

Le Réseau neuronal convolutif

Chaolei CAI

Paris VIII

April 28, 2020

1 Le néocognitron

- Inspiration biologique
- Lien avec Hubel et Wiesel
- Formule de la sortie d'une cellule S
- Exemple de detection numérique

2 Réseau neuronal convolutif

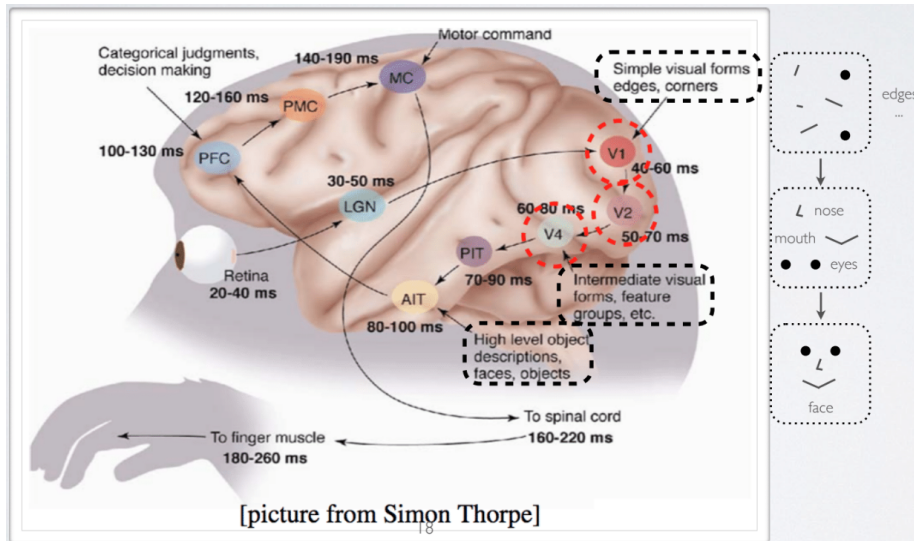
- Architecture du réseau
- Rectified Linear Unites ReLUs
- Couches de convolution
- Couches de pooling
- Présentation des résultats

