DOCTORAL DISSERTATION FROM

THE UNIVERSITÉ BRETAGNE SUD

COMUE UNIVERSITÉ BRETAGNE LOIRE

ÉCOLE DOCTORALE N°601

MAThématiques et Sciences et Technologies de l'Information et de la Communication
Specialization: Computer Science

by

Nicolas AUDEBERT

titled

Classification of big remote sensing data

Defended at Palaiseau on October 17, 2018,
prepared at the Institut de recherche en informatique et systèmes aléatoires (UMR 6074),
and the Office national d'études et de recherches aérospatiales.

Thesis n°: 502

Rewievers before defense:

Jocelyn CHANUSSOT	Professor, Institut polytechnique de Grenoble – GIPSA-Lab
Vincent LEPETIT	Professor, Université de Bordeaux – LaBRI

Jury:

Jocelyn CHANUSSOT	Professor, Institut polytechnique de Grenoble – GIPSA-Lab	Rapporteur
Vincent LEPETIT	Professor, Université de Bordeaux – LaBRI	Rapporteur
Élisa FROMONT	Professor, Université de Rennes 1 – IRISA	Examinatrice
Patrick PÉREZ	Scientific director, Valeo.ai – Paris	Examinateur
Yuliya TARABALKA	Research, Inria Sophia Antipolis, HDR	Examinatrice

Advisor

Sébastien LEFÈVRE	Professor, Université Bretagne-Sud – IRISA

Supervisor

Bertrand LE SAUX	Researcher, ONERA Palaiseau – DTIS

Titre : Classification de données massives de télédétection

Mots clés : apprentissage profond, télédétection, segmentation sémantique, cartographie, réseaux de neurones

Résumé : L'observation de la Terre permet de modéliser et de comprendre son évolution. L'abondance d'images de télédétection aériennes et satellitaires nécessite la mise en œuvre de moyens d'analyse automatiques, capables d'interpréter ces données et de cartographier la surface du globe. Cette thèse traite de la conception, du déploiement et de la validation de stratégies d'apprentissage automatique, en particulier de réseaux de neurones convolutifs profonds, pour la compréhension d'images et la cartographie automatisée. Nous proposons des modèles pour l'interprétation d'images couleur, multispectrales et hyperspectrales, capables de prendre en compte les interactions spatiales entre entités géométriques et produisant des cartes d'une précision permettant la détection d'objets. Nous introduisons des architectures de fusion de données par apprentissage multi-modal et correction résiduelle afin de tirer parti des données ancillaires, comme les modèles numériques de terrain et les connaissances géographiques disponibles a priori. Enfin, nous étudions les capacités de généralisation de ces modèles dans des cas extrêmes de jeux de données limités ou massifs. Nous validons tout au long de cette thèse nos contributions sur de multiples jeux de données aériens et satellitaires pour la classification des sols et de leurs usages, l'extraction de bâtiments et la détection de véhicules.

Title: Classification of big remote sensing data

Keywords: deep learning, remote sensing, semantic segmentation, neural networks, mapping

Summary: Earth Observation allows us to modelize and understand the evolution of our planet. The profusion of aerial and satellite remote sensing images induces the need for automated tools able to semantize such raw data in order to map the Earth. This thesis studies the design, implementation and validation of machine learning strategies, specifically deep convolutional neural networks, for image understanding and automatic mapping. We introduce models for automated interpretation of color, multispectral and hyperspectral images, that are able to exploit spatial relationships between geometrical entities and to produce high precision maps relevant for object detection. We design data fusion architectures using multi-modal learning and residual correction that can leverage ancillary data, such as digital surface models and prior geographical knowledge. Finally, we study the generalization abilities of those networks for extreme cases of both limited and very large datasets. All along this work, we thoroughly validate our contributions on various aerial and satellite datasets for land cover and land use classification, building footprints extraction and vehicle detection.

Contents

List of Figures

List of Tables

Chapter

1 Introduction

[The Computer] was the first machine man built that assisted the power of his brain instead of the strength of his arm.

— Grace Hopper

Contents

1.1 Context

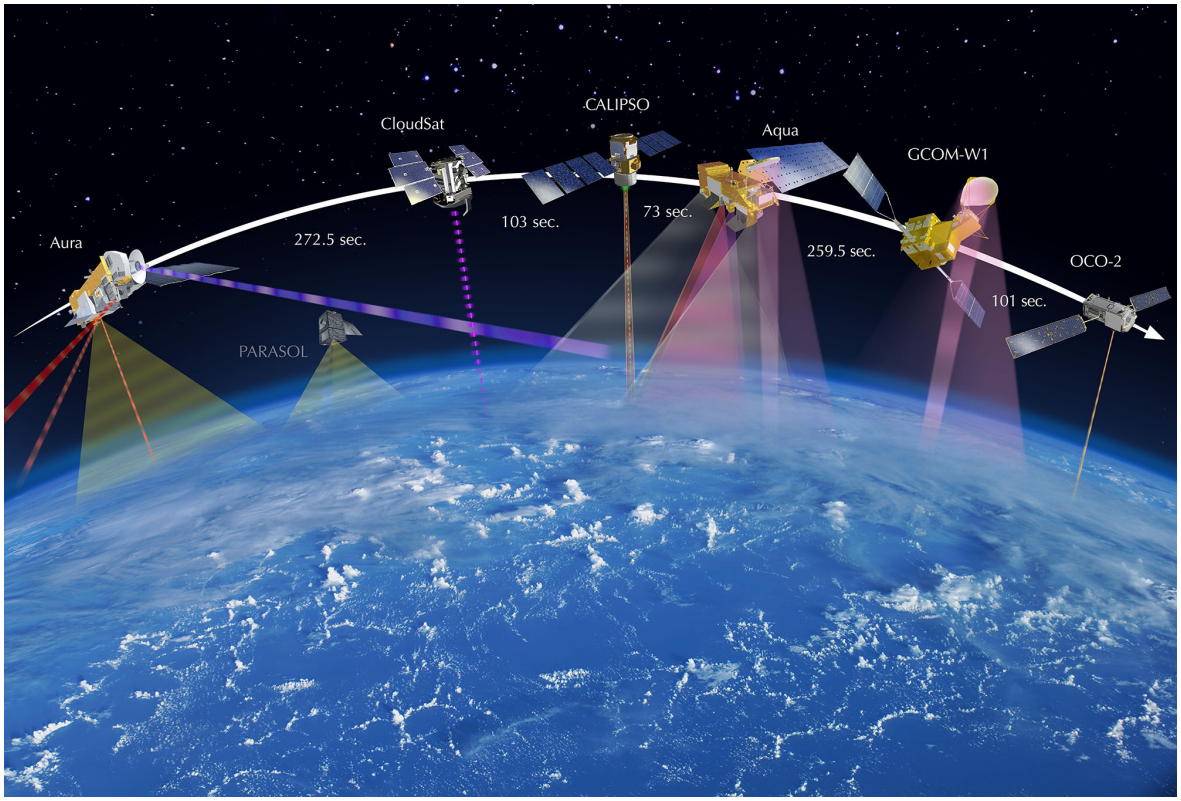


Figure 1.1: Earth Observation satellites from the A-Train constellation in 2018..
Credits: NASA JPL (public domain)

The scientific method is rooted in observation. The understanding of any object comes from a careful examination and Earth does not escape this fact. It is then not a surprise that, since the beginnings of the space race, the first satellite orbiting Earth were turned upon the Earth. Height allowed scientists to gain an entirely new perspective on our planet.

