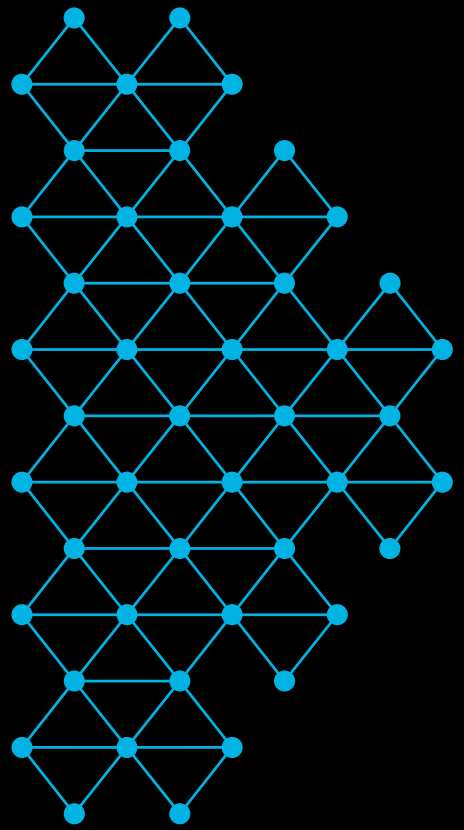
Institut
des algorithmes
d'apprentissage
de Montréal

Application : traitement d'images

Margaux Luck

École d'hiver francophone en apprentissage profond IVADO, MILA 7 mars 2018

Application 1 : sauver les baleines



Sauver les baleines avec l'apprentissage profond

Identification de baleines

https://www.fisheries.noaa.gov/welcome
https://www.kaggle.com/c/noaa-right-whale-recognition

Impact sur :

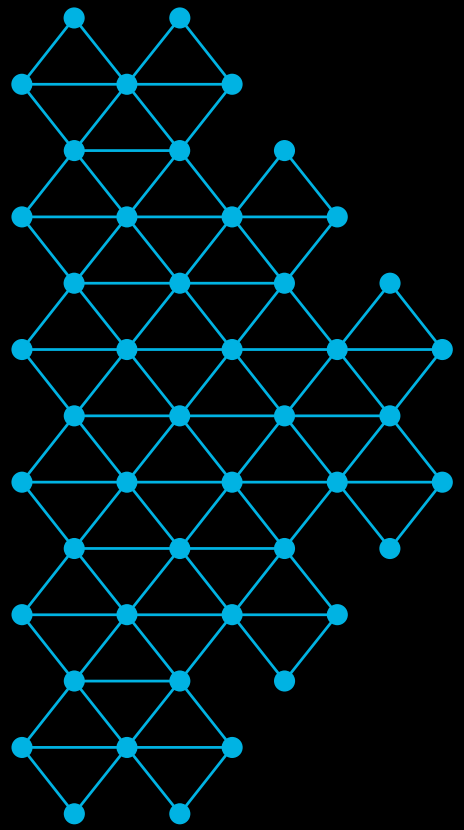
- **L'écologie**
  - Recensement
  - Localisation

- **L'économie**
  - Nouvelles technologies
  - Tourisme

Souffleurs d'Ecume

Hans Bernhard

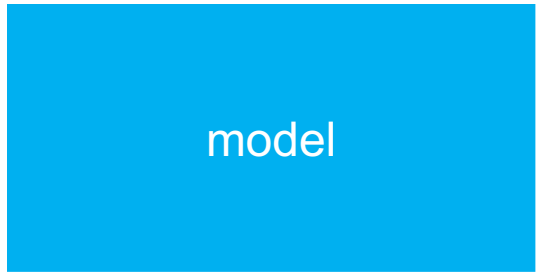
http://baleinesendirect.org/pourquoi-les-baleines-sont-elles-importantes-pourquoi-les-proteger/



Classification

Forme des données et tâche

Image d'entrée représentant l'une des classes



Probabilité d'appartenance aux différentes classes

Tâche : Classification = trouver l'appartenance à une classe

Jeux de données de références : CIFAR10, CIFAR100, MNIST, ImageNet, SVHN

Image source : https://www.kaggle.com/c/noaa-right-whale-recognition

Préparation des données

```
import torch
from torchvision import datasets
import torchvision.transforms as transforms

data_transforms = transforms.Compose([
    transforms.Resize(224),
    transforms.RandomHorizontalFlip(),
    transforms.ToTensor(),
    transforms.Normalize([0.485, 0.456, 0.406], [0.229, 0.224, 0.225])
    ])

data_dir = 'baleines/train/'

image_train = datasets.ImageFolder(data_dir, data_transforms)

dataloaders = torch.utils.data.DataLoader(
    image_train, batch_size=16, shuffle=True, num_workers=4)
```

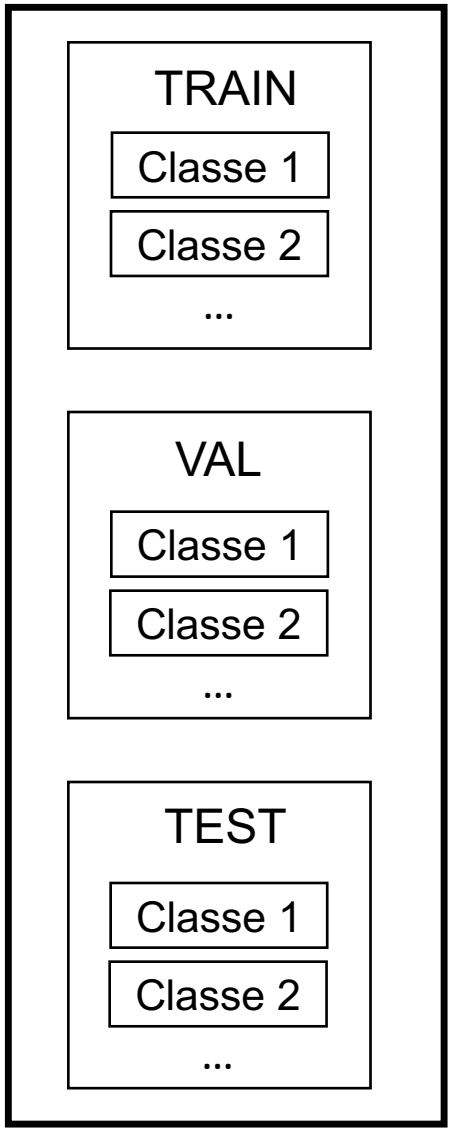
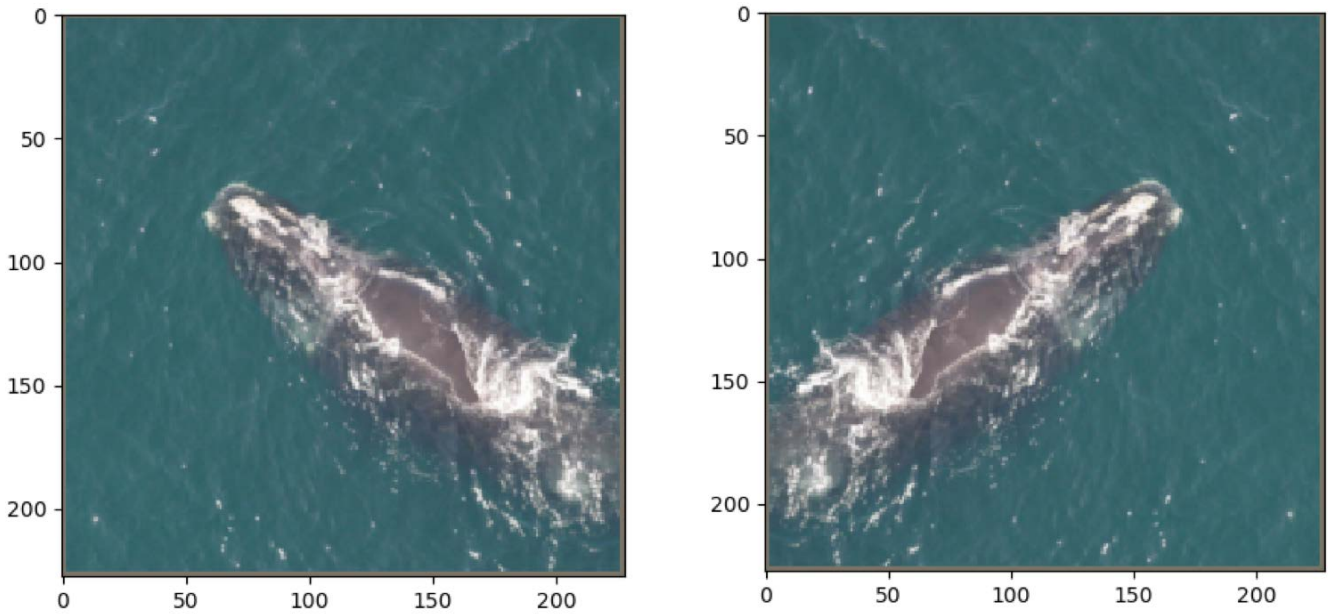


Image originale

Image transformée




```
inputs, classes = next(iter(dataloaders))

for data in dataloaders:
    inputs, classes = data
```

A retenir…

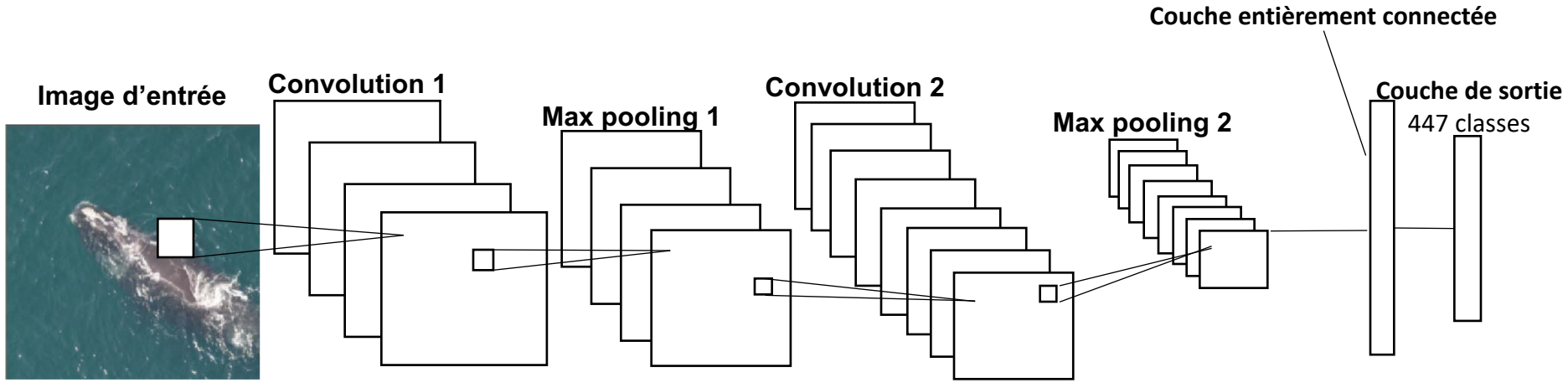
- La préparation des données est jeux de données dépendant
- Penser à normaliser les données
- De préférence utiliser des images de même taille pas trop grosses
- L’augmentation de donnée, ça n’est pas magique

