Deep learning for remote sensing – An introduction

Nicolas Audebert[1, 2], Bertrand Le Saux[1], Sébastien Lefèvre[2]

[1] ONERA, [2] IRISA – Université Bretagne Sud

Atelier DLT Sageo – November 7th, 2018

Introduction

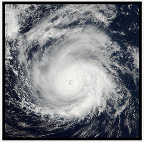
Why Earth Observation?

Disaster management

Emergency services organization, first responders...

CNRS Le Journal (2018), "La cartographie au service des secours"

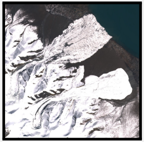
Meteorology



Wind measurements, oceanic temperature monitoring, magnetic field study...

La Tribune (2018), "Avec le satellite Aeolus d'Airbus, Météo-France va mieux mesurer les vents à l'échelle mondiale"

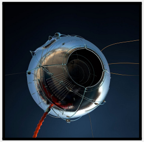
Climate change



Ice cap melt, atmospheric pollution monitoring...

Le Monde (2017), "Un satellite franco-israélien pour lutter contre le réchauffement climatique"
La Croix (2018), "L'Observatoire spatial du climat prend son envol"

Surveillance



Urban expansion monitoring, illegal activities detection...

BBC (2018), "NovaSAR: UK radar satellite launches to track illegal shipping activity"
The Guardian (2016), "New satellite mapping a 'game changer' against illegal logging"

Data volume

In 2017, 620 satellites listed “Earth Observation” as their primary application.[1,2]

Sentinel-2 satellites acquire 6 Tb of data every day. Total volume will reach 1Pb in 2020. A full image of the Earth is acquired every 5 days.[3]

> High-altitude aircraft and satellites will constitute a major primary data acquisition source in the future and will be generating vast amounts of imagery suitable for photomapping. In fact, photomapping would appear to be the only way to take reasonable advantage of these future data sources.
>
> Cartography 1950-2000, Robinson et al., 1977, Transactions of the Institute of British Geographers

[1] UCS Satellite Database

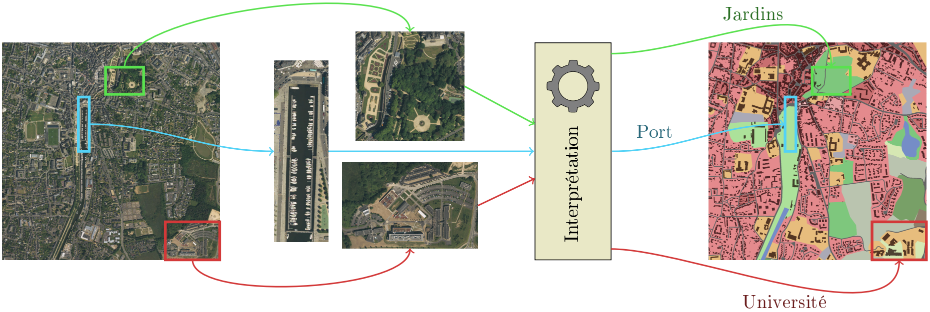
[2] Pixalytics, ”Earth Observation satellites in space in 2017”

[3] Sentinel Data Access Annual Report

Semantic mapping

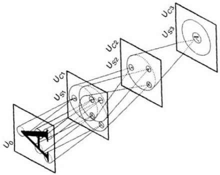
Goal

Automatically map urban or rural areas for thematic classes of interest using aerial or satellite images.



Deep neural networks for computer vision

- 1980: Neocognitron (pattern recognition) (Fukushima et al.)
- 1989: gradient backpropagation (Werbos 75, reintroduced by LeCun et al.)
- 2012: GPU implementation (Krizhevsky et al.)
- 2012: ImageNet challenge (1000 classes, 1 000 000 images) (Deng et al.)

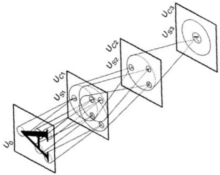


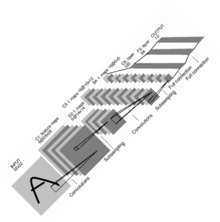
Why is deep learning exciting?

Deep networks are the current state of the art for pattern recognition, object detection, semantic segmentation...in computer vision.

Deep neural networks for computer vision

- 1980: Neocognitron (pattern recognition) (Fukushima et al.)
- 1989: gradient backpropagation (Werbos 75, reintroduced by LeCun et al.)
- 2012: GPU implementation (Krizhevsky et al.)
- 2012: ImageNet challenge (1000 classes, 1 000 000 images) (Deng et al.)



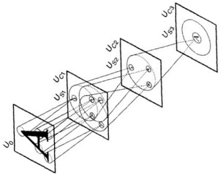


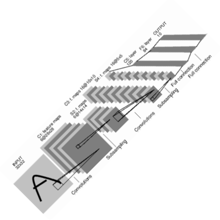
Why is deep learning exciting?

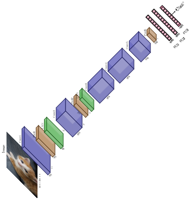
Deep networks are the current state of the art for pattern recognition, object detection, semantic segmentation...in computer vision.

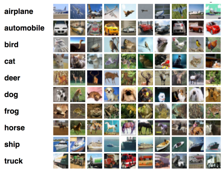
Deep neural networks for computer vision

- 1980: Neocognitron (pattern recognition) (Fukushima et al.)
- 1989: gradient backpropagation (Werbos 75, reintroduced by LeCun et al.)
- 2012: GPU implementation (Krizhevsky et al.)
- 2012: ImageNet challenge (1000 classes, 1 000 000 images) (Deng et al.)







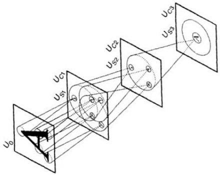


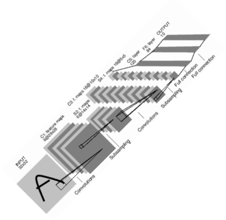
Why is deep learning exciting?

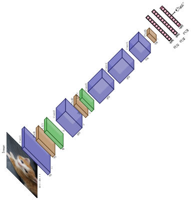
Deep networks are the current state of the art for pattern recognition, object detection, semantic segmentation...in computer vision.

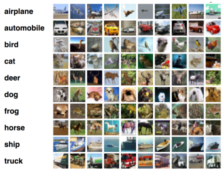
Deep neural networks for computer vision

- 1980: Neocognitron (pattern recognition) (Fukushima et al.)
- 1989: gradient backpropagation (Werbos 75, reintroduced by LeCun et al.)
- 2012: GPU implementation (Krizhevsky et al.)
- 2012: ImageNet challenge (1000 classes, 1 000 000 images) (Deng et al.)