Aerial and satellite imaging is now ubiquitous in modern science. Understanding Earth is a major scientific challenge for which accurate observation is essentiel before any modelization attemp. Meterology, oceanography, ecology and geography all rely on the rich information conveyed by remote sensing data.

For these reasons, it is unsurprising that global-scale Earth imaging efforts intensified these last few years. Satellite constellations such as Landsat, SPOT (*Satellites Pour l'Observation de la Terre*), Sentinel or the A-Train (Fig. 1.1) fly over the planet 24/7. By themselves, Sentinel-2A and 2B acquire more than6Tbof data everyday and cover the entire globe in one week. However, leveraging this big data is far from easy. Image interpretation and scene understanding from remote sensing data more than often require both sensor-related and application-specific expert knowledge.

Yet, many fields would benefit from an automated Earth Observation data mining:

- **Ecology**: forest health monitoring, icecap melt tracking, early detection of oil leaks...
- **Meteorology**: weather forecast, disaster prevention (storms, typhoons), climate warming study...
- **Urban planning**: monitoring of urban and transport network expansions, emergency services organization after an earthquake...
- **Law enforcement**: ensure application of agricultural cycles, survey undeclared buildings, detect contraband ships and unauthorized fishing...

Despite efforts from French institutions such as *Institut national de l'information géographique et forestière* (IGN) and *Centre national d'études spatiales* (CNES), human photointerprets alone cannot process all this data and automation appears to be a seducing alternative. Delegating to machines the task of interpreting Earth Observation images would allow the community to multiply observations to generate knowledge and models. To do so, it is helpful to build on existing tools for artificial perception and image understanding. The current state of the art in computer vision mostly depends on deep artificial neural networks, which significantly overperform traditional approaches in image classification, object detection and segmentation. An ideal iterative mapping strategy based on Earth Observation data is illustrated by Fig. 1.2.

This thesis is structured as follow. We aim to design, implement and validate deep neural networks for automated interpreation of aerial and satellite images. The considered remote sensing data can be produced by multiple sensors on various scenes for several applications.

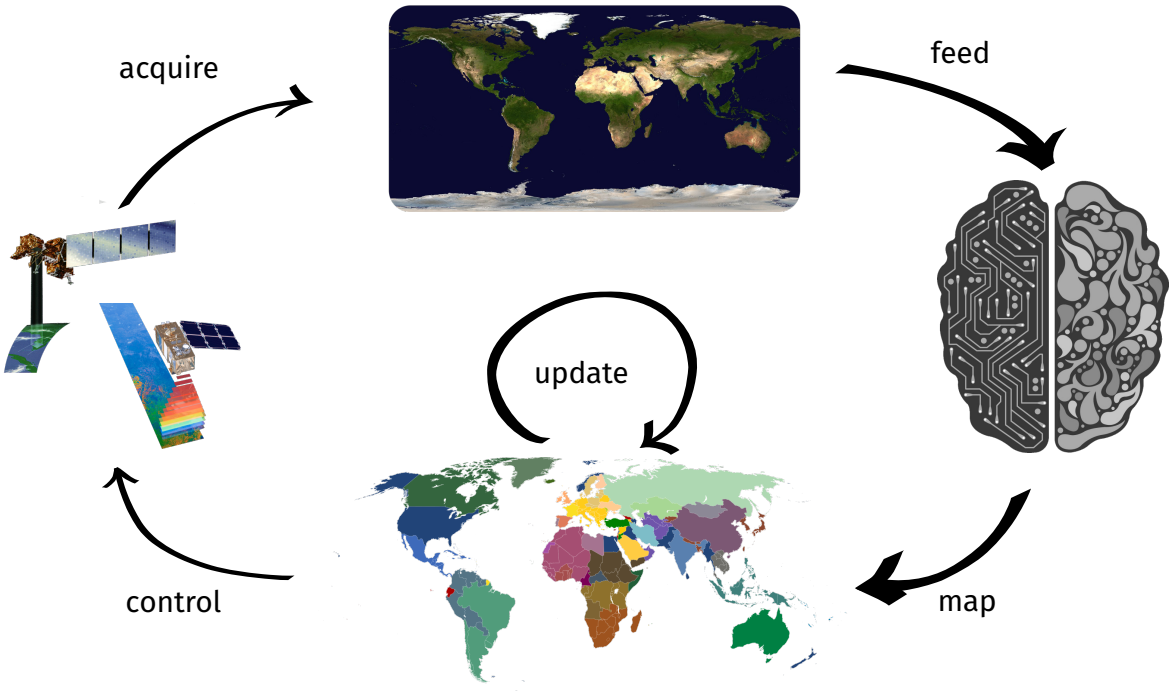


Figure 1.2: Automated iterative cartography based on Earth Observation data.

1.2 Field

This thesis stands at the meeting point of three subfields: remote sensing, computer vision and machine learning. Image interpretation through computer vision has produced a vast literature, even when restricted to aerial and satellite images. Recently, deep learning methods generated significant progress in image understanding. Nonetheless, most of this progress was focused on perception tasks from the everyday life., such as extracting knowledge from images and videos, either indoor or outdoor for smart homes, robotics, multimedia and autonomous driving. Although remote sensing data interpretation benefits from these works, it also has its own specificities both due to geometry, sensor and point-of-view.

1.2.1 Remote sensing data

Remote sensing data encompass a large variety of data acquired either by aircrafts or spacecrafts. Ideal observations are performed at nadir, i.e. perpendicular to the ground. In practice

the embarked sensor is never perfectly oriented, especially in satellites. A geometrical orthorectification step is often necessary to correct errors generated by sensor inclination, terrain relief and parallax.

In all cases, sensors measure the radiative energy emitted by the scene. Active sensors, such as radar or lidar, are their own signal source; they send an electromagnetic wave and capture its reflection. On the contrary, passive sensors measure either the radiative energy emitted by the scene itself (thermal sensors in the infrared) or the solar light it reflects (multispectral sensors). Those require an external light source.

Remote sensing sensors exist in many configurations and deal with much more physical phenomenon than consumer-grade cameras. Despite sensors peculiarities, the remote sensing image processing pipeline does not stem far from the one usually applied in multimedia. The task is the same in both cases: extracting information from images, i.e. computer vision. A community exists at the frontier between remote sensing and artificial vision that design algorithms for Earth Observation image processing.

1.2.2 Machine learning

Converting raw Earth Observation images into data requires automation. The sheer volume of data acquired every year by aerial and satellite sensors prevents human experts from processing it in real time.

Machine learning makes it possible to delegate knowledge extraction to the computer in order to automate it. In most cases, the task at hand is either estimating the value of a parameter (regression) or take a decision in a set of possibilities (classification).

In remote sensing image interpretation, human experts inspect the data to generate semantic maps, i.e. they choose for each area or object the category to which it belongs. In this context, we want to leverage machine learning to model this classification process in order to automate it.

