



Introduction À L'apprentissage Profond

Réseaux de neurones à Convolution (CNN)

S.Boudra, I.Yahiaoui

1

1. Préparation des données et K-fold validation
2. Mesures d'évaluation de classification
3. Optimisation des hyperparamètres
4. Sur-apprentissage et régularisation

Rappel

2

Ce qu'on va voir

Contenu

1. Limites des ANN
2. Cortex visuel
3. Reconnaissance classique vs. automatique
4. Construction d'un CNN
5. Visualisation d'un CNN

3

Classification des images:

représentation du contenu visuel et défis

Source: <http://cs231n.stanford.edu/>

4

Classification des images



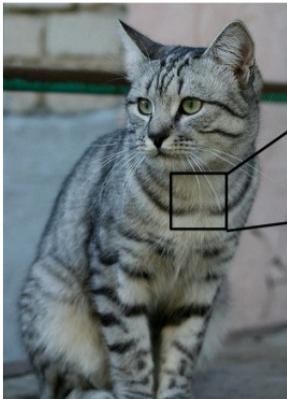
Cat

Source: <http://cs231n.stanford.edu/>

This image by Nikita is
licensed under CC-BY 2.0

5

Classification des images



[1105 112 189 111 184 99 186 99 96 183 112 119 184 97 93 87]
[91 98 182 106 184 79 98 183 99 185 123 136 118 185 94 85]
[76 85 98 182 106 128 185 87 98 95 98 115 112 186 183 99 85]
[89 91 101 106 128 185 87 98 95 98 115 112 186 183 99 85]
[106 91 61 66 69 91 88 85 181 187 189 98 75 84 96 95]
[114 108 85 55 55 69 64 54 64 87 112 129 98 74 84 91]
[133 137 147 103 65 81 88 65 52 54 74 84 182 93 85 82]
[134 138 148 104 65 81 88 65 52 54 74 84 182 93 85 82]
[125 133 149 147 119 121 117 94 65 79 95 65 68 73 86 181]
[127 125 131 147 133 127 126 131 111 98 89 75 61 64 72 84]
[115 114 109 123 158 145 131 118 113 188 92 74 65 72 78]
[89 93 98 97 188 147 131 118 113 114 113 189 186 95 77 80]
[63 67 71 88 101 124 126 119 101 187 114 131 119]
[62 65 82 88 78 71 88 101 124 126 119 101 187 114 131 119]
[63 65 75 88 89 71 62 81 128 130 135 105 81 98 118 118]
[87 65 71 87 186 109 45 76 130 126 107 92 94 105 112]
[118 67 71 87 186 109 45 76 130 126 107 92 94 105 112]
[164 146 112 88 82 120 124 104 76 48 45 66 88 181 182 189]
[157 170 157 120 93 86 114 132 112 97 69 55 78 82 99 94]
[130 120 134 161 139 186 189 118 121 134 114 87 65 53 69 86]
[121 122 121 182 88 82 86 94 117 145 149 153 182 58 78 92 107]
[123 187 96 86 83 112 153 149 122 189 184 75 88 187 112 99]
[122 121 182 88 82 86 94 117 145 149 153 182 58 78 92 107]
[122 164 148 103 71 56 78 83 93 183 119 139 182 61 69 84]

- Ce que voit l'ordinateur
- (une matrice de pixels dont les valeurs d'intensité [0,255])

Source: <http://cs231n.stanford.edu/>

This image by Nikita is
licensed under CC-BY 2.0

6

Défis: présence de fond



[This image is CC0 1.0 public domain](#)



[This image is CC0 1.0 public domain](#)

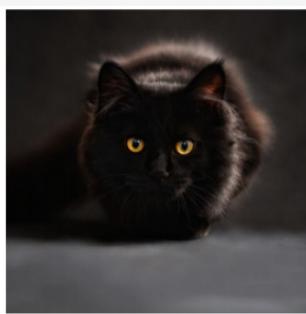
Source: <http://cs231n.stanford.edu/>

7

Défis: changement de liminance



[This image is CC0 1.0 public domain](#)



[This image is CC0 1.0 public domain](#)



[This image is CC0 1.0 public domain](#)



[This image is CC0 1.0 public domain](#)

Source: <http://cs231n.stanford.edu/>

8

Défis: occultation



[This image](#) is CC0 1.0 public domain



[This image](#) is CC0 1.0 public domain



[This image](#) by [jonsson](#) is licensed under CC-BY 2.0

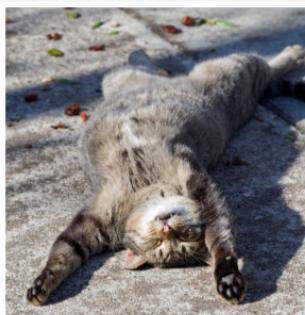
Source: <http://cs231n.stanford.edu/>

9

Défis: déformation



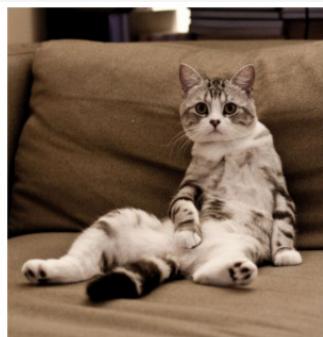
[This image](#) by [Umberto Salvagnin](#) is licensed under CC-BY 2.0



[This image](#) by [Umberto Salvagnin](#) is licensed under CC-BY 2.0



[This image](#) by [sare bear](#) is licensed under CC-BY 2.0



[This image](#) by [Tom ThaI](#) is licensed under CC-BY 2.0

Source: <http://cs231n.stanford.edu/>

10

Défis: variabilité intra-classe



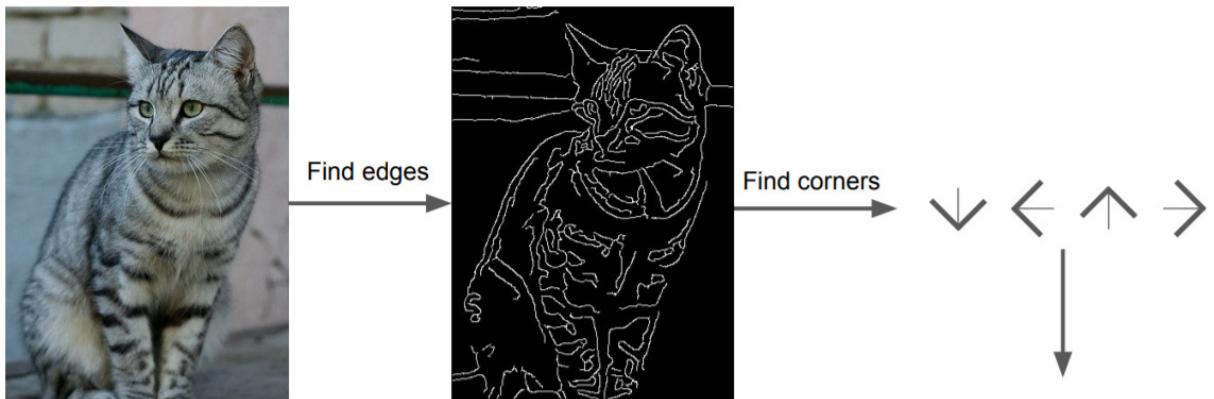
Source: <http://cs231n.stanford.edu/>

This image is CC0 1.0 public domain

11

Représentation du contenu visuel

- Extraire des attributs de bas niveaux,
- Appliquer un algorithme d'apprentissage automatique pour prédire la classe de l'image (KNN,SVM)