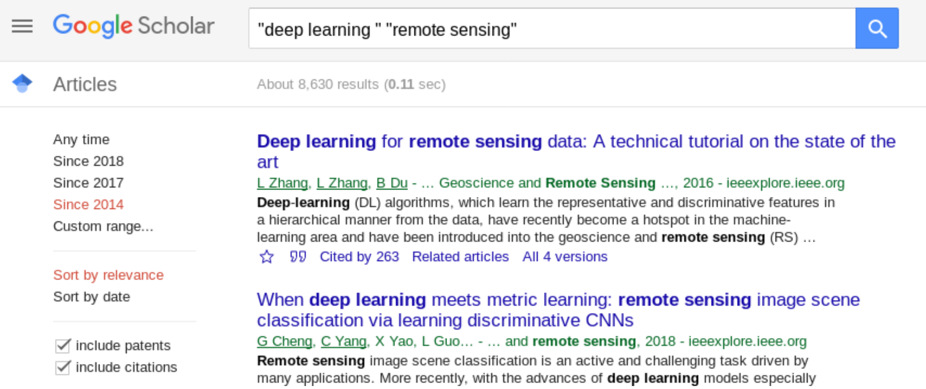




Why is deep learning exciting?

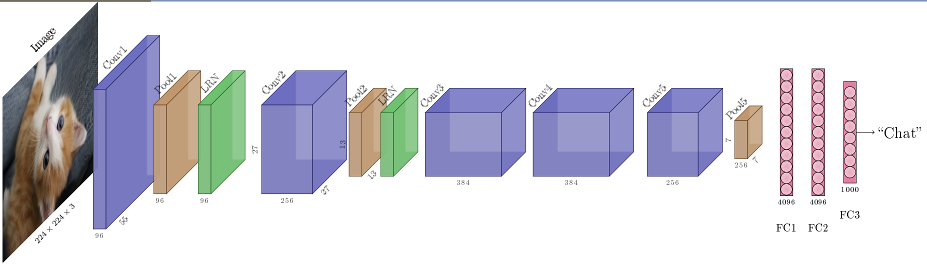
Deep networks are the current state of the art for pattern recognition, object detection, semantic segmentation...in computer vision.

A very, very, very abundant literature



Deep learning for remote sensing image interpretation

Deep networks for image classification



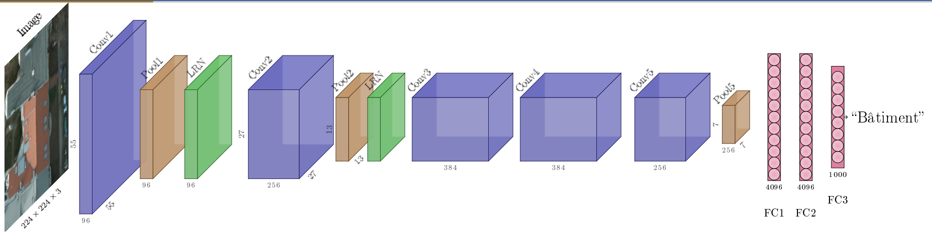
Krizhevsky et al., *ImageNet classification with deep Convolutional Neural Networks*, NIPS 2012

Convolutional Neural Networks

- Convolutional layers: parametrized convolutions to be optimized
- Pooling layers: max or average-pooling
- Fully connected layers: flattening + multiplication w/ weight matrix
- Each layer is followed by a non-linear activation, e.g. *tanh* or *ReLU*

Vakalopoulou et al., *Building detection in VHR multispectral data with deep learning features*, IGARSS 2015

Deep networks for image classification



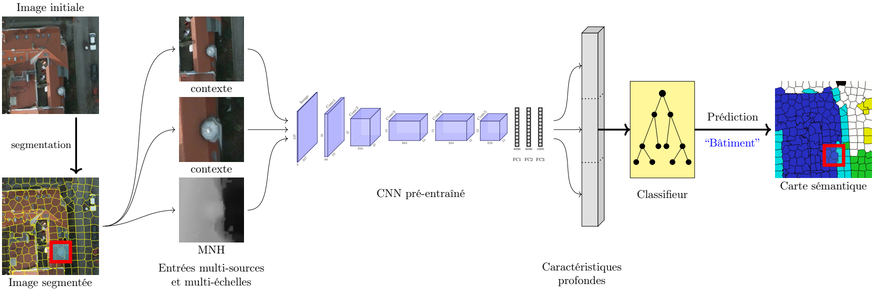
Penatti et al., *Do Deep Features Generalize from Everyday Objects to Remote Sensing and Aerial Scenes Domains?*, CVPRW 2015

Convolutional Neural Networks

- Convolutional layers: parametrized convolutions to be optimized
- Pooling layers: max or average-pooling
- Fully connected layers: flattening + multiplication w/ weight matrix
- Each layer is followed by a non-linear activation, e.g. *tanh* or *ReLU*

Vakalopoulou et al., *Building detection in VHR multispectral data with deep learning features*, IGARSS 2015

Region-based classification



CNN for semantic mapping

1. Unsupervised pre-segmentation
2. Deep features extraction using a pretrained model
3. Statistical model classification

Campos-Taberner et al., *Outcome of the 2015 IEEE GRSS data fusion contest*, JSTARS 2016

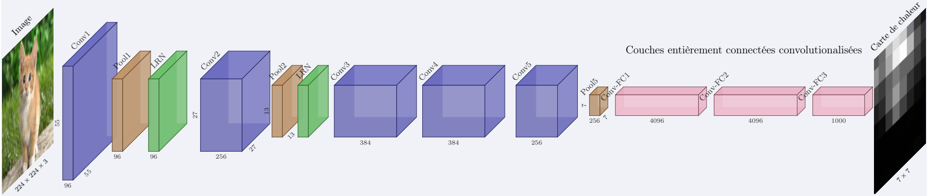
How useful is region-based classification of remote sensing images in a deep learning framework ?, Audebert et al., IGARSS 2016.

Dense semantic mapping

We want a classifier that generates **dense** predictions with an **efficient** inference.

- CNN feature maps are already spatially dense
- we can "convolutionalize" the fully connected layers to keep the spatial dimensions

Fully Convolutional Networks



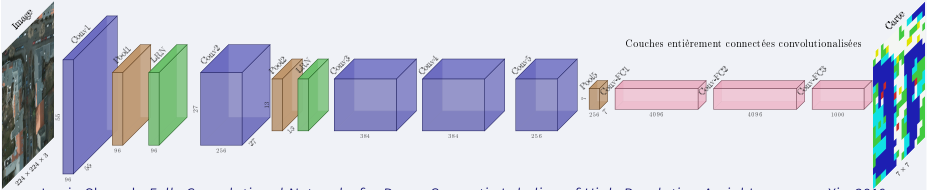
Long, Shellhammer et Darell, *Fully Convolutional Networks for Semantic Segmentation*, CVPR 2015

Dense semantic mapping

We want a classifier that generates **dense** predictions with an **efficient** inference.

- CNN feature maps are already spatially dense
- we can “convolutionalize” the fully connected layers to keep the spatial dimensions

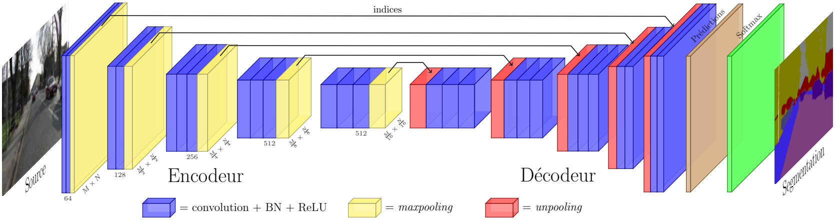
Fully Convolutional Networks



Jamie Sherrah, *Fully Convolutional Networks for Dense Semantic Labeling of High-Resolution Aerial Imagery*, arXiv, 2016

Maggiori et al., *Fully Convolutional Neural Networks For Remote Sensing Image Classification* , IGARSS 2016

Symetrical architectures: SegNet, U-Net, DeconvNet...