3 Elephant in the room

- Présentation de l'expérience

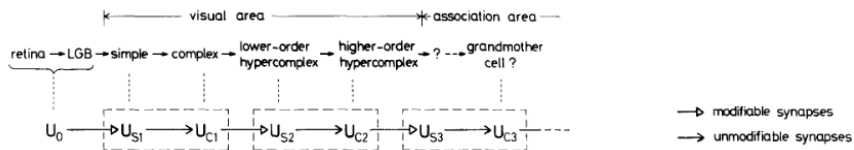
4 bibliographie

Le modèle vivant



Correspondance avec le modèle de Hubel et Wiesel et le Néocognitron

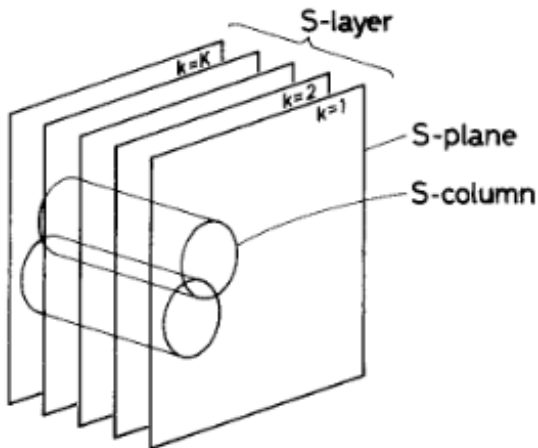
195



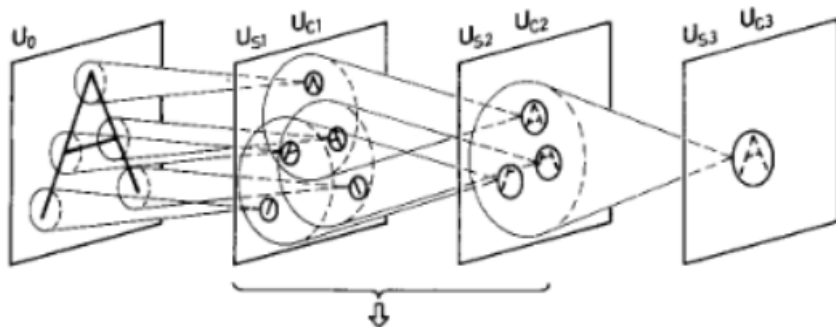
Formule de la sortie d'une cellule S

$$u_{Sl}(k_l, \mathbf{n}) = r_l \cdot \varphi \left[\frac{1 + \sum_{k_{l-1}=1}^{K_{l-1}} \sum_{\mathbf{v} \in S_l} a_l(k_{l-1}, \mathbf{v}, k_l) \cdot u_{Cl-1}(k_{l-1}, \mathbf{n} + \mathbf{v})}{1 + \frac{2r_l}{1+r_l} \cdot b_l(k_l) \cdot v_{Cl-1}(\mathbf{n})} - 1 \right],$$