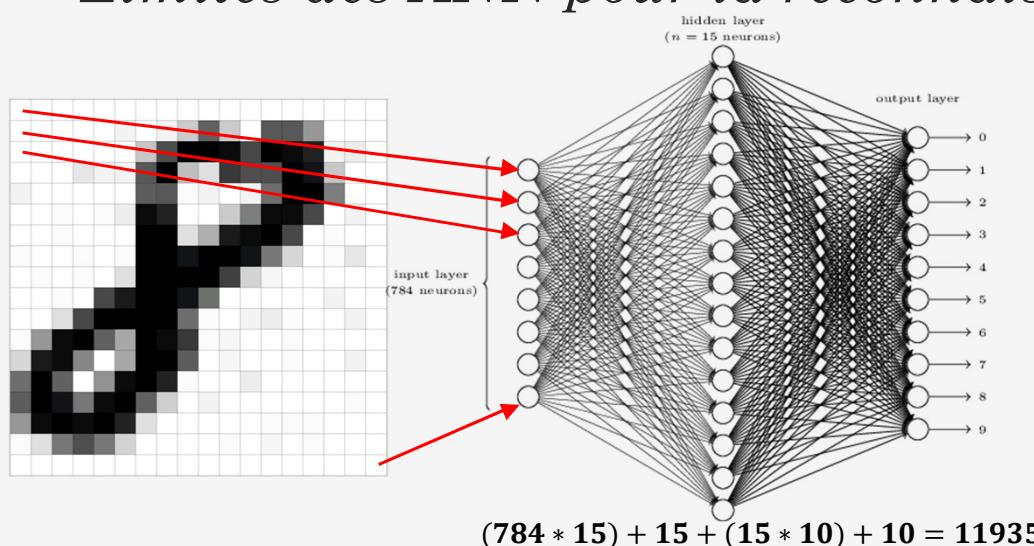
Source: <http://cs231n.stanford.edu/>

12

Limites des ANN pour la reconnaissance

13

Limites des ANN pour la reconnaissance

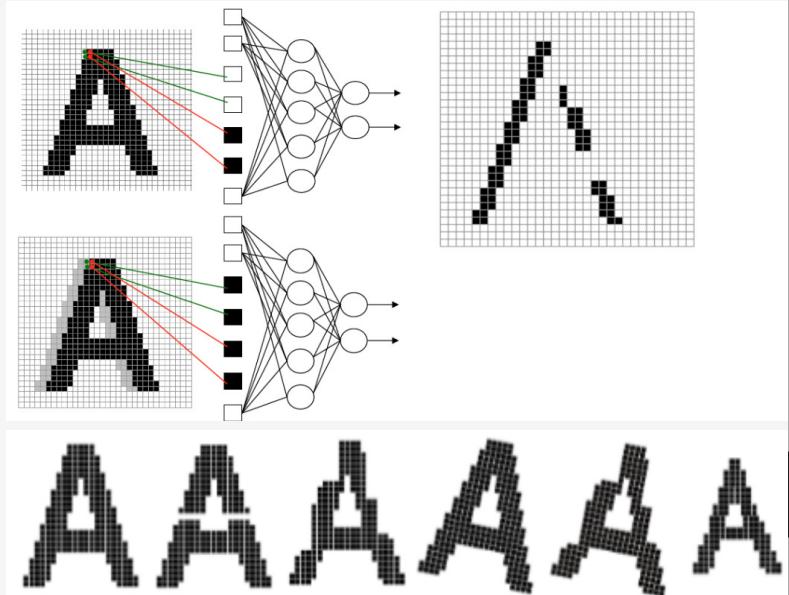


- #nbr de paramètre très grand rien qu'avec une seul couche cachée

14

Limites des ANN pour la reconnaissance

- Problème d'invariance au changement locaux:
 - Changement d'échelle
 - Déformation
 - Rotation locale
 - Translation
- Perte de la « *relation spatiale*» entre les pixels.



15

Limites des ANN pour la reconnaissance

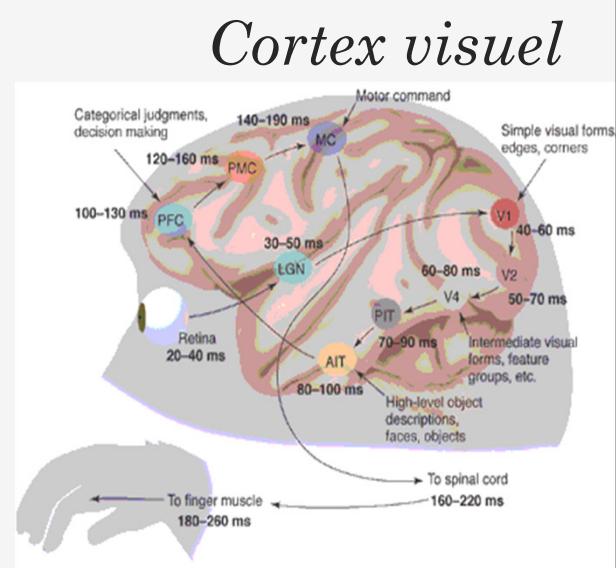
Quelques solutions:

- Limiter le nombre de paramètres à apprendre.
- Représenter la structure local et la relation spatiale.
- Assurer une invariance aux changements locaux.

16

Cortex Visuel

17



- Hubel & Wiesel, 1959

- La reconnaissance d'objet par étapes:

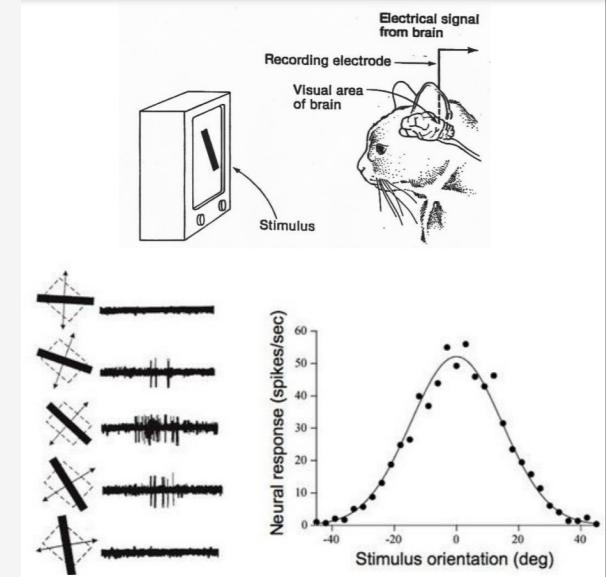
Voir un objet => LGN => V1 => V2 => V4 => allumer l'ensemble de neurone qui représentent le concept de l'objet vu.

- Opération extrêmement rapide, 100ms

18

Cortex visuel

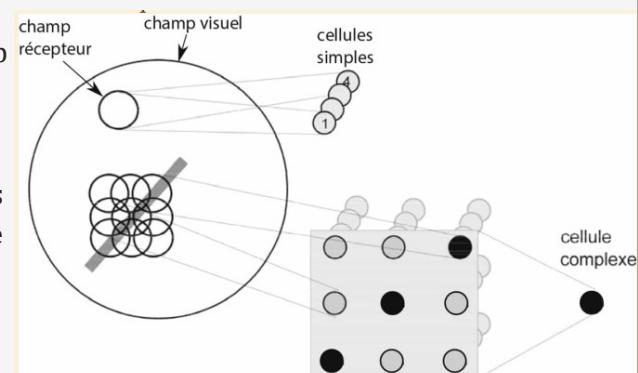
- Hubel & Wiesel, 1959
- Le neurone s'active pour des contours dans un angle donné.
- À regarder:
<https://www.youtube.com/watch?v=8VdFf3egwfg&feature=youtu.be&t=1m10s>



19

Cortex visuel: cellule simple

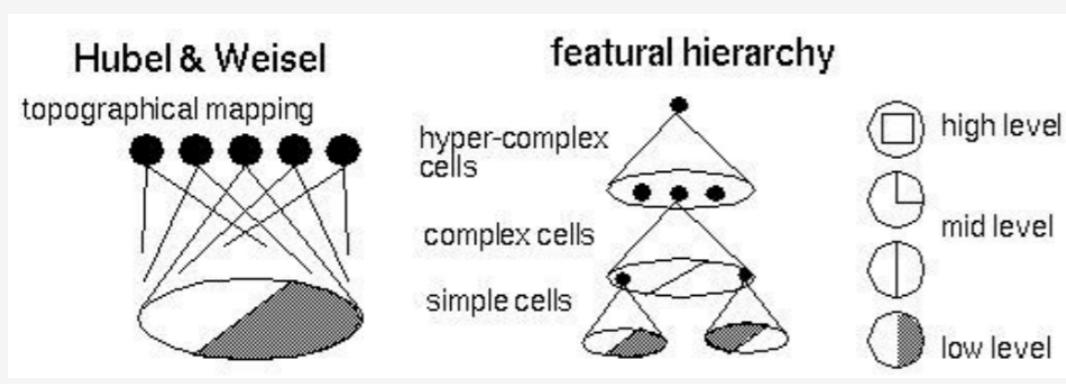
- Hubel & Wiesel, 1959
- Une cellule simple détecte un motif sur le champ récepteur.
- les cellules complexes **agrègent** les réponses des cellules simples voisines d'un même type sur une **petite fenêtre**.
- **Agrégation (pooling)** => calculer une moyenne des réponses, ou simplement prendre la réponse maximale.



