Deep learning for remote sensing – An introduction

Nicolas Audebert\textsuperscript{1, 2}, Bertrand Le Saux\textsuperscript{1}, Sébastien Lefèvre\textsuperscript{2}

\textsuperscript{1} ONERA, \textsuperscript{2} IRISA – Université Bretagne Sud

Atelier DLT Sageo – November 7th, 2018
Introduction
**Why Earth Observation?**

<table>
<thead>
<tr>
<th>Disaster management</th>
<th>Meteorology</th>
</tr>
</thead>
<tbody>
<tr>
<td>Emergency services organization, first responders...</td>
<td>Wind measurements, oceanic temperature monitoring, magnetic field study...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Climate change</th>
<th>Surveillance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ice cap melt, atmospheric pollution monitoring...</td>
<td>Urban expansion monitoring, illegal activities detection...</td>
</tr>
</tbody>
</table>
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.

---

$^1$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

Google Scholar

"deep learning" "remote sensing"

About 8,630 results (0.11 sec)

Deep learning for remote sensing data: A technical tutorial on the state of the art
L Zhang, L Zhang, B Du - ... Geoscience and Remote Sensing ..., 2016 - ieeexplore.ieee.org
Deep-learning (DL) algorithms, which learn the representative and discriminative features in a hierarchical manner from the data, have recently become a hotspot in the machine-learning area and have been introduced into the geoscience and remote sensing (RS) ...

When deep learning meets metric learning: remote sensing image scene classification via learning discriminative CNNs
G Cheng, C Yang, X Yao, L Guo... - ... and remote sensing, 2018 - ieeexplore.ieee.org
Remote sensing image scene classification is an active and challenging task driven by many applications. More recently, with the advances of deep learning models especially
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
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. \( \text{tanh} \) 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**

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**


Maggiori et al., *Fully Convolutional Neural Networks For Remote Sensing Image Classification*, IGARSS 2016
Symmetrical architectures: SegNet, U-Net, DeconvNet...

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

\[
\mathcal{L}(\text{softmax}(z), y) = - \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} \sum_{p=1}^{k} y_p^{(i,j)} \log \left( \frac{\exp(z_p^{(i,j)})}{\sum_{q=1}^{k} \exp(z_q^{(i,j)})} \right)
\]
Symmetrical architectures: SegNet, U-Net, DeconvNet...

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

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

From multimedia to remote sensing images

- 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

Results on the ISPRS Potsdam dataset\(^4\) (\(F_1\) score and accuracy).

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

**Vehicle detection through segmentation**


**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
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
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)

Late fusion by residual correction

Residual correction: one network by sensor + fusion network
Residual correction module

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

<table>
<thead>
<tr>
<th>Model</th>
<th>Roads</th>
<th>Buildings</th>
<th>Low veg.</th>
<th>Trees</th>
<th>Vehicles</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>FCN + expert CRF (Liu et al., 2017)</td>
<td>91.2</td>
<td>94.6</td>
<td>85.1</td>
<td>85.1</td>
<td>92.8</td>
<td>88.4</td>
</tr>
<tr>
<td>FCN (Sherrah, 2016)</td>
<td>92.5</td>
<td>96.4</td>
<td>86.7</td>
<td>88.0</td>
<td>94.7</td>
<td>90.3</td>
</tr>
<tr>
<td>SegNet (IRRG)</td>
<td>92.4</td>
<td>95.8</td>
<td>86.7</td>
<td>87.4</td>
<td>95.1</td>
<td>90.0</td>
</tr>
<tr>
<td>SegNet-CR(^5)</td>
<td>93.3</td>
<td>97.3</td>
<td>87.6</td>
<td>\textbf{88.3}</td>
<td>\textbf{95.8}</td>
<td>\textbf{91.0}</td>
</tr>
<tr>
<td>FuseNet</td>
<td>93.0</td>
<td>97.0</td>
<td>87.3</td>
<td>87.7</td>
<td>95.2</td>
<td>90.6</td>
</tr>
<tr>
<td>V-FuseNet</td>
<td>93.2</td>
<td>97.2</td>
<td>\textbf{87.9}</td>
<td>88.2</td>
<td>95.0</td>
<td>\textbf{91.0}</td>
</tr>
</tbody>
</table>