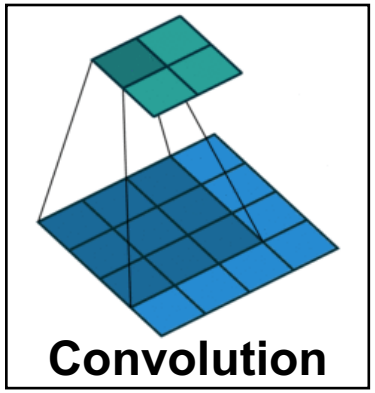
A retenir…

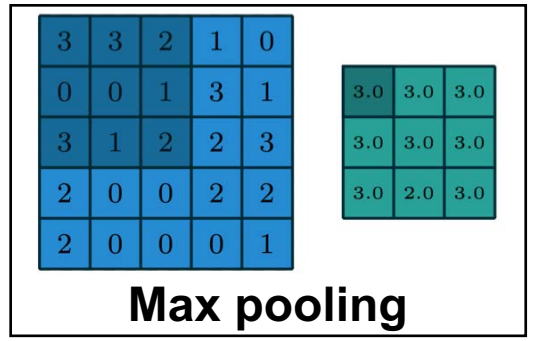
- **Difficulté des données :**
  - Image de tailles différentes
  - Exemple adversariaux
  - Objets coupés
  - Classes qui se ressembles
  - Peu ou pas d’exemple pour certaines des classes dans l’ensemble d’entraînement

Modèles utilisés classiquement

- **LeNet :**



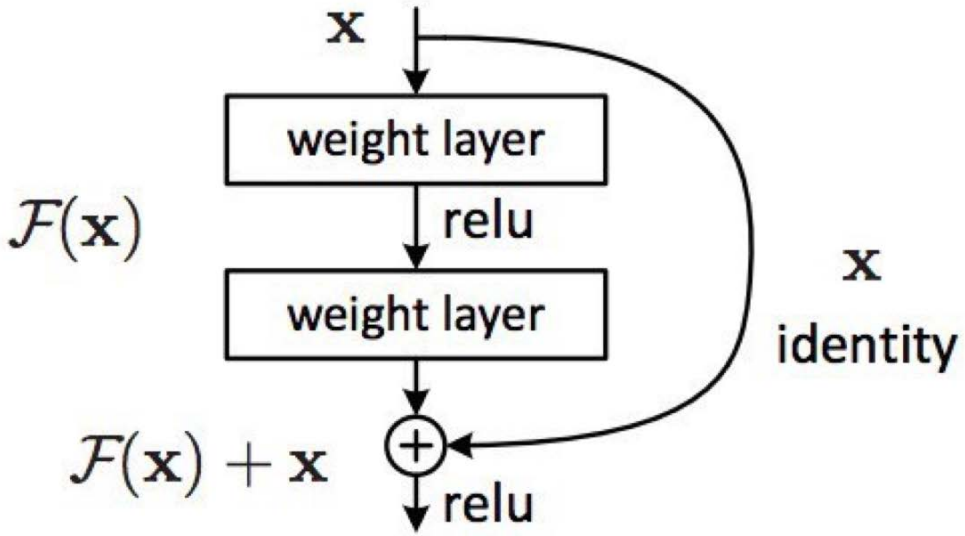




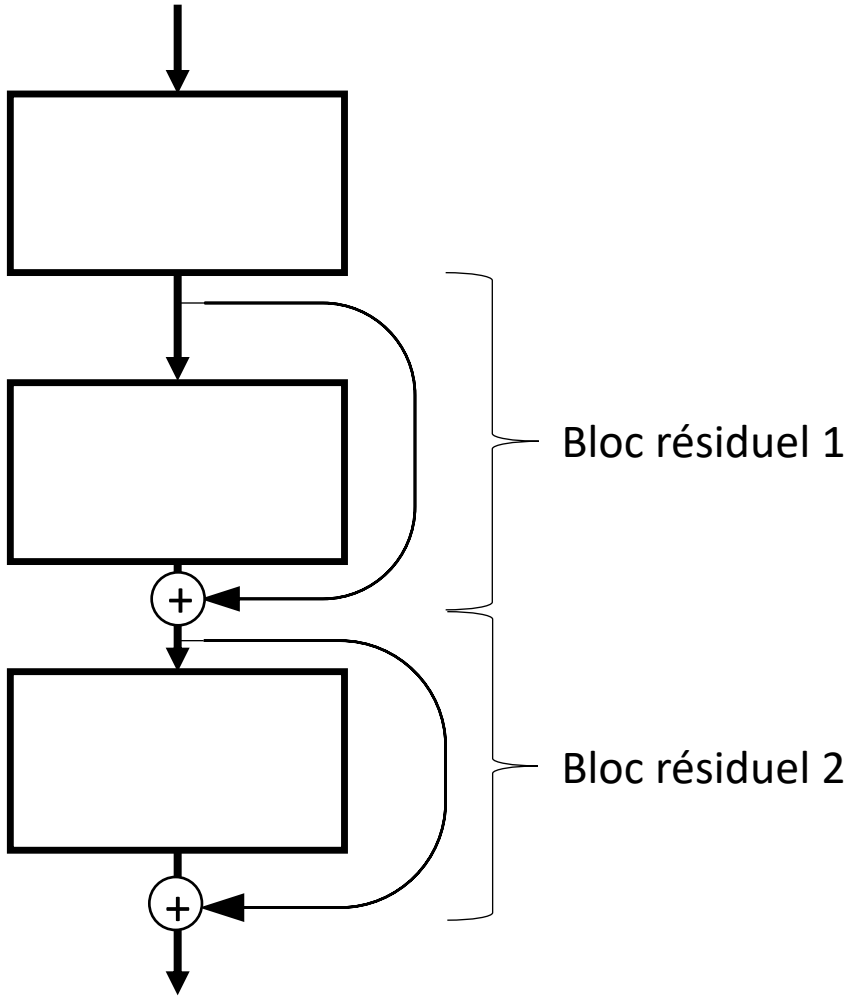
LeNet : http://yann.lecun.com/exdb/publis/pdf/lecun-98.pdf
Image source : https://www.kaggle.com/c/noaa-right-whale-recognition
Animation source : Vincent Dumoulin, Francesco Visin - A guide to convolution arithmetic for deep learning

Modèles utilisés classiquement

- **ResNet :**



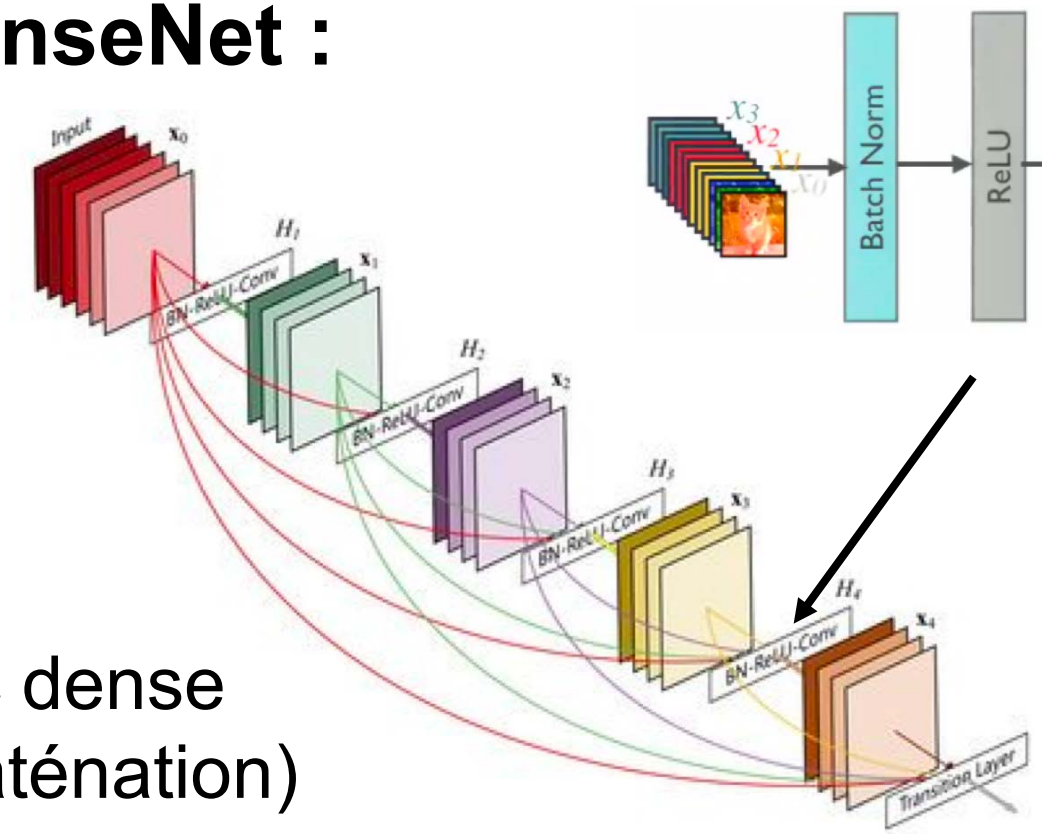
Bloc résiduel
(addition)



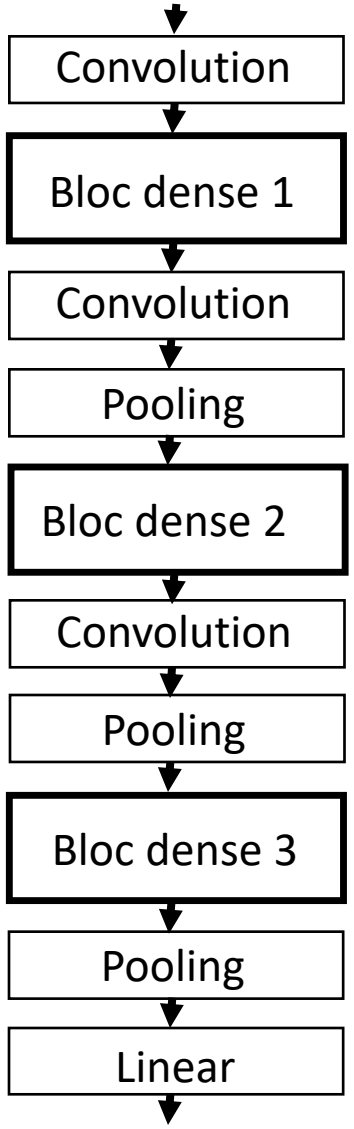
ResNet : https://arxiv.org/abs/1512.03385

Modèles utilisés classiquement

- **DenseNet :**



Bloc dense
(concaténation)



DenseNet : https://arxiv.org/abs/1608.06993

Modèles utilisés classiquement

- **Autres :**
  - AlexNet
  - VGG
  - Inception
  - …

AlexNet : http://www.cs.toronto.edu/~fritz/absps/imagenet.pdf
VGG : https://arxiv.org/abs/1409.1556
Inception : https://arxiv.org/abs/1409.4842

Implémentation

Très souvent le code existe déjà !