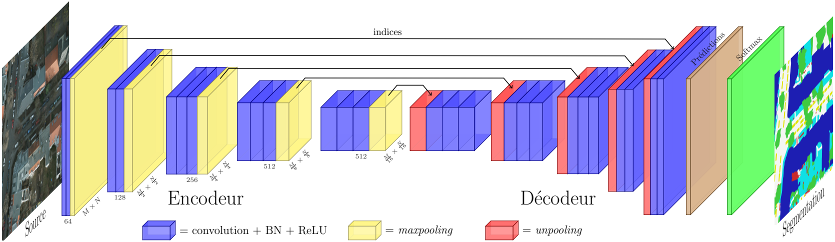


Badrinarayan et al., *SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation, TPAMI, 2017.*

- Symetrical architectures keep the input resolution
- Optimized on usual cross-entropy loss:

$$\mathcal{L}(softmax(z), y) = -\frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} \sum_{p=1}^{k} y_p^{(i,j)} \log \left(\frac{\exp(z_p^{(i,j)})}{\sum_{q=1}^{k} \exp(z_q^{(i,j)})} \right)$$

Symetrical architectures: SegNet, U-Net, DeconvNet...



Audebert et al., *Semantic Segmentation of Earth Observation Data Using Multimodal and Multi-scale Deep Networks* , *ACCV, 2016.*

- Symetrical architectures keep the input resolution
- Optimized on usual cross-entropy loss:

$$\mathcal{L}(softmax(z), y) = -\frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} \sum_{p=1}^{k} y_p^{(i,j)} \log \left(\frac{\exp(z_p^{(i,j)})}{\sum_{q=1}^{k} \exp(z_q^{(i,j)})} \right)$$

From multimedia to remote sensing images

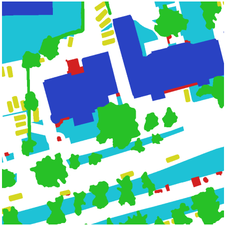
Aerial image (5000 × 5000)

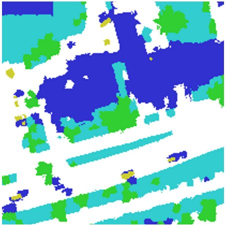
HD image (1920 × 1080)

ImageNet (256 × 256)

- Learn on random patches extracted from high resolution images
- Inference using a sliding window with overlap to smooth discontinuities along the edges
- Data augmentation with random rotations and flipping
- Initialization from pretrained VGG-16 weights (Simonyan et al., ICLR 2014)

A quick benchmark





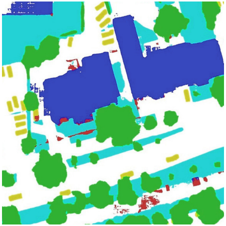


Image (RGB) Ground truth Segmentation + RF SegNet

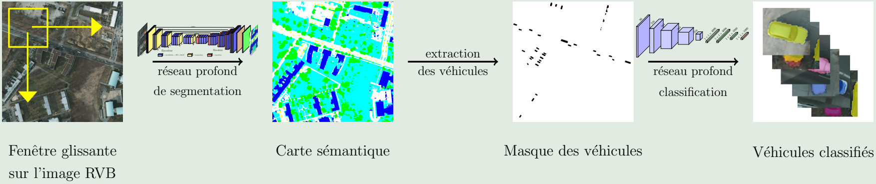
Results on the ISPRS Potsdam dataset[4] (F_1 score and accuracy).

Model	Roads	Buildings	Low veg.	Trees	Vehicles	Accuracy
Random Forest (RF)	77,0 %	79,7 %	73,1 %	59,4 %	58,8 %	74,2 %
FCN (Sherrah, 2016)	91,4 %	95,3 %	85,1 %	87,3 %	88,7 %	89,1 %
SegNet	93,0 %	92,9 %	85,0 %	85,1 %	95,1 %	89,7 %

[4] Rottensteiner et al., *The ISPRS benchmark on urban object classification and 3D building reconstruction*, ISPRS Annals, 2012

Object detection

Vehicle detection through segmentation



Segment-before-Detect: Vehicle Detection and Classification through Semantic Segmentation of Aerial Images, **N. Audebert**, B. Le Saux, S. Lefèvre, Remote Sensing, 2017

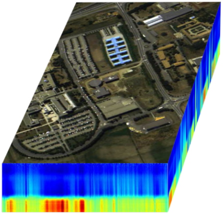
Faster-RCNN, YOLO, SSD...

Deep detection networks adapted for remote sensing

Zhang et al., *A modified faster R-CNN based on CFAR algorithm for SAR ship detection*, RSIP 2017

Cheng et al., *Learning Rotation-Invariant Convolutional Neural Networks for Object Detection in VHR Optical Remote Sensing Images*,

Hyperspectral images



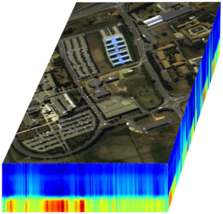
Motivation

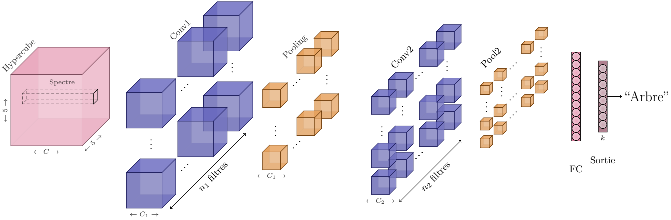
Hyperspectral image = hundreds of wavelength with high discriminative power

→ apply 3D CNN on the hypercube

Li et al., *Spectral–spatial classification of hyperspectral imagery with 3D convolutional neural network*, Remote Sensing, 2017
Chen et al., *Deep feature extraction and classification of hyperspectral images based on convolutional neural networks*, TGRS, 2016

Hyperspectral images





Motivation

Hyperspectral image = hundreds of wavelength with high discriminative power

→ apply 3D CNN on the hypercube

Li et al., *Spectral–spatial classification of hyperspectral imagery with 3D convolutional neural network*, Remote Sensing, 2017
Chen et al., *Deep feature extraction and classification of hyperspectral images based on convolutional neural networks*, TGRS, 2016

Multimodal learning for data fusion

RGB-D fusion in computer vision

Raster/raster fusion has been investigated in computer vision for RGB-D data:

- *Dual stream networks*

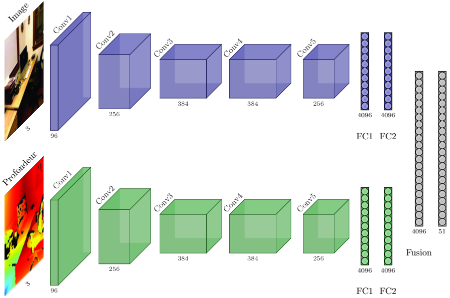
  (Simonyan et al., NIPS 2014, Eitel et al., IROS 2015)