This modelization relies on a training (or learning) phase during which the model builds its knowledge base using examples. When the learning is done, the statistical model can be applied on unseen data to generalize on new observations. The accuracy of this generalization is the critical point of the machine learning workflow. Two obstacles can trouble the model's generalization ability. First, if the model contains too many parameters compared to the number of training samples, then its knowledge will be purely a memorization of the examples: this is called overfitting. Second, if the model learns on many samples but with too few parameters, it will not be able to approximate the decision boundary: this is called underfitting. Finding the right number of parameters based on the training samples to obtain the best model is a challenge in itself.

The rise of deep learning since 2000 has largely renewed the statistical learning literature. Deep neural networks in particular, although designed and first implemented in the 60s, were perfectly in sync with the big data era. As large annotated datasets appeared, efficient parallel implementations of large neural networks allowed the state of the art in artificial perception to jump forward. Data and computing power made deep networks a reality, while training those same models on big data was simply out of reach for computers 50 years ago. Since 2012 and their success in the ImageNet challenge, deep convolutional neural networks have been established as the *de facto* current state of the art in image processing and gradually took over most artificial perception tasks.

1.2.3 Computer vision

Computer vision regroups all techniques designed for automated image interpreation. Since the 60s, artificial intelligence experts worked on emulating human sensorial abilities, starting with the most common of them: sight. As digital cameras reached a larger audience, image

processing went further and further both in automated correction on-device and offline post-processing using dedicated softwares.

The Grail of computer vision is a perfect simulation of the human brain to understand dynamic scenes based on visual cues only, especially identifying objects and their movements. These cognitive functions are necessary for autonomous navigiation and robotics but also benefit data mining. As more and more documents are digitized, online search of similar images or automated speech-to-text will need to rely on image processing steps.

One of the most longstanding task in computer vision is object recognition in images, dealing with both classification and localization. Many visual features have been designed by experts for various applications, from face detection to animal species classification and optical character recognition. The common ground of all these works is extracting meaning from pixels and getting semantics from non-structured image information is also what Earth Observation is about.

This thesis is therefore at the crossing of remote sensing, computer vision and machine learning. We will design, implement and validate deep learning techniques for automated interpretation of Earth Observation data.

1.3 Problem statement

The goal of this thesis is to introduce deep learning techniques for automatic mapping of the Earth by leveraging large volumes of images. More precisely, we aim to perform semantic segmentation of remote sensing images to generate land cover and land use maps. This breaks down in several questions we wish to address:

- What tools can we leverage for automated cartography using Earth Observation data?
- How to deal with multispectral, hyperspectral and Lidar sensors using deep networks?
- Can we leverage multiple observations on the same scene by different sensors to produce richer maps?
- Is is possible to extract spatially structured geographical information from those images?

First, there are many tools for image interpreation. Artificial neural networks are popular in the computer vision state of the art but their application to remote sensing is still new. We will need to start by studying how convolutional neural networks fare against traditional image processing techniques on Earth Observation images. In Chapter 2, we will remind the theorical principles behind deep learning and convolutional neural networks. In Chapter 3, we will show how the usual region-based classification strategies has reached its limit for semantic mapping of aerial images and how fully convolutional networks can efficiently replace it.

Howerver, as we have seen earlier, Earth Observation sensors vastly differ from the usual digital cameras. Moreover, optical sensors may be completed by Lidar sensors that do not measure the same physical properties. Indeed, Earth Observation often relies on hetereogeneous but complementary sensors to produce richer observations. In addition, many geographical databases are openly available and contain a knowledge yet to be used. Merging all this data to benefit from the joined strenghts of many sensors would be a significant advance for automated cartography. Consequently, Chapter 4 will extend results obtained on red-green-blue (RGB) data to multispectral and hyperspectral acquisitions, and digital elevation models rasterized from Light Detection And Ranging (Lidar) point clouds. In Chapter 5, we will introduce multimodal deep network architectures for data fusion, both to deal with heteregeneous sensors and to inject prior knowledge.

As generalizing statistical models is a critical roadblock, Chapter 6 will investigate how well deep convolutional networks perform on very large remote sensing datasets, especially since mapping the whole globe require robustness to environmental, temporal and weather variations. We will also study how to deal with small datasets in which few samples are labeled, and how generative models can help perform data augmentation to learn supervised models.

Finally, if segmentation of remote sensing images allows us to generate semantic maps, end users are often more interested in the relationships between geographical objects and entities. Object-based image analysis is a fundamental topic in remote sensing since it makes it possible to model structures at local and global scales. In Chapter 7, we will explore various strategies to enforce spatial structure to the pixel-wise semantic maps produced by our classification networks.

Chapter 8 will conclude this manuscript and discuss future research topics.

1.4 Contributions

This thesis is built on 4 major contributions.

1. We participate in establishing convolutional neural networks as the new state of the art for semantic segmentation of remote sensing data.

2. We show that convolutional deep networks can be extended to deal with all usual optical, such as multispectral and hyperspectral cameras.

3. We introduce novel multimodal architectures for fusion of several heterogeneous sensors and data inputs.

4. We validate our findings on large datasets which cover significant portions of the globe.

These works have been published in several publications:

Published works in international peer-reviewed journals

Nicolas Audebert, Bertrand Le Saux, and Sébastien Lefèvre. "Segment-before-Detect: Vehicle Detection and Classification through Semantic Segmentation of Aerial Images". In: *Remote Sensing* 9.4 (Apr. 13, 2017), p. 368. DOI: 10.3390/rs9040368

Nicolas Audebert, Bertrand Le Saux, and Sébastien Lefèvre. "Beyond RGB: Very High Resolution Urban Remote Sensing with Multimodal Deep Networks". In: *ISPRS Journal of Photogrammetry and Remote Sensing* (Nov. 23, 2017). ISSN: 0924-2716. DOI: 10.1016/j.isprsjprs.2017.11.011

Nicolas Audebert, Bertrand Le Saux, and Sébastien Lefèvre. "Deep Learning for Classification of Hyperspectral Data: A Comparative Review". In: *IEEE Geoscience and Remote Sensing Magazine* in press (Mar. 2019)

Published works in international peer-reviewed conferences

Nicolas Audebert, Bertrand Le Saux, and Sébastien Lefèvre. "How Useful Is Region-Based Classification of Remote Sensing Images in a Deep Learning Framework?" In: *2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. July 2016, pp. 5091–5094. DOI: 10.1109/IGARSS.2016.7730327

Nicolas Audebert, Bertrand Le Saux, and Sébastien Lefèvre. "Semantic Segmentation of Earth Observation Data Using Multimodal and Multi-Scale Deep Networks". In: *Computer Vision – ACCV 2016*. Springer, Cham, Nov. 20, 2016, pp. 180–196. DOI: 10.1007/978-3-319-54181-5_12

Nicolas Audebert, Bertrand Le Saux, and Sébastien Lefèvre. "Fusion of Heterogeneous Data in Convolutional Networks for Urban Semantic Labeling". In: *2017 Joint Urban Remote Sensing Event (JURSE)*. Mar. 2017, pp. 1–4. DOI: 10.1109/JURSE.2017.7924566