LeNet est dans tous les tutoriaux !
http://pytorch.org/tutorials/beginner/blitz/neural_networks_tutorial.html

```
import torchvision.models as models
import torch.nn as nn

# ResNet
resnet18 = models.resnet18(pretrained=False)
resnet34 = models.resnet34(pretrained=False)
resnet50 = models.resnet50(pretrained=False)
resnet101 = models.resnet101(pretrained=False)
resnet105 = models.resnet152(pretrained=False)

# DenseNet
densenet121 = models.densenet121(pretrained=False)
densenet169 = models.densenet169(pretrained=False)
densenet161 = models.densenet161(pretrained=False)
densenet201 = models.densenet201(pretrained=False)

model = models.resnet18(pretrained=True)
num_ftrs = model.fc.in_features
num_classes = 447
model.fc = nn.Linear(num_ftrs, num_classes)
```

Penser à redéfinir la couche de sortie !

Scores et fonctions de coût utilisés

Scores :

- Accuracy = (VP + VN) / (VP + FP + FN + VN)
- Precision = VP / (VP + FP)
- Rappel = VP / (VP + FN)
- Score F1 = 2 * (precision * rappel) / (precision + rappel)

Fonctions de coûts :

- Entropie croisée binaire = $-c\log(p) - (1-c)\log(1-p)$
- Entropie croisée = $-\sum_j c_{i,j}\log(p_{i,j})$

Scores et fonctions de coût utilisés

Scores :

- Accuracy

```
import torch

_, predicted = torch.max(outputs.data, 1)
total = labels.size(0)
correct = torch.sum(predicted == labels.data)
accuracy  = 100 * correct / total
```

- Score F1

```
from sklearn.metrics import (
        accuracy_score, f1_score)

accuracy = accuracy_score(labels, predicted)
f1 = f1_score(labels, predicted, average='macro')
```

Fonctions de coûts :

- Entropie croisée binaire

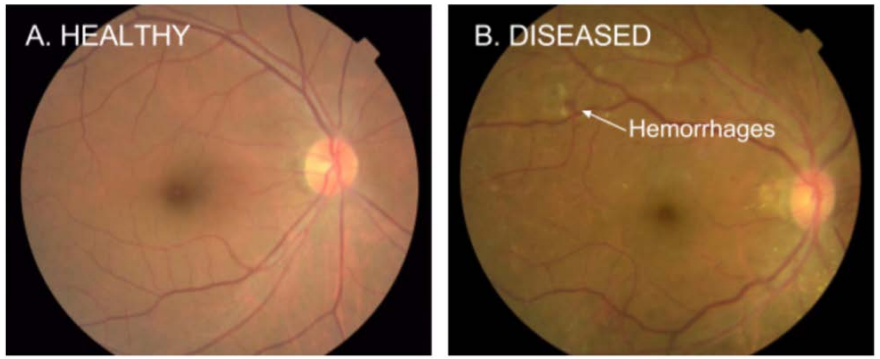
```
import torch.nn as nn

criterion = nn.BCELoss()
loss = criterion(outputs, labels)
running_loss = loss.data[0]
```

- Entropie croisée

```
criterion = nn.CrossEntropyLoss()
```

Autres exemples d'applications

Détection de la rétinopathie diabétique

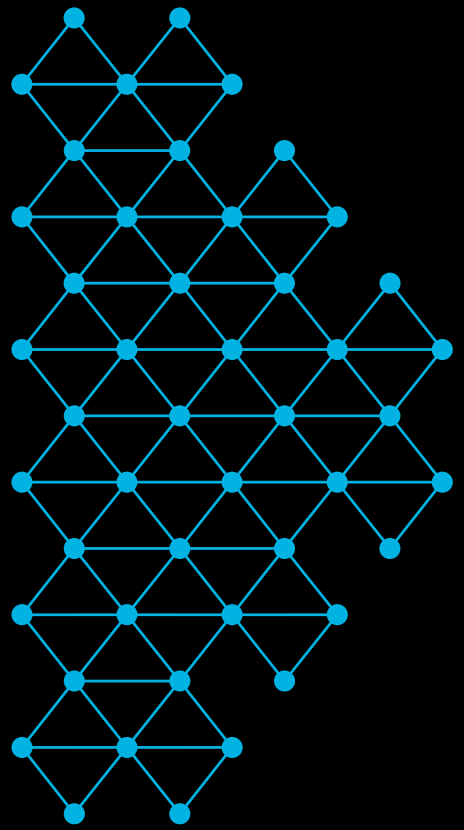


https://research.googleblog.com/2016/11/deep-learning-for-detection-of-diabetic.html
https://jamanetwork.com/journals/jama/fullarticle/2588763

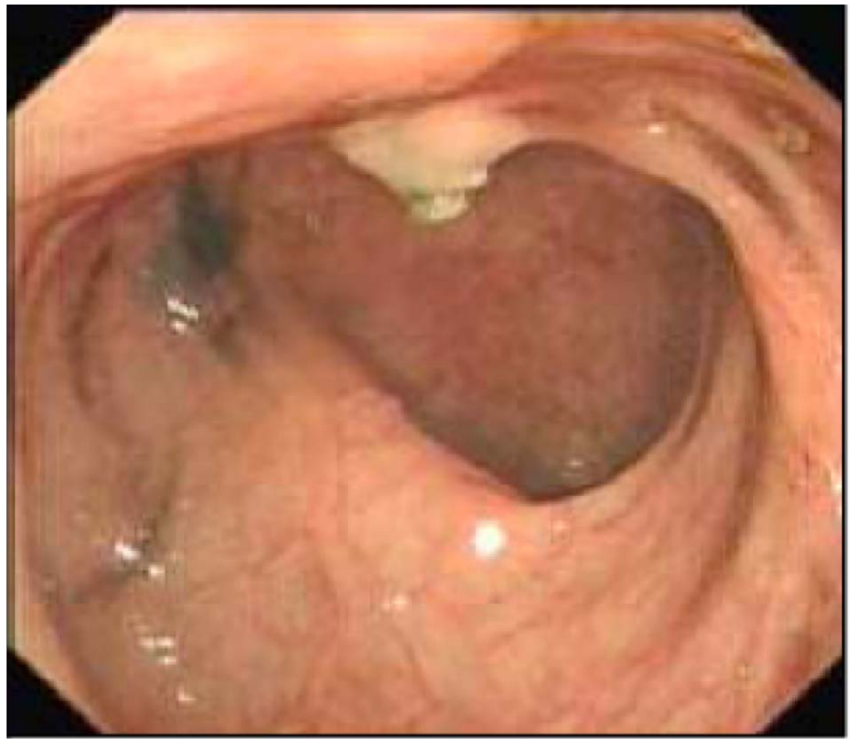
Détection de la pornographie

https://github.com/yahoo/open_nsfw

Application 2 : médecine personnalisée



Médecine personnalisée



Adapté de Gilo1969

Détection de polypes dans des images de colonoscopie

Deep Radiomics : découverte de biomarqueurs des données d'imageries prédictifs de l'état de santé des patients et de leur évolution.

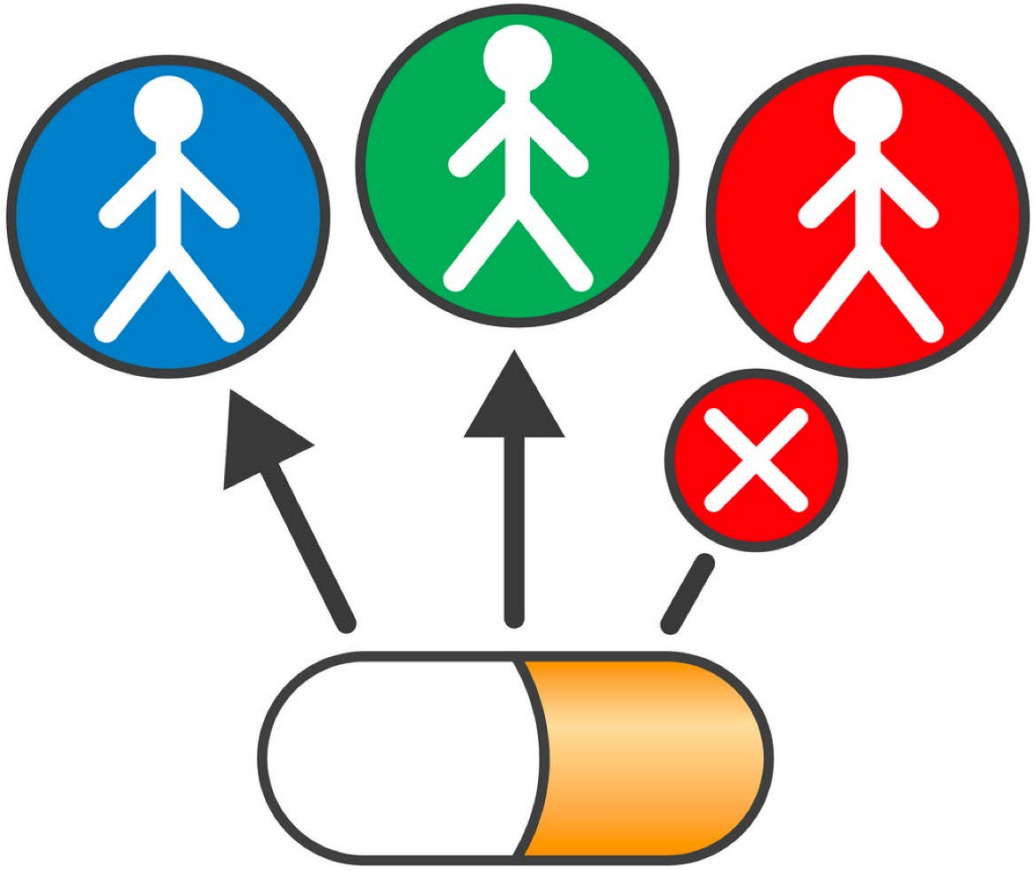
https://www.imagia.com/fr/accueil
https://www.imagia.com/research
https://arxiv.org/pdf/1612.00799.pdf

Impact sur :

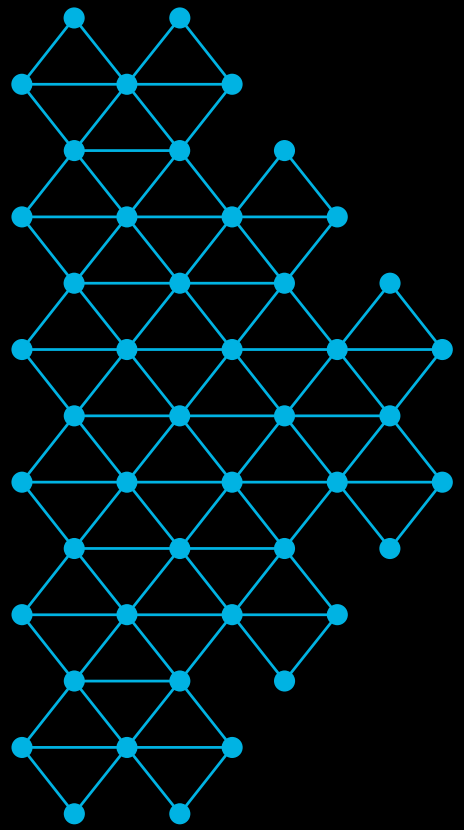
- **Le système de santé**
  - Facilite les collaborations
  - Prise de décision rapide

