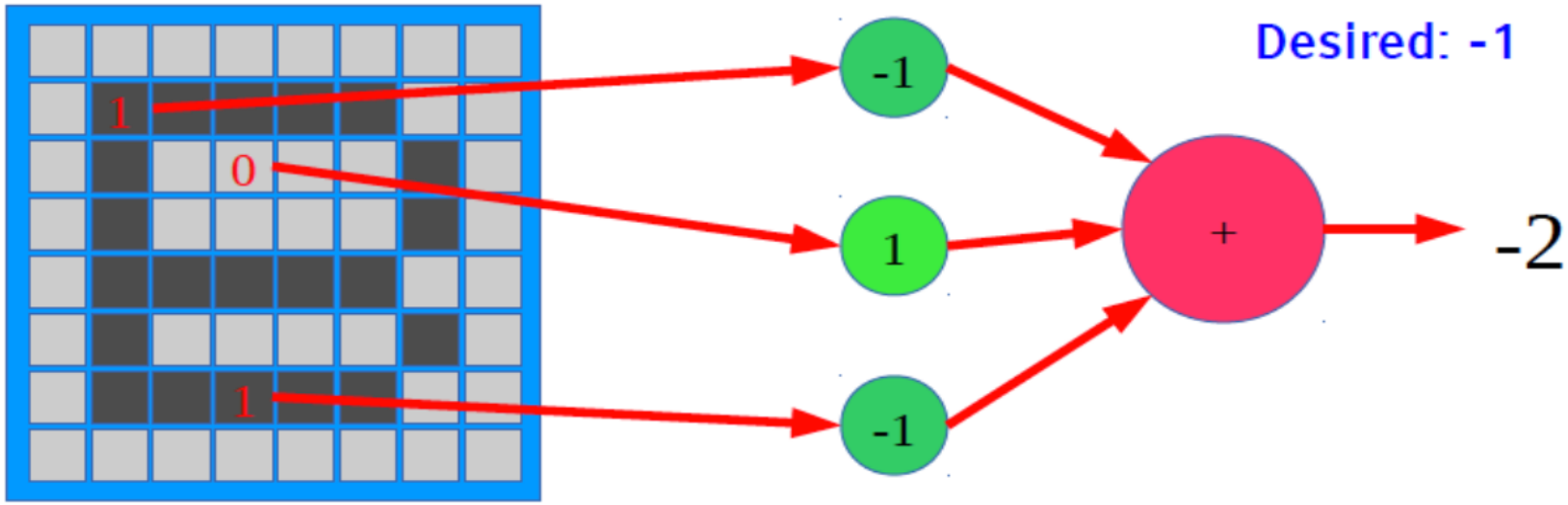


(cf Y.Lecun - Collège de France – 2016)

VIII – Introduction au Deep Learning

Apprentissage des poids synaptiques

- Adjusting the weights when the the output is incorrect
  - If the desired output is -1, subtract pixel values from the weights.

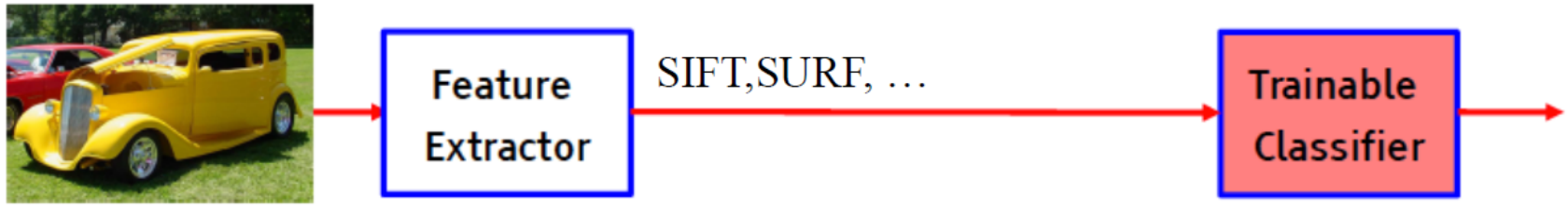


(cf Y.Lecun - Collège de France – 2016)

VIII – Introduction au Deep Learning

Modèles de vision artificielle

Traditional Pattern Recognition: Fixed/Handcrafted Feature Extractor



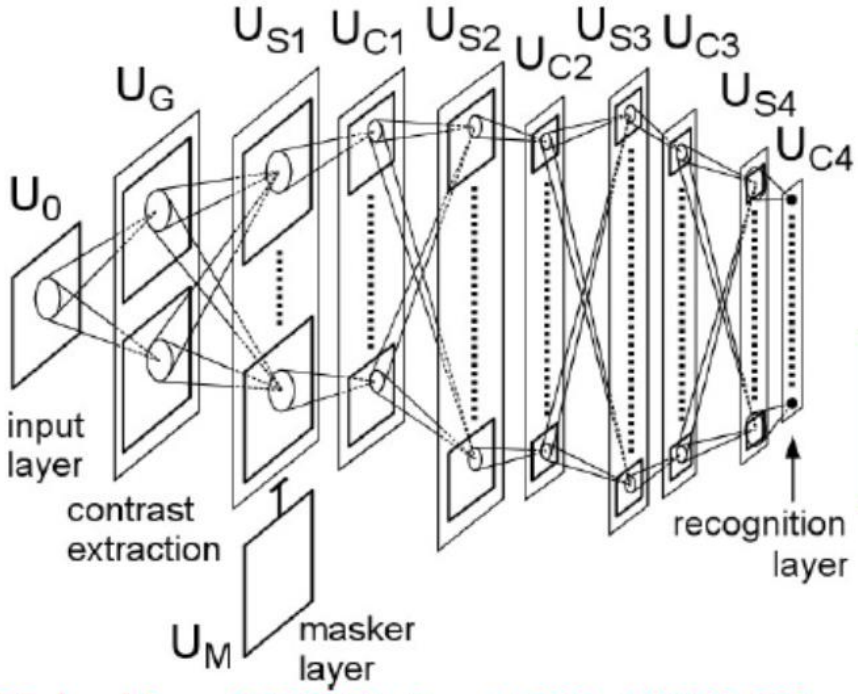
Deep Learning: Representations are hierarchical and trained

