

Réseaux de neurones profonds

Guillaume Bourmaud

PLAN

I. CNN profond

II. Modèles de fondation

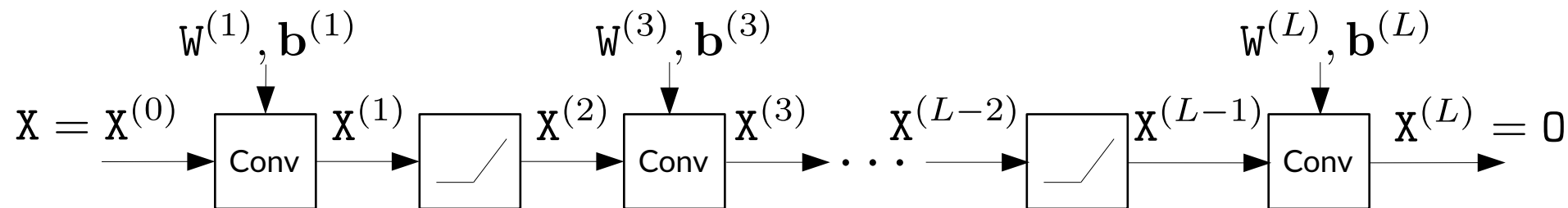
I) CNN profond

Rappel des ingrédients du « Deep Learning »

- 1) Grande base de données étiquetées
- 2) Grande capacité de calculs en parallèle (GPU)
- 3) « Bonne » architecture de réseau de neurones profond

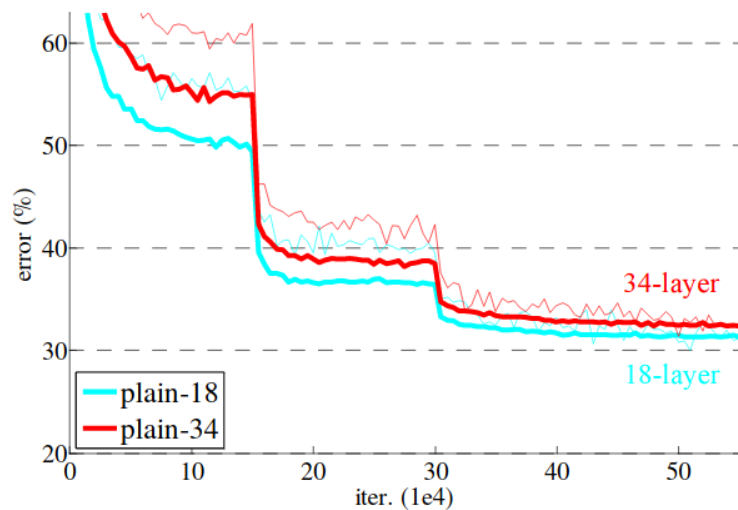
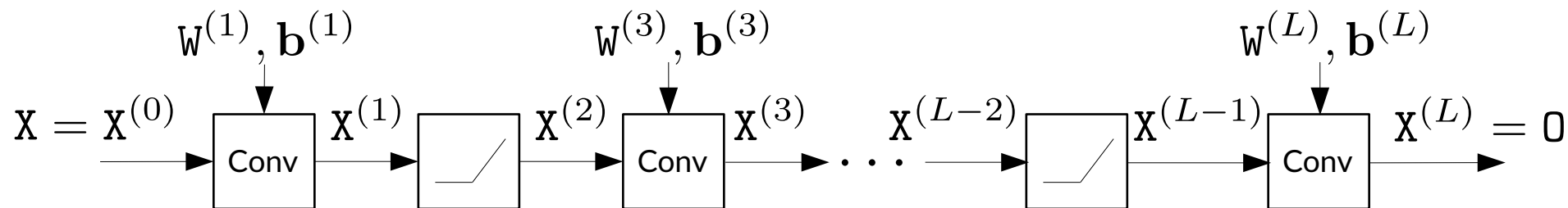
1)

Limites du CNN « classique »



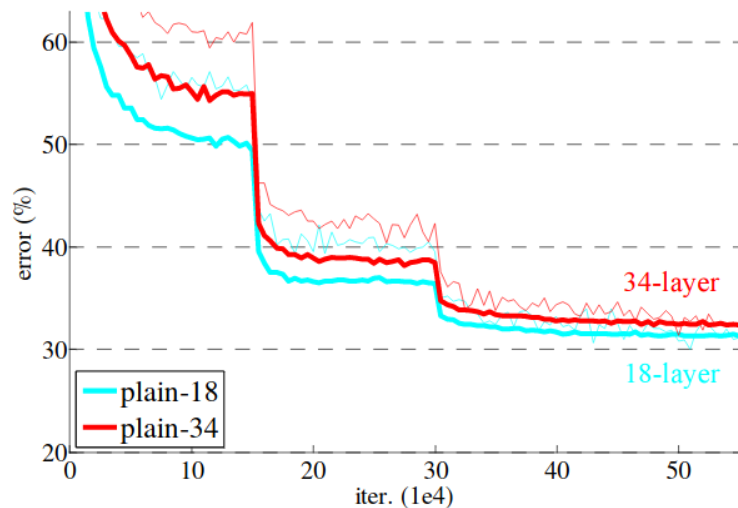
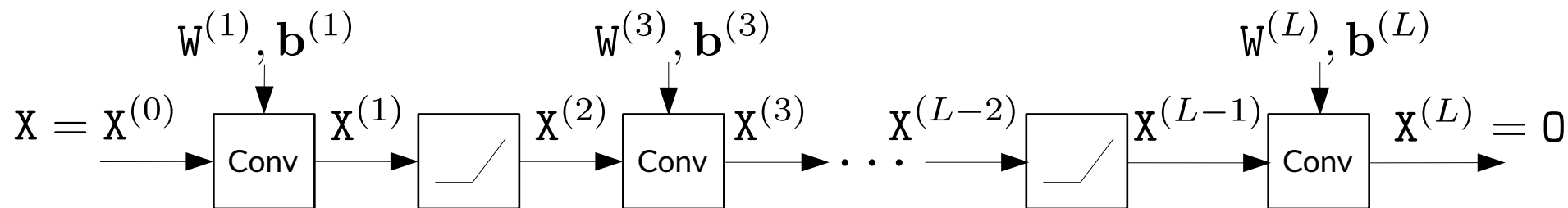
1)

Limites du CNN « classique »



1)

Limites du CNN « classique »



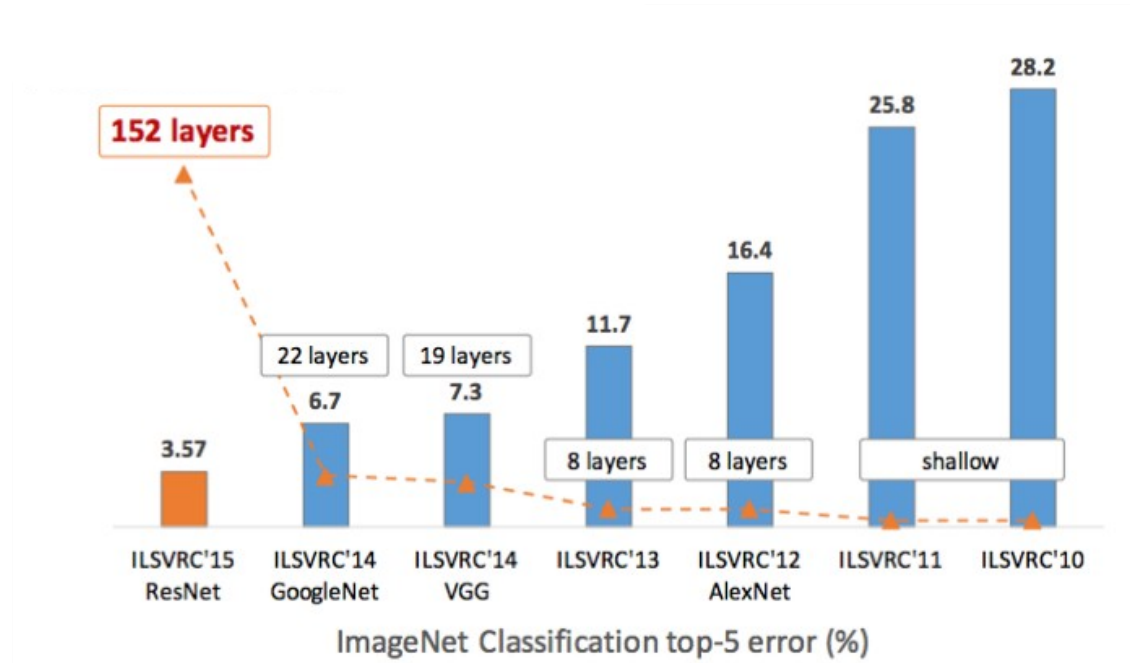
Ingrédient limitant les performances



Architecture du CNN

1)