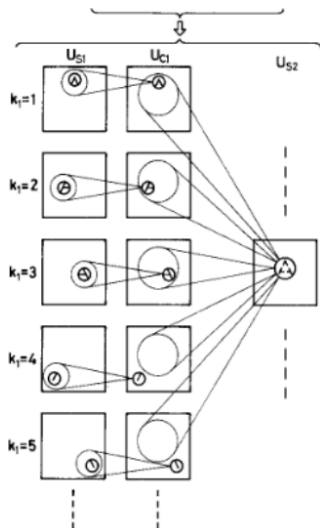
Auto-organisation du réseau



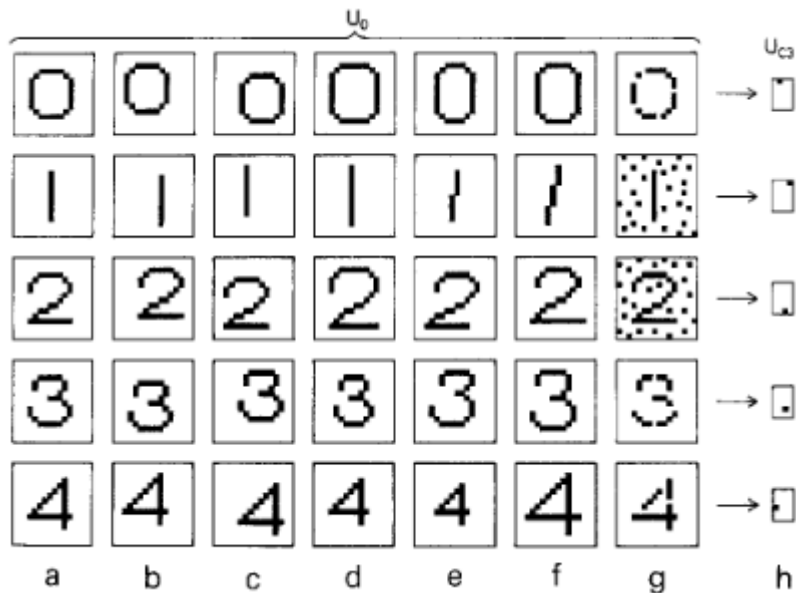
Auto-organisation du réseau



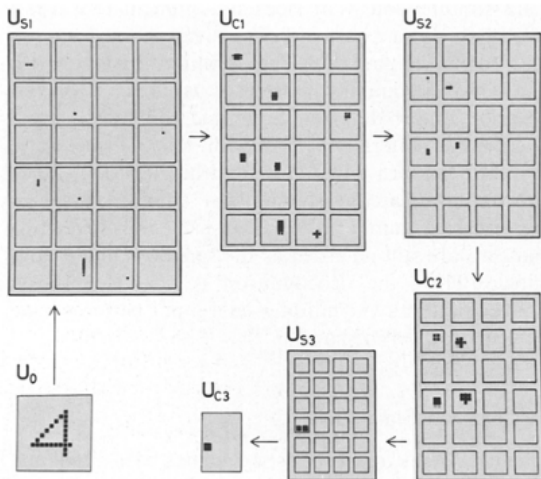
Auto-organisation du réseau



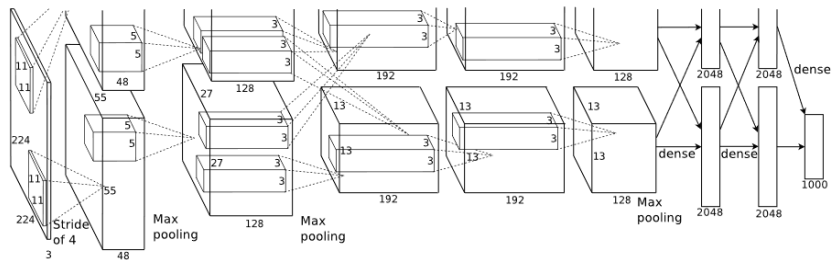
Exemple de detection numérique



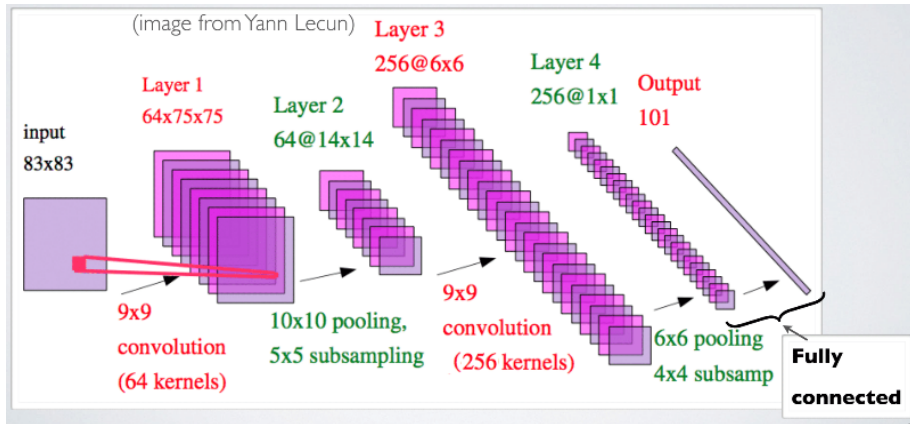
Exemple de detection numérique



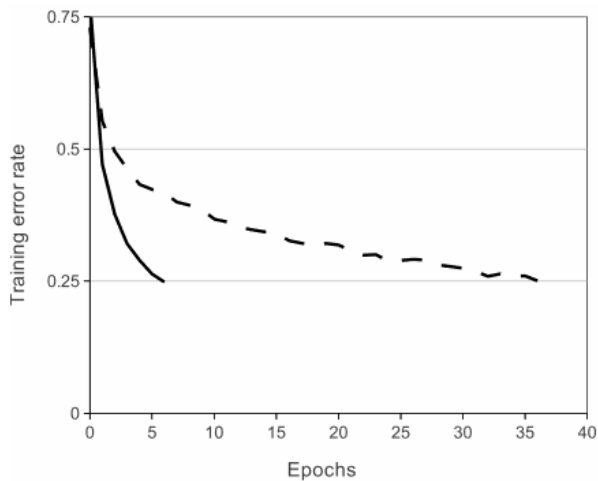
Architecture du réseau



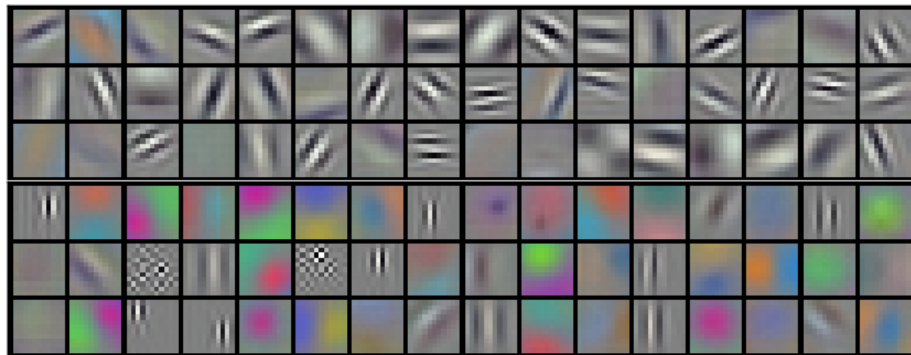
Architecture du réseau | modèle initial



Rectified Linear Unites ReLUs



Couches de convolution



Produit de convolution

Input Volume (+pad 1) (7x7x3)