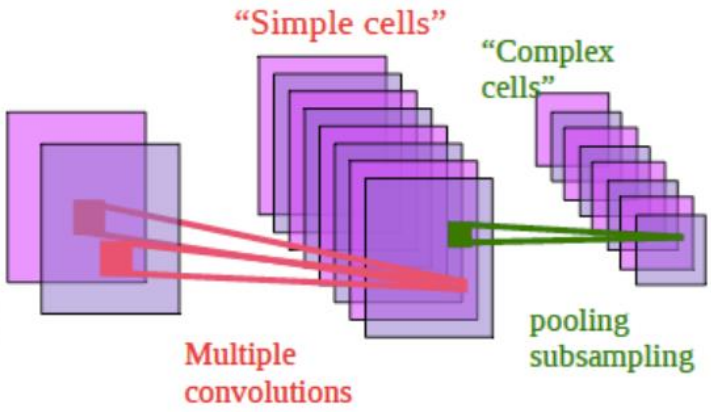
(cf Y.Lecun - Collège de France – 2016)

VIII – Introduction au Deep Learning

L'apprentissage profond : >50 ans

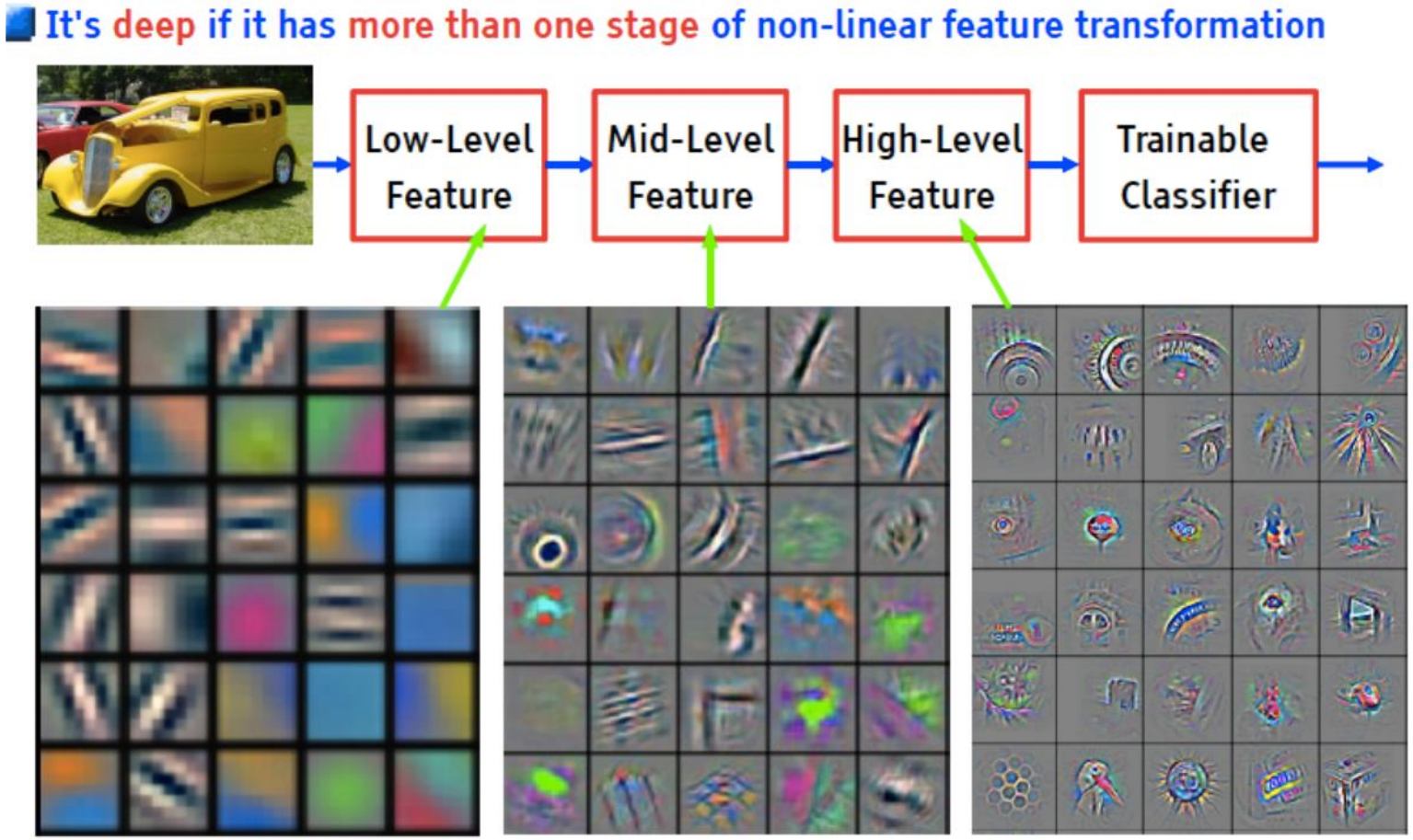
- [Hubel & Wiesel 1962]:
  - simple cells detect local features
  - complex cells "pool" the outputs of simple cells within a retinotopic neighborhood.





[Fukushima 1982] [LeCun 1989, 1998], [Riesenhuber 1999].....

VIII – Introduction au Deep Learning

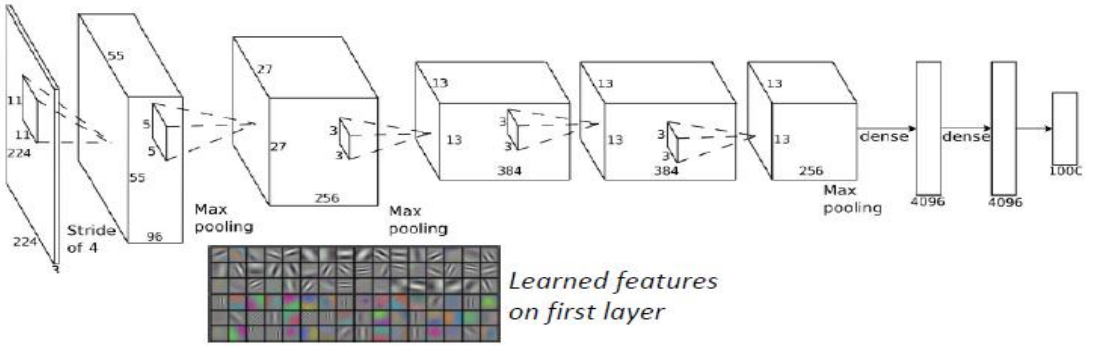


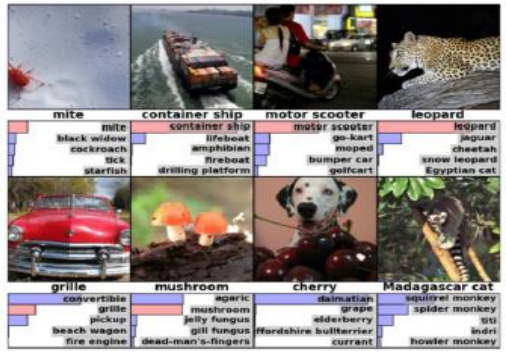
Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]