Nicolas Audebert, Bertrand Le Saux, and Sébastien Lefèvre. "Joint Learning from Earth Observation and OpenStreetMap Data to Get Faster Better Semantic Maps". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. Honolulu, United States, July 2017, pp. 1552–1560. DOI: 10.1109/CVPRW.2017.199

Nicolas Audebert, Bertrand Le Saux, and Sébastien Lefèvre. "Generative Adversarial Networks for Realistic Synthesis of Hyperspectral Samples". In: *2018 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. July 2018, pp. 5091–5094

Nicolas Audebert et al. "A Real-World Hyperspectral Image Processing Pipeline for Vegetation and Hydrocarbon Characterization". In: *Proceedings of the 9th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*. Sept. 2018

Chapter

2 Related work

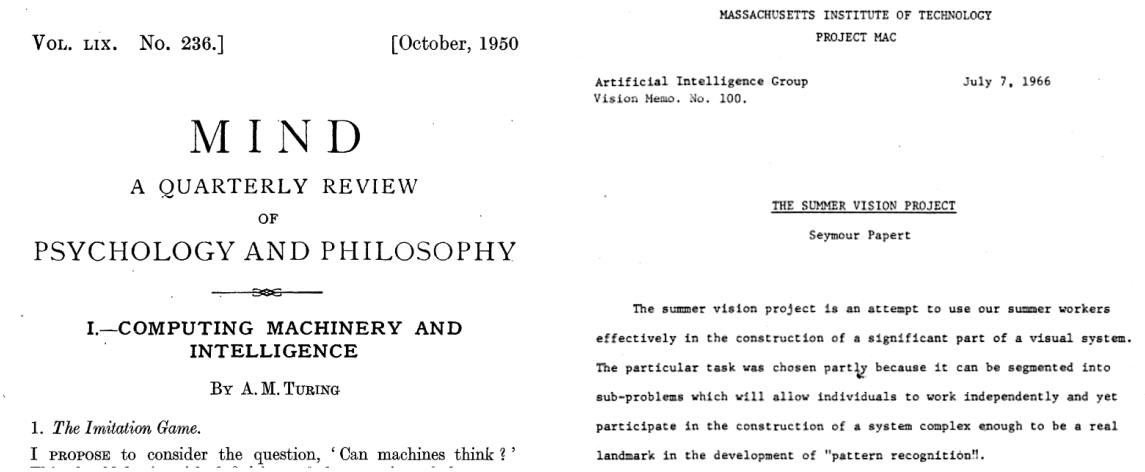
An algorithm must be seen to be believed.

— Donald Knuth

Contents

Summary:

This chapter consists in a introduction to the theory of deep learning for computer vision, on which the rest of this manuscript relies. We begin with a brief history of the motivations and the seminal works behind modern artificial neural networks, with a focus on convolutional neural networks. Then, we study more in depth the applications of those networks for computer vision and more specifically for semantic segmentation. Finally, we do a short tour of classical techniques for remote sensing image classification using machine learning, highlighting the specificities of these data.



Vol. lix. No. 236.] [October, 1950

MIND

A QUARTERLY REVIEW

OF

PSYCHOLOGY AND PHILOSOPHY

I.—COMPUTING MACHINERY AND INTELLIGENCE

By A. M. Turing

1. *The Imitation Game.*

I propose to consider the question, 'Can machines think?'

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

PROJECT MAC

Artificial Intelligence Group July 7, 1966

Vision Memo. No. 100.

THE SUMMER VISION PROJECT

Seymour Papert

The summer vision project is an attempt to use our summer workers effectively in the construction of a significant part of a visual system. The particular task was chosen partly because it can be segmented into sub-problems which will allow individuals to work independently and yet participate in the construction of a system complex enough to be a real landmark in the development of "pattern recognition".

Figure 2.1: Introductions of Turing [167] and Papert [131], two documents which pioneered artificial intelligence and computer vision.

The foundations of this thesis lay on many anterior works in computer vision, artificial intelligence and Earth Observation. In this chapter, we draw the scientific framework on which we will be able to build later. To begin, we will present the fundamental principles of deep learning and the mathematical background of artificial (convolutional) neural networks. Then, we will detail how these statistical models are used for image understanding in a task called semantic segmentation. Finally, we will discuss the modern approaches for remote sensing image understanding using machine learning and their peculiarities related to Earth Observation data.

2.1 Deep learning for computer vision

2.1.1 A brief history of deep learning

Giving cognitive abilities to computers was first theorized by Alan Turing in 1950 [167] (cf. Fig. 2.1). In *Computing Machinery and Intelligence*, Turing designs a theorical experiment that could answer to the following question: "can a machine think?". According to Turing, an intelligent computer would be defined by its ability to mimick a human, so that other individuals would be unable to discern its true nature. However Turing does not look into "how" a computer could achieve this goal, and therefore leaves the problem unsolved.

Yet, as early as 1943, Warren McCulloch and Walter Pitts already introduced artificial boolean neuron systems [112] with two states: active and inactive. They define a neuron as an automata associated to a transfer function that transform a set of inputs into an output value. Some neurons do not receive any input from another neuron but hold in themselves the input signal. Other neurons compute logical circuits based on their inputs. McCulloch and Pitts [112] prove that many predicates from temporal logic can be computed by these boolean networks. More importantly, an extension of this theory by Stephen Kleene studies cyclic boolean networks, i.e. *recurrent* networks. Kleene [82] proves that these networks, which are actually finite state automata, can model any regular language[1].

Concurrently, neurpsychologist Donald Hebb studies the cognitive mechanisms that allow the brain to learn. He introduces the Hebbian learning theory, according to which a connection between two neurones strengthens every time they are simultaneously activated [65]. Hebb also sugggests that neurons cluster themselves into "cell assemblies" for which the activation are synchronized. These clusters would therefore code a neuronal representation of the signals the brain receives. As we will see, these two ideas have been since a formidable source of inspiration for artificial intelligence and statistical learning.

[1] I.e. any language defined by a regular expression.

In 1957, Frank Rosenblatt defines the *perceptron* [141], an acyclic neural network similar to those from McCulloch and Pitts [112]. Inputs and outputs are boolean values and the network has only one layer. The weights of the connections between neurons are automatically determined using Hebb's rule [65]. At the same time, Bernard Widrow builds the *Adaptive Linear Neuron* (ADALINE) [179], a computer based on memistors also inspired by McCulloch and Pitts.The ADALINE has a design close to the perceptron: a linear network with one layer operating on the weighted sum of its inputs. However, Widrow automatically adjust the weights using a gradient descent algorithm that minimizes the summed squared error. Yet these two models have a significant drawback. Although the perceptron is guaranteed to find an optimal border between the data, this holds true only if the data is linearily separable. Indeed the ADALINE and the perceptron are linear classifiers and cannot solve problems where the data is not linearily separable. In the *Perceptrons* book, Minsky and Papert [115] prove that a perceptron with one hidden layer cannot reproduce the XOR function, despite its simplicity and so independently of the number of neurons used. At the time, there is no suitable policy to automatically find the optimal weights of a perceptron with several hidden layers that could possess an non-linear behaviour. Artificial neural networks are abandoned for several years. In his PhD thesis defened in 1975 [178], Paul Werbos introduces a gradient descent algorithm to minimize the error of a multilayer neural networks that leverages the derivation theorem of composed function – the *chain rule*. Werbos names this the *backpropagation* algorithm. Still, ten years are needed before gradient backpropagation is used to train multilayer perceptrons [143, 91].[2]