$x[:, :, 0]$

0	0	0	0	0	0	0
0	0	0	1	2	0	0
0	1	1	2	0	2	0

0	1	0	0	1	2	0
0	2	2	1	0	1	0
0	1	2	1	0	1	0
0	0	0	0	0	0	0

$x[:, :, 1]$

0	0	0	0	0	0	0
0	2	0	1	2	1	0
0	1	1	0	2	2	0

0	2	0	1	1	0	0
0	2	0	1	1	0	0
0	2	0	0	1	2	0
0	0	0	0	0	0	0

$x[:, :, 2]$

0	0	0	0	0	0	0
0	1	2	0	1	1	0
0	1	2	0	1	2	0

Filter W0 (3x3x3)

$w0[:, :, 0]$

1	1	-1
-1	1	1
0	0	1

$w0[:, :, 1]$

0	-1	-1
1	-1	-1
-1	0	0

$w0[:, :, 2]$

-1	-1	1
-1	0	1
-1	0	-1

Bias b0 (1x1x1)

$b0[:, :, 0]$

1

Filter W1 (3x3x3)

$w1[:, :, 0]$

-1	1	0
-1	-1	1
1	-1	0

$w1[:, :, 1]$

-1	1	1
0	0	0
0	1	0

$w1[:, :, 2]$

0	-1	-1
0	0	-1
0	0	0

Bias b1 (1x1x1)

$b1[:, :, 0]$

0

Output Volume (3x3x2)

$o[:, :, 0]$

0	-4	-4
1	-1	-2
2	-3	-1

$o[:, :, 1]$

-2	-1	-2
-2	3	-4
1	-4	-3

toggle movement

Produit de convolution

Input Volume (+pad 1) (7x7x3)

$x[:, :, 0]$

0	0	0	0	0	0	0
0	0	0	1	2	0	0
0	1	1	2	0	2	0
0	1	0	0	1	2	0
0	2	2	1	0	1	0
0	1	2	1	0	1	0
0	0	0	0	0	0	0

$x[:, :, 1]$

0	0	0	0	0	0	0
0	2	0	1	2	1	0
0	1	1	0	2	2	0
0	2	0	1	1	0	0
0	2	0	1	1	0	0
0	2	0	0	1	2	0
0	0	0	0	0	0	0

$x[:, :, 2]$

0	0	0	0	0	0	0
0	1	2	0	1	1	0

Filter W0 (3x3x3)

$w0[:, :, 0]$

1	1	-1
-1	1	1
0	0	1

$w0[:, :, 1]$

0	-1	-1
1	-1	-1
-1	0	0

$w0[:, :, 2]$

-1	-1	1
-1	0	1
-1	0	-1

Bias b0 (1x1x1)

$b0[:, :, 0]$

1

Filter W1 (3x3x3)

$w1[:, :, 0]$

-1	1	0
-1	-1	1
1	-1	0

$w1[:, :, 1]$

-1	1	1
0	0	0
0	1	0

$w1[:, :, 2]$

0	-1	-1
0	0	-1
0	0	0

Bias b1 (1x1x1)

$b1[:, :, 0]$

0

Output Volume (3x3x2)

$o[:, :, 0]$

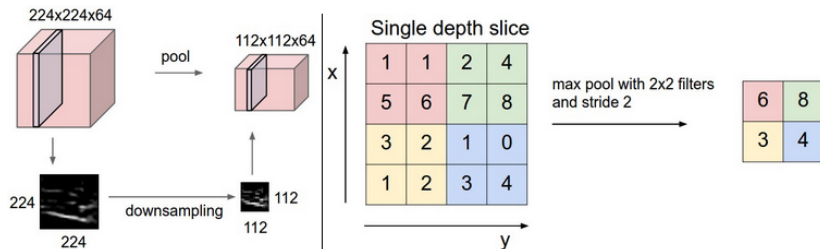
0	-4	-4
1	-1	-2
2	-3	-1

$o[:, :, 1]$

-2	-1	-2
-2	3	-4
1	-4	-3

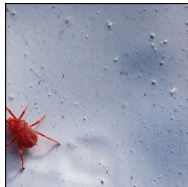
toggle movement

Couches de pooling



Pooling layer downsamples the volume spatially, independently in each depth slice of the input volume. **Left:** In this example, the input volume of size $[224 \times 224 \times 64]$ is pooled with filter size 2, stride 2 into output volume of size $[112 \times 112 \times 64]$. Notice that the volume depth is preserved. **Right:** The most common downsampling operation is max, giving rise to **max pooling**, here shown with a stride of 2. That is, each max is taken over 4 numbers (little 2×2 square).