VIII – Introduction au Deep Learning

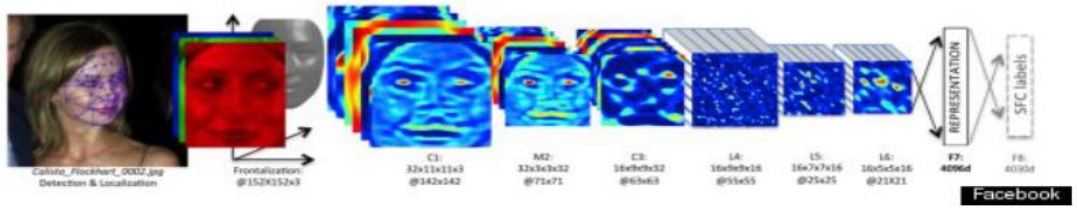
Modèles récents de DNN

- ImageNet classification (Hinton's team, hired by Google)
  - 1.2 million high res images, 1,000 different classes
  - Top-5 17% error rate (huge improvement)



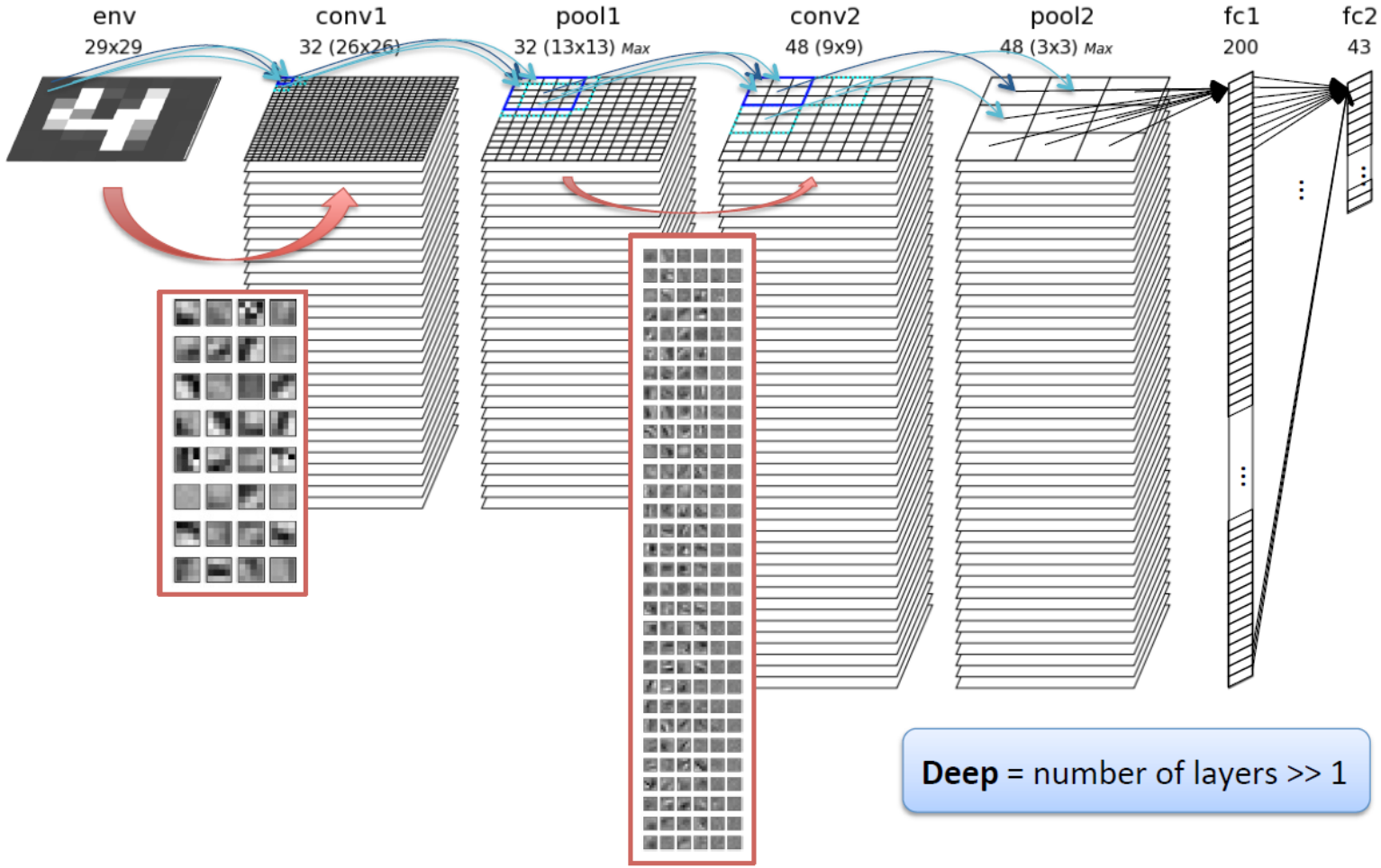


- Facebook's 'DeepFace' Program (labs head: Y. LeCun)
  - 4 million images, 4,000 identities
  - 97.25% accuracy, vs. 97.53% human performance