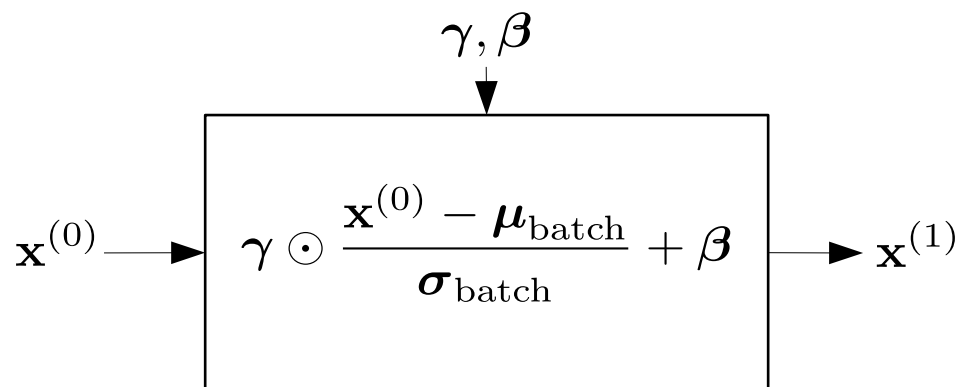
Limites du CNN « classique »



Source : <https://medium.com/@Lidinwise/the-revolution-of-depth-fac174924f5>

1)

Couche de “Batch Normalization”



Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1\dots m}\}$;

Parameters to be learned: γ, β

Output: $\{y_i = \text{BN}_{\gamma, \beta}(x_i)\}$

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m} \sum_{i=1}^m x_i \quad // \text{ mini-batch mean}$$

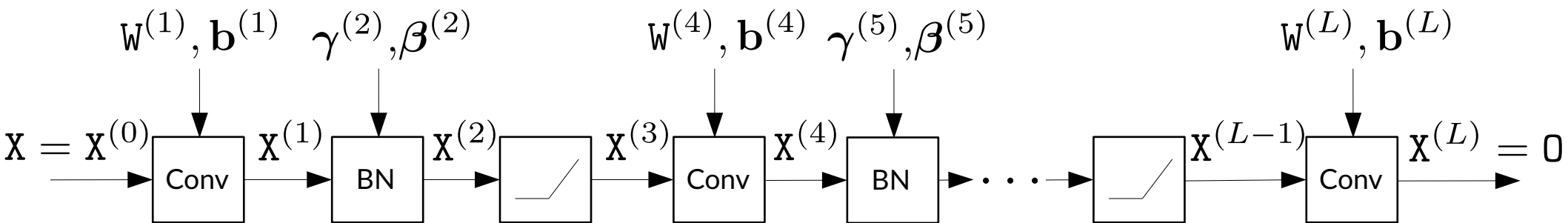
$$\sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m} \sum_{i=1}^m (x_i - \mu_{\mathcal{B}})^2 \quad // \text{ mini-batch variance}$$

$$\hat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} \quad // \text{ normalize}$$

$$y_i \leftarrow \gamma \hat{x}_i + \beta \equiv \text{BN}_{\gamma, \beta}(x_i) \quad // \text{ scale and shift}$$

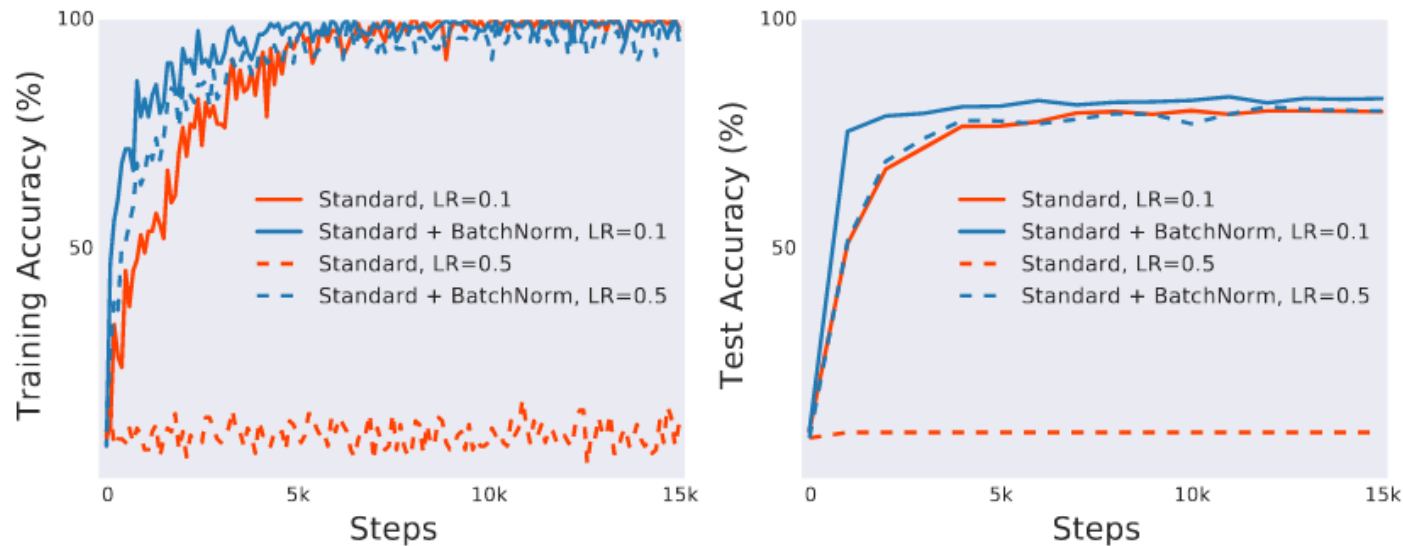
1)

CNN + BN



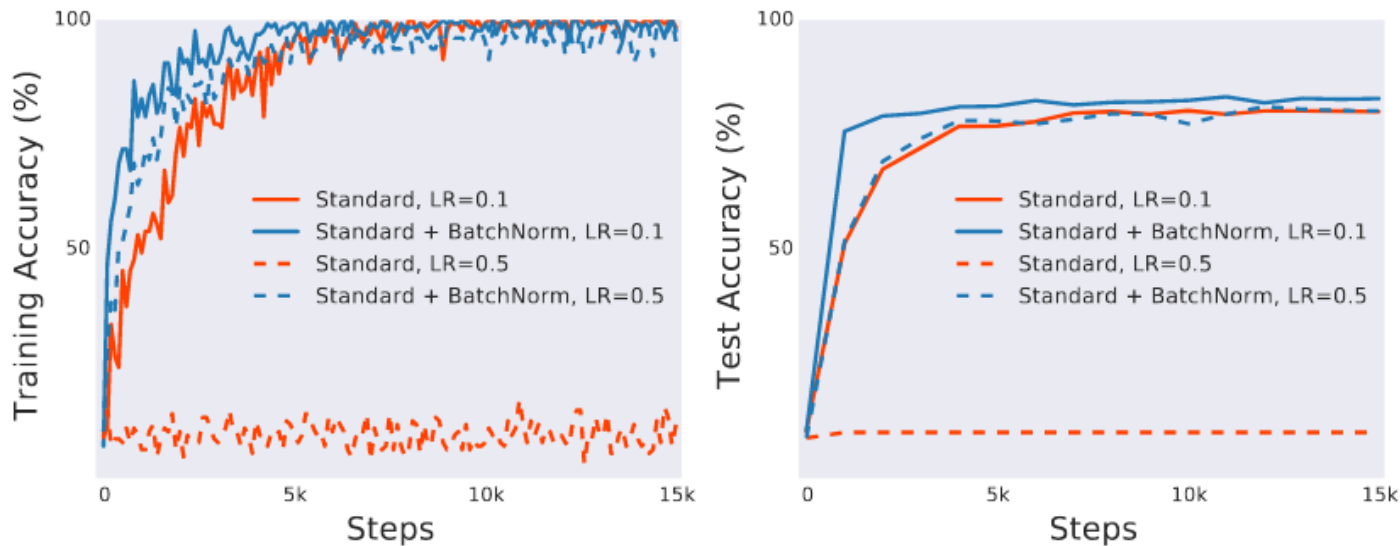
I)

CNN + BN (suite)



I)

CNN + BN (suite)



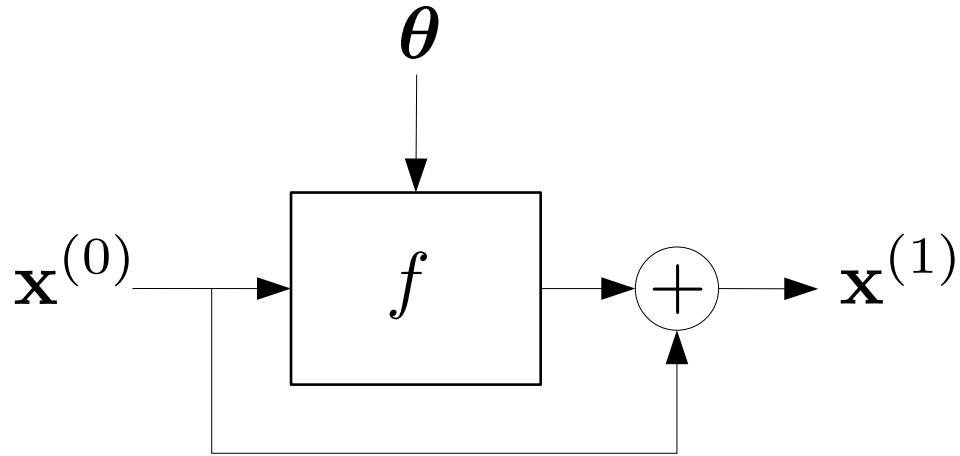
Rend le problème d'optimisation plus « lisse » :