Présentation des résultats



mite



container ship



motor scooter



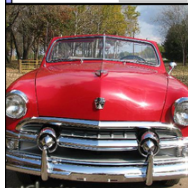
leopard

	mite
	black widow
	cockroach
	tick
	starfish

	container ship
	lifeboat
	amphibian
	fireboat
	drilling platform

	motor scooter
	go-kart
	moped
	bumper car
	golfcart

	leopard
	jaguar
	cheetah
	snow leopard
	Egyptian cat



grille



mushroom



cherry



Madagascar cat

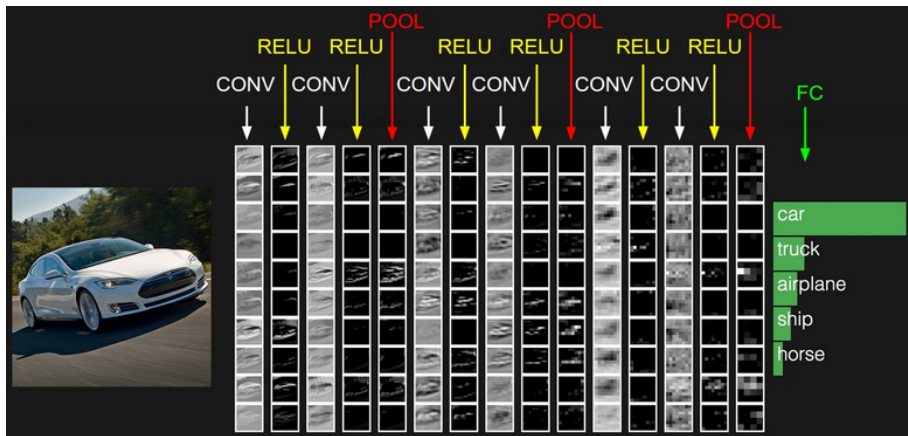
	convertible
	grille
	pickup
	beach wagon
	fire engine

	agaric
	mushroom
	jelly fungus
	gill fungus
	dead-man's-fingers

	dalmatian
	grape
	elderberry
	ffordshire bullterrier
	currant

	squirrel monkey
	spider monkey
	titi
	indri
	howler monkey

Présentation des résultats



Et après...