The theory of feed-forward neural networks becomes interesting again, especially the multilayer perceptrons. Cybenko [34] proves in 1989 the universal approximation theorem that states that the set of the functions computable by a perceptron is a dense set of piecewise continuous functions, provided that the activation function is the sigmoid. This result is extend by Hornik [69] to all usual activation functions two years later. The formal statement is given below:

Theorem 1. *Let φ be a bounded function, monotically increasing and not constant. Let C_0^n denote the set of continuous functions defined on $[0,1]^n$. Then:*

$$\forall \epsilon > 0, \forall F \in C_0^n, \exists N \in \mathbb{N}^*, \text{ real numbers } v_i, b_i \in \mathbb{R} \text{ and vectors } \mathbf{w}_i \in \mathbb{R}^n \text{ where } i \in [\![1, n]\!] \text{ such as}$$

$$\hat{F} : \mathbf{x} \rightarrow \sum_{i=1}^{N} v_i \varphi\left(\mathbf{w}_i^t \mathbf{x} + b_i\right)$$

is an ϵ-approximation of F, *i.e.:*

$$\forall \mathbf{x} \in [0,1]^n, \ \left|F(\mathbf{x}) - \hat{F}(\mathbf{x})\right| < \epsilon \, .$$

This means that any smooth function (piecewise continuous on a set of compact spaces) can be approximated with an arbitrary precision by a perceptron. It proves that artificial neural networks can reproduce nearly any function, altough it does not give any construction method. Outside of theorical results, practical uses of neural networks start to appear for artificial vision and pattern recognition. Handwritten character recognition, especially digits and letters, is particularly popular. In 1980 Fukushima [52] introduces the *Neocognitron*, a multilayer perceptron with a bioinspired structure taking its roots in the works from Hubel and Wiesel [75, 76] on cats' and monkeys' visual cortex. The *Neocognitron* extracts local features from the image that are robust to small disturbances. These features are combined using a cascade architecture. Thanks to this structure, the model can learn from

[2] The ADALINE is also transformed in a multilayer variant: the MADALINE [180], which uses a specific optimization algorithm. Indeed, MADALINE uses sign activation function whose derivative is zero nearly everywhere. Widrow and Lehr converge two years later with a Madaline structure based on the sigmoid activation, trainable with backpropagation.

the pixel values, but also recognize local patterns and their variations, which mimicks how the mammal brain works [96]. In 1989, LeCun et al. [92] propose a multilayer perceptron architecture for handwritten digit recognition whose first layer is *convolutional* and trained by backpropagation. They build on this principle to design the LeNet-5 model [94], the first modern Convolutional Neural Network (CNN). In 2004 Convolutional Neural Network (CNN)-based object detection and recognition approaches are competitive – and occasionally superior – to other approaches su as pixel-based Support Vector Machine (SVM). The first works using CNN-learnt representations to replace usual *ad hoc* image features such as SIFT (Scale-Invariant Feature Transform) [106] and Histograms of Oriented Gradients (HOG) [35] for object classification appear in the 2000s [150, 72].

In 2006 Hinton and Salakhutdinov [67] introduce autoencoder neural networks that can compress a dataset by embedding the samples in a space with a lower dimensionality. Their approach for dimension reduction use a stack of Restricted Boltzmann Machines (RBM) [1, 145] trained layerwise iteratively. This hybrid model is explored in an article from 2006 [66] which names it Deep Belief Networks (DBN). Bengio et al. [10] extend this layerwise pretraining to DBN for regression a year later. Their work suggest that pretraining initializes deeper layers based on better representations of the abstract features compared to a random initialization. Yoshua Bengio argues that a good machine learning algorithm should be able to learn relevant semantic features at various levels of abstraction based on data both labeled and unlabeled, i.e. in a unsupervised setting [8]. He defends deeper models as more expressive thanks to their ability to learn representations based on data and justifies it with recent progress in neurosciences in understanding the visual cortex [151]. The introduction non-saturating activation functions such as Rectified Linear Unit (ReLU) [56] makes it easier to optimize a network without pretraining by alleviating the exploding gradient problems which made the training of very deep networks practically impossible.

2006 was also the year of the first CNN implented on Graphics Processing Unit (GPU) [22] for automated document processing, unsupervised DBN training [139] and optical character recognition [27]. In 2011, Dan Cireşan used several CNN methods to reach the first place in two challenges: chinese character recognition [99] and traffic sign classification [160]. These CNNs also obtained state of the art performances in latin character recognition and object classification in small images from the CIFAR-10 dataset [26]. The ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) image classification challenge starts in 2010 and is based on the ImageNet dataset [40]. One million images are annotated for a thousand classes of interest. In 2012 the challenge is won by Krizhevsky, Sutskever, and Hinton [84] using a GPU implementation of the new AlexNet convolutional neural network, using the Compute Unified Device Architecture (CUDA) library. AlexNet reaches a 15% *top-5*[3] error-rate, while the second best method only achieves 26%. This unexpected success is commonly pointed as the beginning of the renewed popularity of deep networks and deep learning in general in the computer vision community. The ILSVRC competition has been won every year by CNN-based techniques for object recognition, localization and segmentation. The effectiveness of deep convolutional networks since 2012 is due to an alignement of three factors: fundamental progress (ReLU, convolutional networks) that allowed researchers to design deeper networks, the availability of new large annotated datasets for supervised learning (e.g. ImageNet) and efficient GPU implementation that made computations tractable.

In the following, we draw the theoretical framework of artificial neural networks and their optimization for various tasks. We then focus on convolutional models.

2.1.2 Artificial neural networks

The formal definition of an artificial neuron was introduced by McCulloch and Pitts [96] in 1959. A neuron with a transfer function φ operates on a set of n input neurones all of

[3] A *top-5* prediction is rich if the actual label is in the set of the five first predictions given by the model.