- **La santé du patient**
  - Médecine personnalisé

- **L’économie**
  - Réduction de coût



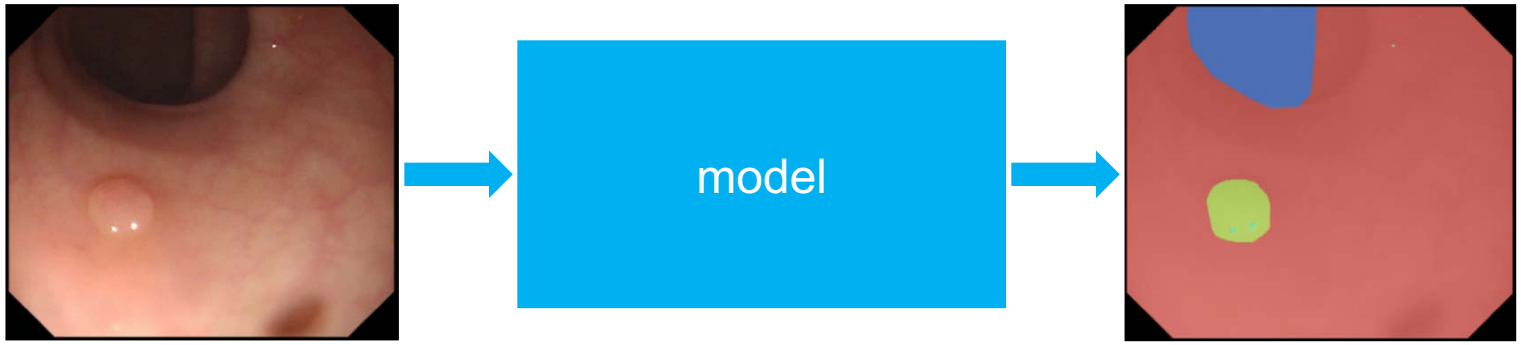
Michael J. Ermarth



Segmentation sémantique

Forme des données et tâches

Image d'entrée contenant des objets représentant une ou plusieurs classes



Segmentation de l'image d'entrée : regroupement des pixels entre eux pour séparer les objets du fonds

Tâches : Segmentation sémantique = associer à chaque partie de l'image (à chaque pixel) une classe.

Jeux de donnée de référence : PASCAL VOC, COCO, CamVid

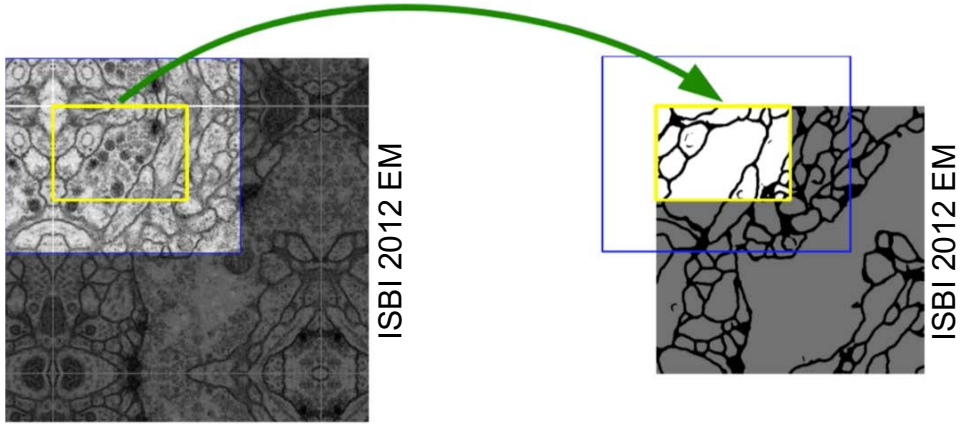
https://arxiv.org/pdf/1612.00799.pdf

Préparation des données

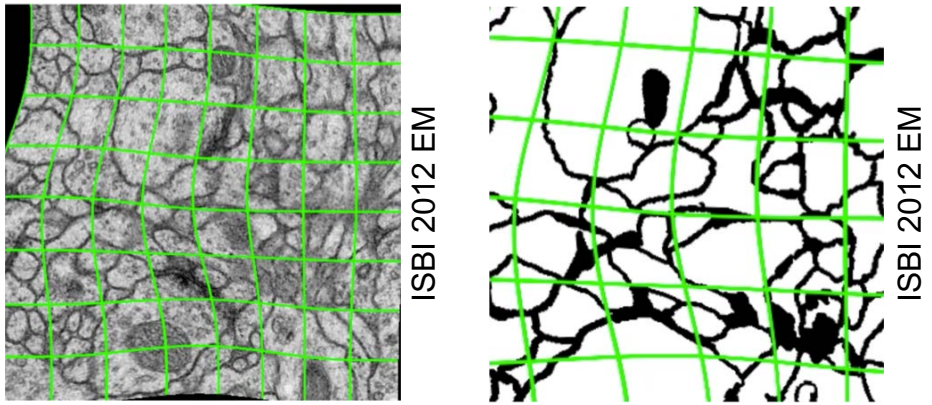
- Souvent jeux de données dépendant
- Des couples {images, masque}
- Normalisation des images mais pas du masque
- Penser à appliquer les transformations sur les images et le masque

Ex. de préparation des données

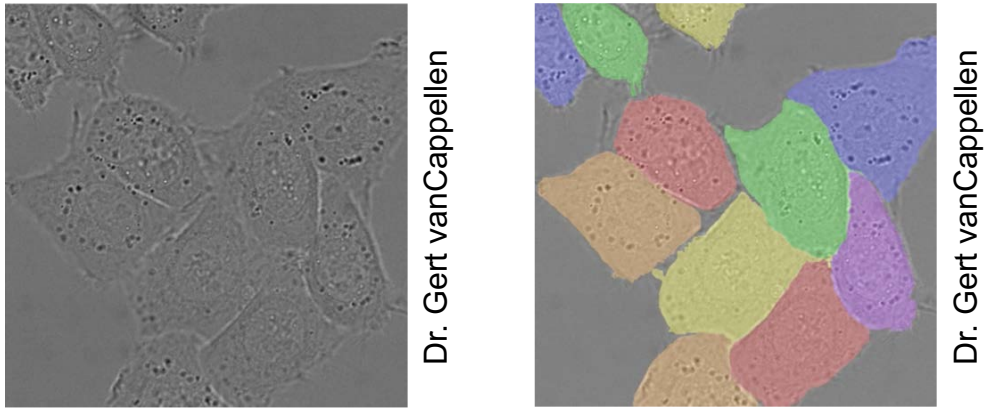
Coupe de l'image

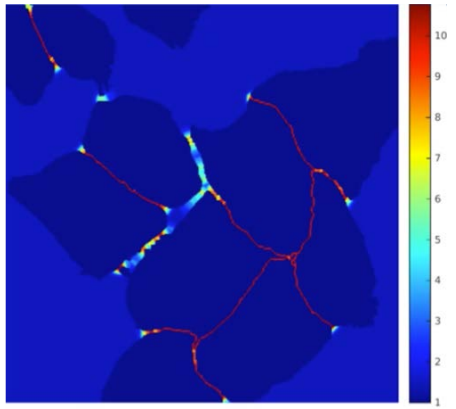


Augmentation de données par déformations



Séparation des objets agglomérés

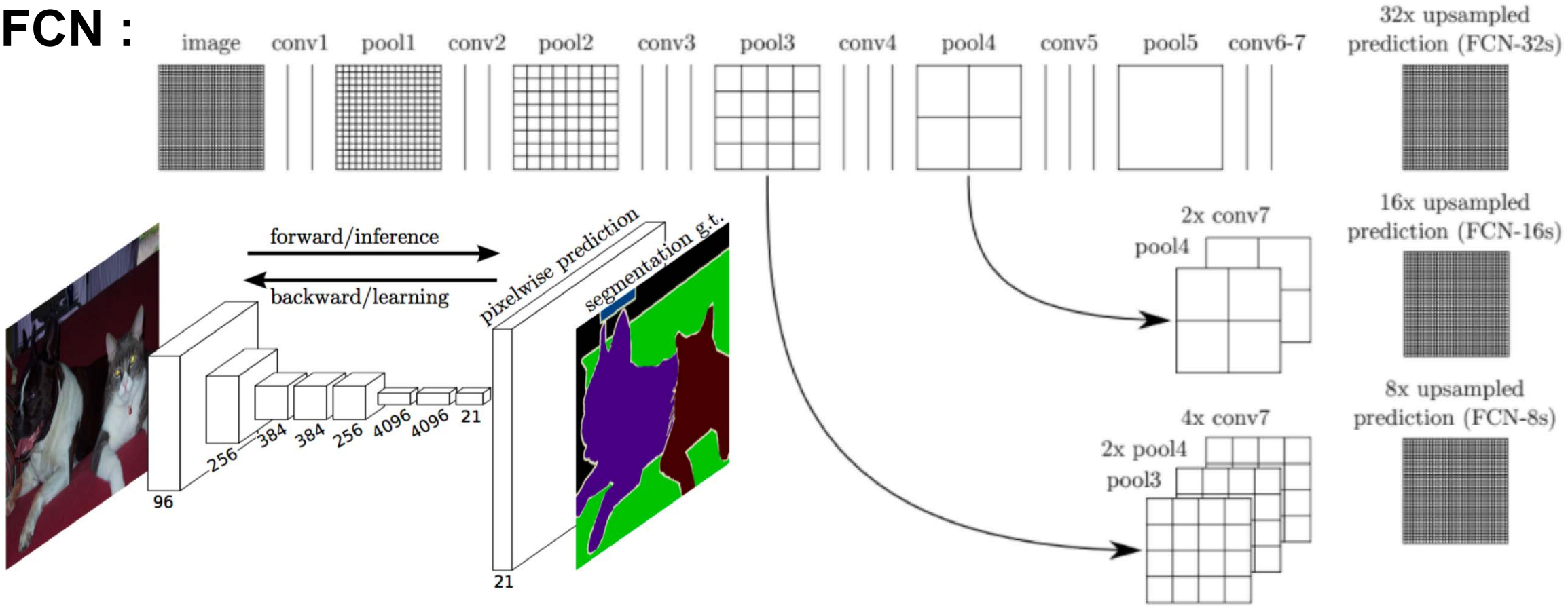




Olaf Ronneberg, Philipp Fisher, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. MICCAI 2015.
Slide adaptée de la vidéo de présentation du papier par Olaf Ronneberg, University of Freiburg, Germany.