→ Initialisation des paramètres moins critique

→ Possibilité d'utiliser un plus grand pas d'apprentissage

l)

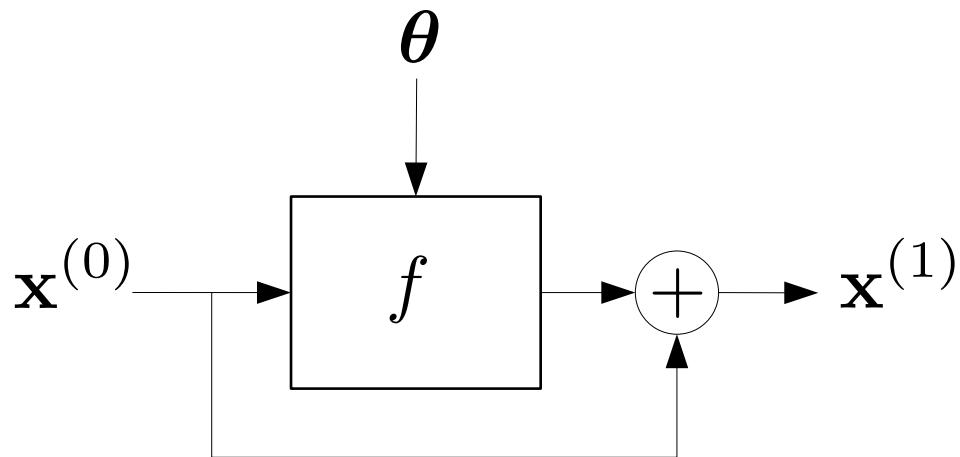
Connexion résiduelle



$$\mathbf{x}^{(1)} = \mathbf{x}^{(0)} + f(\mathbf{x}^{(0)}; \theta)$$

l)

Connexion résiduelle



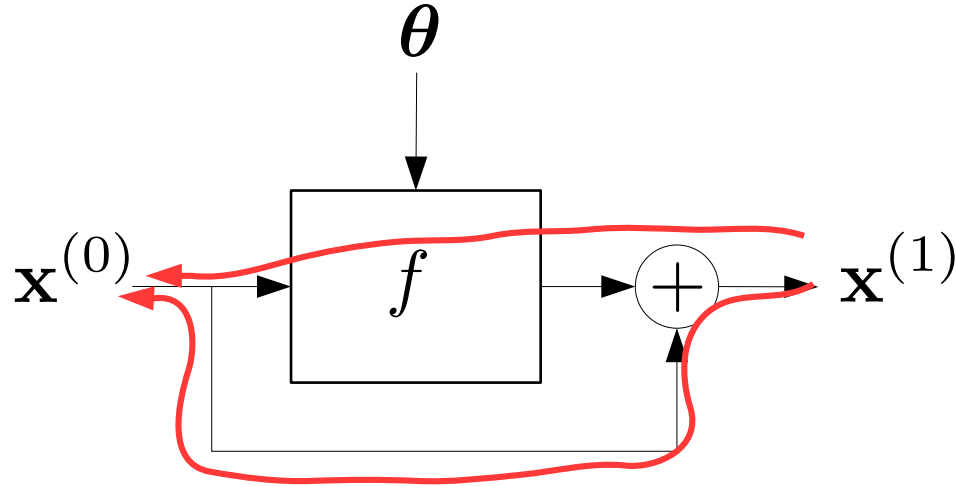
$$\mathbf{x}^{(1)} = \mathbf{x}^{(0)} + f(\mathbf{x}^{(0)}; \boldsymbol{\theta})$$

Rend la fonction plus « linéaire » :

→ Réduit sa « capacité » → augmentation du nombre de couches pour un même résultat

l)

Connexion résiduelle (suite)



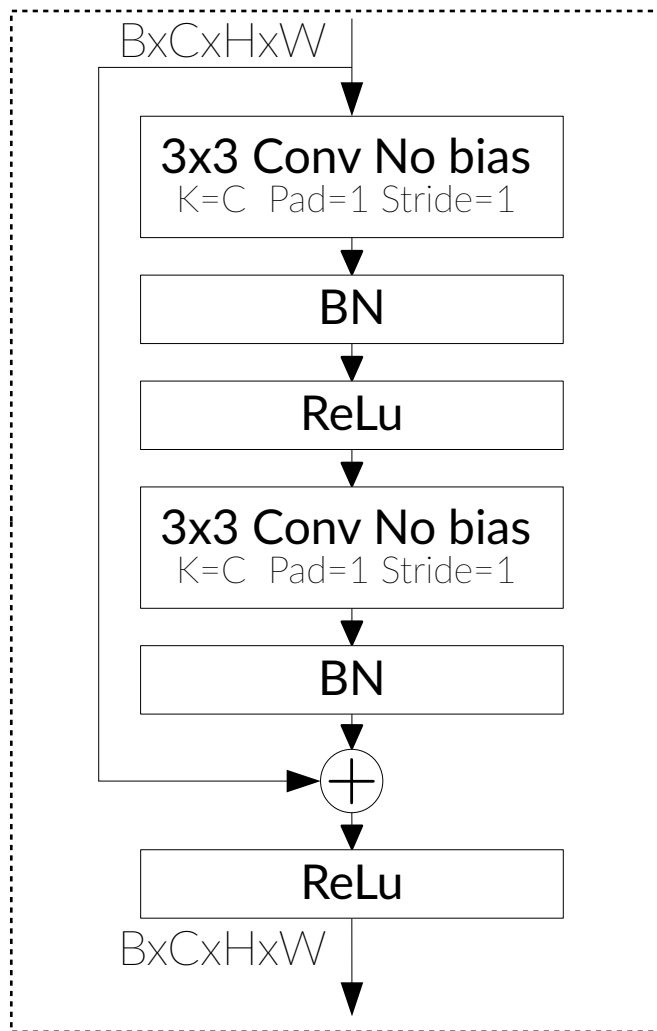
$$\frac{\partial \mathbf{x}^{(1)}}{\partial \mathbf{x}^{(0)}} = \mathbf{I} + \frac{\partial f(\mathbf{x}^{(0)}; \boldsymbol{\theta})}{\partial \mathbf{x}^{(0)}}$$

Rend la fonction plus « linéaire » :

→ Réduit sa « capacité » → augmentation du nombre de couches pour un même résultat

→ Facilite la propagation du gradient → plus de couches conduit à de meilleurs résultats (en théorie)

1)



ResBlock A

Transforme le tenseur d'entrée

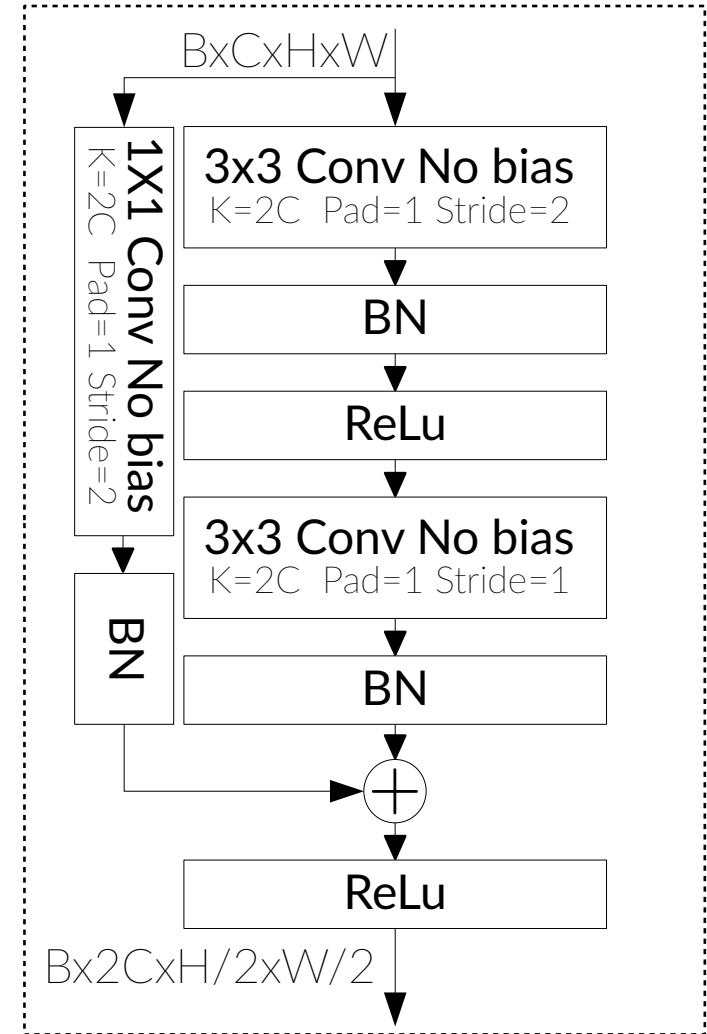
- en préservant la résolution
- et en préservant le nombre de canaux