\(^5\)CR: correction résiduelle

---

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

<table>
<thead>
<tr>
<th>Model</th>
<th>Roads</th>
<th>Buildings</th>
<th>Low veg.</th>
<th>Trees</th>
<th>Vehicles</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>SegNet (RGB)</td>
<td>93,0%</td>
<td>92,9%</td>
<td>85,0%</td>
<td>85,1%</td>
<td>95,1%</td>
<td>89,7%</td>
</tr>
<tr>
<td>SegNet-CR (RGB + OSM)</td>
<td>93,9%</td>
<td>92,8%</td>
<td>85,1%</td>
<td>85,2%</td>
<td>95,8%</td>
<td>90,6%</td>
</tr>
<tr>
<td>FuseNet (RGB + OSM)</td>
<td>95,3%</td>
<td>95,9%</td>
<td>86,3%</td>
<td>85,1%</td>
<td>96,8%</td>
<td>92,3%</td>
</tr>
</tbody>
</table>

![Image](attachment:results_image.png)

Atelier DLT – 2018/11/07
Nicolas Audebert
3D data
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 Simonovsky, Large-scale Point Cloud Semantic Segmentation with Superpoint Graphs, CVPR 2018
3D reconstruction: stereo matching

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 →)
- SpaceNet, Dstl Kaggle, DeepGlobe...: multiple challenges

Object detection:

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

Random train/test

Disjoint train/test
**Network design**

### Keep it simple

- Start small, add layers until it starts overfitting
- SOTA models use stacked $3 \times 3$ convolutions with stride 1
- Use a non-saturating 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)
- ↓ learning rate when validation loss plateaus (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
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
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.
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.
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.
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 addressed here...

- Change detection and time series  Russwurm and Körner, *Multi-Temporal Land Cover Classification with Sequential Recurrent Encoders*, Remote Sensing 2018


- 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 Synthesis of Hyperspectral Samples*, IGARSS 2018
Worthwhile references

- PyTorch tutorials: [https://pytorch.org/tutorials/](https://pytorch.org/tutorials/)
- TensorFlow tutorials: [https://www.tensorflow.org/tutorials/](https://www.tensorflow.org/tutorials/)
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)

<table>
<thead>
<tr>
<th>Initialisation</th>
<th>Aléatoire</th>
<th>VGG-16 (ImageNet)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variabilité de l’encodeur $\frac{\alpha_e}{\alpha_d}$</td>
<td>1</td>
<td>1 0.5 0.1 0</td>
</tr>
<tr>
<td>Exactitude</td>
<td>87.0%</td>
<td>87.2% 87.8% 86.9% 86.5%</td>
</tr>
</tbody>
</table>

\(^6\) 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
Expériences préliminaires

Cas multispectral

Régularisation par carte de distances

SegNet/UNet

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

Extension au cas multispectral
Expériences préliminaires

Cas multispectral

Régularisation par carte de distances

Résultats

D1 (sans nuage)

D2 (avec nuages)

Couleur naturelle

Prédiction

Vérité terrain

Atelier DLT – 2018/11/07

Nicolas Audebert
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

ACP sur les spectres réels

ACP sur les spectres synthétiques

- Asphalte
- Arbre
- Bitume
- Prairie
- Plaque de métal
- Brique
- Gravier
- Sol nu
- Ombre
Interpolations dans l’espace latent

Interpolation dans l’espace latent

Interpolation 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.

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

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

\[ y[i, j] = (0, \ldots, 0, 1, 0, \ldots, 0) \quad \text{vs} \quad d[i, j] = (-1, -0.3, \ldots, 0.8, -1, \ldots, -0.3) \]
La régression des CDS est utilisée comme proxy pour la classification.

La fonction de perte totale est donnée par:

$$\mathcal{L}_{total} = \text{NLLLoss}(\text{softmax}(\hat{y}), \text{softmax}(y)) + \lambda \cdot |\hat{D}_y - D_y|$$

où $\lambda$ est la force de la régularisation et $\mathcal{L}_1$ est la distance $L_1$ sur les distances.
Apprentissage multitâche

Architecture multitâche

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

\[ L_{\text{total}} = \text{NLLLoss}(\text{softmax}(\hat{y}), \text{softmax}(y)) + \lambda \cdot |\hat{D}_y - D_y| \]

force de la régularisation

\[ L_{\text{dist}} = L1 \]

\[ L_{\text{seg}} = \text{NLL} \]

Segmentation network
Expériences préliminaires

Cas multispectral

Régularisation par carte de distances

Inria Aerial Image Labeling

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

![Image RVB](image1.png)

- SegNet (classification)
- SegNet (multitâche)
- Vérité terrain


Atelier DLT – 2018/11/07
Nicolas Audebert