20

Organisation hiérarchique

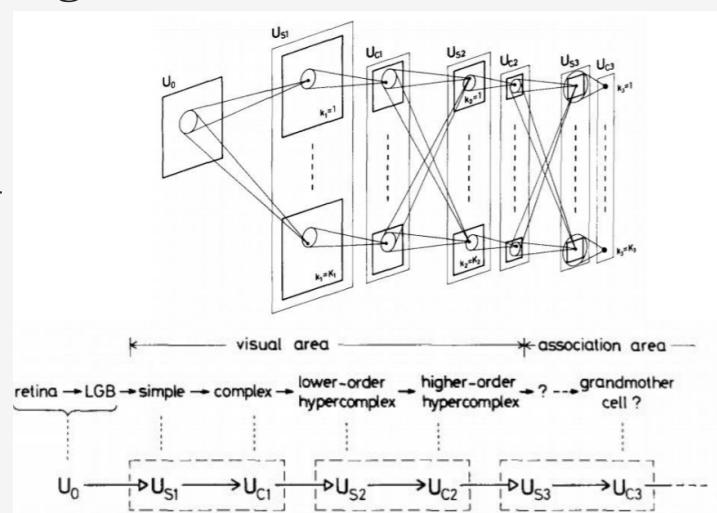
- Hubel & Wiesel, 1962



21

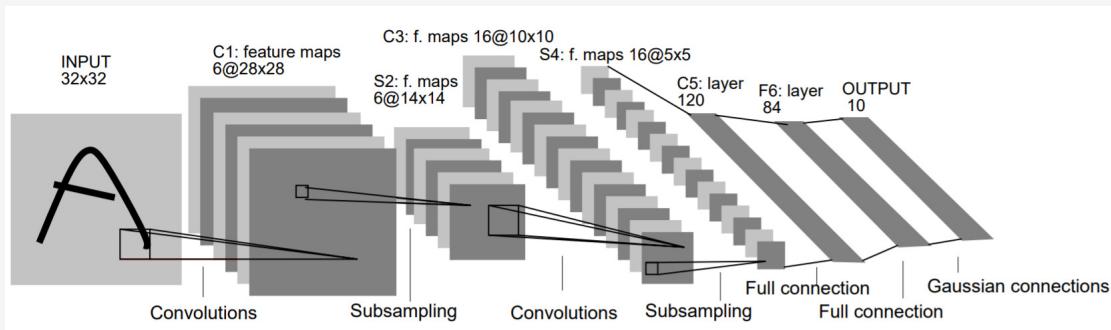
Modélisation: Neurocognitron: Fukushima 1980

- Succession de cellule simple-cellules complexes
- Pas d'apprentissage supervisé de bout-en-bout.
- Pas de rétropropagation (pas encore inventée !)



22

Modélisation: Rétropropagation: Gradient descent



Y.LeCun et al. "GradientBased Learning Applied to Document Recognition", proceedings of the IEEE.1998

23

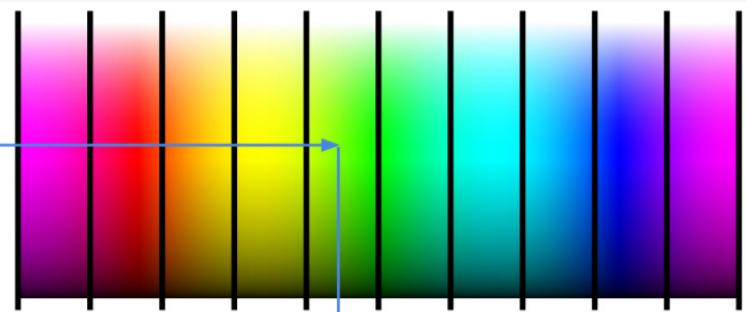
Modèle de reconnaissance



- Les caractéristique ne sont pas apprises automatiquement à partir de l'image.
- Un classifieurs générique, comme SVM.

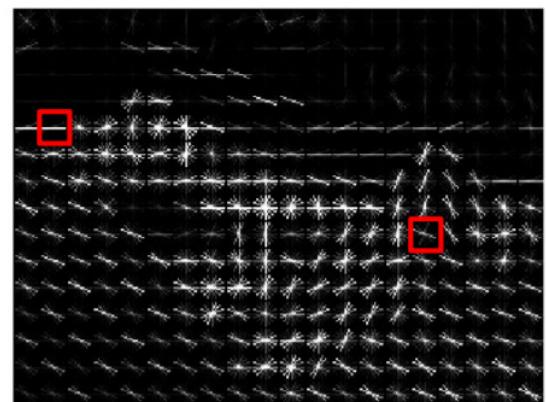
24

Histogramme de couleur



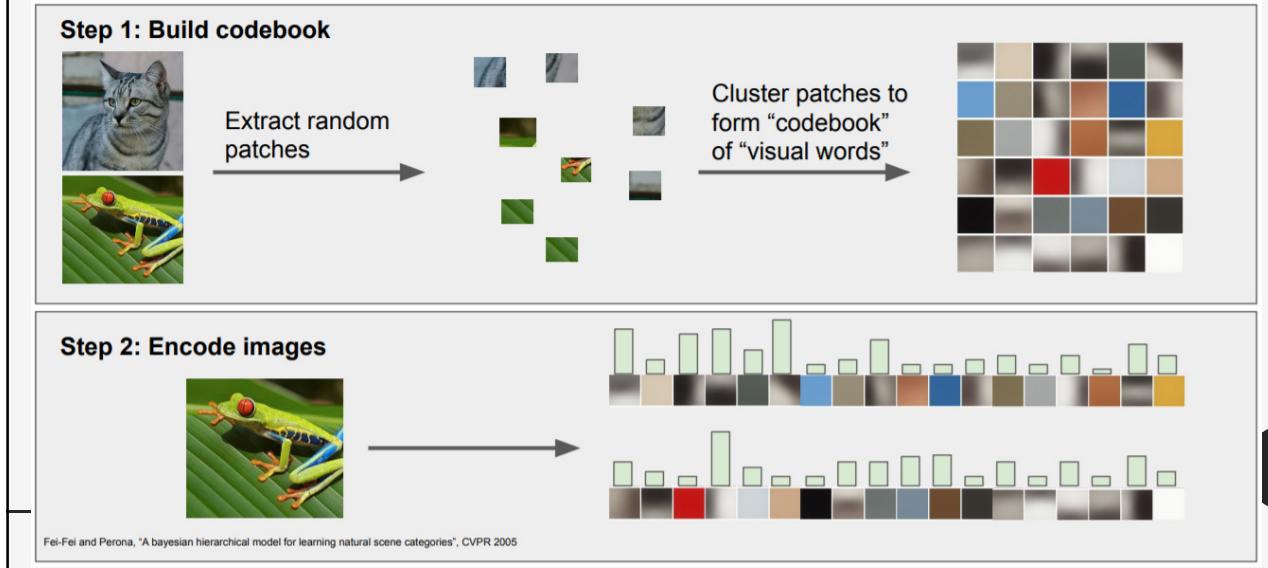
25

Histogram of Oriented Gradients



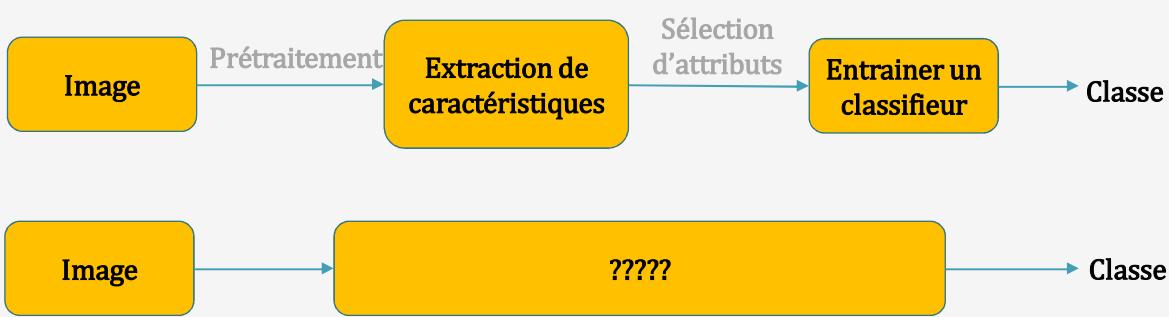
26

Bag of Words



27

Modèle de reconnaissance



28

Introduction au réseaux de neurones convolutifs

29

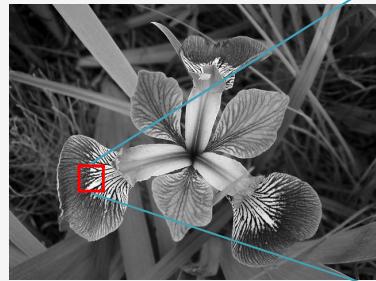
Notion de filtre



$$\begin{aligned}
 &= \frac{1}{9}((-1) * 138 + (-1) * 171 + (-1) * 125 \\
 &\quad + (-1) * 117 + (8) * 191 + (-1) * 167 \\
 &\quad + (-1) * 44 + (-1) * 88 + (-1) * 136) = \mathbf{60}
 \end{aligned}$$