I)

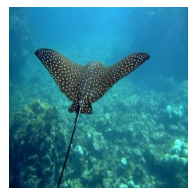
ResBlock B

Transforme le tenseur d'entrée

- en divisant la résolution par 2
- et en augmentant le nombre de canaux par 2



1)

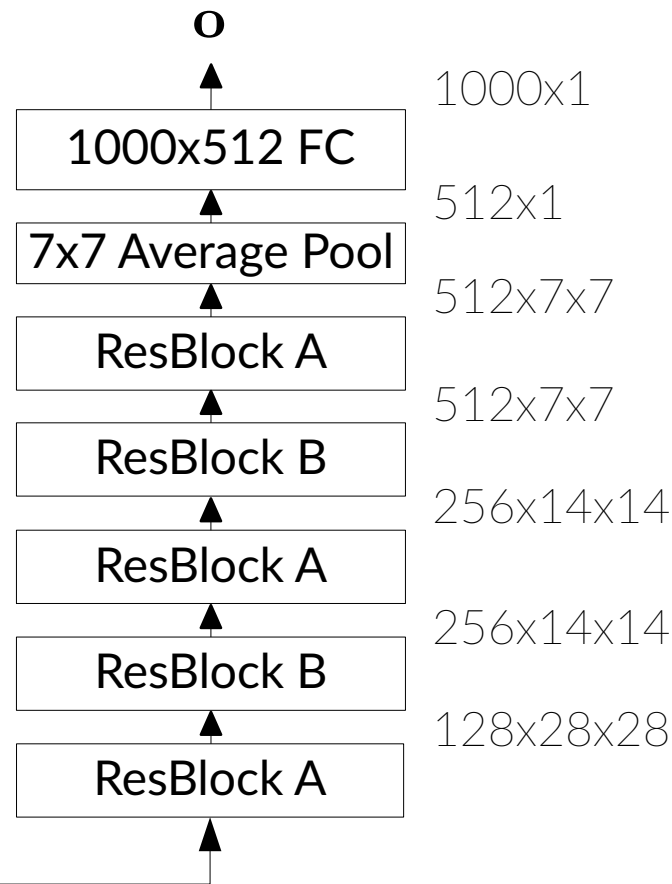
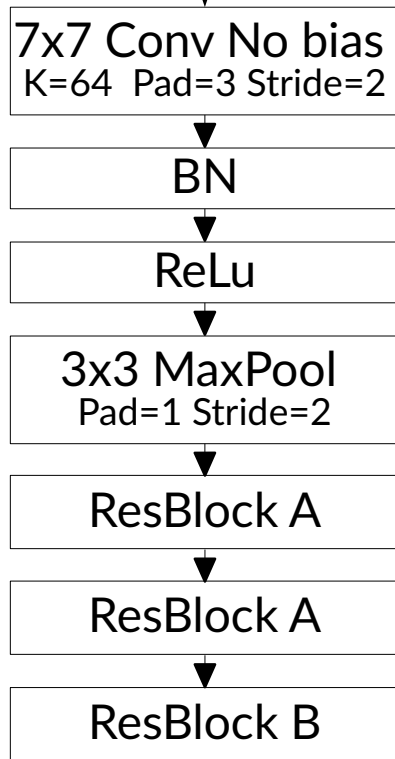


ResNet 18

$\arg \max(\mathbf{o}) = 748 \doteq \text{"raie"}$

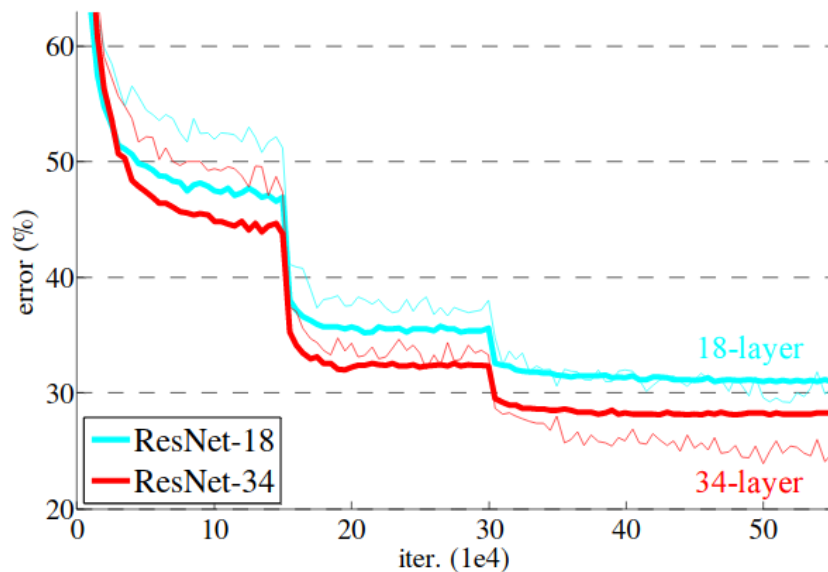
Passage
en grande
dimension

3x224x224
64x112x112
64x112x112
64x112x112
64x56x56
64x56x56
64x56x56
128x28x28



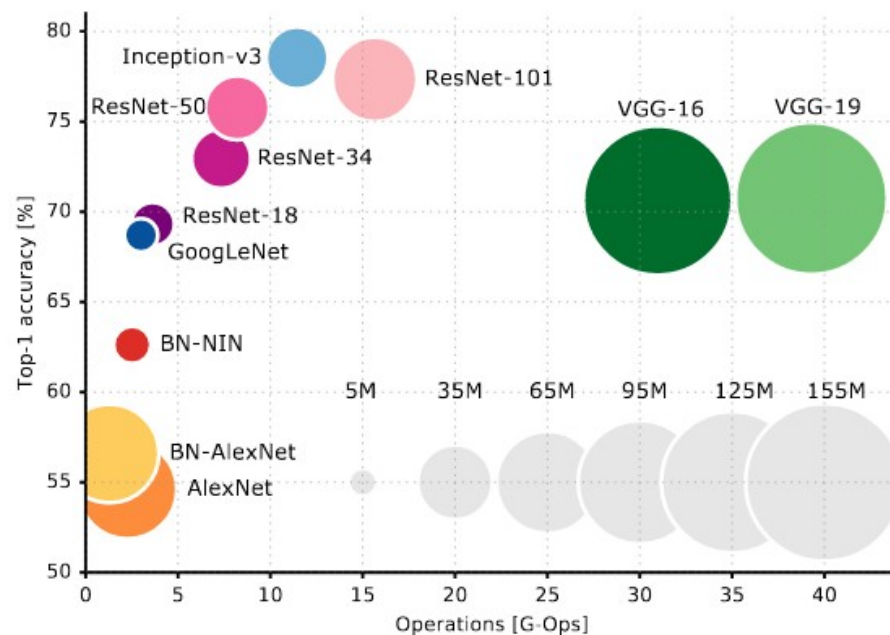
1)

ResNet 18 < ResNet 34



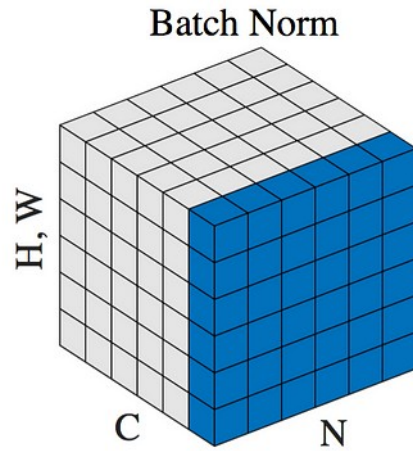
1)