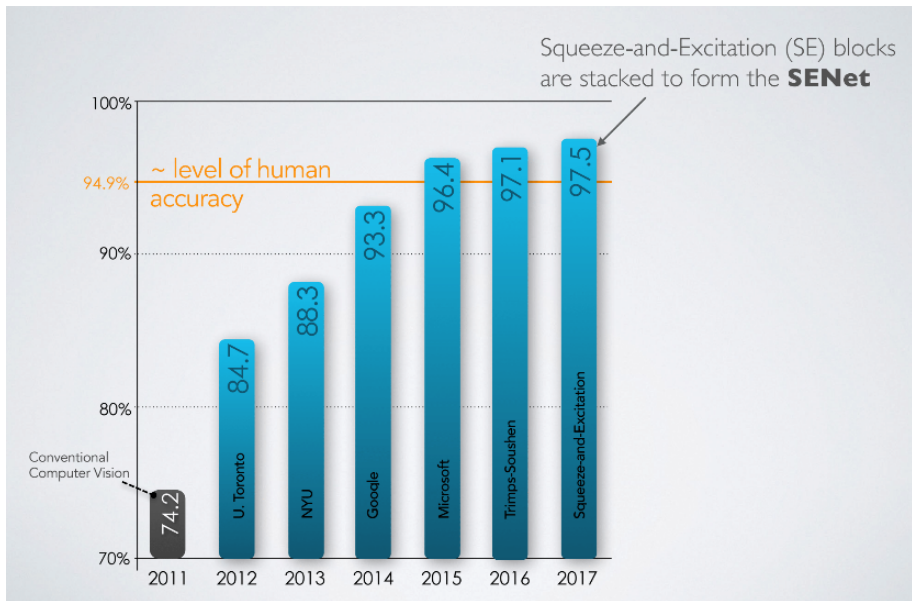


Image initial

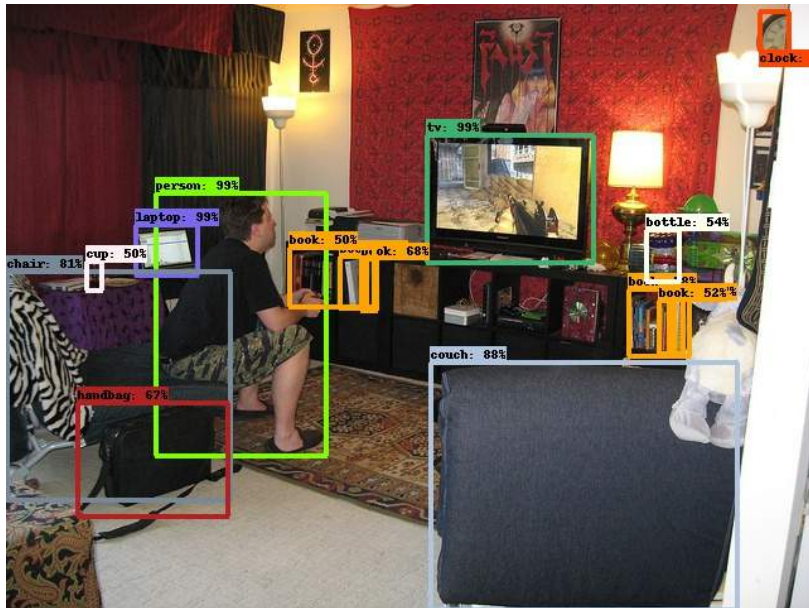


Image après transplânt

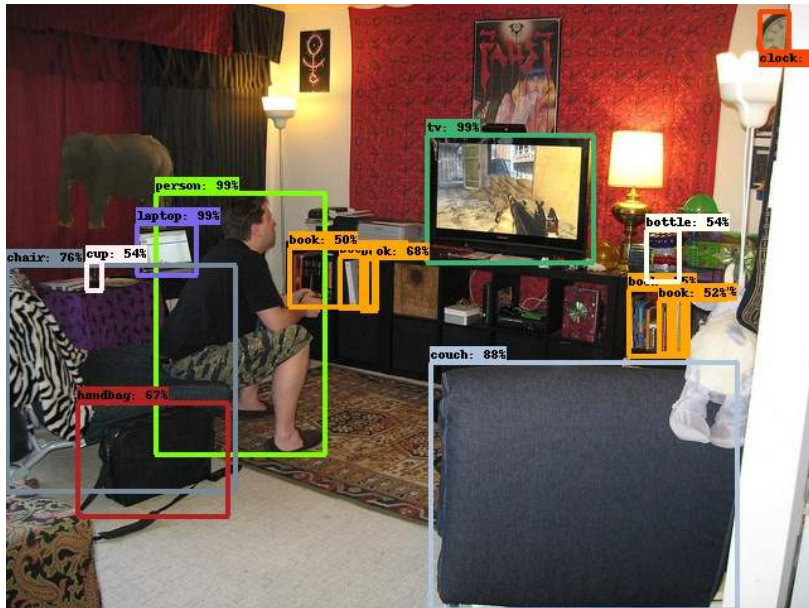


Image après transplant

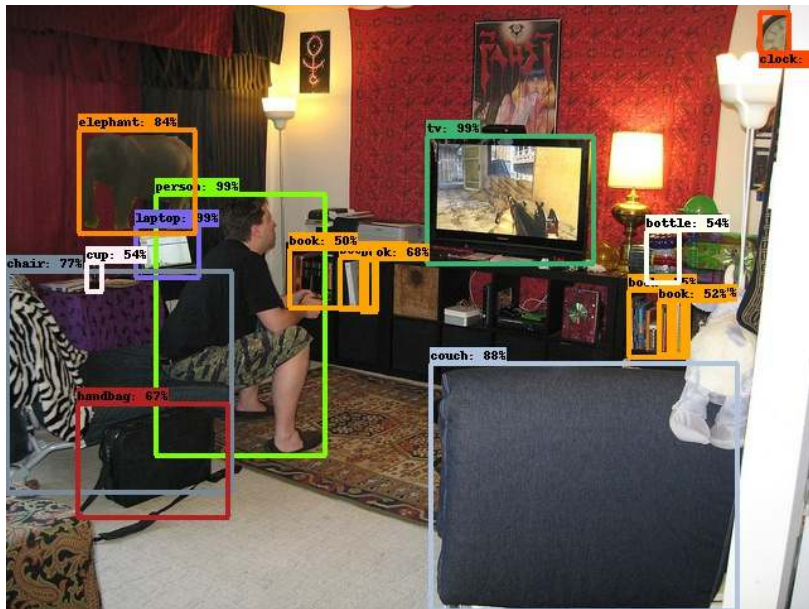


Image après transplânt

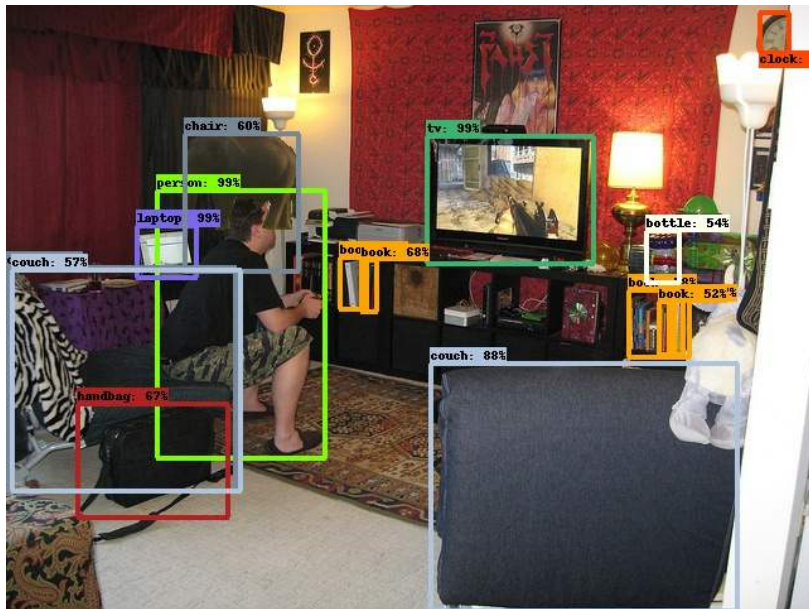


Image initial

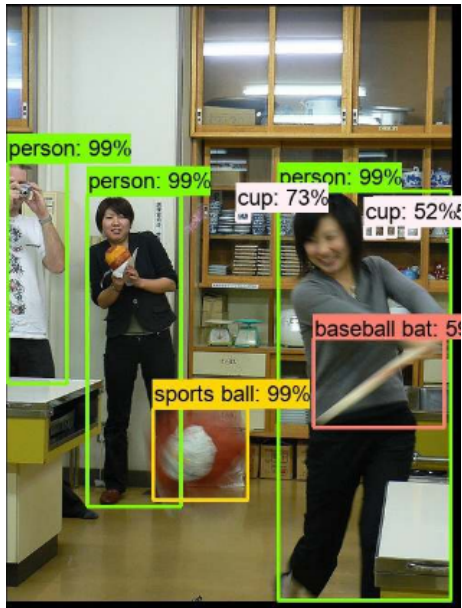


Image après transplânt



Image après transplant



[2] [3] [4] [6] [7] [5] [1]



Danfei Xu Fei-Fei Li Ranjay Krishnam. *CS231n: Convolutional Neural Networks for Visual Recognition*. URL: <http://cs231n.stanford.edu/>.



Kunihiko Fukushima. “Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position”. In: *Biological Cybernetics* (Apr. 1980). URL: https://lrn.no-ip.info/other/books/neural/Neocognitron/1980_Neocognitron%20A%20Self-organizing%20Neural%20Network%20Model%20for%20a%20Mechanism%20of%20Pattern%20Recognition%20Unaffected%20by%20Shift%20in%20Position.pdf.



Hubel and Wiesel. “Receptive fields of single neurones in the cat’s striate cortex.”. In: *MEDLINE* (Oct. 1959).



Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “ImageNet Classification with Deep Convolutional Neural Networks”. In: *Advances in Neural Information Processing Systems 25*. Ed. by F. Pereira et al. Curran Associates, Inc., 2012, pp. 1097–1105. URL: <http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf>.



Hugo Larochelle. *Support de cours Master Informatique*. URL: <https://sites.google.com/view/ift-6135-h2020/lectures?authuser=0>.



V. Nair and G. E. Hinton. “Rectified linear units improve restricted boltzmann machines”. In: ed. by F. Pereira et al. 27th International Conference on Machine Learning, 2010.



Amir Rosenfeld, Richard Zemel, and John K. Tsotsos. *The Elephant in the Room*. 2018. arXiv: 1808.03305 [cs.CV].