30

Notion de filtre



138	171	125	62	31	90	153
117	191	167	128	53	38	65
44	88	136	149	87	52	46
42	44	49	70	75	60	44
80	46	44	45	42	54	52
192	128	81	62	47	38	42

$$\begin{array}{|c|c|c|} \hline 171 & 125 & 62 \\ \hline 191 & 167 & 128 \\ \hline 88 & 136 & 149 \\ \hline \end{array} * \begin{array}{|c|c|c|} \hline -1 & -1 & -1 \\ \hline -1 & 8 & -1 \\ \hline -1 & -1 & -1 \\ \hline \end{array}$$

$$= \frac{1}{9}((-1) * 171 + (-1) * 125 + (-1) * 62 + (-1) * 191 + (8) * 167 + (-1) * 128 + (-1) * 88 + (-1) * 136 + (-1) * 149) = 31$$

31

Notion de filtre



$$* \begin{array}{|c|c|c|} \hline -1 & -1 & -1 \\ \hline -1 & 8 & -1 \\ \hline -1 & -1 & -1 \\ \hline \end{array} =$$



32

Notion de filtre

- Filtre:

+++

- Facile à calculer,
- Extraire des attributs de bas niveaux, comme les contours, coins, jonction Y et X,, etc
- Encoder la relation spatiale entre les pixels,
- Efficace pour la représentations du contenu visuel.

- - -

- Difficile à concevoir,
- Basés sur l'expertise,
- Spécifique à l'application et au données.



33

Top Sobel $\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$	Bottom Sobel $\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$	left Sobel $\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$	right Sobel $\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$
Blur $\begin{bmatrix} 0.0625 & 0.125 & 0.0625 \\ 0.125 & 0.25 & 0.125 \\ 0.0625 & 0.125 & 0.0625 \end{bmatrix}$	Emboss $\begin{bmatrix} -2 & -1 & 0 \\ -1 & 1 & 1 \\ 0 & 1 & 2 \end{bmatrix}$	Outline $\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	Outline $\begin{bmatrix} -3 & -2 & -1 \\ 4 & 8 & 4 \\ -3 & -2 & -1 \end{bmatrix}$

34

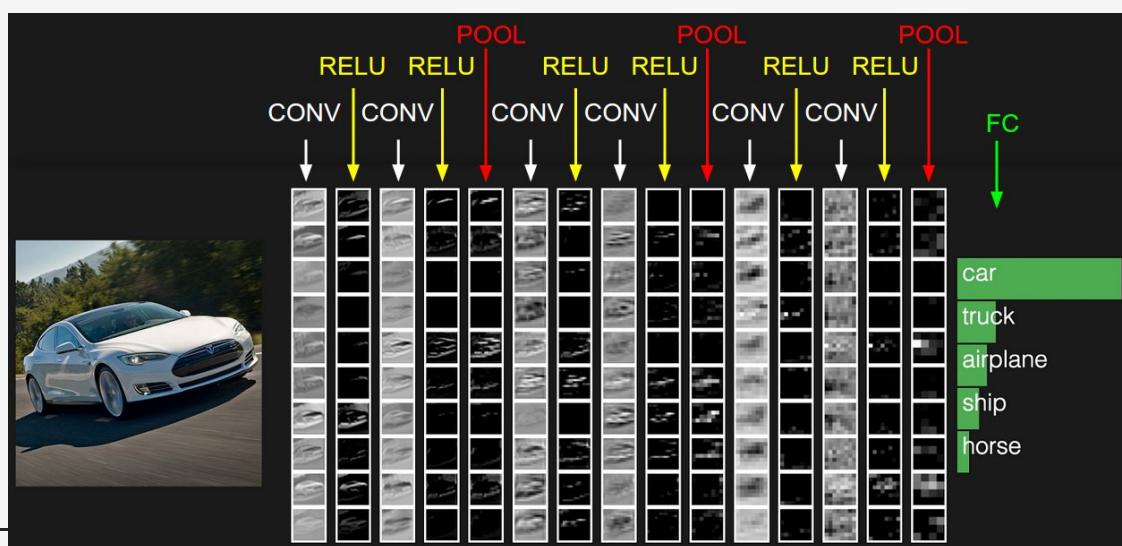
De ANN à CNN

- Et si on peut apprendre ces filtre automatiquement pour représenter l'image?



35

Architecture



36

Convolution

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

1	0	1
0	1	0
1	0	1

1 _{x1}	1 _{x0}	1 _{x1}	0	0
0 _{x0}	1 _{x1}	1 _{x0}	1	0
0 _{x1}	0 _{x0}	1 _{x1}	1	1
0	0	1	1	0
0	1	1	0	0

Image

4		

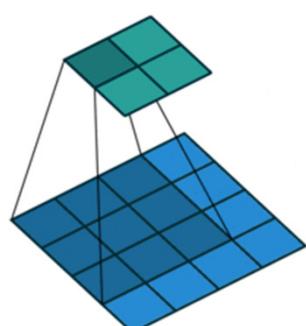
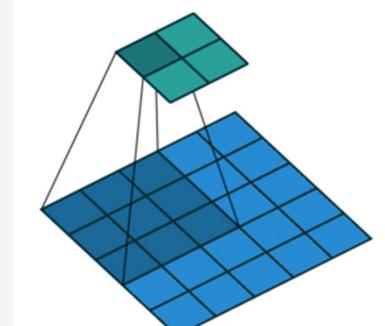
Convolved Feature

37

Convolution

Bleu: entrée (input)

Vert: carte de caractéristiques (features map)

no padding
no strideno padding
stridehttps://github.com/vdumoulin/conv_arithmetic

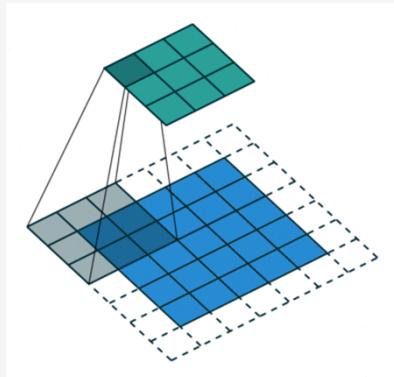
38

19

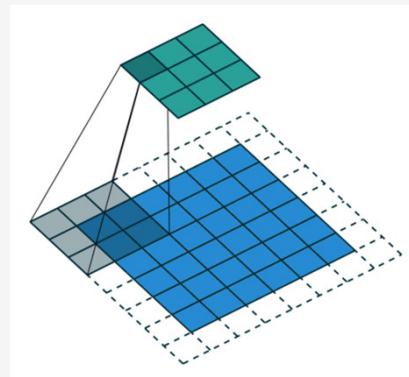
Convolution

Bleu: entrée (input)

Vert: carte de caractéristiques (features map)



padding
stride



padding
stride

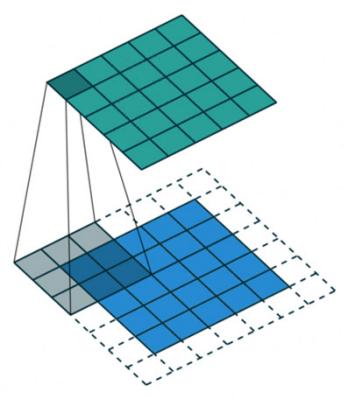
https://github.com/vdumoulin/conv_arithmetic

39

Convolution

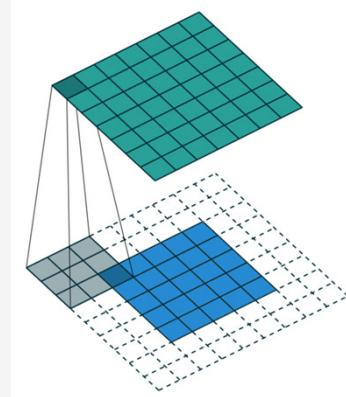
Bleu: entrée (input)