Précision vs Nombre de paramètres vs Nombre d'opérations



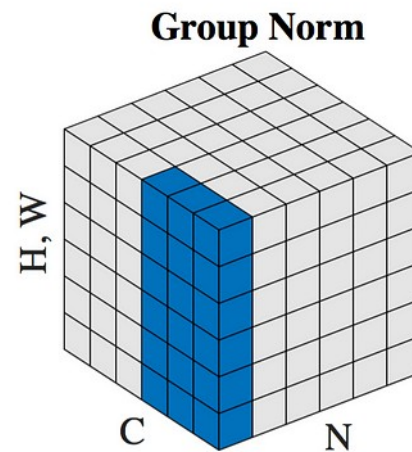
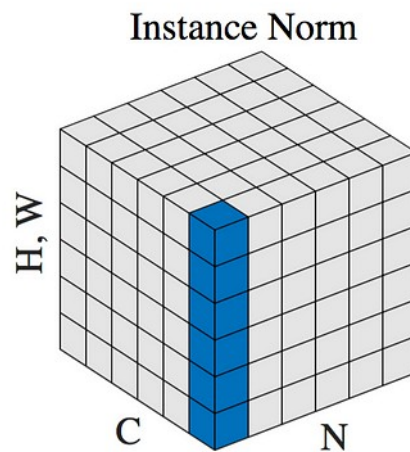
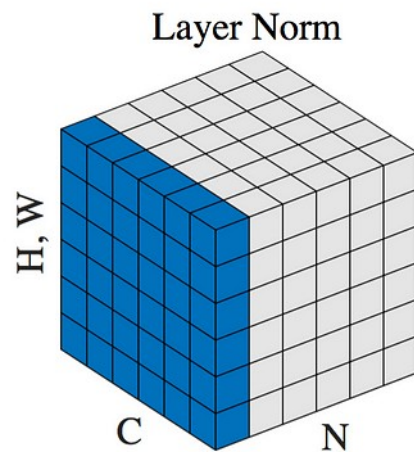
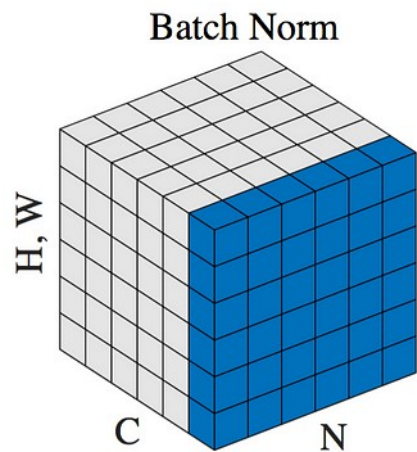
1)

Différentes couches de normalisation



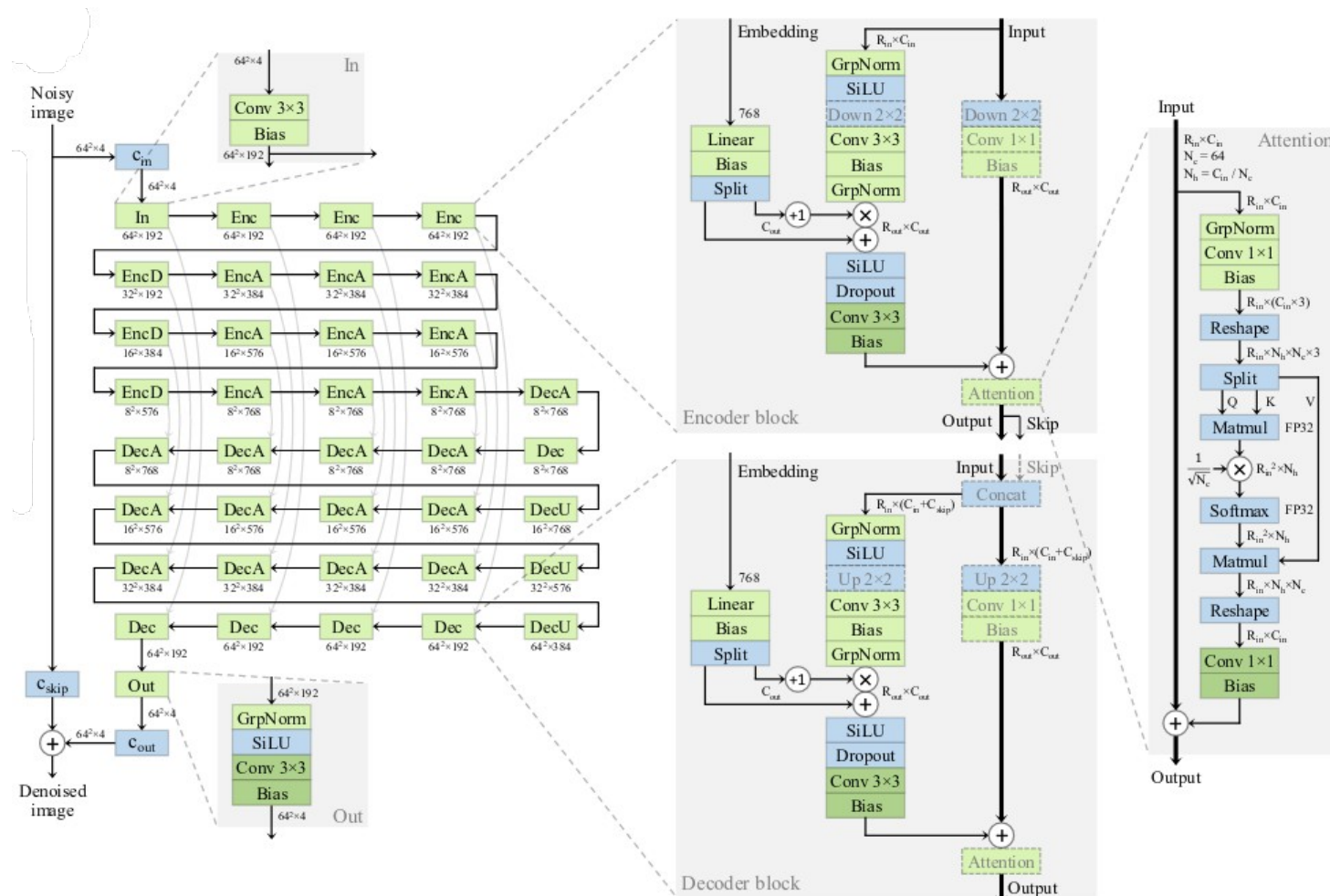
1)

Différentes couches de normalisation



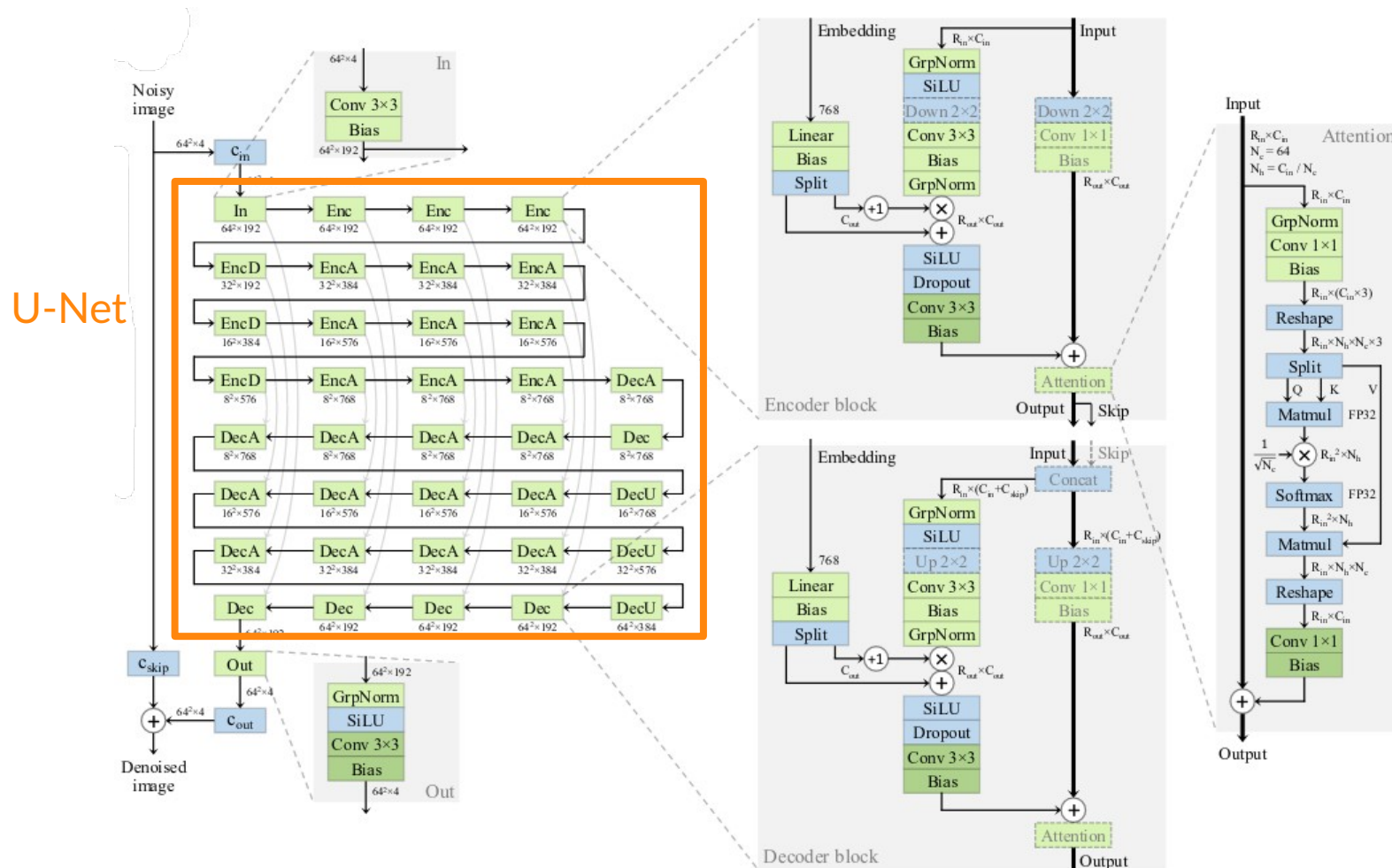
1)

En 2024 : U-Net + ResBlock toujours là !