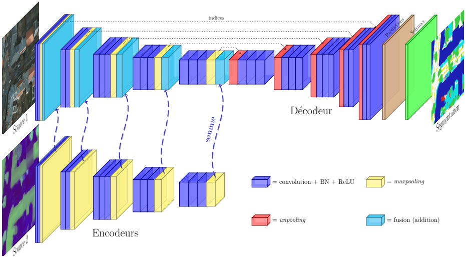
- Stochastic ensembles

  (Neverova et al., TPAMI 2015)

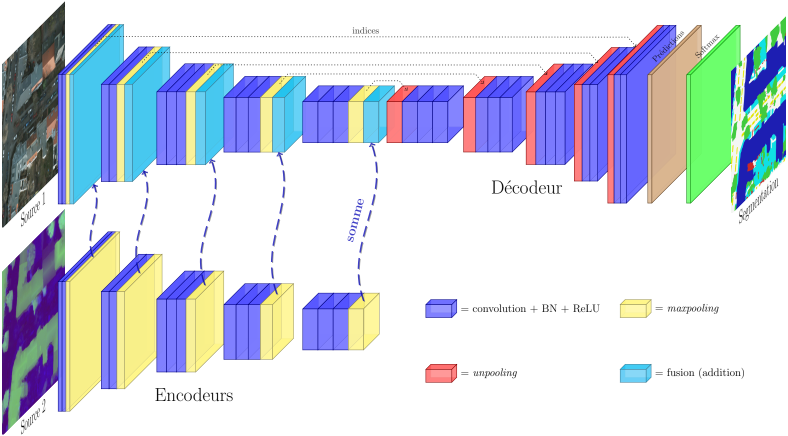
- Joint learning

  (Hazirbas et al., ACCV 2016)





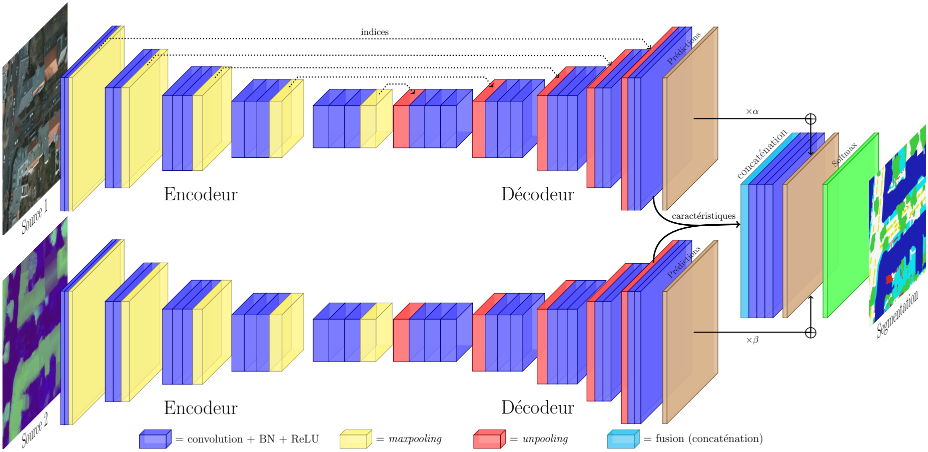
Joint learning



Multimodal fusion on both sensors (RGB and depth)

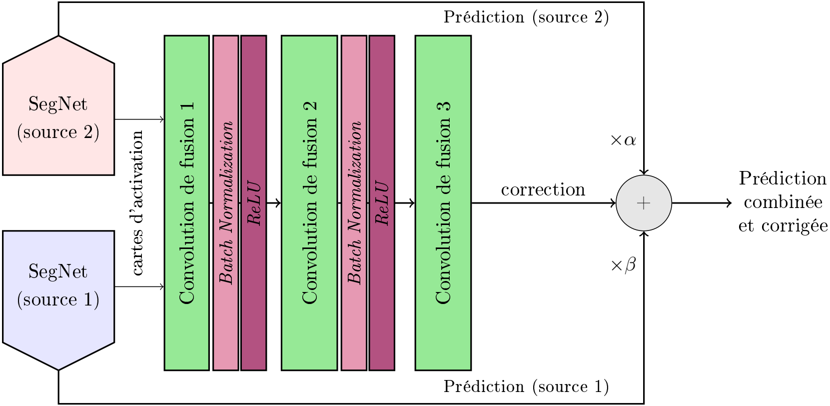
Hazirbas et al., *FuseNet: Incorporating Depth into Semantic Segmentation via Fusion-based CNN Architecture*, ACCV 2016

Late fusion by residual correction



Residual correction: one network by sensor + fusion network

Residual correction module



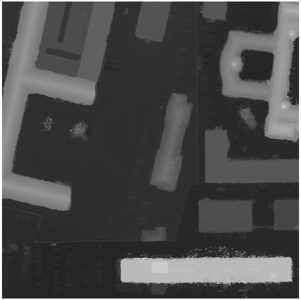
N. Audebert, B. Le Saux, S. Lefèvre, *Semantic Segmentation of Earth Observation Data Using Multimodal and Multi-scale Deep Networks*, ACCV 2016

RGB/DSM fusion

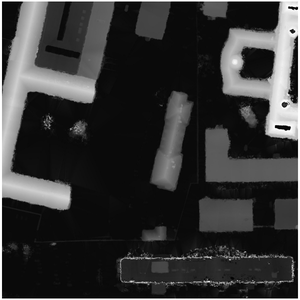
Context

Digital Surface Model can be computed from the Lidar point cloud.
How to take this information into account when mapping the RGB image?

RGB image



DSM



Normalized DSM

Résultats quantitatifs: ISPRS Potsdam

Model	Roads	Buildings	Low veg.	Trees	Vehicles	Accuracy
FCN + expert CRF (Liu et al., 2017)	91.2	94.6	85.1	85.1	92.8	88.4
FCN (Sherrah, 2016)	92.5	96.4	86.7	88.0	94.7	90.3
SegNet (IRRG)	92.4	95.8	86.7	87.4	95.1	90.0
SegNet-CR[5]	**93.3**	**97.3**	*87.6*	**88.3**	**95.8**	**91.0**
FuseNet	93.0	97.0	87.3	87.7	*95.2*	*90.6*
V-FuseNet	*93.2*	*97.2*	**87.9**	*88.2*	95.0	**91.0**

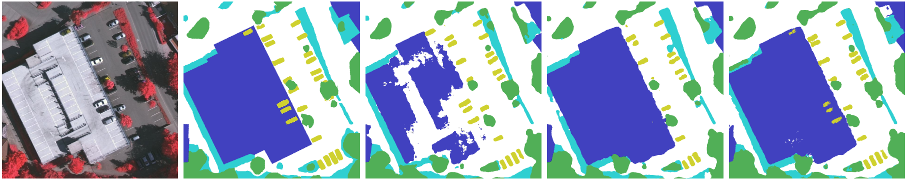
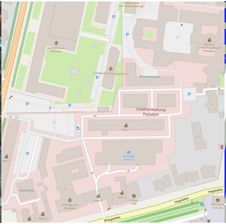


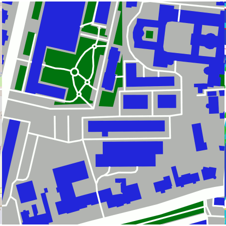
Image (IRRG) Ground truth SegNet (IRRG) FuseNet SegNet-CR

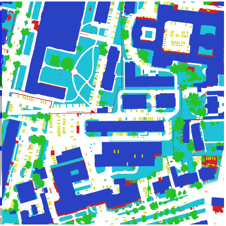
[5] CR: correction résiduelle

OpenStreetMap and color image fusion


Image


OSM


OSM raster


Ground truth

Prior geographical information

Data from OpenStreetMap establish a strong prior regarding the objects in the observed image, yet:

- OSM classes are necessarily those that we want to use,
- There is a temporal shift between OSM and the images.