Vert: carte de caractéristiques (features map)



half padding

no stride



full padding

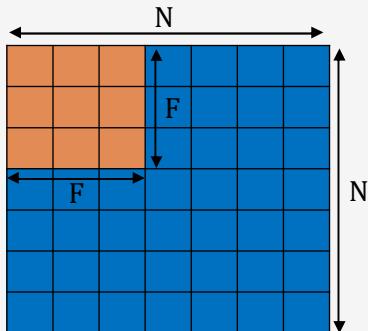
no stride

https://github.com/vdumoulin/conv_arithmetic

40

20

Convolution



Taille de la carte de caractéristiques (features map)
 $(N-F)/\text{stride}+1$

Exp: $N=7; F=3$

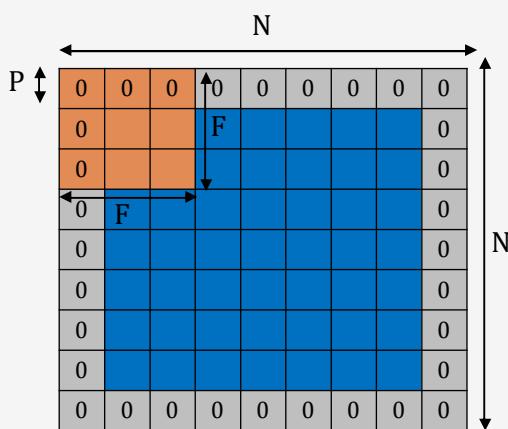
$$\text{stride } 1 \Rightarrow (7-3)/1+1=5$$

$$\text{stride } 2 \Rightarrow (7-3)/2+1=3$$

$$\text{stride } 3 \Rightarrow (7-3)/3+1=2.22$$

41

Convolution



Taille de la carte de caractéristiques (features map)
 $(N-F+2P)/\text{stride}+1$

Exp: $N=7;$

$$\text{stride } 1, \text{zero pad } 1, F=3 \Rightarrow (7-3+2*1)/1+1=7$$

$$\text{stride } 1, \text{zero pad } 2, F=5 \Rightarrow (7-5+2*2)/1+1=7$$

$$\text{stride } 1, \text{zero pad } 3, F=7 \Rightarrow (7-7+2*3)/1+1=7$$

Taille de la feature map préservée

42

Convolution



43

Taille de feature map et n°paramètres

Taille de la carte de caractéristiques (features map) **(N-F+2P)/stride+1**

Taille de volume: $32 \times 32 \times 3$

K=10 filtres de 5×5 avec stride **S=1** et zero pad **P=2**

Taille de la feature map:

???

44

Taille de feature map et n°paramètres

Taille de la carte de caractéristiques (features map) **(N-F+2P)/stride+1**

Taille de volume: $32 \times 32 \times 3$

K=10 filtres de 5×5 avec stride **S=1** et zero pad **P=2**

Taille de la feature map:

$$(N-F+2P)/stride+1 \Rightarrow (32 - 5 + 2*2)/1 + 1 = 32 \quad \text{Donc: } 32 \times 32 \times 10$$

45

Taille de feature map et n°paramètres

Taille de la carte de caractéristiques (features map) **(N-F+2P)/stride+1**

Taille de volume: $32 \times 32 \times 3$

K=10 filtres de 5×5 avec stride **S=1** et zero pad **P=2**

Taille de la feature map:

$$(N-F+2P)/stride+1 \Rightarrow (32 - 5 + 2*2)/1 + 1 = 32 \quad \text{Donc: } 32 \times 32 \times 10$$

Communément:
K = puissance de 2 : 32, 64, 128,
 512

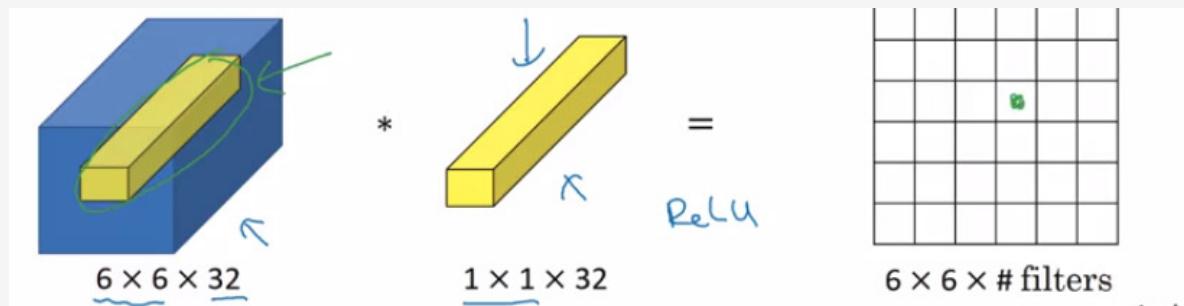
F = 3, S=1, P=1

F = 5, S=1, P=2

F = 1, S=1, P=0

46

(1×1) Conv



47

Taille de feature map et n°paramètres

Taille de la carte de caractéristiques (features map) $(N-F+2P)/\text{stride}+1$

Taille de volume: $32 \times 32 \times 3$

K=10 filtres de 5×5 avec stride **S=1** et zero pad **P=2**

Nombres de paramètres:

???

48

Taille de feature map et n°paramètres

Taille de la carte de caractéristiques (features map) $(F \times F \times D)K$

Taille de volume: $32 \times 32 \times 3$ (D=3)

K=10 filtres de 5×5 avec stride **S=1** et zero pad **P=2**

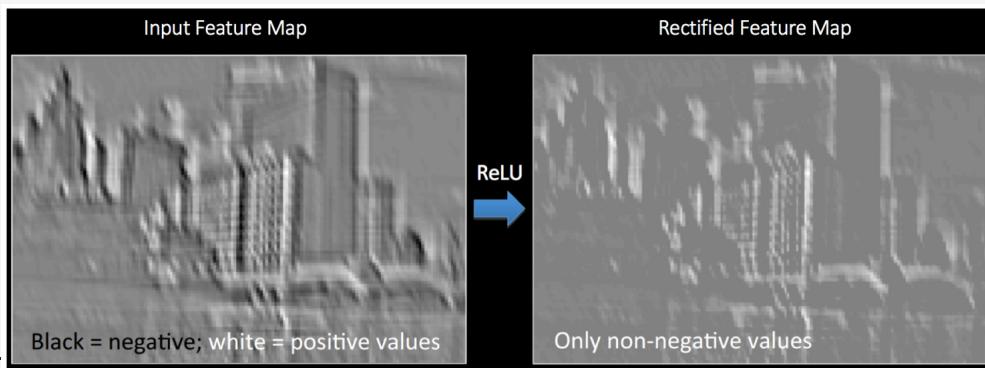
Nombres de paramètres:

Chaque filtre a $5 \times 5 \times 3 + 1$ (pour le biais) = 76
Et on a 10 filtres