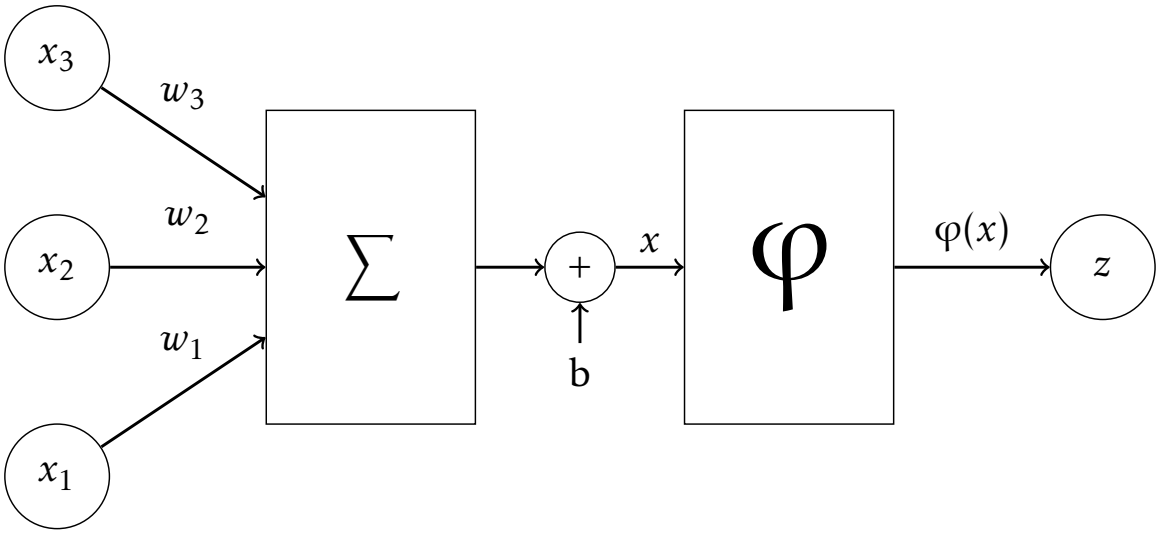
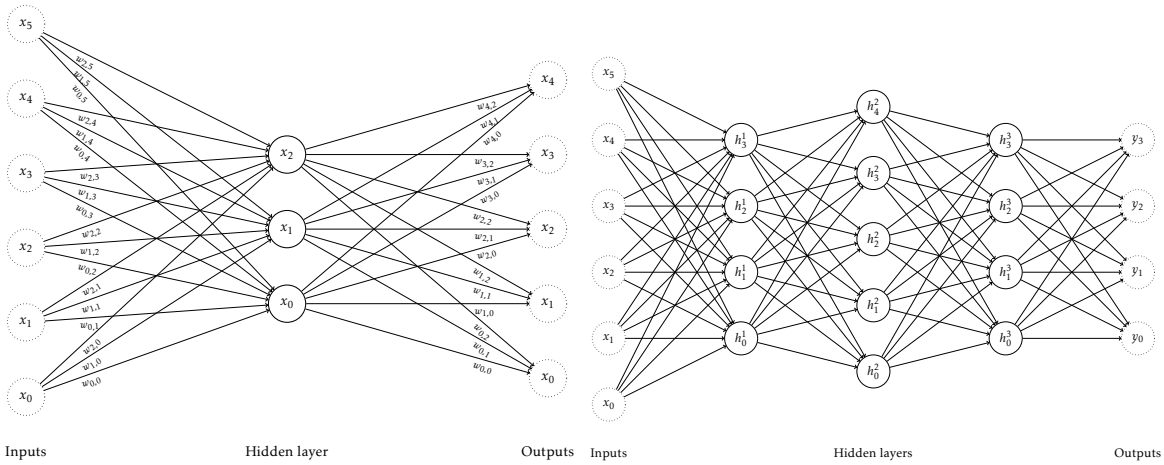


Figure 2.2: Model of an artificial neuron.



(a) Perceptron with one hidden layer.

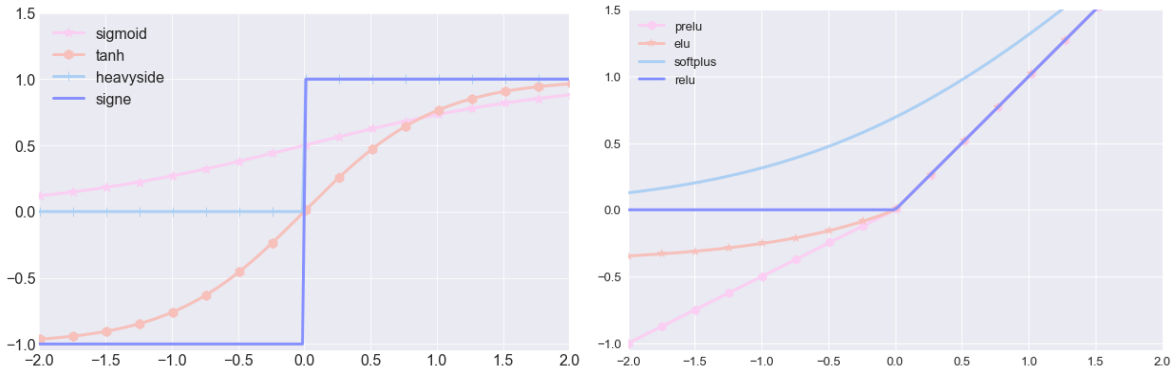
(b) Multilayer perceptron.

Figure 2.3: Perceptron with one and several hidden layers. Inputs and outputs can have any dimension and are pictured as neurons.

which emit a signal $x_1 \dots x_n$. The neuron is connected to its inputs by synapses of weight w_i. The input signal x the neuron receives is the weighted sum of the input signals from each neuron, based on the synaptic weights. The neuron outputs a signal $z = \varphi(x)$ i.e. the image of its input by its tranfer function. Fig. 2.2 illustrates this model. The activation of a neuron is given by the formula:

$$z = \varphi\left(\sum_{i=1}^{n} w_i x_i + b\right). \tag{2.1}$$

Several neurons can be connected to each other, forming an oriented weighted graph. A feed-forward neural network is an acyclic neural graph. In practice these graphs are k-partite: neurons can be grouped in "layers" that connect to each other. For simplicity, inputs and outputs of a multilayer perceptron are placed in specific layers. The actual learnable weights are synaptic connections for which at least one end neuron is in the "hidden layers". These hidden layers are the interface between intput and output. These networks have a fixed topology and are parametrized by the set of synaptic weights. Multilayer perceptrons with one or more hidden layers are described in Fig. 2.3. A layer for which all input neurons are connected to all neurons from the next layer is called "fully connected". These layers are one of the main layer type one can encounter in modern deep networks.



(a) Saturating activation functions.

(b) Non-saturating activation functions.

Figure 2.4: Examples of activation functions.

Many activation functions can be used in neural networks. The only requirements are non-linearity – otherwise the network would reduce to a perceptron – and that its derivative exists nearly everywhere to train with the gradient backpropagation algorithm. The activation φ is often chosen so that φ and its derivative φ' are monotically increasing. Several commonly used activation function are pictured in Fig. 2.4a:

- **sigmoid**, or logistic, function: $\sigma(x) = \frac{1}{1+e^{-x}}$.
- **hyperbolic tangent**: $\tanh(x) = \frac{1-e^{-2x}}{1+e^{-2x}}$.
- **Heaviside step function**: $\mathrm{H}(x) = 0$ if $x < 0$ and 1 if $x \geq 0$.
- **sign** function: $\mathrm{sign}(x) = +1$ if $x > 0$ and -1 if $x < 0$.

The Heaviside step and the sign functions have null gradients nearly everywhere since their derivative is the Dirac δ function. This makes these functions unpractical and rarely used since the backpropagation algorithm does not apply. The sigmoid function was commonly used despite the vanishing gradients problem it entailed. While not specific to the sigmoid, it is particularly strong in reccurent neural networks [68]. The multiplication of consecutive layers in the network entails a geometric evolution of the gradient norm during backpropagation. The cumulative product of n gradients through the contractive activation function (derivative < 1) produces a $n+1$þgradient with a smaller amplitude, and so on. On the contrary, gradients can explode with an exponentially increasing amplitude for certain activation functions. This problem worsens with saturating functions such as sigmoid or hyperbolic tangent since their gradient are bounded in $[0,1]$. Some works either encouraged or discouraged the use of these functions. LeCun et al. [93] recommended to use a tweaked hyperbolic tangent $f(x) = 1.7159\tanh(\frac{2}{3}x)$ because it is bounded by $[-1,+1]$ and centered in 0, which is suitable for normalized centered data.