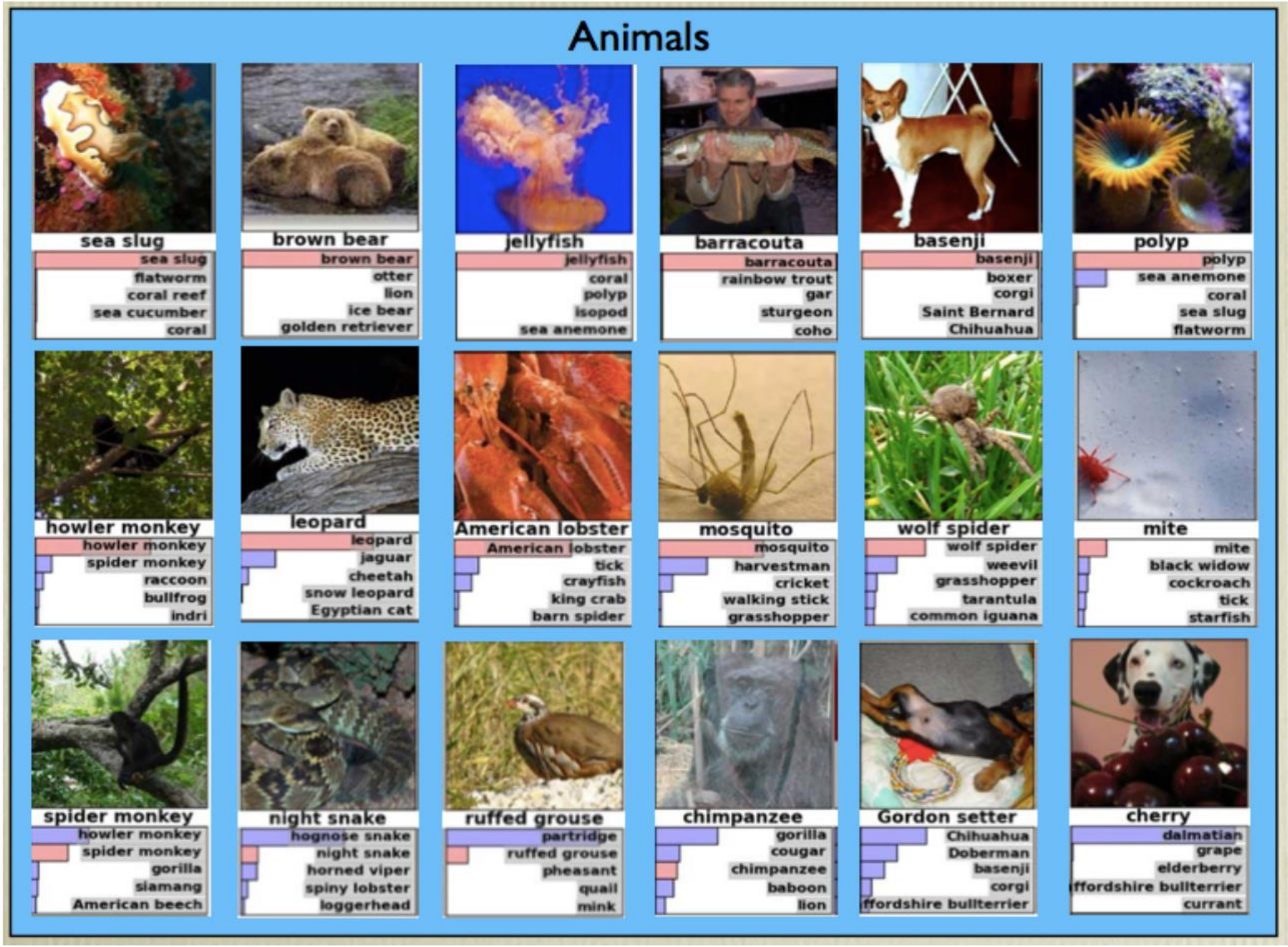
VIII – Introduction au Deep Learning

Organisation des CNNs



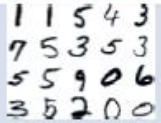
VIII – Int

Supervision par CNNs



VIII – Introduction au Deep Learning

Etat de l'art dans la reconnaissance

Database	# Images	# Classes	Best score
MNIST *Handwritten digits* 	60,000 + 10,000	10	99.79% [3]
GTSRB *Traffic sign* 	~ 50,000	43	99.46% [4]
CIFAR-10 *airplane, automobile, bird, cat, deer, dog, frog, horse, ship, truck* 	50,000 + 10,000	10	91.2% [5]
Caltech-101 	~ 50,000	101	86.5% [6]
ImageNet 	~ 1,000,000	1,000	Top-5 83% [1]
DeepFace 	~ 4,000,000	4,000	97.25% [2]

INCREASING COMPLEXITY

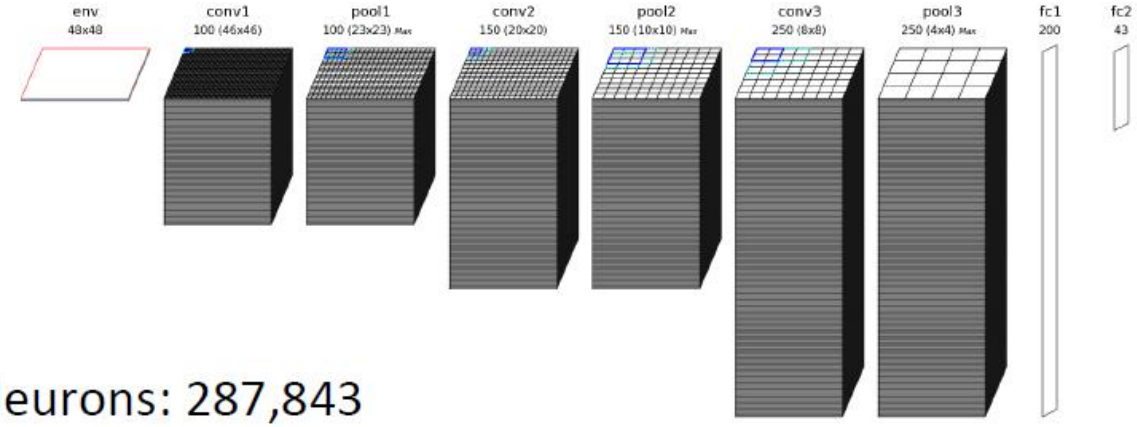
- **State-of-the-art are Deep Neural Networks *every time***

VIII – Introduction au Deep Learning

Etat de l'art ex de CNNs

The German Traffic Sign Recognition Benchmark (GTSRB)