Modèles utilisés classiquement

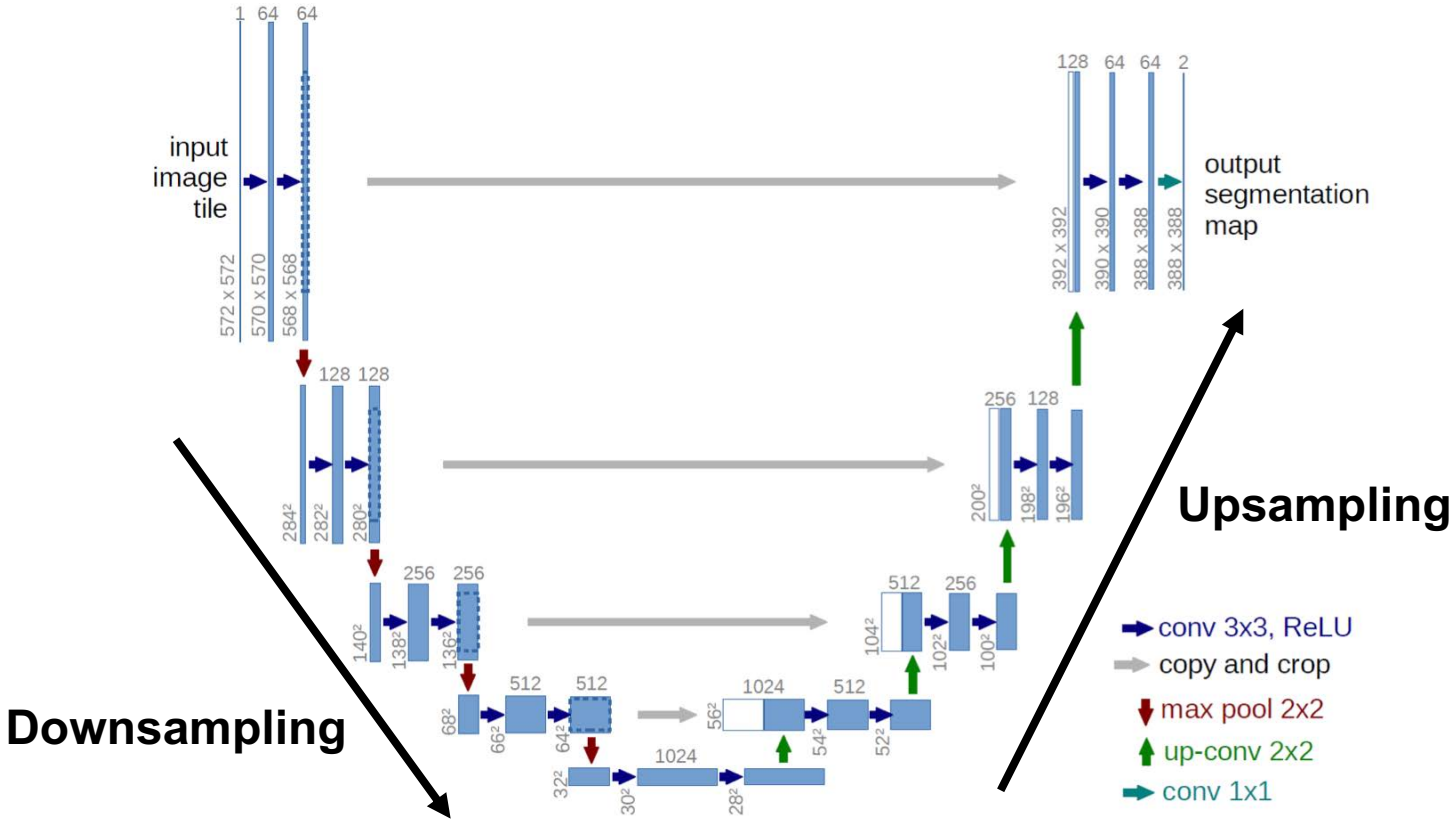
- **FCN :**

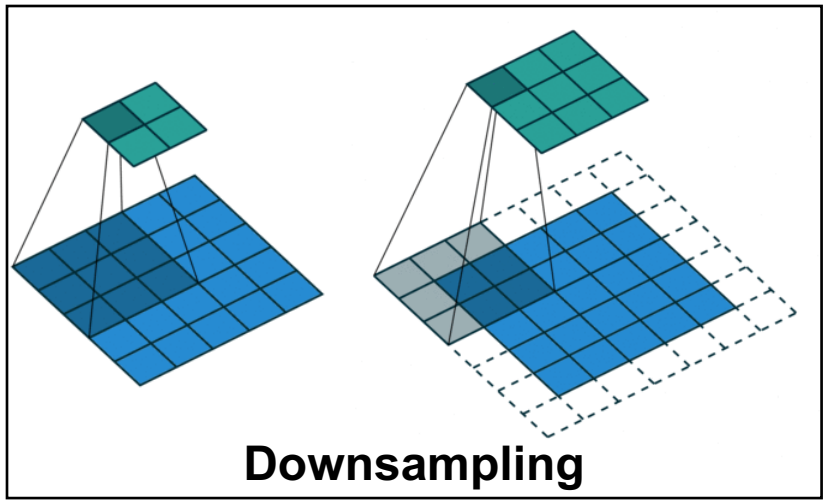


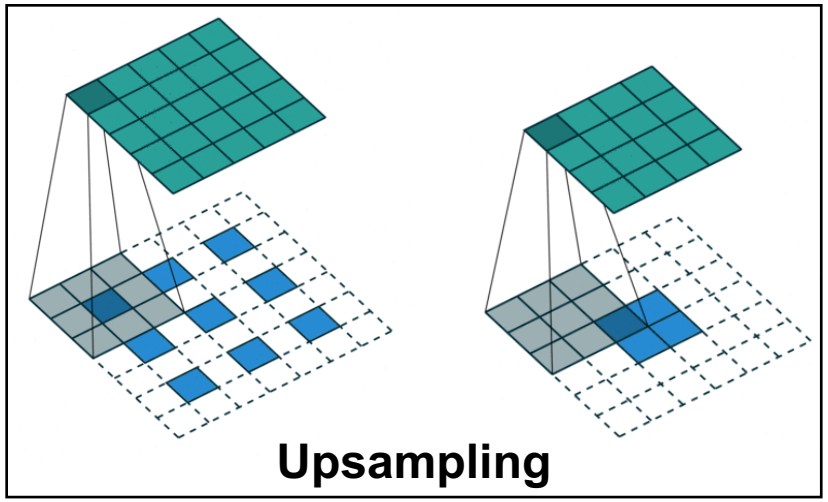
FCN : https://people.eecs.berkeley.edu/~jonlong/long_shelhamer_fcn.pdf

Modèles utilisés classiquement

- **U-Net :**







U-Net : https://arxiv.org/abs/1505.04597
Vincent Dumoulin, Francesco Visin - A guide to convolution arithmetic for deep learning : https://arxiv.org/abs/1603.07285

MILA

Scores et fonctions de coût utilisés

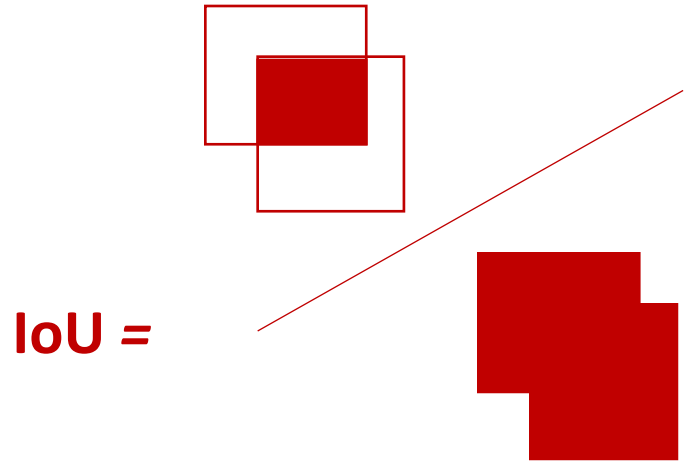
Scores :

- Intersection over union
  ou indice de Jaccard

Fonctions de coûts :

- Entropie croisée binaire
- Entropie croisée
- Dice loss

Intersection over Union (non différentiable)



Plus c'est proche de 1 mieux c'est !

Autres exemples d'applications

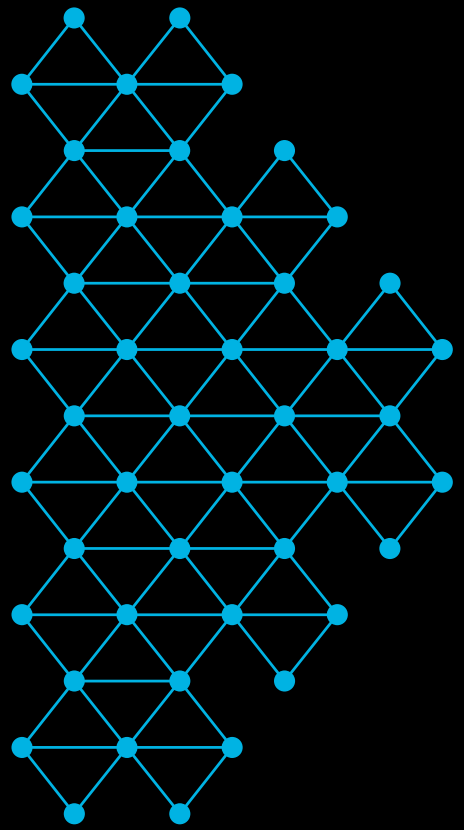
Edition de photos

https://blog.photoeditorsdk.com/deep-learning-for-photo-editing-943bdf9765e1

Génération de données cartographiques

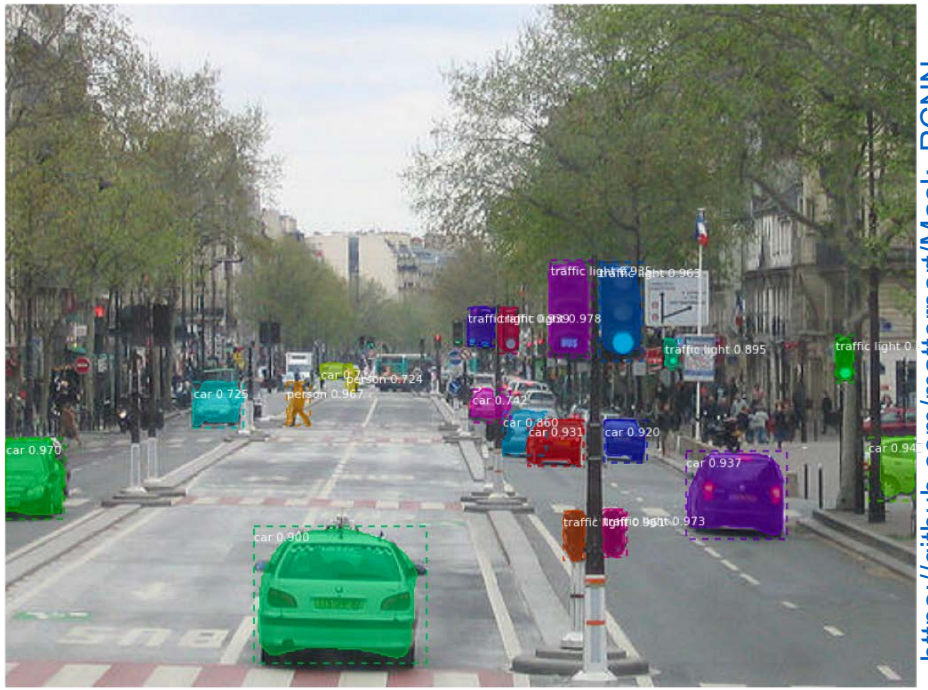
https://blogs.nvidia.com/blog/2017/12/22/ai-maps/
https://www.mapillary.com/