Donc: $76 \times 10 = 760$ paramètres

49

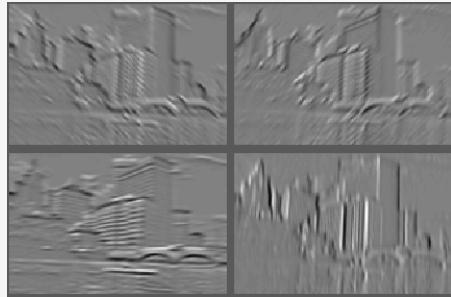
Activation non-linéaire



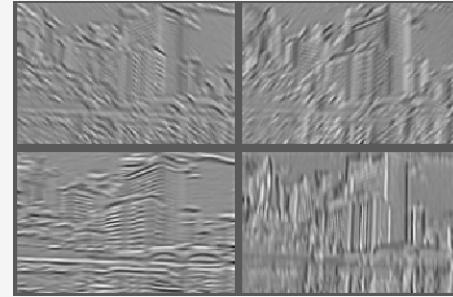
50

Normalisation du contraste

- mean = 0, std. = 1



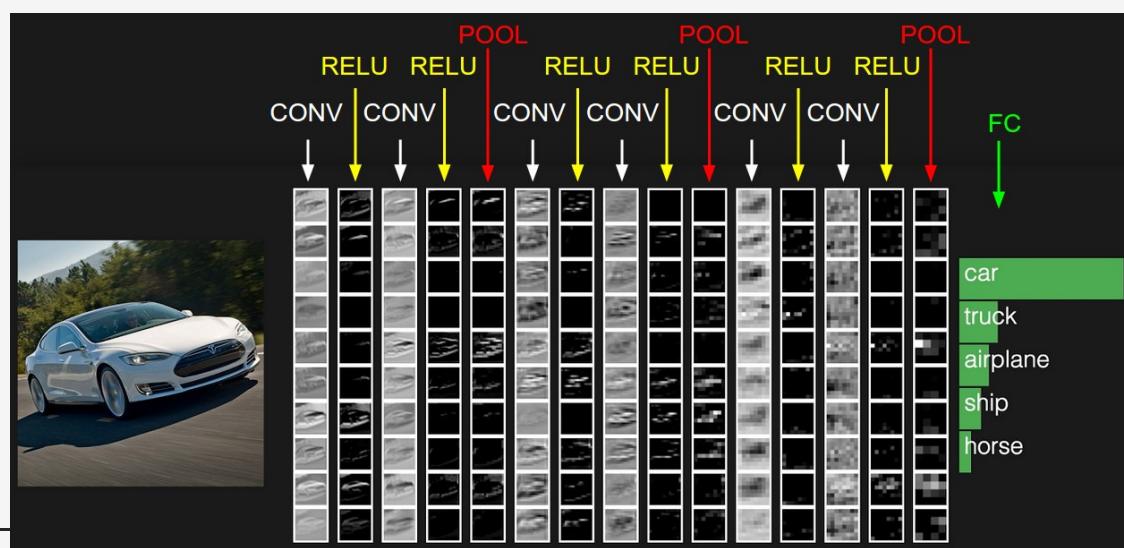
Feature Maps



Feature Maps
apres normalization du contraste

51

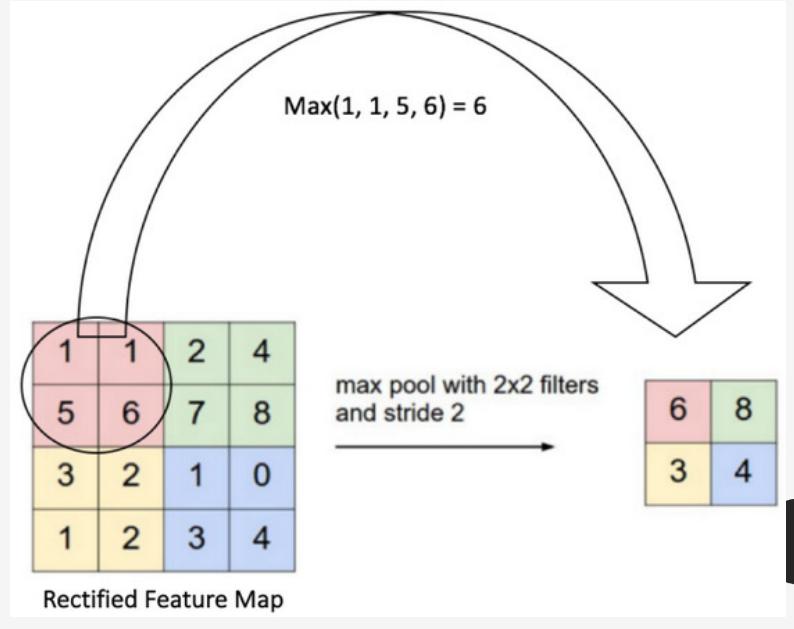
Architecture



52

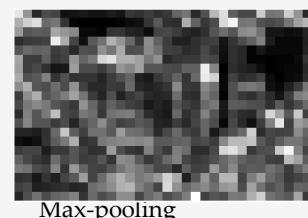
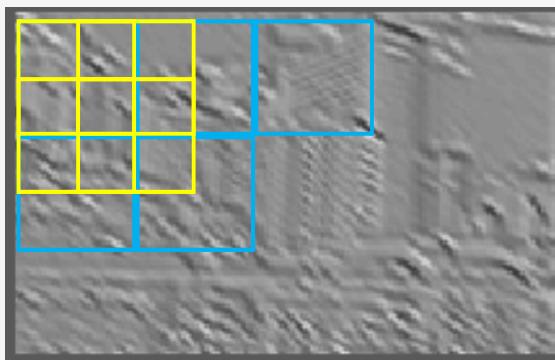
Pooling

- Agrégation statistique:
- max (max-pooling)
- moyenne (average-pooling)
- Réduire la taille de la feature map (sub-sampling)



53

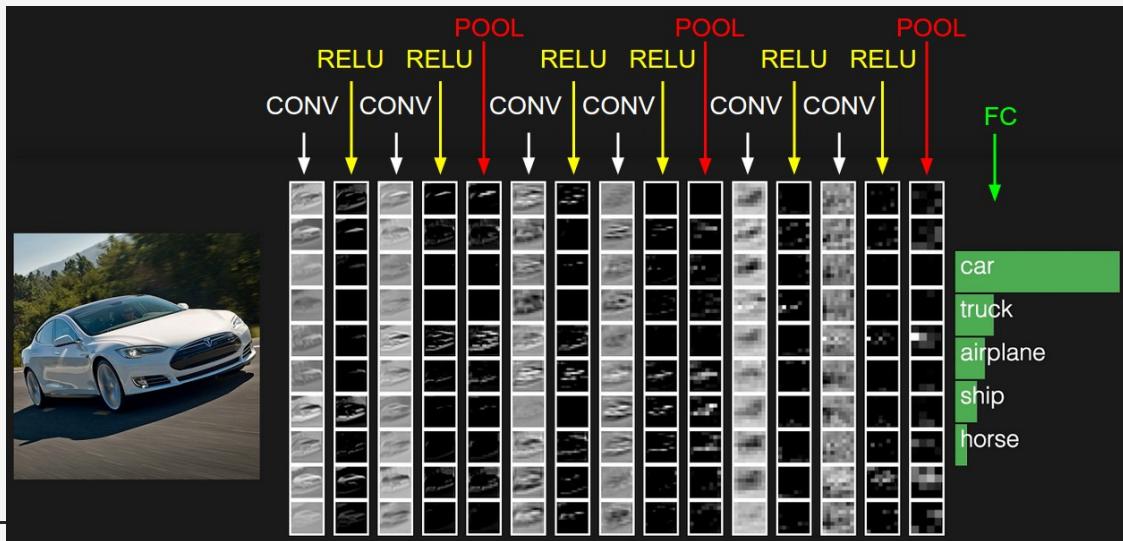
Pooling



- Avec ou sans recouvrement des fenêtres
- Assurer une invariance aux changements locaux.