43 traffic sign types
> 50,000 images



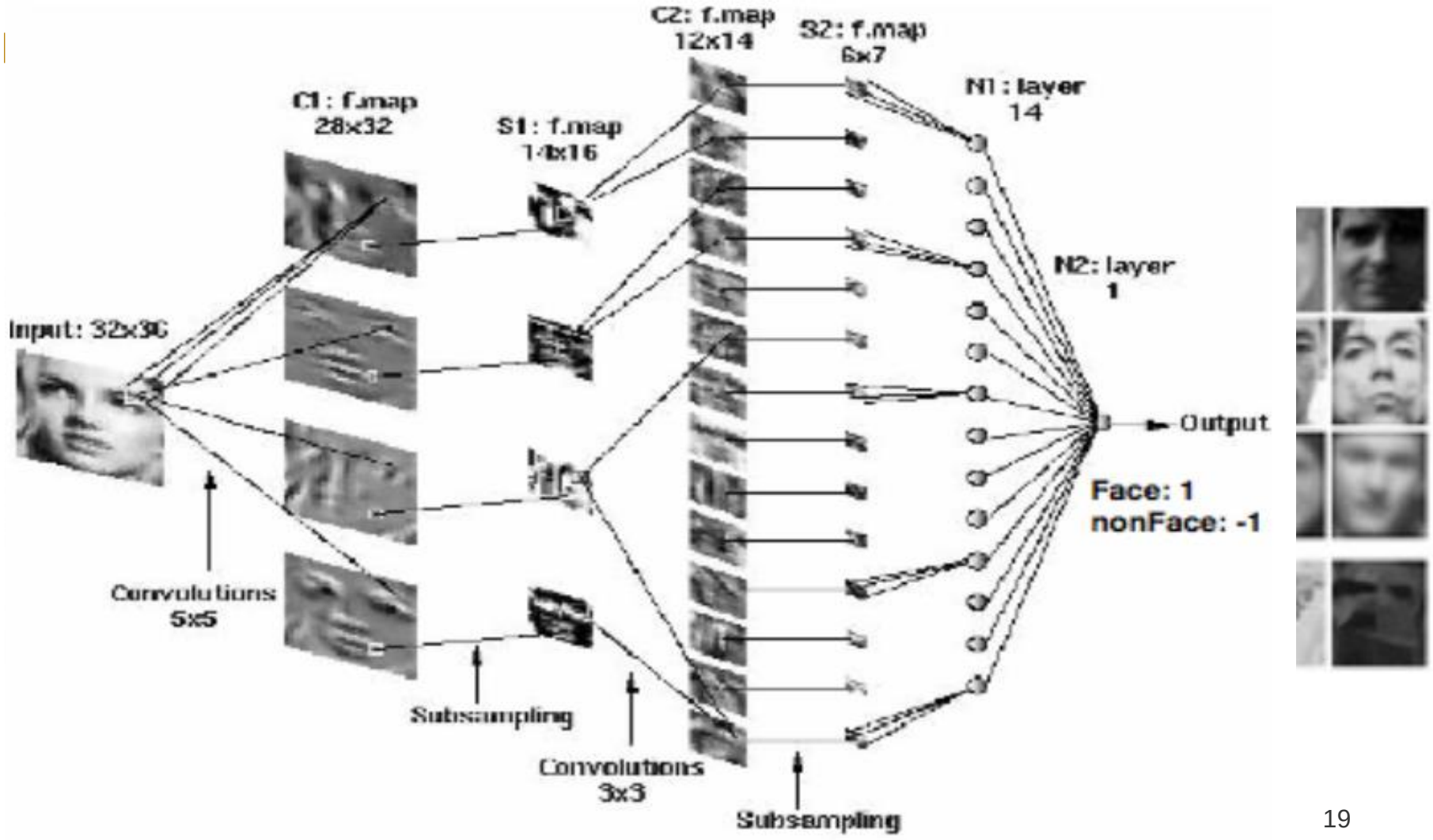
- Neurons: 287,843
- Synapses: 1,388,800
  - Total memory: 1.5MB (with 8 bits synapses)
- Connections: 124,121,800

Near human recognition (> 98%) [3]

[3] D. Ciresan, U. Meier, J. Masci, J. Schmidhuber, Multi-column deep neural network for traffic sign classification, Neural Networks (32), pp. 333-338, 2012