Application 3 : voitures autonomes



Voiture autonome

https://www.nvidia.com/en-us/self-driving-cars/

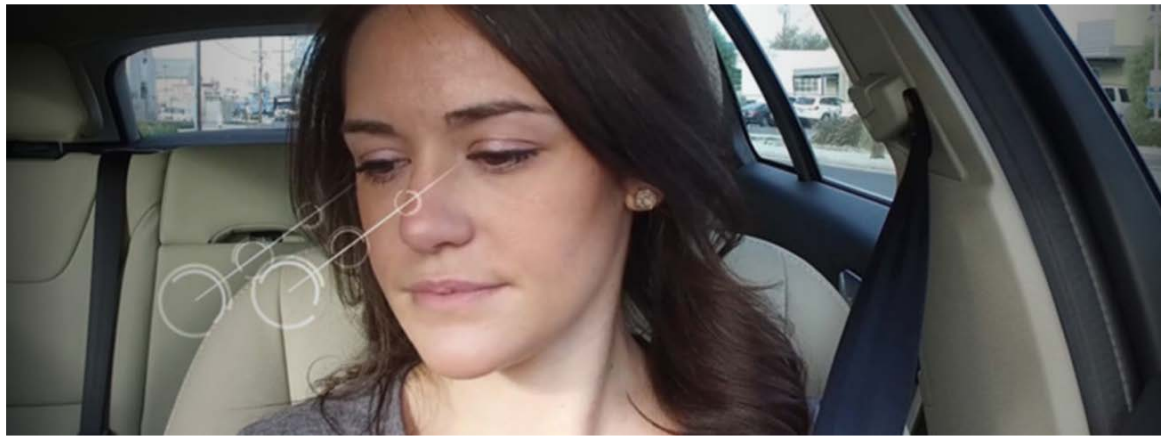


https://github.com/matterport/Mask_RCNN

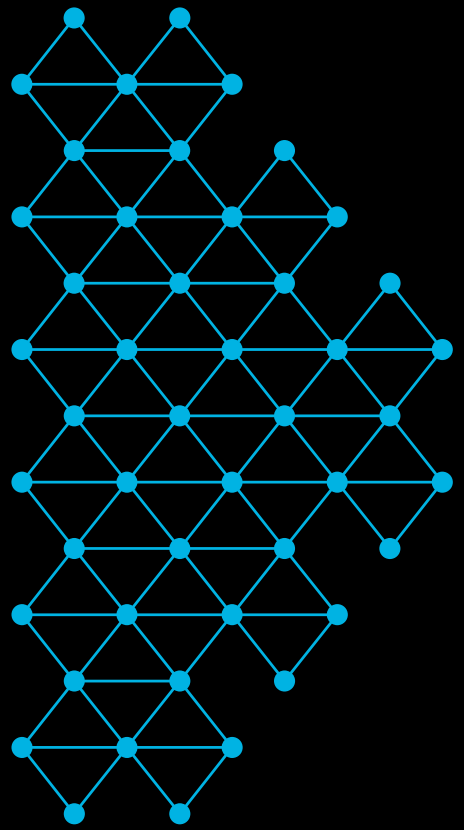
Capter ce qui se passe dans l'environnement

Impact sur :

- **L’écologie**
  - Moins de pollution
- **La société**
  - Moins de bruit
  - Plus de temps
- **La sécurité/santé**
  - Moins d’accident
- **Economique**
  - Moins de dépenses



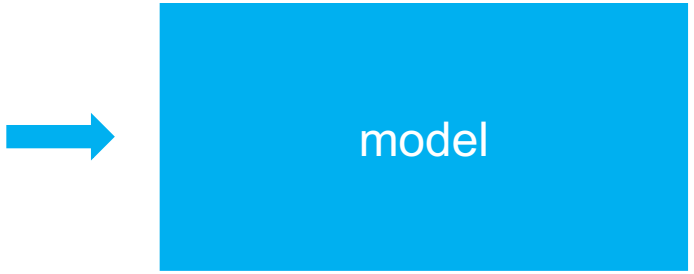
https://www.nvidia.com/en-us/self-driving-cars/



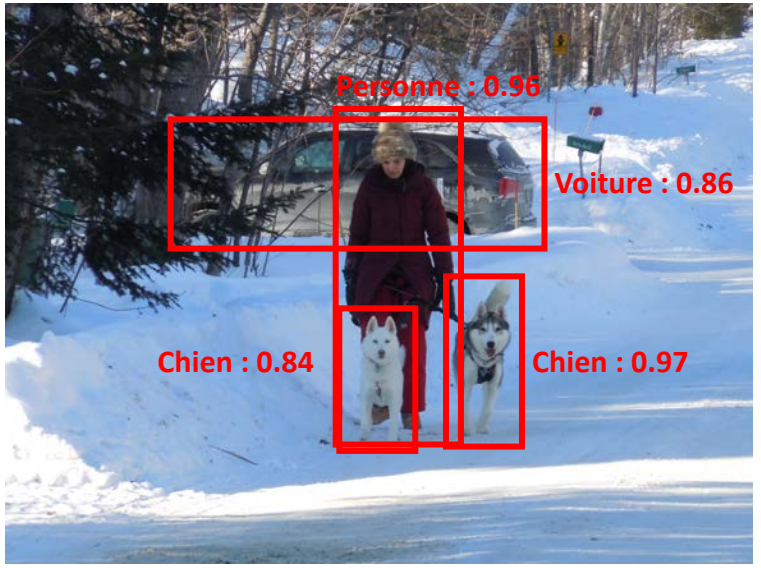
Détection d’objets
Classification
(Segmentation)

Forme des données et tâches

Image d'entrée représentant l'une ou plusieurs des classes d'objets



Boîtes de délimitation + Probabilité d'appartenance aux classes

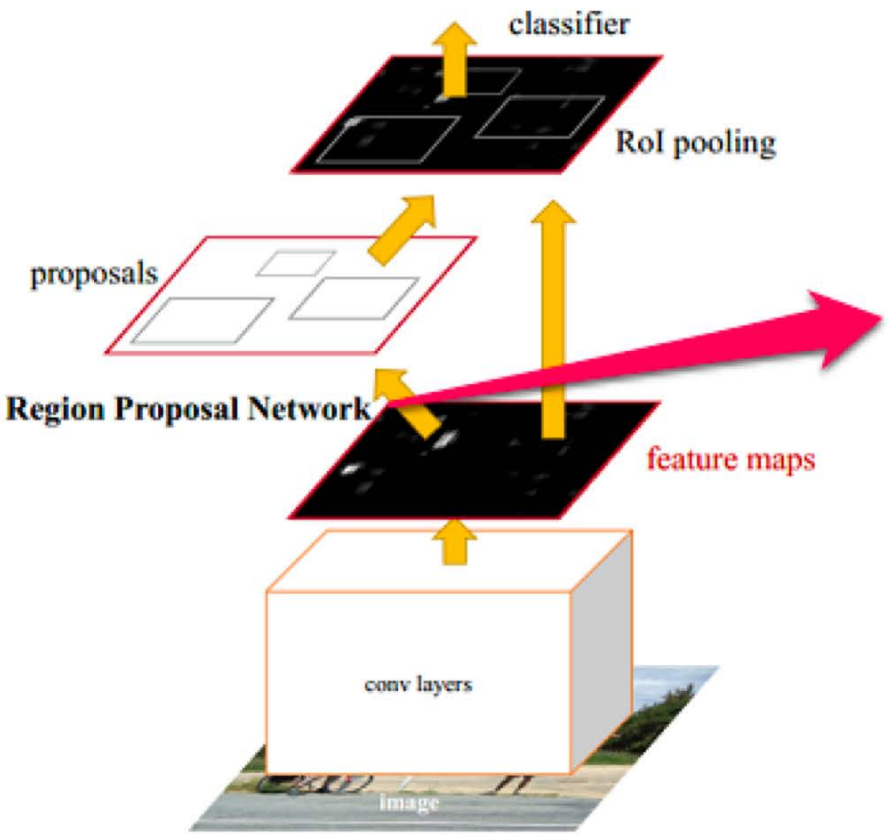


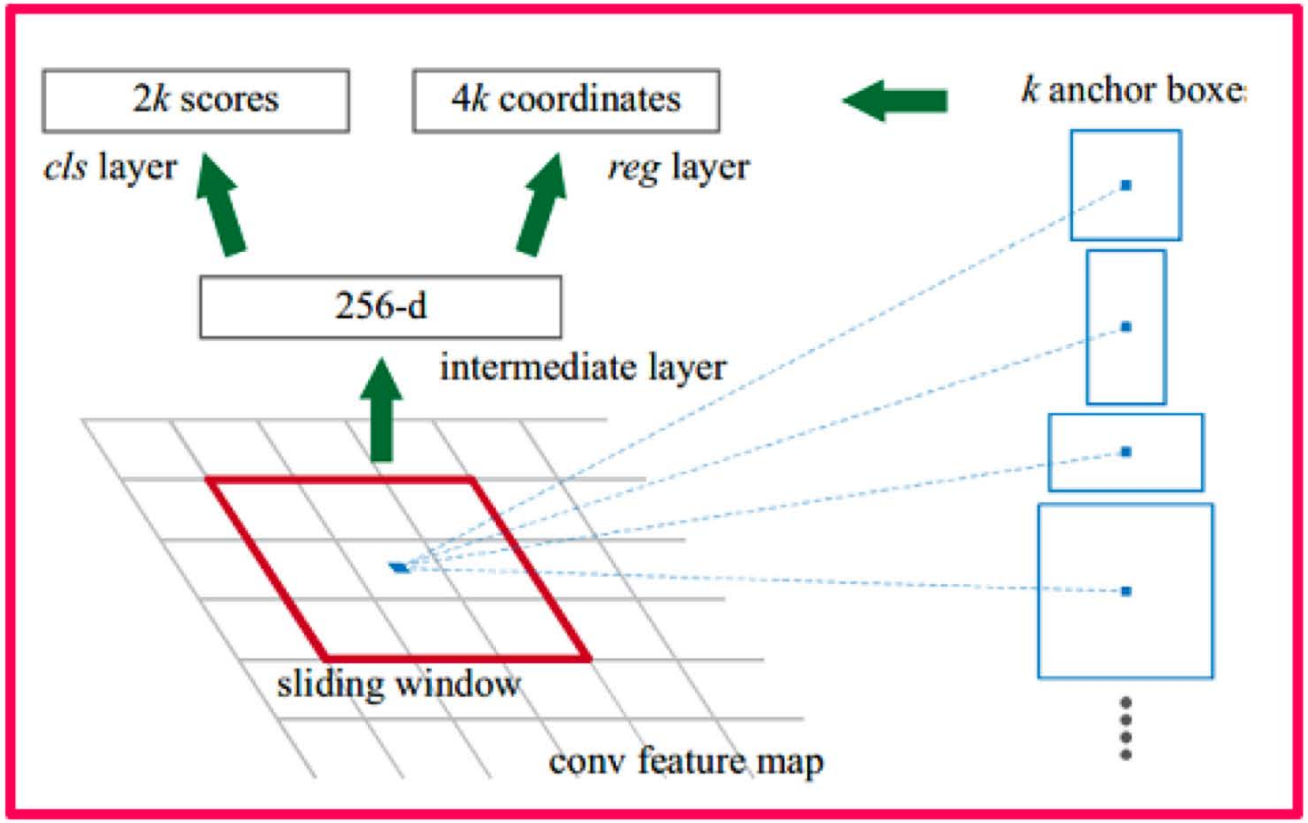
Tâches : localiser et classifier des objets