Results on the ISPRS Potsdam dataset

Model	Roads	Buildings	Low veg.	Trees	Vehicles	Accuracy
SegNet (RGB)	93,0%	92,9%	85,0%	85,1%	95,1%	89,7%
SegNet-CR (RGB + OSM)	93,9%	92,8%	85,1%	**85,2%**	95,8%	90,6%
FuseNet (RGB + OSM)	**95,3%**	**95,9%**	**86,3%**	85,1%	**96,8%**	**92,3%**

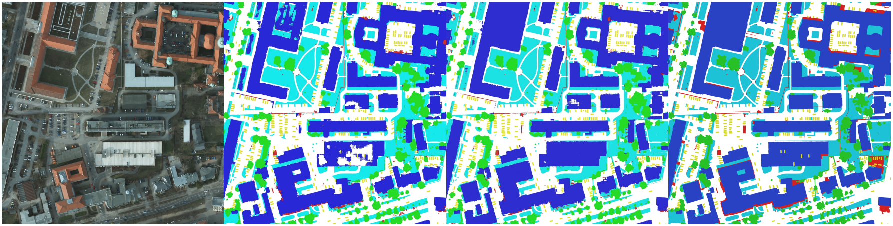
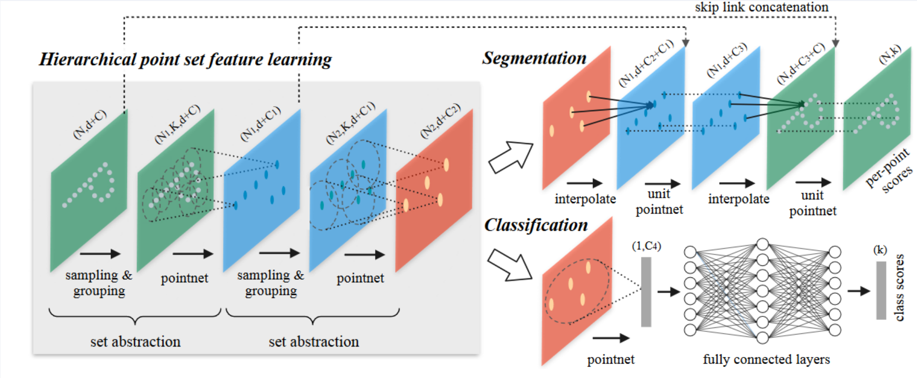


Image SegNet FuseNet (+ OSM) Ground truth

3D data

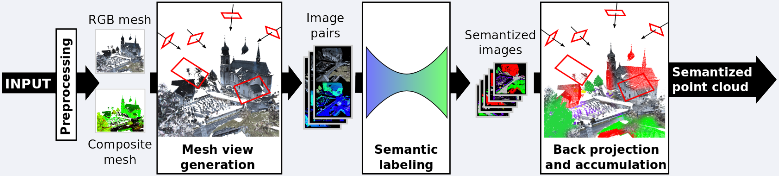
Point clouds

PointNet/PointNet++



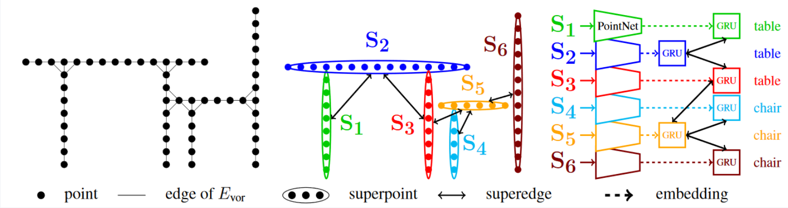
3D semantic segmentation

SnapNet



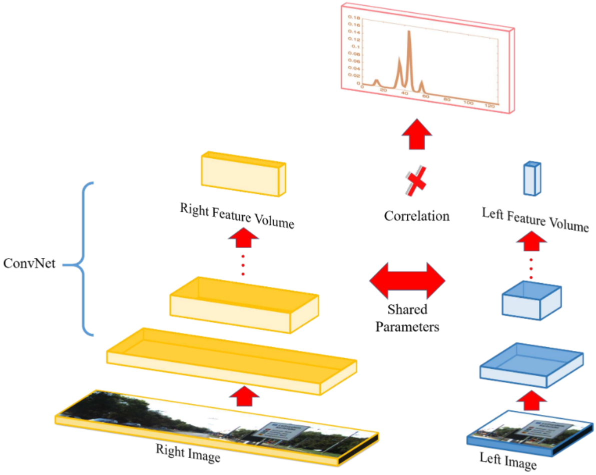
Boulch et al., *SnapNet: 3D point cloud semantic labeling with 2D deep segmentation networks*, Computer & Graphics, 2018

Superpoint Graphs



Landrieu and Simonovosky, *Large-scale Point Cloud Semantic Segmentation with Superpoint Graphs*, CVPR 2018

3D reconstruction: stereo matching



Luo et al., *Efficient Deep Learning for Stereo Matching*, CVPR 2016

Best practices

Frameworks

- TensorFlow/Keras (Google, Python/C++)
- PyTorch (Facebook, Python/C++)
- MXNet (Apache, C++, Julia, Python, R, ...)
- Chainer (Preferred Networks, Python)
- DeepLearning4j (Skymind, Java)
- CNTK (Microsoft, C#, C++, Python)
- MATLAB (Mathworks)
- ...

Fundamentals

All frameworks are designed for GPU computing, most leverage the CUDNN library from NVidia. Lots of wrappers (skorch, keras, tf.slim, ...) exist to ease the use of low-level libraries.