Current activation functions are non-linear and non-saturating to avoid vanishing gradients. Indeed they are considered state of the art since Glorot, Bordes, and Bengio [56] introduced the ReLU and the SoftPlus functions for deep networks – adapting the idea of rectified linear units for DBN [118]. They analyzed how these activation functions influenced the network training and drew three conclusions. First, networks non-saturating activation functions generalize better than models using *tanh*. Second, networks trained with ReLU do not require an unsupervised layerwise pretraining which speeds up greatly the learning phase. Finally, these models are often sparser than their usual equivalents. ReLU has been widely adopted since it is simple to implement and efficient to compute $(\max(0,x))$.

Overall most activation functions commonly used are continuous, monotonically increasing, conctractive and pointwise, although all these hypotheses might not be actually required [125]. Several variants have been introduced around the idea of rectified linear units,

such as a parametrized counterpart with an $\alpha > 0$ slope in the negative part, either chosen by the user (*Leaky ReLU* [107]) or learnable (Parametrized Rectified Linear Unit (PReLU) [63]). An everywhere-differentiable alternative has also been introduced: Exponential Linear Unit (ELU) [29]. These declinations are illustrated in Fig. 2.4b:

- **ReLU** : $\mathrm{ReLU}(x) = \max(0, x)$.
- **SoftPlus** : $s^+(x) = \ln(1 + e^x)$.
- ***Leaky ReLU*** : $\mathrm{LReLU}_\alpha(x) = max(0, x) - \alpha max(0, -x)$, where α is a hyperparameter.
- **PReLU** : $\mathrm{PReLU}(x, \alpha) = max(0, x) - \alpha max(0, -x)$, where α is learnable.
- **ELU** : $\mathrm{ELU}_\alpha(x) = x$ if $x > 0$ and $\alpha(exp(x) - 1)$ otherwise.

The universal approximation theorem [34, 69] states that the set of functions computable with a perceptron is a dense set in the piecewise continuous (on compact spaces) function set. Another way to frame it is to say that any function $f : \mathrm{E} \to \mathbb{R}^m$ where $\mathrm{E} = \bigcup_k \mathrm{C}_k$ is a union of compact subspaces of $\mathbb{R}^n$, continuous on every compact space, can be approximated with an arbitrary precision $\epsilon > 0$ by a perceptron. Howerver this statement comes with two strong limitations. On the one hand, the theorem generalized by Hornik only covers bounded and monotically increasing activation function, which excludes rectified linear functions such as ReLU. Sonoda and Murata [158] removed this limitation and proved that the universal approximation theorem still holds for unbounded activation functions.

On ther other hand, there is a fundamental limitation to the theorem. Altough it guarantees that there is a set of parameters that can approximate the requested function, it does not give any method to find such a set. Nothing ensures that these weights are reachable to gradient descent, for example, and there is no indication whatsoever to the network structure that could achieve the approximation. The theorem holds on shallow networks with one hidden layers while practical successes are obtained with networks that exhibit an ever-increasing depth. Especially, deeper networks can approximate more complex functions using less neurons compared to perceptrons [11, 114]. The hierarchical structure of multilayer networks seems to be particularly suited to approximated composed functions while working around the curse of dimensionality [136]. Howerver this complexifies the network architecture and adds many new hyperparameters that one has to fiddle with. As there is no automated construction strategy for deep networks, trial-and-error remains the main way to design a deep network – or meta-learning in some promising works [193].

2.1.3 Training a deep network

Since there is no systematic way to find the optimal weights of an artificial neural networks, we need to look for optimization heuristics. The backpropagation algorithm [178, 143, 91] relies on the famous gradient descent to find the right weights. While nothing ensures that the local minima found this way are equivalent to the optimal weights hinted at by the universal approximation theorem, this is the best practical method we can rely on.

The gradient descent algorithm [20] is applied to the model to minimize the total error by updating the synaptic weights. This "strongest slope" algorithme approximates local minima of a differantiable function f by looking for stationary points, i.e. points where its gradient is zero. The fundamental principle is that f decreases more quickly in the opposite direction to its gradient and works as described in the following.

Definition 1. *Gradient descent algorithm:*

Let $f : \mathbb{R}^n \to \mathbb{R}$ be a differentiable function and ∇f its gradient. Let $x_0 \in \mathbb{R}^n$ be an initial point, $\epsilon > 0$ a tolerance threshold and $\alpha > 0$ the descent rate. We define the sequence $(x_i)_{i \geq 0} \in \mathbb{R}^{\mathbb{N}}$ such as:

$$x_{i+1} = x_i - \alpha \nabla f(x_i) \, .$$

The algorithm stops when $\nabla f(x_i) \leq \epsilon$ and returns x_i.

In deep neural networks, the function to be minimized is called the "cost function" or "loss function" denoted $\mathcal{L}$ in the following. Most of the time, $\mathcal{L}$ is a measure of the total error the model makes when predicting over the full training set Ω. Optimizing the network is finding an approximation of the solution to the equation:

$$W^* = \text{argmin}_W \mathcal{L}(W, \Omega) \tag{2.2}$$

where the model is parametrized by its synaptic weights $W = \{w_1, \ldots, w_m\}$ and a cost function $\mathcal{L}$.

We can apply the gradient descent algorithm as long as the los function $\mathcal{L}$ is differentiable. Indeed, we compute the update to apply to the weights by propagating the gradient value from a layer to its predecessor: this is the backpropagation algorithm [178, 93, 143]. The weights update in the opposite direction of the gradient error with respect to these weights $\nabla_W \mathcal{L}(W, \Omega)$.

Definition 2. *Gradient descent algorithm applied to a neural network:*

1. *Assign random values to the weights* W.
2. *Compute* $\nabla_W \mathcal{L}(W, \Omega)$ *on the whole dataset.*
3. *While* $\nabla_W \mathcal{L}(W, \Omega) > \epsilon$ *:*
   - $W := W - \alpha \nabla_W \mathcal{L}(W, \Omega)$

Practically speaking, the training datasets can contain millions of samples and Ω can be quite large. For this reason, we generally use an online version of the algorithm: the *stochastic* gradient descent. This version performs a weight update for each training sample by approximating the average error based on the error on one sample.

Definition 3. *Stochastic gradient descent algorithm:*

1. *Assign random values to the weights* W.
2. *While the stopping criterion is not reached:*
   - *Randomly choose a sample* $\omega \in \Omega$
   - $W := W - \alpha \nabla_W \mathcal{L}(W, \omega)$

The algorithm stops when the stopping criterion is verified, often after a predefined number of iterations.