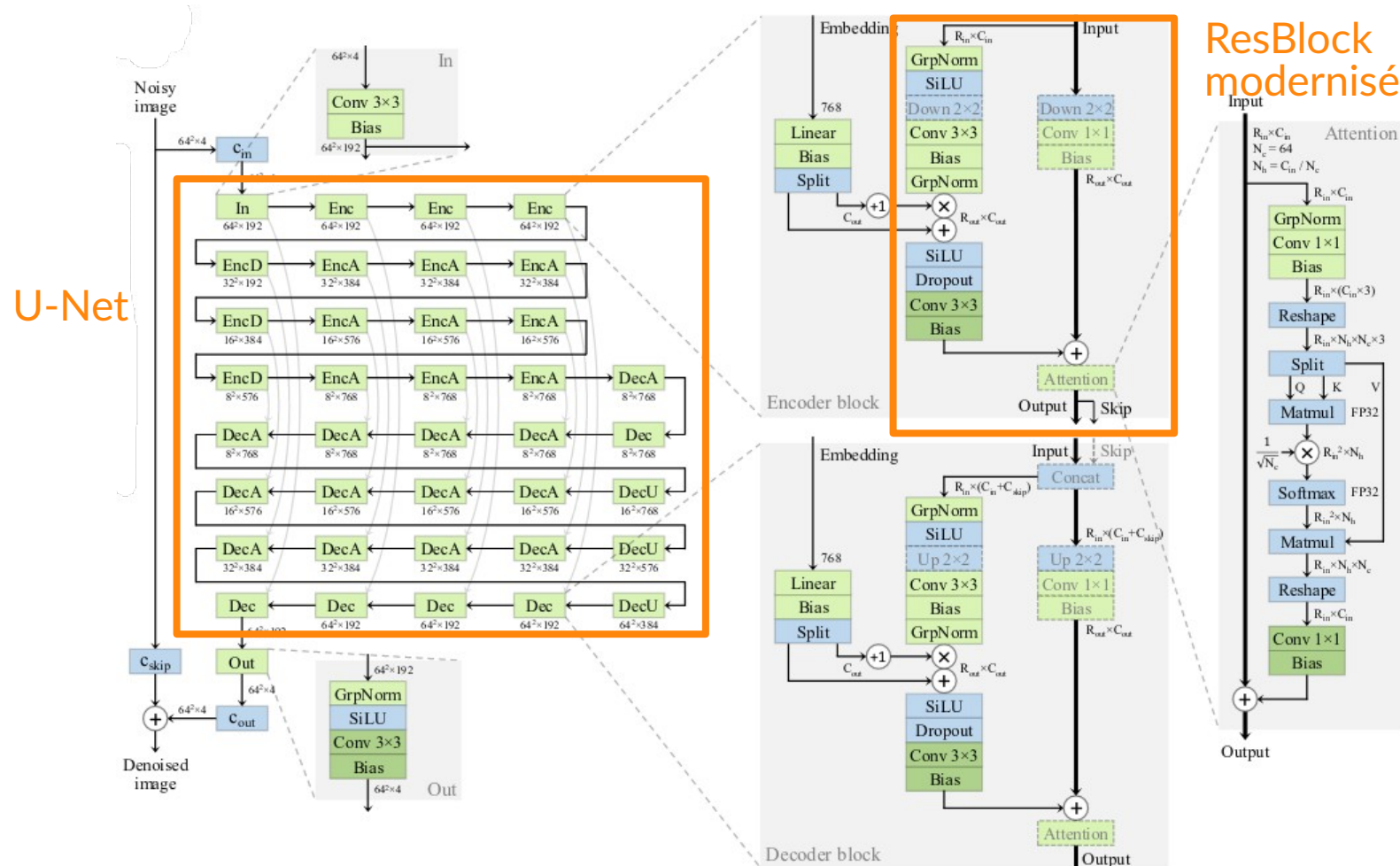
1)

En 2024 : U-Net + ResBlock toujours là !



1)

En 2024 : U-Net + ResBlock toujours là !



II) Modèles de fondation

II)

Comment faire avec une petite base de données étiquetées ?

Exemple : détection du frelon asiatique



Présence (1042 images)



Absence (1844 images)

Ingrédient limitant les performances

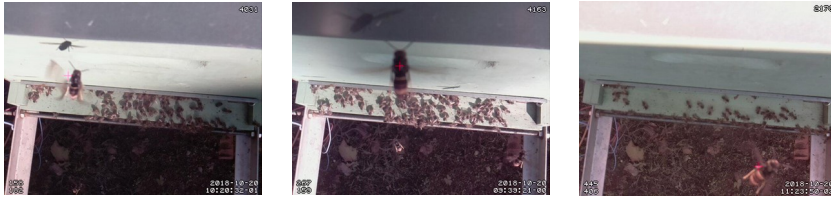


Taille de la base de données étiquetées

II)

Comment faire avec une petite base de données étiquetées ?

Exemple : détection du frelon asiatique



Présence (1042 images)



Absence (1844 images)

Ingrédient limitant les performances



Taille de la base de données étiquetées

Solution

1) Récupérer un modèle de fondation

2) Spécialiser ce modèle de fondation sur sa petite base de données étiquetées

II)

Modèle de fondation « historique »

Historiquement (~ à partir de 2012), un modèle de fondation (le terme « foundation model » n'est introduit qu'en 2021) est :

Modèle de fondation « historique »

Historiquement (~ à partir de 2012), un modèle de fondation (le terme « foundation model » n'est introduit qu'en 2021) est :

- un CNN (e.g ResNet)

Modèle de fondation « historique »

Historiquement (~ à partir de 2012), un modèle de fondation (le terme « foundation model » n'est introduit qu'en 2021) est :

- un CNN (e.g ResNet)
- qui est **pré-entraîné** = entraîné sur ImageNet1k

Rappel : ImageNet1k = 1.2M d'images étiquetées sur 1000 classes qui représentent une grande diversité d'images **issues de notre monde**



II) Exemple de spécialisation (« fine-tuning ») d'un ResNet 18 pré-entraîné sur ImageNet1k

Image « normalisée »



3x224x224

64x112x112

64x112x112

64x112x112

64x56x56

64x56x56

64x56x56

128x28x28

7x7 Conv No bias
K=64 Pad=3 Stride=2

BN

ReLu

3x3 MaxPool
Pad=1 Stride=2

ResBlock A

ResBlock A

ResBlock B

$\arg \max(o) = 2 \doteq \text{''Présence''}$

2x512 FC

7x7 Average Pool

ResBlock A

ResBlock B

ResBlock A

ResBlock B

ResBlock A

2x1

512x1

512x7x7

512x7x7

256x14x14

256x14x14

128x28x28

Modèles de fondation « modernes »

Pré-entraînement sur une **très** grande base de données.

Exemple : ~2012 ImageNet 1k = 1.2×10^6 images étiquetées
~2022 LAION-5B = 5×10^9 images étiquetées

Modèles de fondation « modernes »

Pré-entraînement sur une **très** grande base de données.

Exemple : ~2012 ImageNet 1k = 1.2×10^6 images étiquetées
~2022 LAION-5B = 5×10^9 images étiquetées

Rendu possible grâce à :

- l'augmentation continue de la puissance des cartes graphiques
- l'apparition des couches d'attention (« Transformer »)

Modèles de fondation « modernes »

Pré-entraînement sur une **très** grande base de données.

Exemple : ~2012 ImageNet 1k = 1.2×10^6 images étiquetées
~2022 LAION-5B = 5×10^9 images étiquetées

Rendu possible grâce à :

- l'augmentation continue de la puissance des cartes graphiques
- l'apparition des couches d'attention (« Transformer »)

Apparition de modèles de fondation dans un grand nombre de domaines :

- Imagerie médicale
- Imagerie astronomique
- Etc.

Modèles de fondation « modernes »

Pré-entraînement sur une **très** grande base de données.

Exemple : ~2012 ImageNet 1k = 1.2×10^6 images étiquetées
~2022 LAION-5B = 5×10^9 images étiquetées

Rendu possible grâce à :

- l'augmentation continue de la puissance des cartes graphiques
- l'apparition des couches d'attention (« Transformer »)

Apparition de modèles de fondation dans un grand nombre de domaines :

- Imagerie médicale
- Imagerie astronomique
- Etc.
- Et bien-sûr les LLM (« Large Language Model ») avec GPT !