Datasets

Semantic segmentation:

- ISPRS Potsdam & Vaihingen, Zurich Summer: aerial VHR
- Onera Change Detection: multispectral satellite
- Inria Aerial Image Labeling: VHR building extraction
- Massachussets Buildings & Roads: VHR urban mapping
- IEEE GRSS Data Fusion Contest (2013 $\rightarrow$)
- SpaceNet, Dstl Kaggle, DeepGlobe...: multiple challenges

Object detection:

- XView: aerial multiple objects detection
- VEDAI: aerial vehicle detection

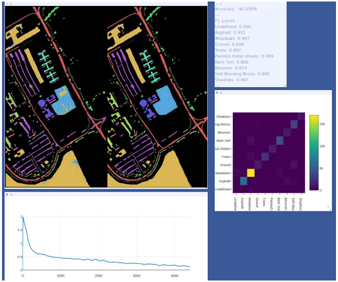
Research code

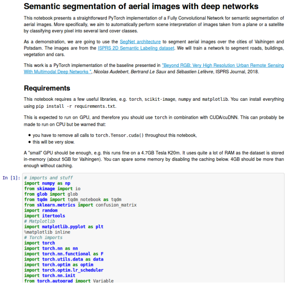
DeepHyperX, deep learning toolbox for hyperspectral classification:

https://gitlab.inria.fr/naudeber/DeepHyperX

Deep Nets for Earth Observation, pretrained networks for semantic segmentation of aerial and satellite color and multispectral images:

https://github.com/nshaud/DeepNetsForEO





Semantic segmentation of aerial images with deep networks

This notebook presents a straightforward PyTorch implementation of a Fully Convolutional Network for semantic segmentation of aerial images. More specifically, we aim to automatically perform scene interpretation of images taken from a plane or a satellite by classifying every pixel into several land cover classes.

As a demonstration, we are going to use the SegNet architecture to segment aerial images over the cities of Vaihingen and Potsdam. The images are from the ISPRS 2D Semantic Labeling dataset. We will train a network to segment roads, buildings, vegetation and cars.

This work is a PyTorch implementation of the baseline presented in "Beyond RGB: Very High Resolution Urban Remote Sensing With Multimodal Deep Networks", *Nicolas Audebert, Bertrand Le Saux and Sébastien Lefèvre*, ISPRS Journal, 2018.

Requirements

This notebook requires a few useful libraries, e.g. torch, scikit-image, numpy and matplotlib. You can install everything using pip install -r requirements.txt.

This is expected to run on GPU, and therefore you should use torch in combination with CUDA/cuDNN. This can probably be made to run on CPU but be warned that:

- you have to remove all calls to torch.Tensor.cuda() throughout this notebook,
- this will be very slow.

A "small" GPU should be enough, e.g. this runs fine on a 4.7GB Tesla K20m. It uses quite a lot of RAM as the dataset is stored in-memory (about 5GB for Vaihingen). You can spare some memory by disabling the caching below. 4GB should be more than enough without caching.

In [1]:
```
# imports and stuff
import numpy as np
from skimage import io
from glob import glob
from tqdm import tqdm_notebook as tqdm
from sklearn.metrics import confusion_matrix
import random
import itertools
# Matplotlib
import matplotlib.pyplot as plt
%matplotlib inline
# Torch imports
import torch
import torch.nn as nn
import torch.nn.functional as F
import torch.utils.data as data
import torch.optim as optim
import torch.optim.lr_scheduler
import torch.nn.init
from torch.autograd import Variable
```

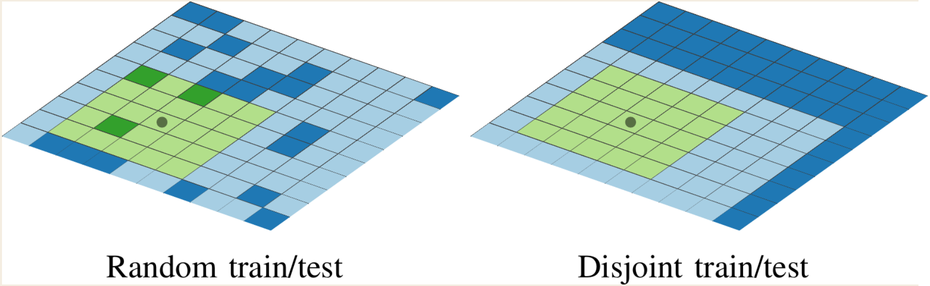
Dataset splitting

Train/validation/test (e.g. 80%/10%/10%)

- Tune hyperparameters and architecture on validation
- Assess final accuracy **once** on test
- Avoid uniform sampling

Why?

Realistic usecases help us avoid overfitting and optimistic metrics.



Network design

Keep it simple

- Start small, add layers until it starts overfitting
- SOTA models use stacked 3×3 convolutions with stride 1
- Use a non-saturing activation function, e.g. ReLU unless explicitly required (avoids vanishing gradients)
- Use standard losses, e.g. L_2 (regression), cross-entropy (classif.)

Tips for a better optimization

- Use Dropout in the fully connected layers (reduces overfitting)
- $\searrow$ learning rate when validation loss plateaues (no manual tuning)
- Use a standard optimizer, e.g. Adam or SGD + momentum
- Data augmentation goes far for small datasets
- Batch Normalization helps in most cases
- Choose the right initialization, fine-tune if possible

Validating the model

Choose a metric

- Overall accuracy is biased in unbalanced datasets
- F_1 score or *IoU* for pixelwise labeling
- Anything domain relevant...

Validating the model

- **Ideal:** Cross-validate over a K-fold train/val/test split
- Several random runs on the same test set
- Be skeptical if you reach 99% accuracy

Conclusion

Conclusion

Deep neural networks are excellent tools for automated remote sensing image processing.

- Deep networks are the new *de facto* state of the art for semantic mapping and object detection.
- Deep nets are applicable to EO-specific sensors such as multispectral and hyperspectral cameras.
- New research allows us to deal with 3D data and heterogeneous sensors.
- Tools and datasets are more and more easily available.

Conclusion

Deep neural networks are excellent tools for automated remote sensing image processing.

- Deep networks are the new *de facto* state of the art for semantic mapping and object detection.
- Deep nets are applicable to EO-specific sensors such as multispectral and hyperspectral cameras.
- New research allows us to deal with 3D data and heterogeneous sensors.
- Tools and datasets are more and more easily available.

Conclusion

Deep neural networks are excellent tools for automated remote sensing image processing.

- Deep networks are the new *de facto* state of the art for semantic mapping and object detection.
- Deep nets are applicable to EO-specific sensors such as multispectral and hyperspectral cameras.
- New research allows us to deal with 3D data and heterogeneous sensors.
- Tools and datasets are more and more easily available.

Conclusion

Deep neural networks are excellent tools for automated remote sensing image processing.

- Deep networks are the new *de facto* state of the art for semantic mapping and object detection.
- Deep nets are applicable to EO-specific sensors such as multispectral and hyperspectral cameras.
- New research allows us to deal with 3D data and heterogeneous sensors.
- Tools and datasets are more and more easily available.

Some topics not adressed here...

- Change detection and time series Russwurm and Körner, *Multi-Temporal Land Cover Classification with Sequential Recurrent Encoders*, Remote Sensing 2018
- Super-resolution and pansharpening Galliani et al., *Learned Spectral Super-Resolution*, ICCV 2017; Masi et al., *Pansharpening by CNN*, Remote Sensing 2016
- SAR image processing Gong et al., *Change detection in synthetic aperture radar images based on deep neural networks*, IEEE NNLS, 2016
- Generative models for data synthesis Audebert et al., *Generative Adversarial Networks for Realistic Synthesisof Hyperspectral Samples*, IGARSS 2018

Worthwhile references

- Zhu et al., *Deep Learning in Remote Sensing: A Review*, IEEE Geoscience and Remote Sensing Magazine, 2018.
- Goodfellow, Courville, Bengio, *Deep Learning*, MIT Press, 2015.
- PyTorch tutorials: `https://pytorch.org/tutorials/`
- TensorFlow tutorials: `https://www.tensorflow.org/tutorials/`

Questions and contact

Acknowledgements

Part of this work has been funded by the NAOMI project, a Total-ONERA collaboration.