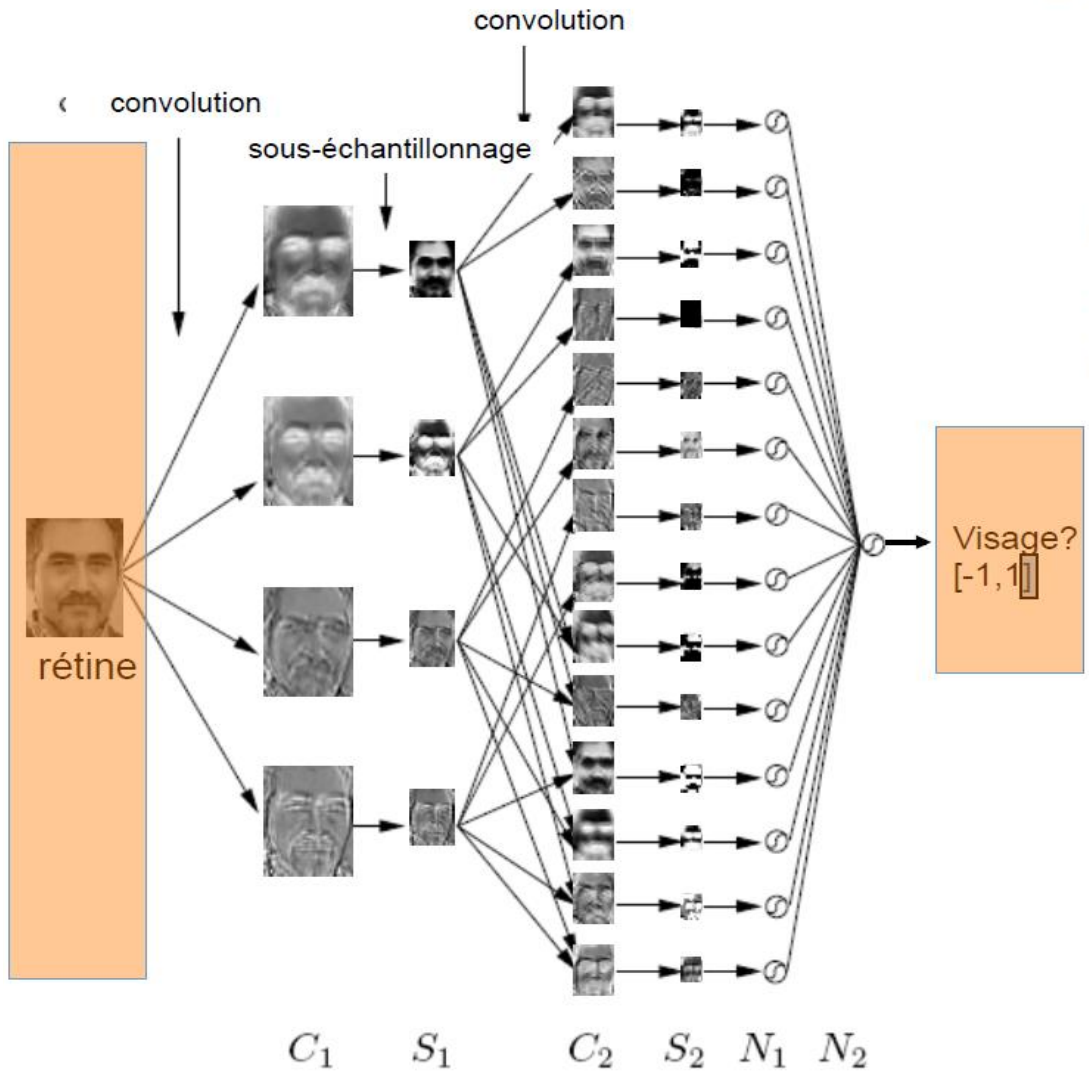
VIII – Introduction au Deep Learning

Adaptation des CNNs pour les visages



VIII – Introduction au Deep Learning

Architecture de CFF => Apprentissage



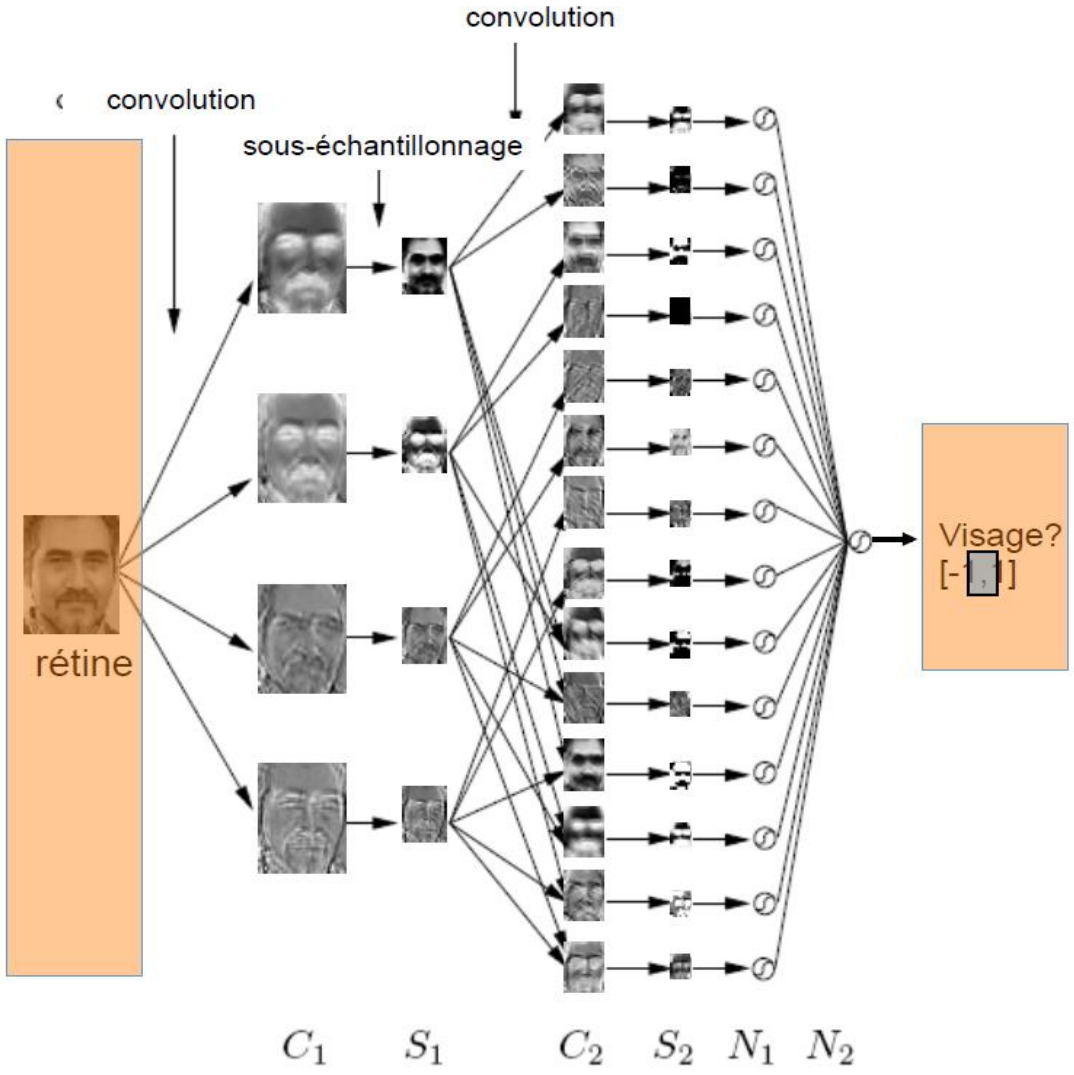
3702 images de visages (32x36)
(web, scannées,etc.)

Transformées artificiellement
(rotation, translation, contraste, lissage, etc.)