II) Exemple de modèle de fondation pour la segmentation d'images

« Segment Anything »

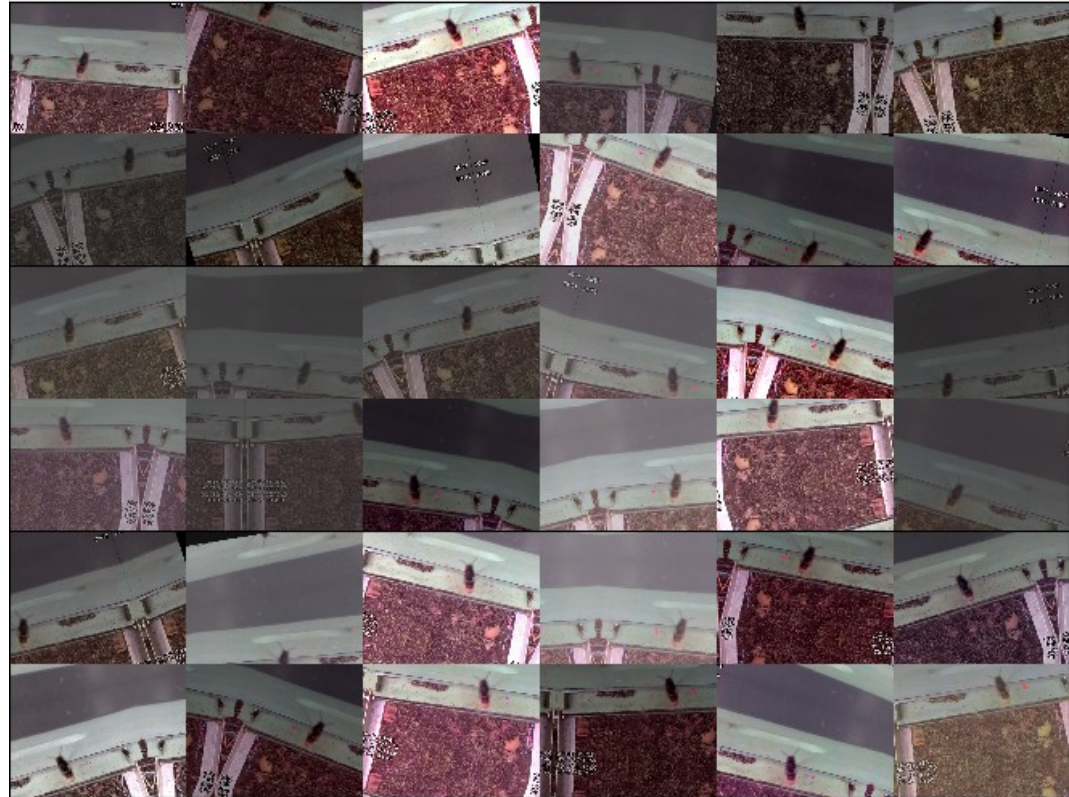


Entraînement sur « SA-1B » : une base de 11×10^6 images avec 1.1×10^9 masques de segmentation

Annexes

Augmentation de données

- Mirroir
- Transformation affine
- Perturbation couleur
- Effacement
- Bruit
- ...



Cubuk et al. (2019). Autoaugment: Learning augmentation strategies from data. CVPR

Cubuk et al. (2020). Randaugment: Practical automated data augmentation with a reduced search space. CVPR

“Adversarial examples”



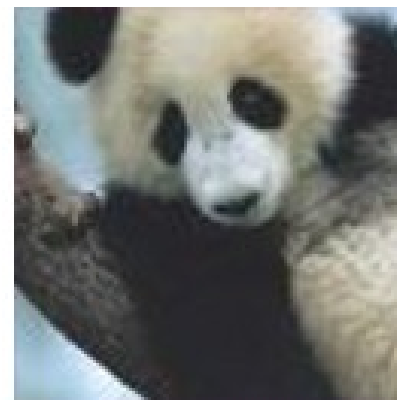
“panda”

57.7% confidence

+ ϵ



=

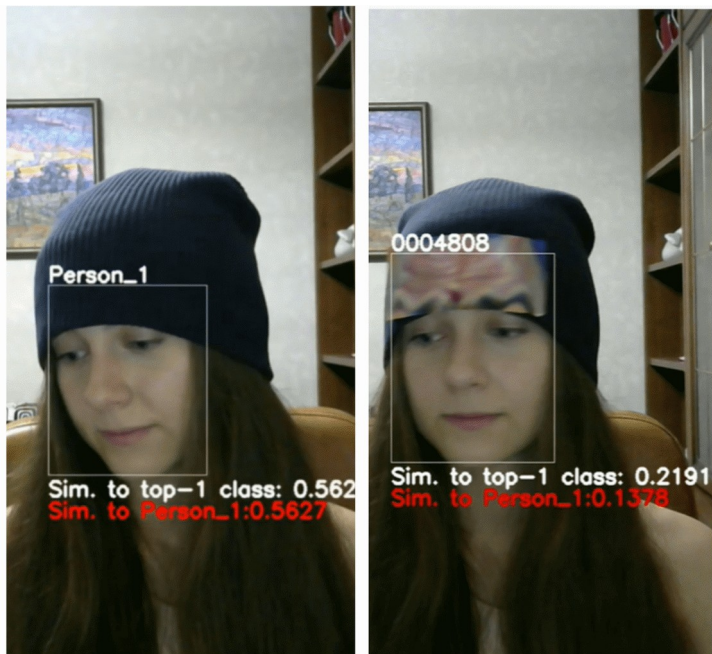


“gibbon”

99.3% confidence

Source : <https://openai.com/blog/adversarial-example-research/>

“Adversarial patches”



Sources : “ADVHAT: Real-world adversarial attack on ArcFace face ID system” (<https://arxiv.org/pdf/1908.08705.pdf>)
“Robust Physical-World Attacks on Deep Learning Visual Classification” (<https://arxiv.org/pdf/1707.08945.pdf>)