Contact info

https://nicolas.audebert.at

nicolas.audebert@onera.fr

Expériences préliminaires

Transfert ImageNet → télédétection

- ImageNet: vie quotidienne (animaux, objets, personnes...)
  → symétrie gauche/droite, perspective et changements d'échelle
- Télédétection: structures au nadir (bâtiments, forêts, véhicules...)
  → équivariance à l'azimut, aucune perspective, échelle fixe

Les filtres appris sur ImageNet se transfèrent au moins partiellement sur des images de télédétection.[6,7]

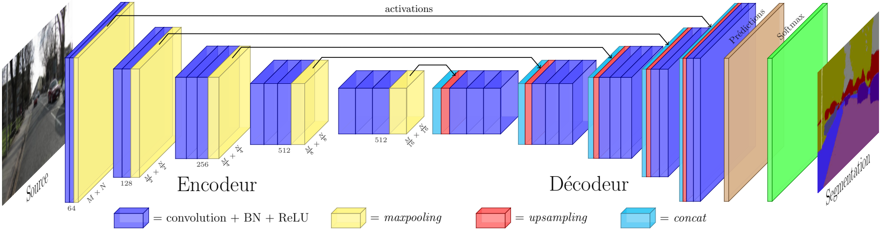
Comparaison de différentes initialisations (ISPRS Vaihingen)

Initialisation	Aléatoire	VGG-16 (ImageNet)			
Variabilité de l'encodeur $\frac{\alpha_e}{\alpha_d}$	1	1	0.5	0.1	0
Exactitude	87.0%	87.2%	**87.8%**	86.9%	86.5%

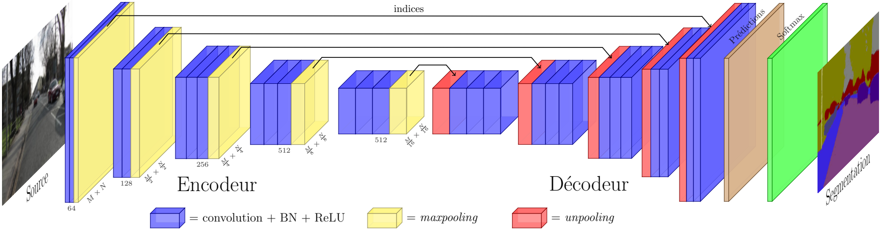
[6] Penatti et al., *Do Deep Features Generalize from Everyday Objects to Remote Sensing and Aerial Scenes Domains?*, CVPRW 2015

[7] Razavian et al., *CNN Features off-the-shelf: an Astounding Baseline for Recognition*, CVPRW 2014

SegNet/UNet



Ronneberger et al., *U-Net: Convolutional Networks for Biomedical Image Segmentation*, MICCAI 2015



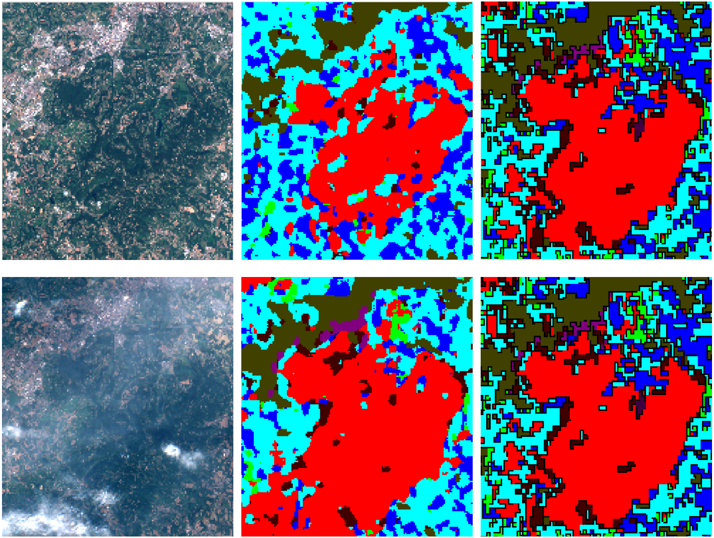
Badrinarayanan et al., *SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation*, TPAMI 2017, arXiv 2015

Extension au cas multispectral

Résultats

D1 (sans nuage)

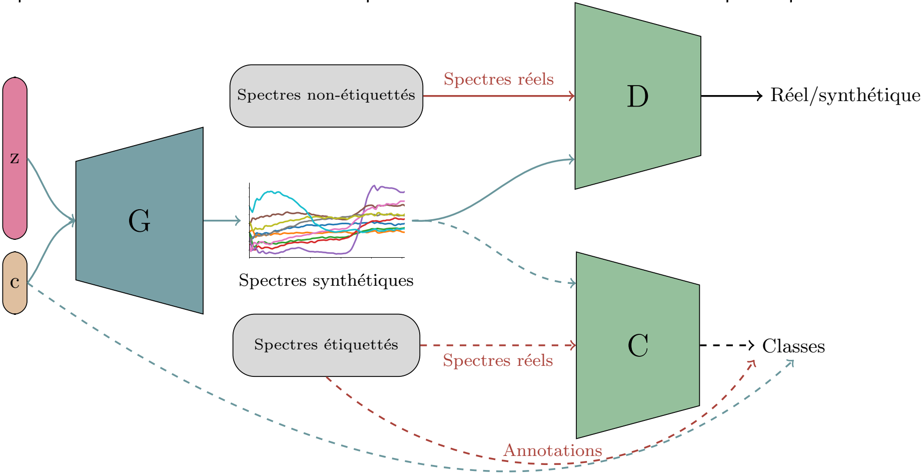
D2 (avec nuages)



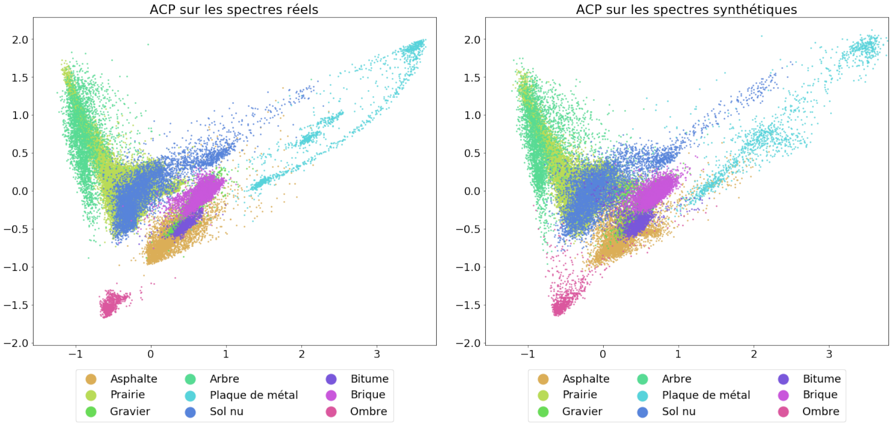
Couleur naturelle | Prédiction | Vérité terrain

Génération de données

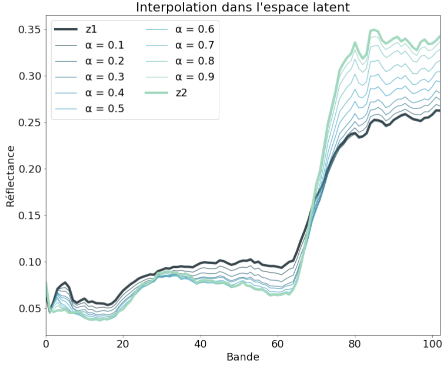
Idée: Générer des exemples d'apprentissage synthétiques, *i.e.* des spectres artificiels réalistes pour entraîner des réseaux plus profonds.