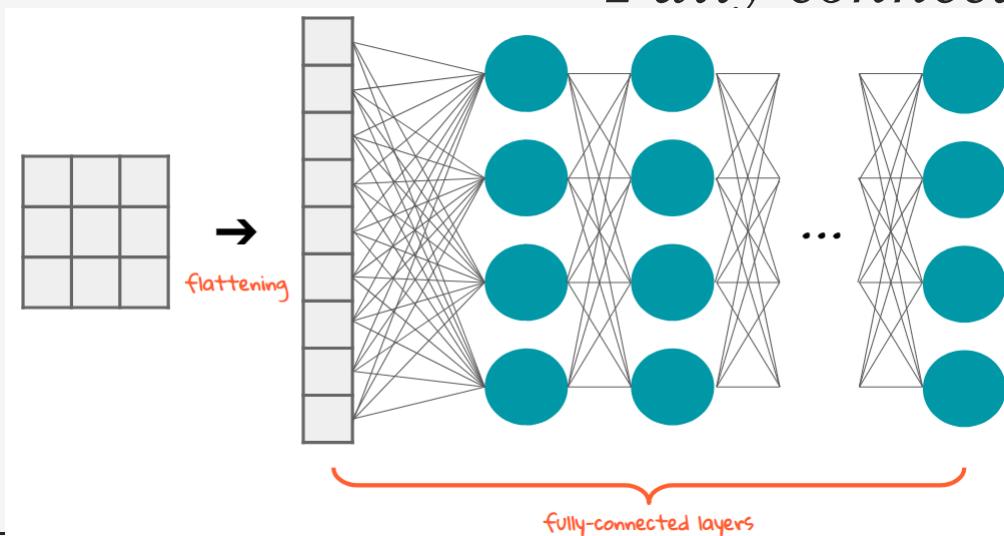
54

Architecture



55

Fully connected



56

Fully connected

1	1	0
4	2	1
0	2	1

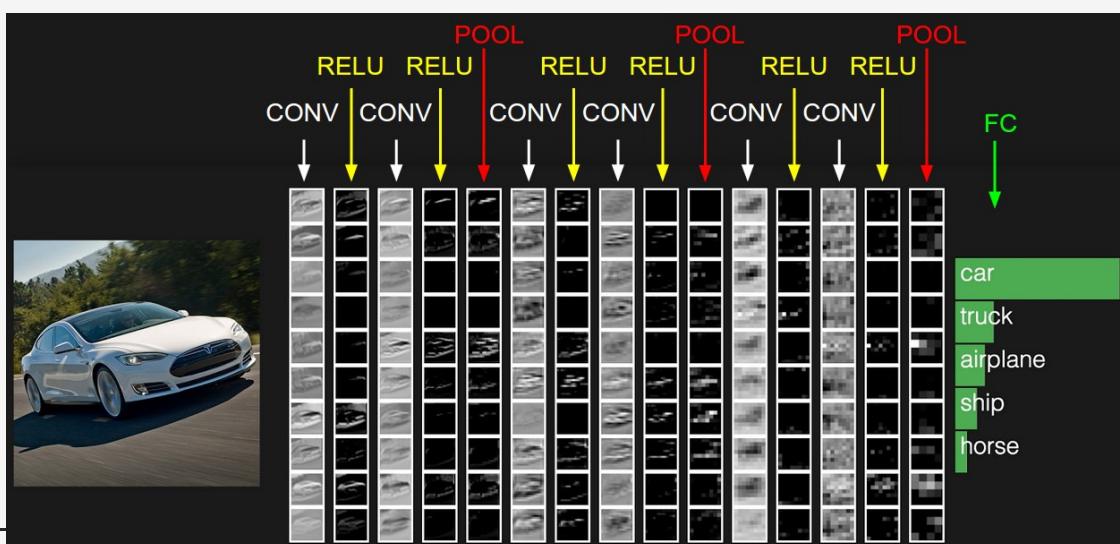
Pooled Feature Map

Flattening

1
1
0
4
2
1
0
2
1

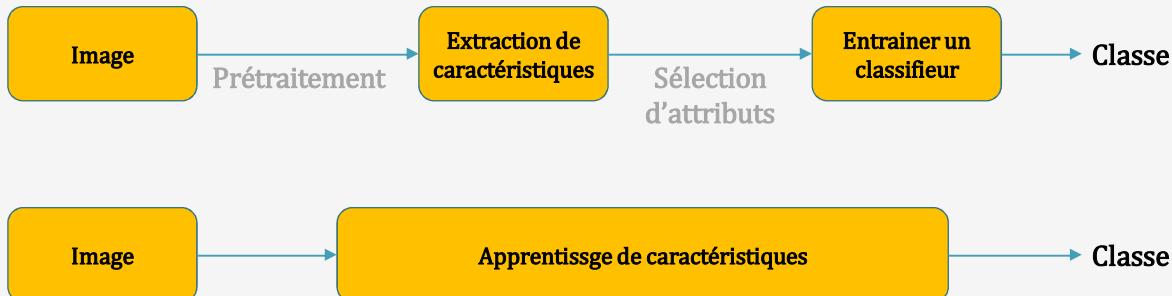
57

Architecture



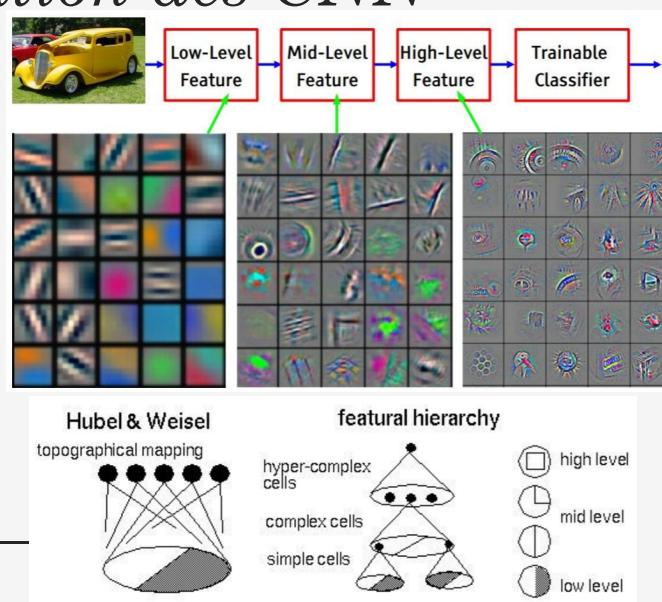
58

Modèle de reconnaissance



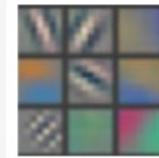
59

Visualisation des CNN



60

Visualisation des CNN



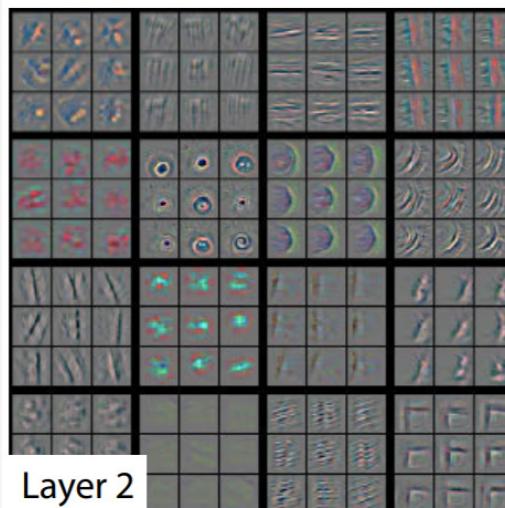
Layer 1



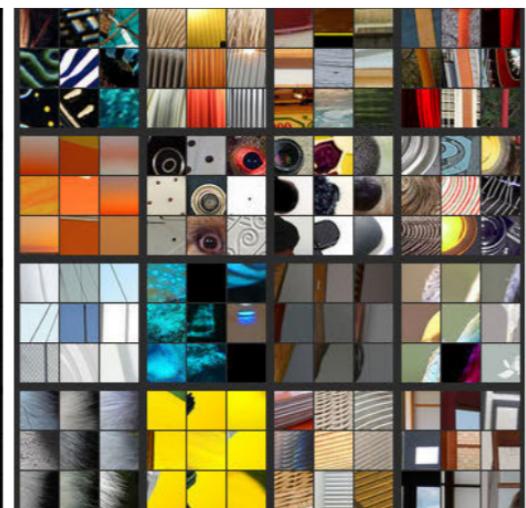
Zeiler, Matthew D., and Rob Fergus. "Visualizing and understanding convolutional networks." *ECCN*. Springer, Cham, 2014.

61

Visualisation des CNN



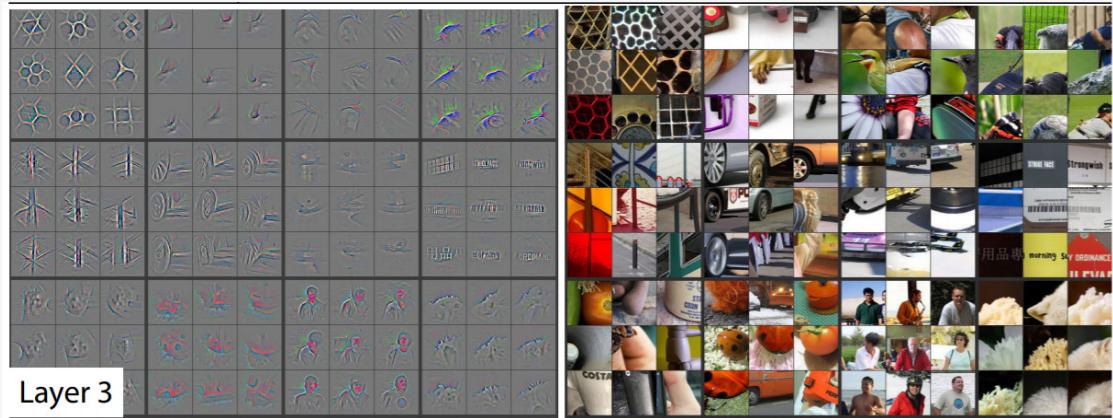
Layer 2



Zeiler, Matthew D., and Rob Fergus. "Visualizing and understanding convolutional networks." *ECCN*. Springer, Cham, 2014.