Annexe : Application à la détection d'objets dans une image

Détection d'objets

Classification



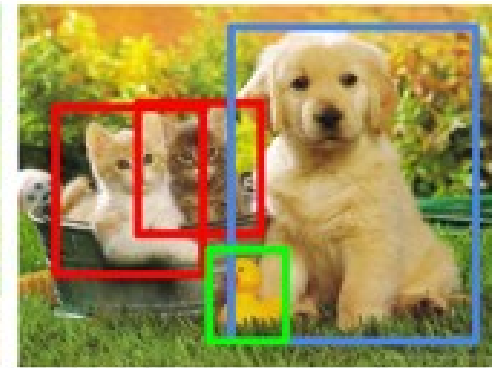
CAT

**Classification
+ Localization**



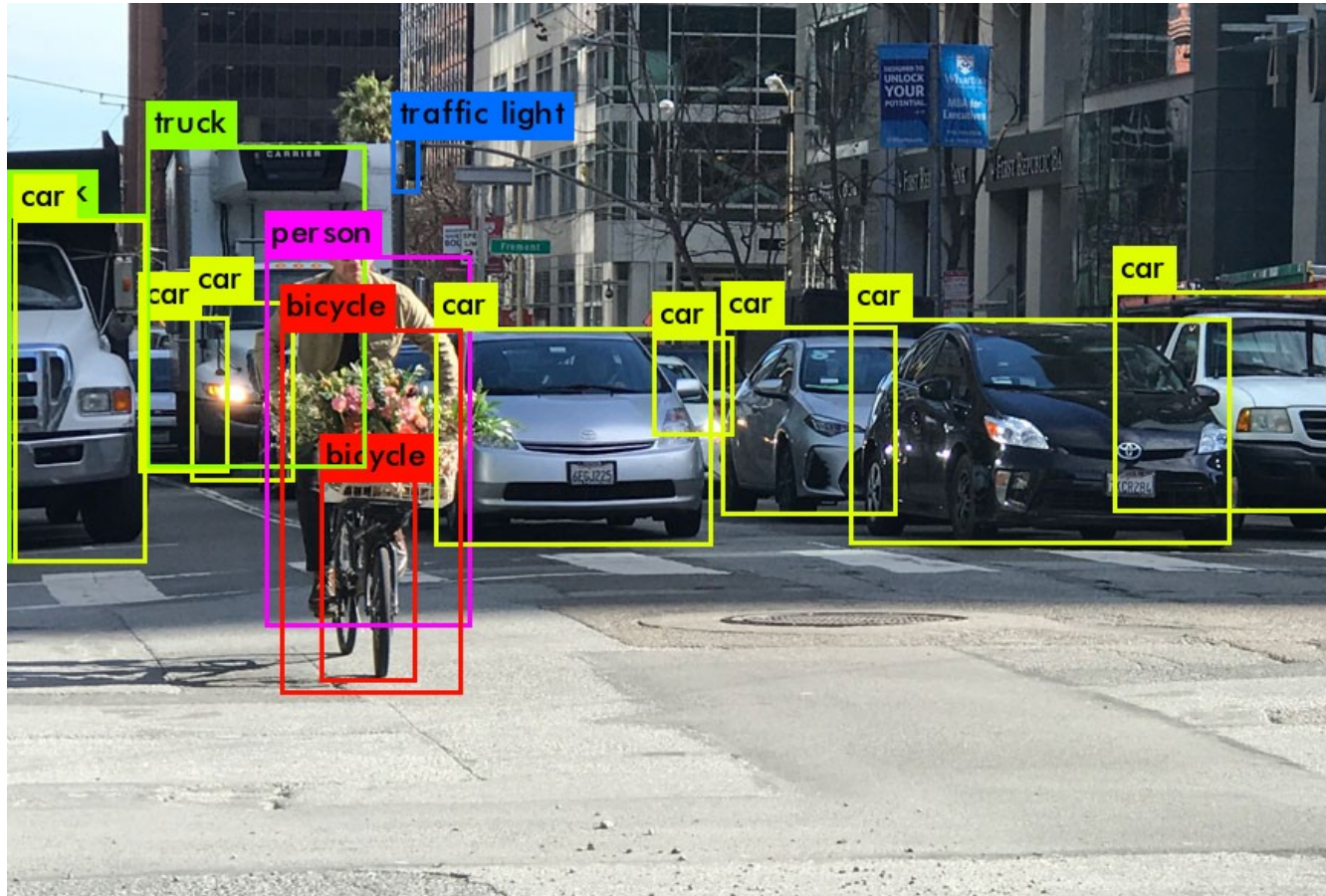
CAT

Object Detection



CAT, DOG, DUCK

Exemple de détection d'objets



Formulation du problème

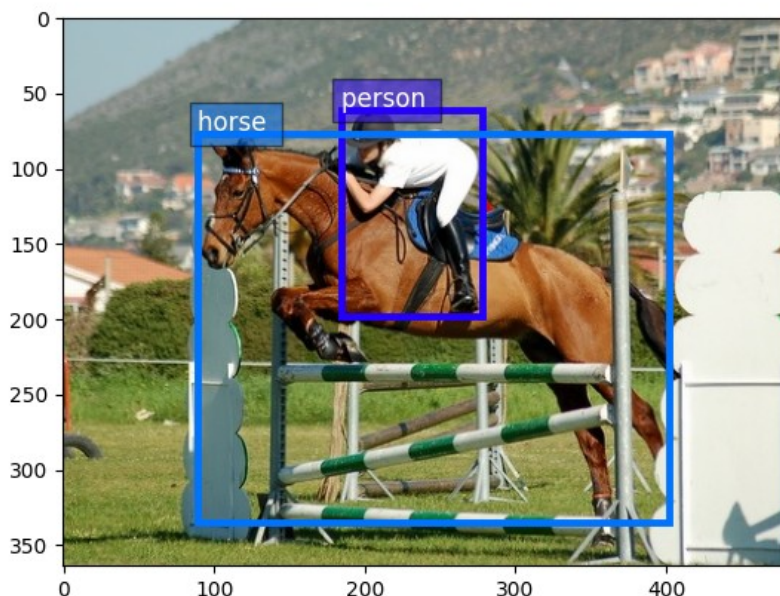
- Objectif
 - Prédire une boîte englobante autour de chaque objet,
 - Prédire la classe de l'objet,
 - En utilisant un réseau de neurones en apprentissage supervisé.
- Questions
 - Quelles données annotées disponibles ?
 - Quelle architecture ?
 - Quelle fonction de coût ?

Quelles données annotées disponibles ?

Microsoft COCO dataset (80 classes)

Pascal VOC dataset (20 classes)

1: 'person',	31: 'skis',	61: 'dining table',
2: 'bicycle',	32: 'snowboard',	62: 'toilet',
3: 'car',	33: 'sports ball',	63: 'tv',
4: 'motorcycle',	34: 'kite',	64: 'laptop',
5: 'airplane',	35: 'baseball bat',	65: 'mose',
6: 'bs',	36: 'baseball glove',	66: 'remote',
7: 'train',	37: 'skateboard',	67: 'keyboard',
8: 'trck',	38: 'srfboard',	68: 'cell phone',
9: 'boat',	39: 'tennis racket',	69: 'microwave',
10: 'traffic light',	40: 'bottle',	70: 'oven',
11: 'fire hydrant',	41: 'wine glass',	71: 'toaster',
12: 'stop sign',	42: 'cp',	72: 'sink',
13: 'parking meter',	43: 'fork',	73: 'refrigerator',
14: 'bench',	44: 'knife',	74: 'book',
15: 'bird',	45: 'spoon',	75: 'clock',
16: 'cat',	46: 'bowl',	76: 'vase',
17: 'dog',	47: 'banana',	77: 'scissors',
18: 'horse',	48: 'apple',	78: 'teddy bear',
19: 'sheep',	49: 'sandwich',	79: 'hair drier',
20: 'cow',	50: 'orange',	80: 'toothbrsh',
21: 'elephant',	51: 'broccoli',	
22: 'bear',	52: 'carrot',	
23: 'zebra',	53: 'hot dog',	
24: 'giraffe',	54: 'pizza',	
25: 'backpack',	55: 'dont',	
26: 'mbrella',	56: 'cake',	
27: 'handbag',	57: 'chair',	
28: 'tie',	58: 'coch',	
29: 'sitcase',	59: 'potted plant',	
30: 'frisbee',	60: 'bed',	



Person:
1: person

Animal:
2: bird
3: cat
4: cow
5: dog
6: horse
7: sheep

Vehicle:
8: aeroplane
9: bicycle
10: boat
11: bus
12: car
13: motorbike
14: train

Indoor:
15: bottle
16: chair
17: dining table
18: potted plant
19: sofa
20: tv/monitor

Comment prédire le nombre d'objets et pour chaque objet sa boîte et sa classe ?

Exemple de solution : CenterNet (Zhou et. al, Objects as points, 2019)



keypoint heatmap [C]

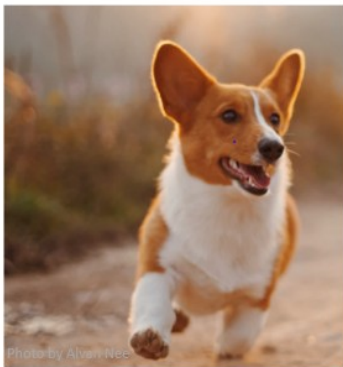


local offset [2]

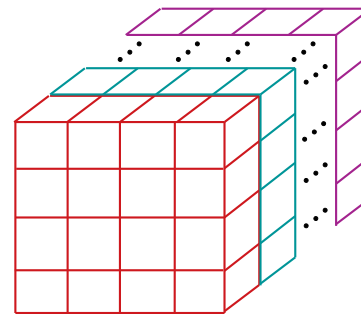


object size [2]

CenterNet



$3 \times H \times W$



$(C+2+2) \times H/4 \times W/4$

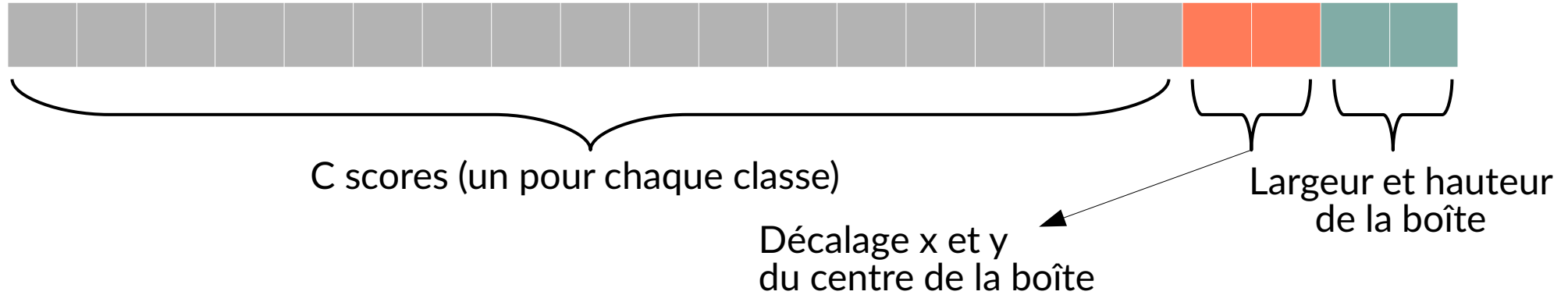


C scores (un pour chaque classe)

Décalage x et y
du centre de la boîte

Largeur et hauteur
de la boîte

CenterNet (suite)



Fonction de coût = somme de trois fonctions de coût

- Pour les scores : « pixelwise logistic regression » (i.e. sur chaque case grise)
- Pour le décalage : régression L1 (sur les cases oranges s'il y a une boîte sinon rien)
- Pour la largeur et la hauteur : régression L1 (sur les cases vertes s'il y a une boîte sinon rien)

CenterNet (suite)