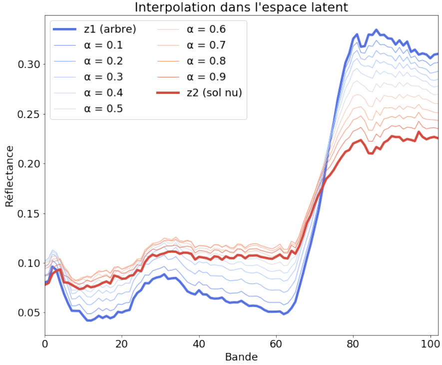


Comparaison avec la distribution réelle



Interpolations dans l'espace latent





Régularisation par carte de distances

Transformée de distance euclidienne

Idée

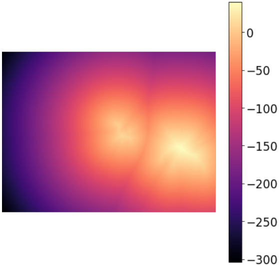
Estimer la transformée de distance euclidienne tronquée (Ye et al, ICPR 1998) pour chaque classe afin d'incorporer le voisinage spatial dans la fonction de coût.

$y[i, j] = (0, \dots, 0, 1, 0, \dots, 0)$ vs $d[i, j] = (-1, -0.3, \dots, 0.8, -1, \dots, -0.3)$

Transformée de distance euclidienne

Idée

Estimer la transformée de distance euclidienne tronquée (Ye et al, ICPR 1998) pour chaque classe afin d'incorporer le voisinage spatial dans la fonction de coût.

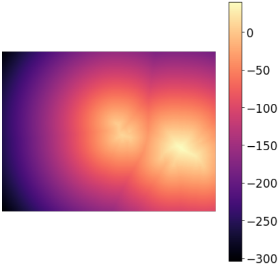


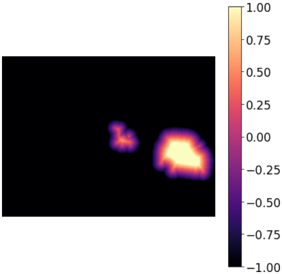
$$y[i,j] = (0, \ldots, 0, 1, 0, \ldots, 0) \quad \text{vs} \quad d[i,j] = (-1, -0.3, \ldots, 0.8, -1, \ldots, -0.3)$$

Transformée de distance euclidienne

Idée

Estimer la transformée de distance euclidienne tronquée (Ye et al, ICPR 1998) pour chaque classe afin d'incorporer le voisinage spatial dans la fonction de coût.



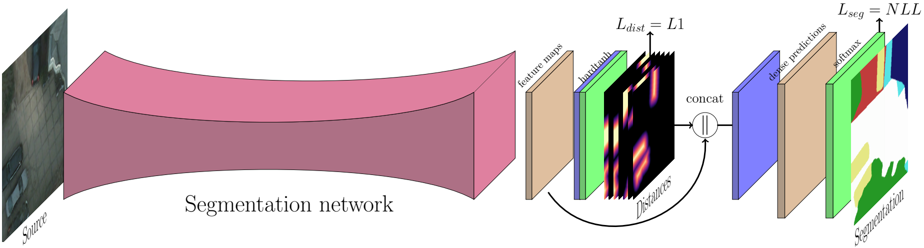


$$y[i,j] = (0,\dots,0,1,0,\dots,0) \quad \text{vs} \quad d[i,j] = (-1,-0.3,\dots,0.8,-1,\dots,-0.3)$$

Apprentissage multitâche

Architecture multitâche

La régression des CDS est utilisée comme proxy pour la classification.

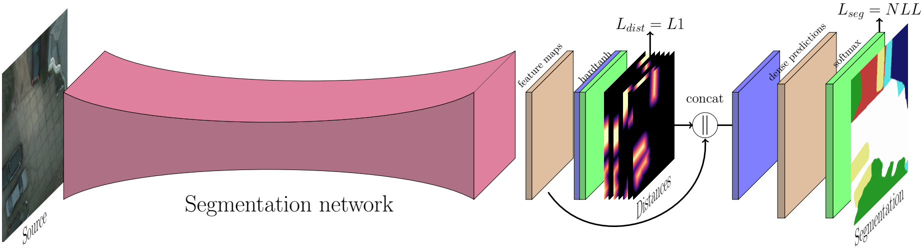


$$\mathcal{L}_{total} = \overbrace{\text{NLLLoss}(\text{softmax}(\hat{y}), \text{softmax}(y))}^{\text{entropie croisée}} + \underbrace{\lambda}_{\text{force de la régularisation}} \cdot \overbrace{|\hat{\mathcal{D}}_y - \mathcal{D}_y|}^{L_1 \text{ sur les distances}}$$

Apprentissage multitâche

Architecture multitâche

La régression des CDS est utilisée comme proxy pour la classification.



$$\mathcal{L}_{total} = \overbrace{\mathrm{NLLLoss}(\mathrm{softmax}(\hat{y}), \mathrm{softmax}(y))}^{\text{entropie croisée}} + \underbrace{\lambda}_{\text{force de la régularisation}} \cdot \overbrace{|\hat{\mathcal{D}}_y - \mathcal{D}_y|}^{L_1 \text{ sur les distances}}$$

Inria Aerial Image Labeling

- SegNet: 71.02% IoU (+6.98%), 95.63% OA (+0.89%)

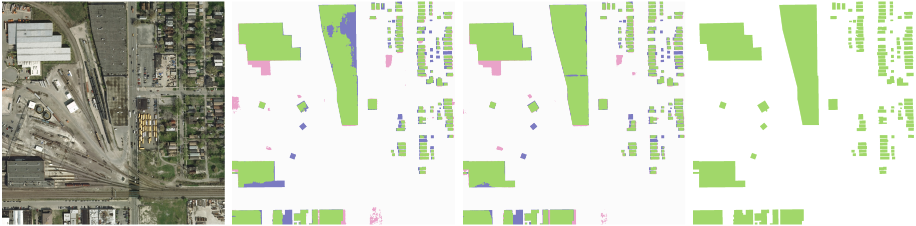


Image RVB | SegNet (classification) | SegNet (multitâche) | Vérité terrain

Vert: vrais positifs, rose: faux positifs, bleu: faux négatifs, blanc: vrais négatifs.