Jeux de données de références : PASCAL VOC, COCO

Modèles utilisés classiquement

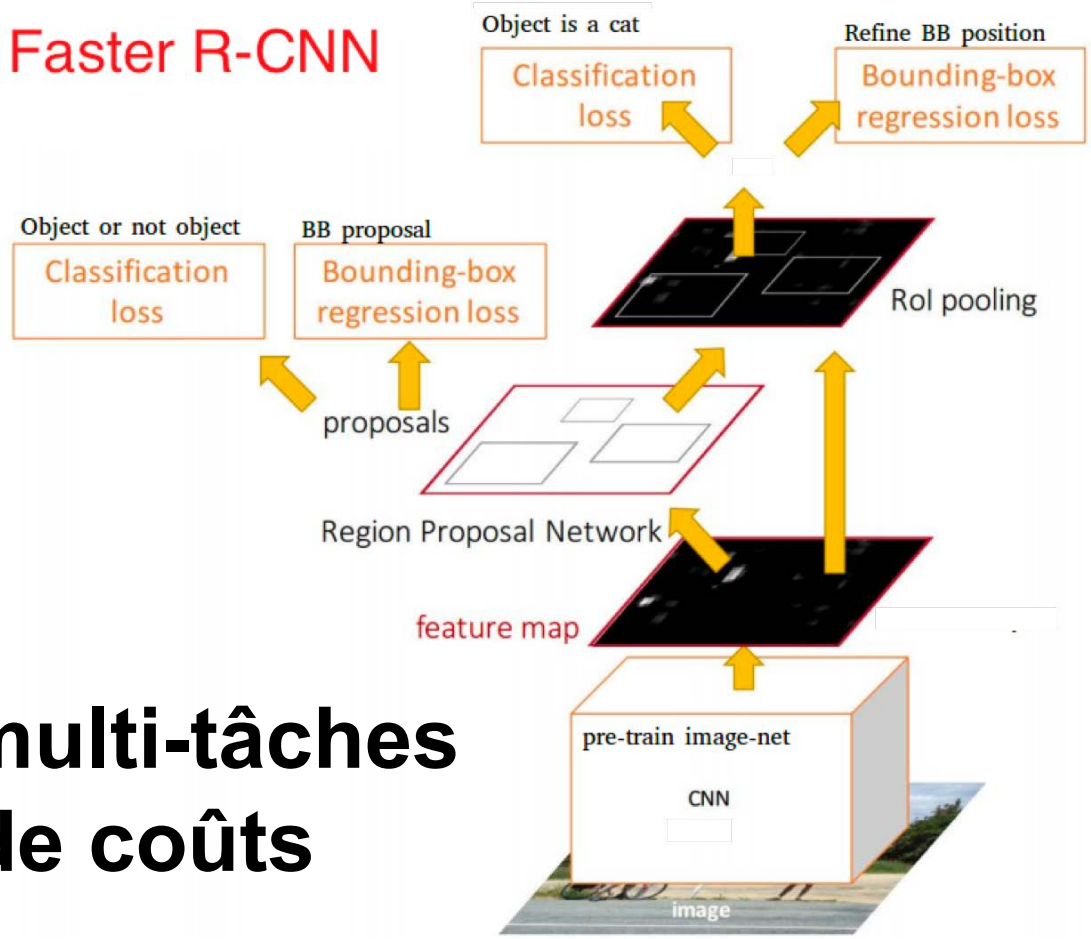
- **Faster R-CNN :**





Faster R-CNN : Ren, Shaoqing, et al. "Faster r-cnn: Towards real-time object detection with region proposal networks." *Advances in neural information processing systems*. 2015.

Scores et fonctions de coût utilisées

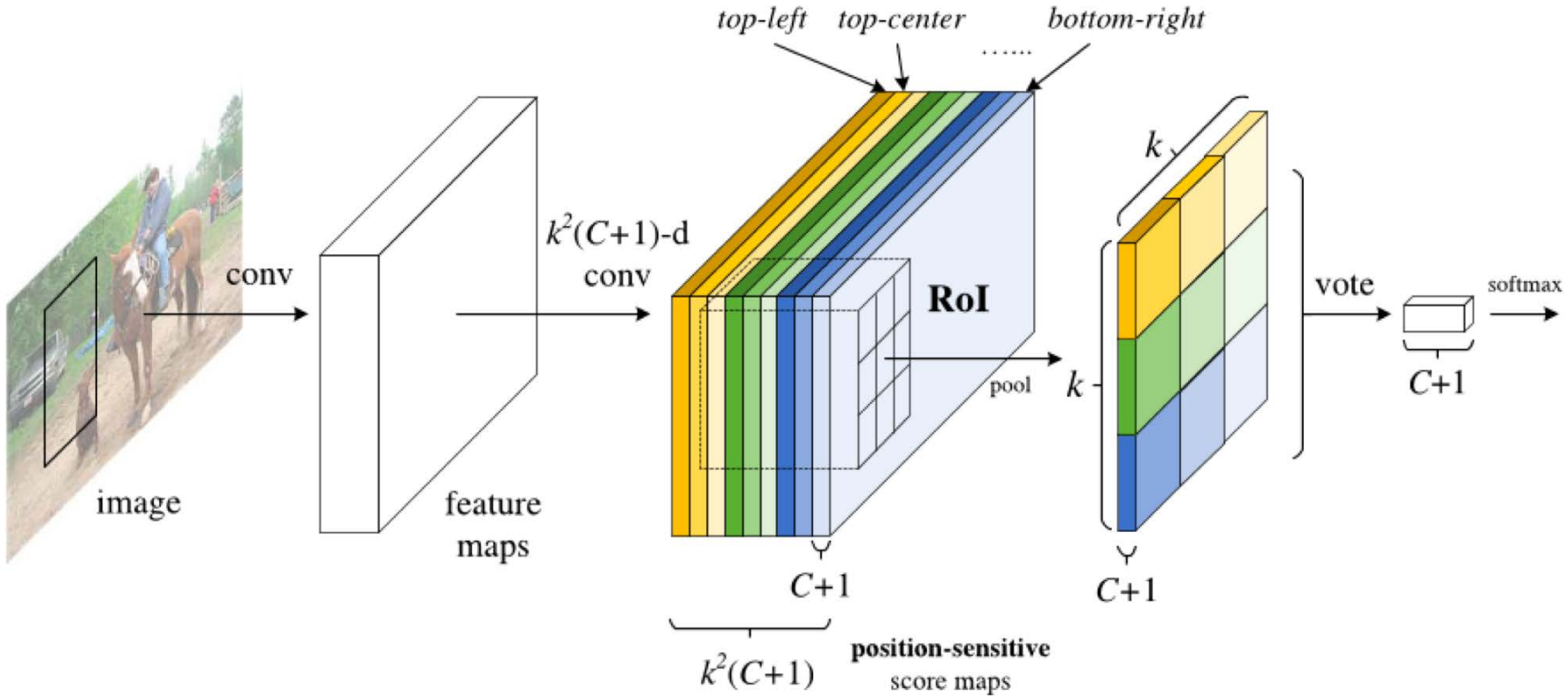


Apprentissage multi-tâches
4 fonctions de coûts

Faster R-CNN : Ren, Shaoqing, et al. "Faster r-cnn: Towards real-time object detection with region proposal networks." *Advances in neural information processing systems*. 2015.

Modèles utilisés classiquement

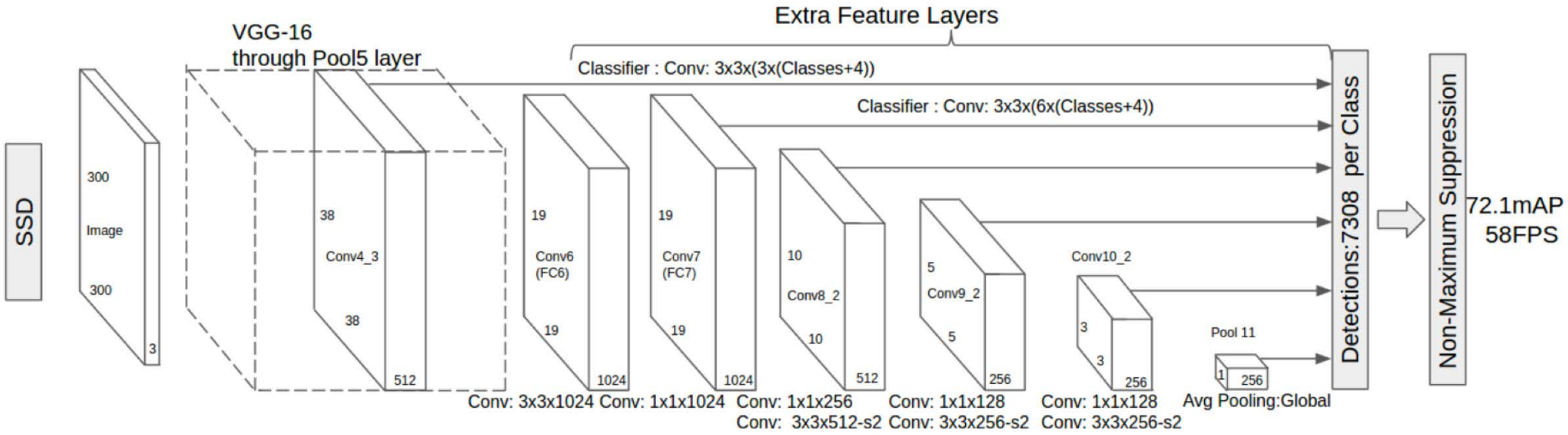
- **R-FCN :**



Dai, Jifeng, et al. "R-fcn: Object detection via region-based fully convolutional networks." *Advances in neural information processing systems*. 2016.