25.212 exemples de visages

VIII – Introduction au Deep Learning

Architecture de CFF => Apprentissage



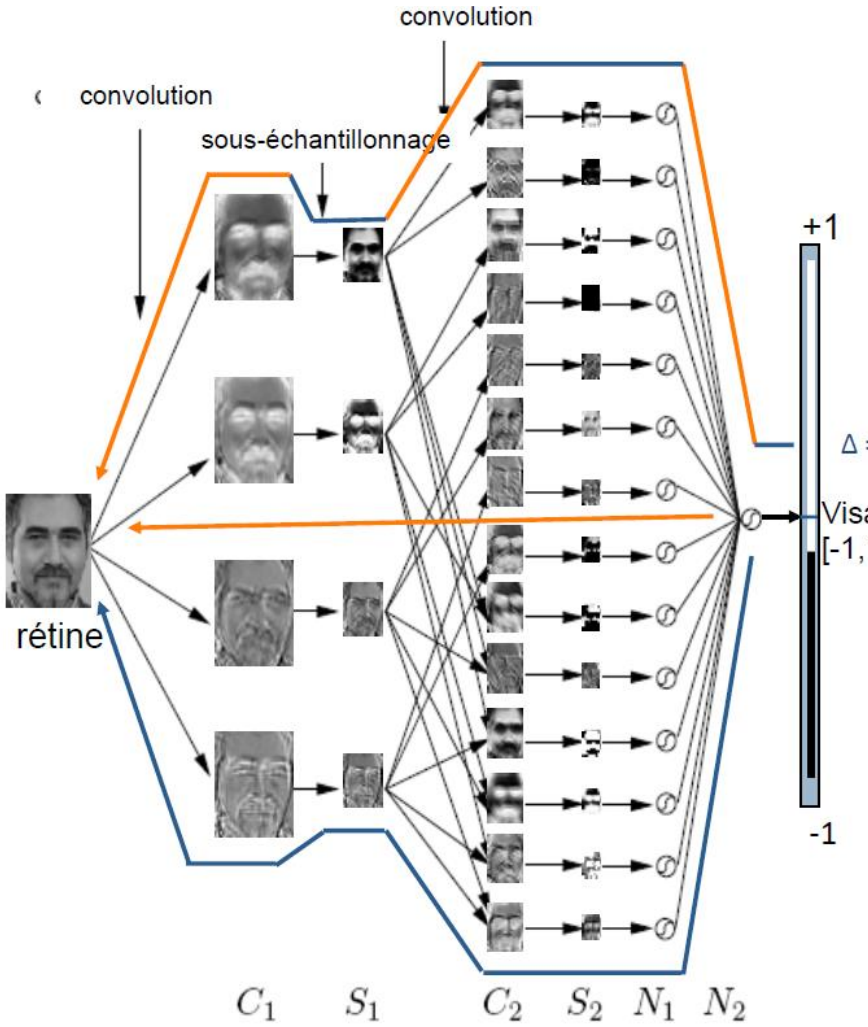
6422 exemples de «non-visages»

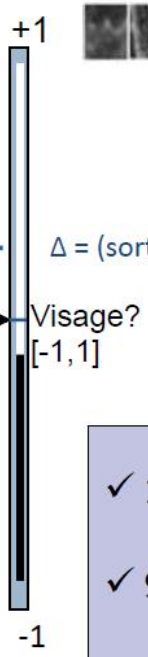
Boostrap: 19.065 fausses alarmes

25.487 exemples de «non-visages»

VIII – Introduction au Deep Learning

Architecture de CFF => Apprentissage





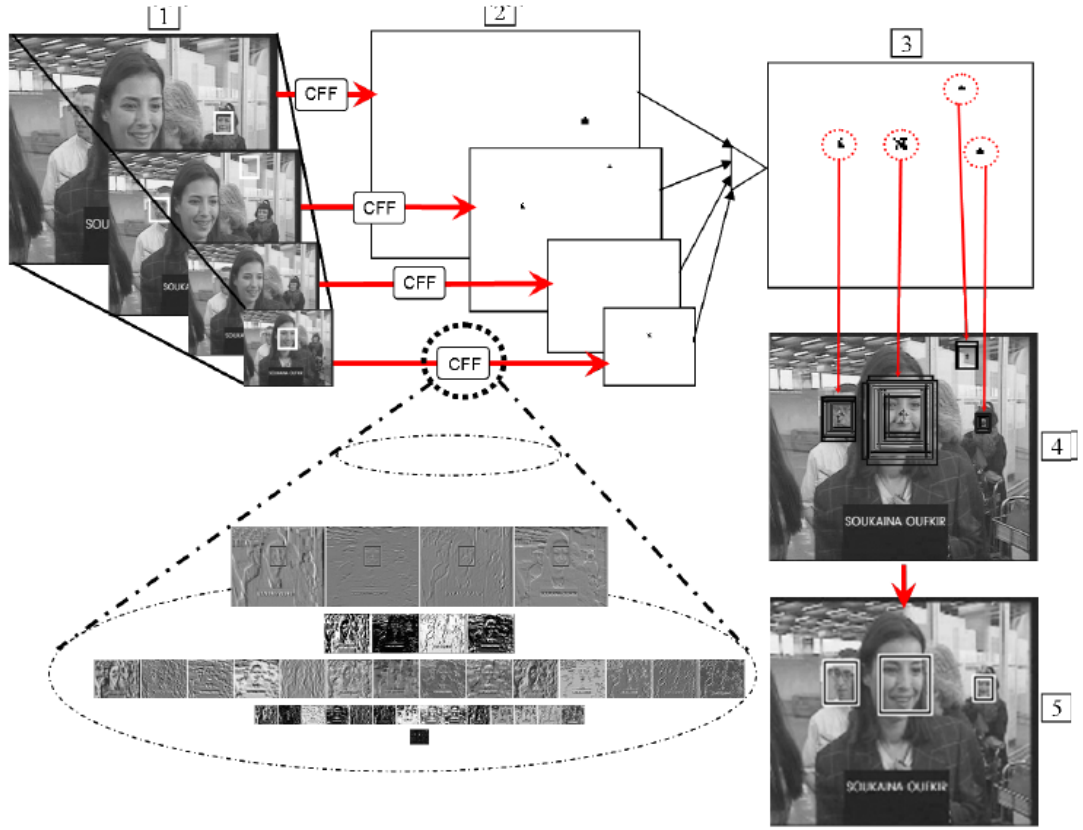
Sortie > 0 → visage!

- ✓ 131.475 poids synaptiques
- ✓ 951 poids W à apprendre
- ✓ par rétropropagation du gradient

VIII – Introduction au Deep Learning

Système de détection CFF

1. Construire une pyramide d'images (facteur 1.2)
2. Appliquer CFF sur chaque image de la pyramide
   → Positions des visages candidates
3. Projeter et fusionner les positions candidates dans l'image originale
4. Appliquer CFF autour des positions candidates (position et échelle)
5. Filtrer les résultats en fonction du «volume» des réponses positives (*ThrVol*)