62

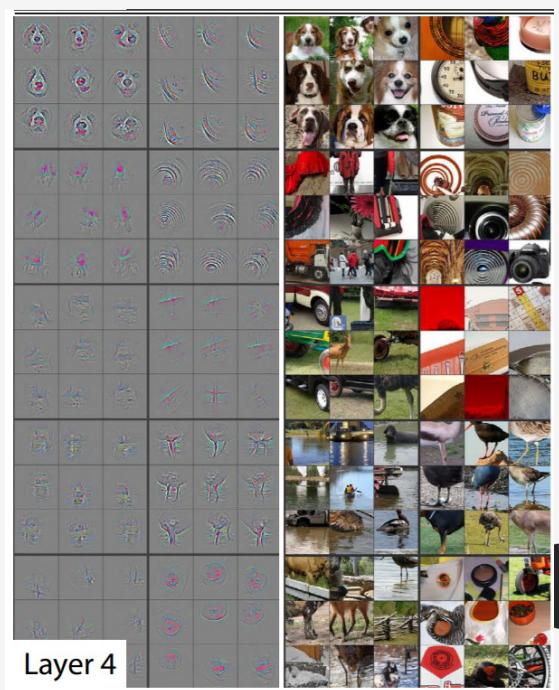
Visualisation des CNN



Zeiler, Matthew D., and Rob Fergus. "Visualizing and understanding convolutional networks." *ECCN*. Springer, Cham, 2014.

63

Visualisation des CNN

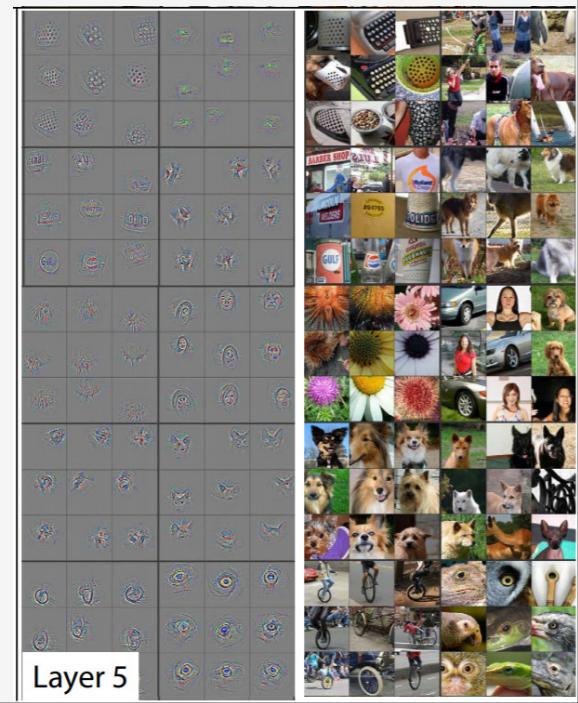


Zeiler, Matthew D., and Rob Fergus. "Visualizing and understanding convolutional networks." *ECCN*. Springer, Cham, 2014.

64

Visualisation des CNN

Zeiler, Matthew D., and Rob Fergus. "Visualizing and understanding convolutional networks." *ECCN*. Springer, Cham, 2014.

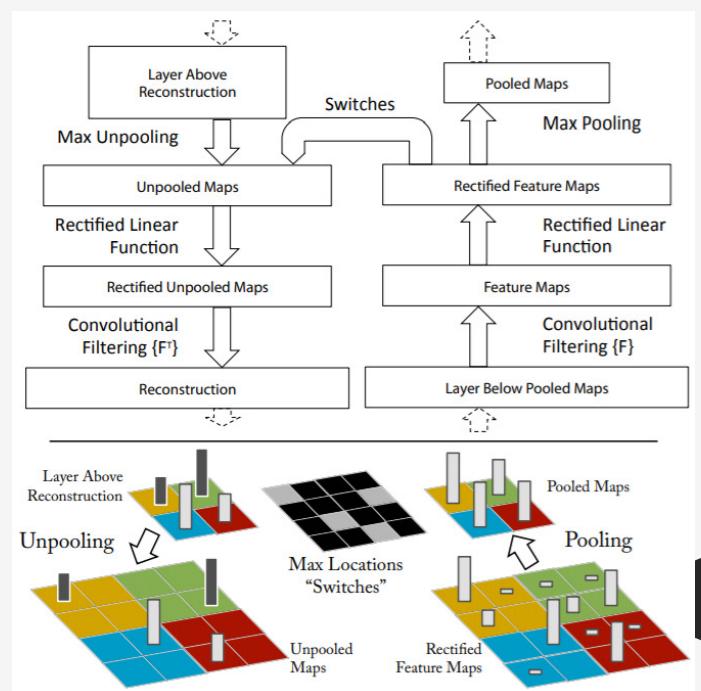


65

Visualisation des CNN

- **Unpooling**
- Enregistrer les emplacements des maxima.
- **Rectification**
- ReLu non-linearity
- **Filtering**
- Convolution transposée

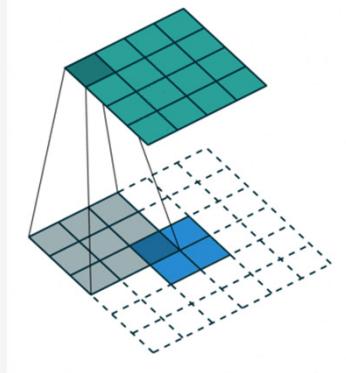
https://github.com/vdumoulin/conv_arithmetic



66

Visualisation des CNN

- Filtering
- Convolution transposée
- Pas d'apprentissage des filtres
- Appliquer sur les rectifications
- Le filtre est transposé:
 - Retournement vertical
 - Retournement horizontal
- Il s'agit d'une opération de convolution!



https://github.com/vdumoulin/conv_arithmetic

67

Au final

- De ANN à CNN pour la description automatique du contenu visuel.
- Notion de convolution pour capter des patterns locaux dans l'image.
- Entraîner un CNN => apprendre les poids des filtres convolutifs.
- Un CNN procède par une représentation hiérarchique du contenu visuel
 - Caractéristiques du bas niveau (contours, coins,...)
 - Caractéristiques des parties des objets présents dans l'image
 - Caractéristiques à sens sémantique pour chaque catégorie (classe)

68

Références

- <https://towardsdatascience.com/image-classification-in-10-minutes-with-mnist-dataset-54c35b77a38d>
- <http://scs.ryerson.ca/~aharley/vis/conv/flat.html>
- <https://medium.com/@ageitgey/machine-learning-is-fun-part-3-deep-learning-and-convolutional-neural-networks-f40359318721>
- http://mlss.tuebingen.mpg.de/2015/slides/fergus/Fergus_1.pdf
- <http://cs231n.github.io/assets/conv-demo/index.html>
- <http://cs231n.github.io/convolutional-networks/>
- <https://uijwalkarn.me/2016/08/11/intuitive-explanation-convnets/>
- <http://neuralnetworksanddeeplearning.com/chap1.html>
- <http://cedric.cnam.fr/vertigo/Cours/ml2/preamble.html>
- <http://cs231n.github.io/convolutional-networks/#case>
- <https://www.learnopencv.com/number-of-parameters-and-tensor-sizes-in-convolutional-neural-network/>
- <https://medium.com/@RaghavPrabhu/cnn-architectures-lenet-alexnet-vgg-googlenet-and-resnet-7c81c017b848>
- <https://towardsdatascience.com/illustrated-10-cnn-architectures-95d78ace614d#a253>
- <https://www.youtube.com/watch?v=c1RBQzKsDCk&list=PLkDaE6sCZn6Gl29AoE31iwdVwSG-KnDzF&index=17&t=0s&pbjreload=10>

69

Références

- <http://cs231n.github.io/convolutional-networks/#case>
- <https://www.learnopencv.com/number-of-parameters-and-tensor-sizes-in-convolutional-neural-network/>
- <https://medium.com/@RaghavPrabhu/cnn-architectures-lenet-alexnet-vgg-googlenet-and-resnet-7c81c017b848>
- <https://towardsdatascience.com/illustrated-10-cnn-architectures-95d78ace614d#a253>
- <https://www.youtube.com/watch?v=c1RBQzKsDCk&list=PLkDaE6sCZn6Gl29AoE31iwdVwSG-KnDzF&index=17&t=0s&pbjreload=10>
- <https://www.pyimagesearch.com/2019/07/08/keras-imagedatagenerator-and-data-augmentation/>
- <https://machinelearningmastery.com/how-to-configure-image-data-augmentation-when-training-deep-learning-neural-networks/>
- <https://keras.io/preprocessing/image/>
- <https://towardsdatascience.com/a-comprehensive-hands-on-guide-to-transfer-learning-with-real-world-applications-in-deep-learning-212bf3b2f27a>
- <http://cs231n.github.io/transfer-learning/>
- <https://towardsdatascience.com/what-is-deep-transfer-learning-and-why-is-it-becoming-so-popular-91acdcc2717a>
- <https://towardsdatascience.com/a-comprehensive-hands-on-guide-to-transfer-learning-with-real-world-applications-in-deep-learning-212bf3b2f27a>
- <http://cedric.cnam.fr/vertigo/Cours/ml2/docs/coursDeep5.pdf>

70