Modèles utilisés classiquement

- **SSD :**

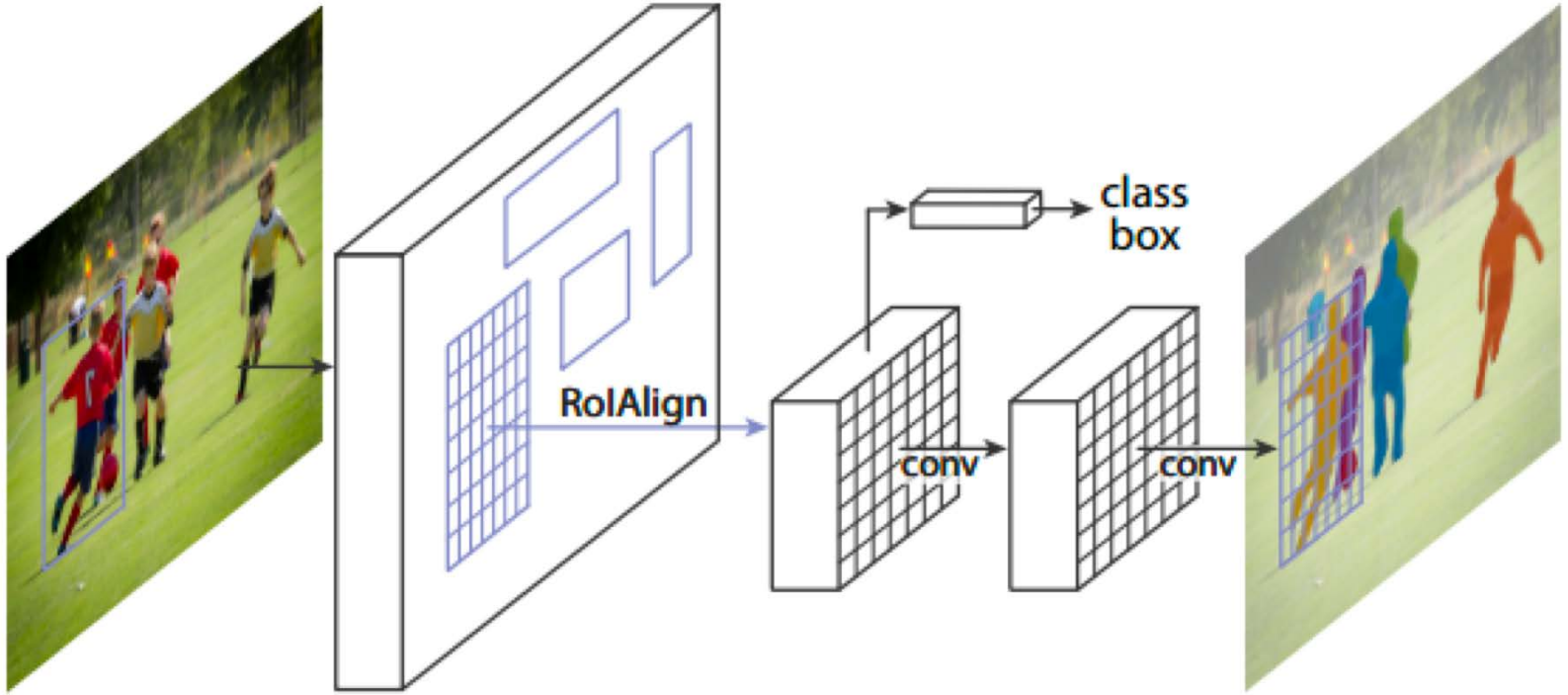


- **YOLO**

SSD : Liu, Wei, et al. "Ssd: Single shot multibox detector." *European conference on computer vision*. Springer, Cham, 2016.
YOLO : Redmon, Joseph, et al. "You only look once: Unified, real-time object detection." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

Modèles utilisés classiquement

- **Mask RCNN :**

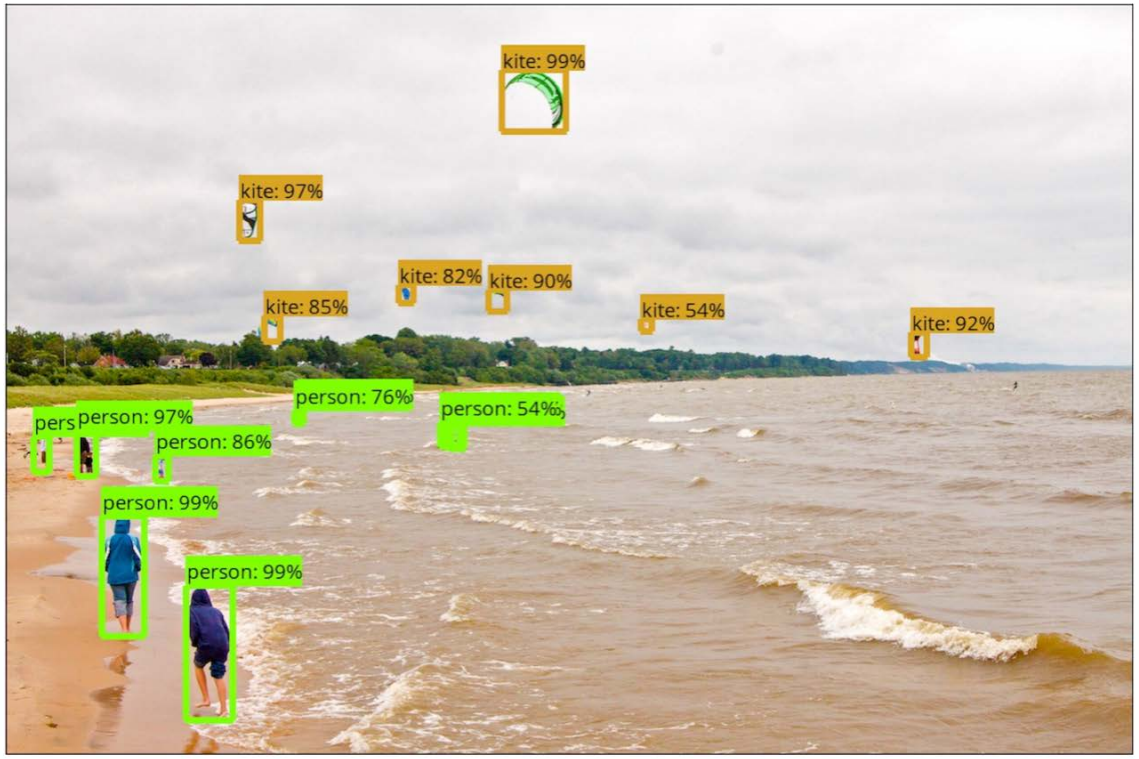


https://arxiv.org/pdf/1703.06870.pdf
https://github.com/matterport/Mask_RCNN

Implémentation

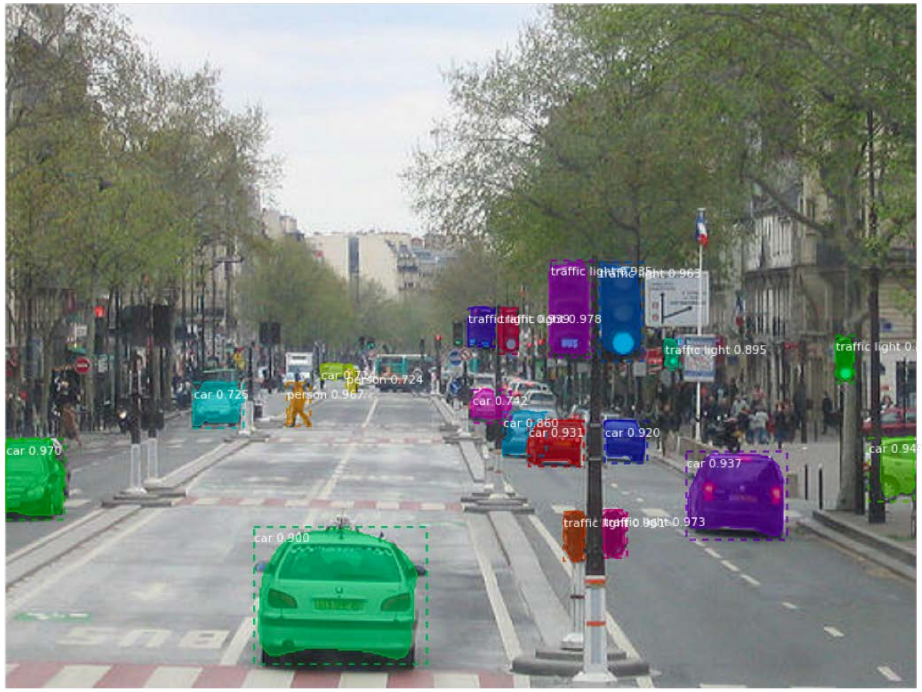
https://research.googleblog.com/2017/06/supercharge-your-computer-vision-models.html

https://github.com/tensorflow/models/tree/master/research/object_detection

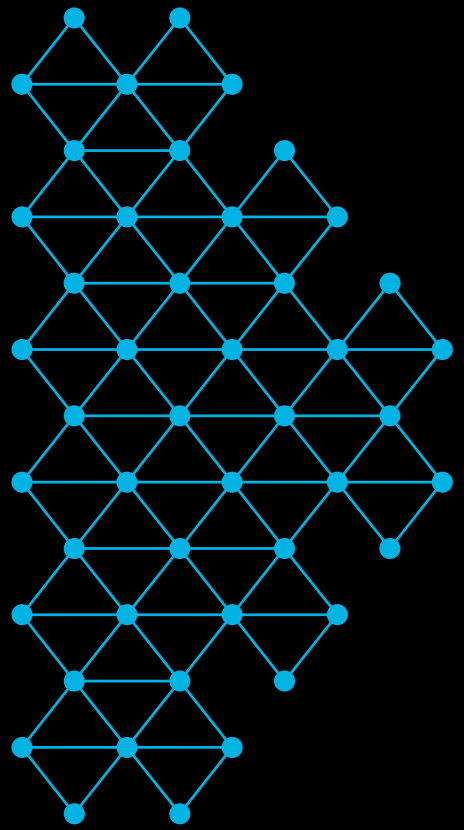


Autres exemples d'applications

Vidéo de surveillance intelligente



Détection de visages



Mais encore…

Autres applications possibles

VQA

Horaciotorquez

De quelle couleur sont les yeux du chat ?

A. Bleue et marron

B. Bleue et vert

Image captionning

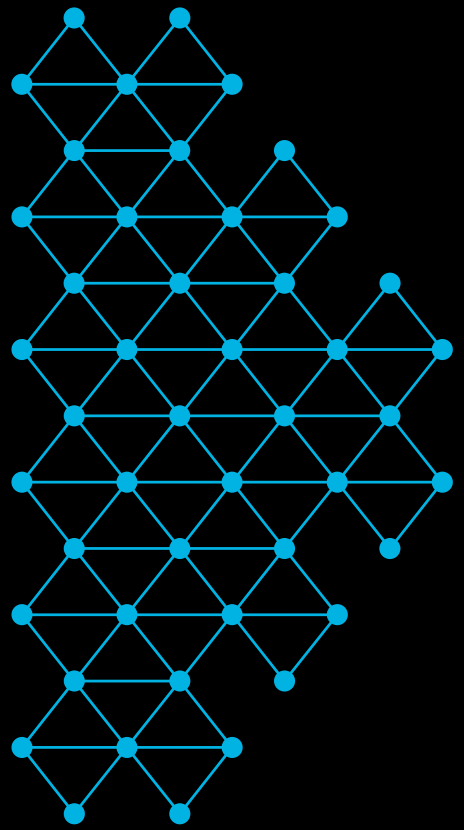
Ron Armstrong

Un chien noir et blanc saute un obstacle

Image generation

Quelques applications : https://deepai.org/ai-image-processing

Merci de votre attention !
Des questions ?



Contact

http://mila.umontreal.ca

Margaux Luck • margaux.luck@rd.mila.quebec

Contact business :

Myriam Côté • myriam.cote@rd.mila.quebec