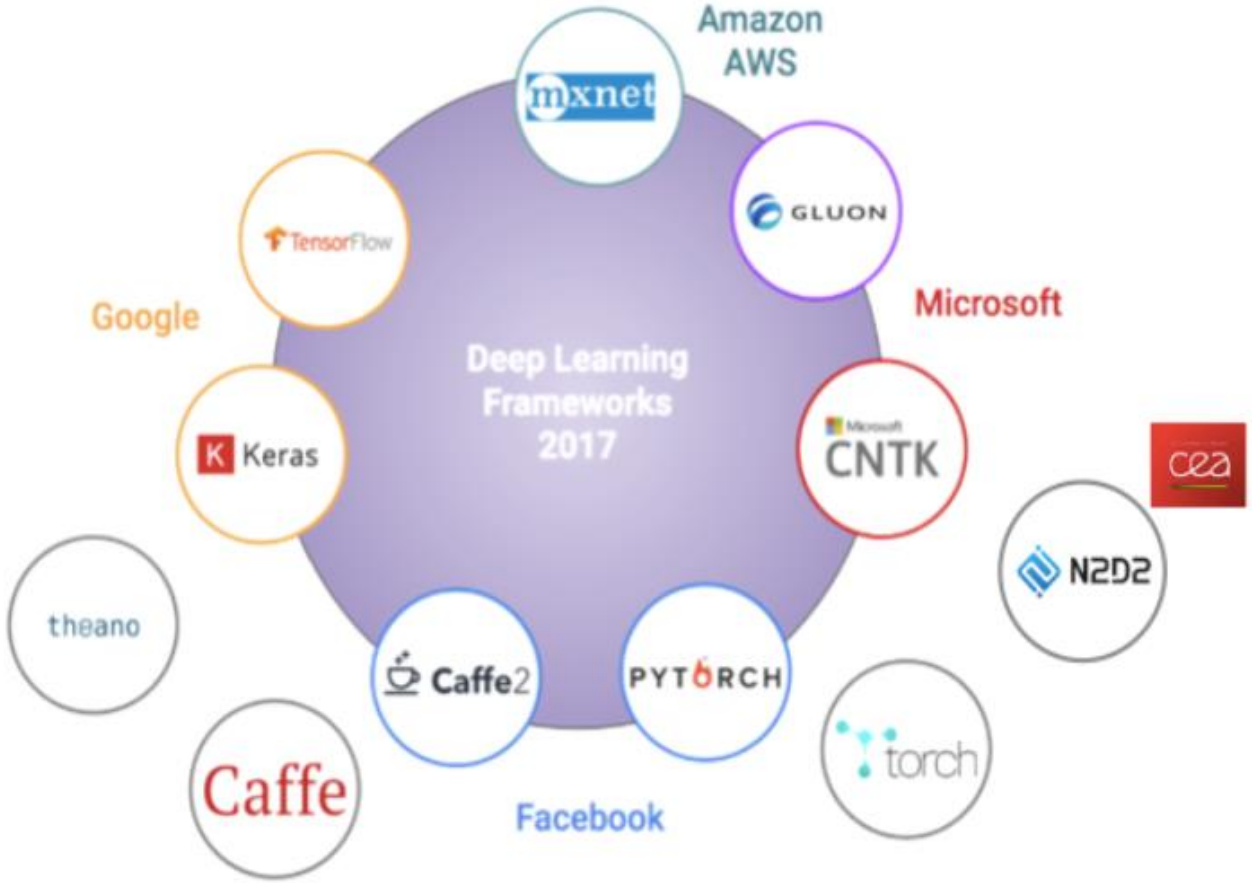


VIII – Introduction au Deep Learning

Frameworks pour développer des Deep Networks:

Numerous frameworks

- Most popular ones are open source
- Some are powered by “GAFAM”



VIII – Introduction au Deep Learning

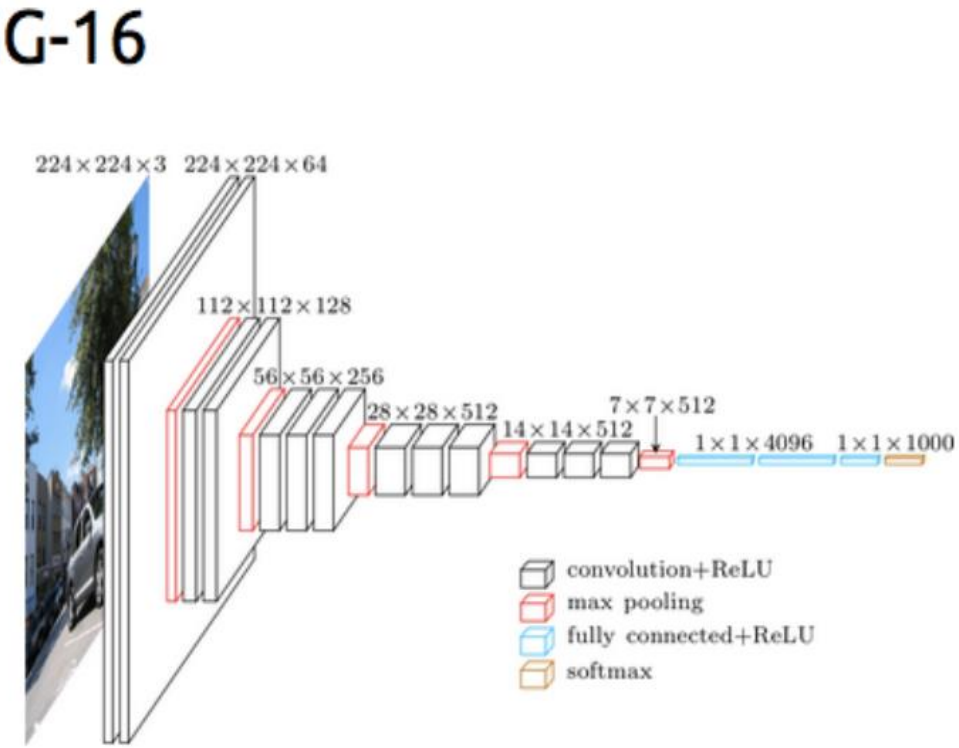
Utilisation de Keras:

Why use Keras…

- **Keras prioritizes developer experience**
- **Keras has broad adoption in the industry and the research community**
- **Keras makes it easy to turn models into products**
  - On iOS, via Apple's CoreML (Keras support officially provided by Apple).
  - On Android, via the TensorFlow Android runtime.
  - In the browser, via GPU-accelerated JavaScript runtimes such as Keras.js and WebDNN.
  - On Google Cloud, via TensorFlow-Serving.
  - In a Python webapp backend (such as a Flask app).
  - On the JVM, via DL4J model import provided by SkyMind.
  - On Raspberry Pi (direct Keras installation).

VIII – Introduction au Deep Learning

Ex de Deep Network



Simonyan, Karen, and Zisserman. "Very deep convolutional networks for large-scale image recognition." (2014)