Yet, estimating the gradient $\nabla_W \mathcal{L}(W, \omega)$ based on a unique sample is noisy and the descent can suffer from large direction changes between consecutive iterations. To stabilize the descent and ease the convergence, we mostly use the *mini-batch* stochastic gradient descent. The global error on the dataset is then estimated on a batch (or *mini-batch*) of samples, i.e. averaged over a group of k samples:

Definition 4. *Batch stochastic gradient descent algorithm:*

1. *Assign random values to the weights* W.
2. *While the stopping criterion is not reached:*
   - *Randomly choose* k *training samples* $(\omega_1, \ldots, \omega_k) \in \Omega^k$
   - $W := W - \alpha \frac{1}{k} \sum_{i=1}^{k} \nabla_W \mathcal{L}(W, \omega_i)$

The algorithm stops when the stopping criterion is verified, often after a predefined number of iterations.

Since the update is applied on all layers, we need to compute $\frac{\partial \mathcal{L}}{\partial w_i}$ for all weight vectors w_i that parametrize the i^{th}layer. Howerver directly computing the gradient $\nabla \mathcal{L}$ is possible only on the last layer. To rewind the computation up to the partial derivative with respect to the previous layers, we can use the backpropagation algorithm. This algorithm actually uses the chain rule to compute the derivative of composed functions [38, 87]:

Theorem 2. *Let f et g be two functions such as $f : \mathrm{I} \to \mathrm{J} \subset \mathbb{R}$ and $g : \mathrm{J} \to \mathbb{R}$. Let $x \in \mathrm{I}$ such as f has a derivative in x. Then, the function $h = g \circ f : \mathrm{I} \to \mathbb{R}$ has a derivative in x which value is:*

$$h'(x) = (g \circ f)'(x) = f'(x) \times g'(f(x)) \,.$$

If f and g are differentiable on I *and* J *respectively, then:*

$$(g \circ f)' = f' \times (g' \circ f) \,.$$

or using Leibniz notation with $z = g(y)$ and $y = f(x)$:

$$\frac{\mathrm{d}z}{\mathrm{d}x} = \frac{\mathrm{d}z}{\mathrm{d}y} \times \frac{\mathrm{d}y}{\mathrm{d}x} \,.$$

This theorem still holds true for partial derivative of multivariate functions in $\mathbb{R}^n$.

To decrease the error, we can update the weights w^k in the opposite direction of the gradient $\frac{\mathrm{d}e}{\mathrm{d}w^k}$. If z^k denotes the output activations of the k^{th}layer, the chain rule gives,:

$$\frac{\partial \mathcal{L}}{\mathrm{d}w^k} = \frac{\partial \mathcal{L}}{\mathrm{d}z^k} \times \frac{\partial z^k}{\partial w^k} = \frac{\partial \mathcal{L}}{\partial z^{(k+1)}} \times \frac{\partial z^{(k+1)}}{\partial z^k} \times \frac{\partial z^k}{\partial w^k} \,.$$

This means that we can go backward into the network from the deepest layers to the earliest ones to backpropagate the gradient $\frac{\partial \mathcal{L}}{\partial w^k}$. To compute the error gradient with respect to the weights of a specific layer, we need to compute the gradient of its outputs with respect to its weights and the gradient of its outputs with respect to its inputs. This step is called the backward pass.

Actual neural networks do not operate on scalar values but on tensors $\mathbf{x}, \mathbf{y}, \mathbf{z}$. Nonetheless the chain rule still applies: we only need to rewrite it using the jacobian matrices $\mathbf{J}$:

$$\mathbf{J}_{\mathrm{F} \circ \mathrm{G}} = \mathbf{J}_{\mathrm{F}} \circ \mathrm{G} \cdot \mathrm{J}_{\mathrm{G}}$$

and the backpropagation algorithm still applies.

This is for this reason that vanishing and explosive gradients are a problem. The sequence of consecutive gradient norms in the backward pass becomes nearly geometrical due to the successive multiplications. If the jacobian norm is mostly less than 1, the gradient norm goes to 0. Convergence is either slow or impossible. If the norm is more than 1, then the gradient increase exponentially and the weight updates become unstable. In practice, we will look for normalized jacobians, especially during initialization [147].

The model aims to approximate a specific function $\mathcal{F}$. To do so we introduce a proxy loss function $\mathcal{L}$ that measures the approximation error of the network, such as:

$$\mathcal{L}(\widehat{\mathcal{F}_{\mathrm{W}}}(x) - \mathcal{F}(x)) \to 0 \Rightarrow \hat{\mathcal{F}} \to \mathcal{F} \quad ,$$

i.e. minimizing the loss entails model convergence to the actual function.

The exact loss function to use depends on the task at hand. For regerssion problems – $\mathcal{F}$ takes its values in a continuous set – we often use a distance in the function space such as the L_1 or L_2 norm. For each sample we compare the prediction $\hat{y}$ to the actual label y with

$$\mathrm{L}_1(y, \hat{y}) = |\hat{y} - y|$$

$$\text{ou} \quad L_2(\hat{y}, y) = \|\hat{y} - y\| \ .$$

Using the L_2 norm means approximating $\mathcal{F}$ using the least square error method. The L_1 norm is often more robust to outliers which can be explosive when using the euclidean distance. In comparison, the L_2 distance is differntiable everywhere and is more tolerant to small errors – since it is contractive on $[-1, +1]$. Hybrid losses such as the Huber loss can mix both for an improved robustness.

When $\mathcal{F}$ is discrete – e.g. for classification – y is encoded in the *one-hot* fashion. For a multi-label classification problem with n classes, if y belongs to class k then the label is represented by $y_i = \delta_{i,k}$ where δ is the Kronecker delta. This means that y is encoded with the pattern $(0, \dots, 0, 1, 0, \dots, 0)$, i.e. zeroes everywhere except for the component number k. The same cost functions could still be used but we often prefer the cross-entropy:

$$H(z, y) = -\sum_{i=1}^{n} y_i \log(z_i) \ . \tag{2.3}$$

The cross-entropy is especially interesting since minimizing its value also means minimizing the Kullback-Leibler divergence between two statistical distributions: $\hat{y}$ and y, i.e. the image of $\mathcal{F}$ and the image of $\hat{\mathcal{F}}$. The required condition is that $\hat{y}$ is a probability vector, i.e. $\hat{y}_i \in [0, 1]$ and $\sum_i \hat{y}_i = 1$. To achieve this, we can feed the output activations into the *softmax* function:

$$\hat{y}_i = z_i = softmax(x)_i = \frac{\exp(x_i)}{\sum_j \exp(x_j)} \tag{2.4}$$

which generalizes the sigmoid to multiple classes.

Let us stress that the gradient descent algorithm ensures convergence only if $\mathcal{L}$ is convex which is almost never the case for non-trivial deep networks. Several variations of the stochastic gradient descend have been introduced to improve its convergence properties. The descent step α in Definition 1 plays an imoprtant role in the optimization of deep neural networks. It is often named learning rate, since α controls the amplitude of the weight updates. If α is too high, each update will be large and convergence will be unstable. If α is too low, it slows down convergence and the algorithm can get stuck in poor local minima or saddle points. Variations of the gradient descent algorithm introduce specific heuristic to update the weights. The "momentum" methods are inspired by the kinetic energy conservation mechanical principle. The gradient "speed" – its norm – is partially kept between each iteration to reduce oscillations around the level sets of the error surface [138, 120]. Sutskever et al. [161] showed that momentum stochastic gradient descent improved the model accuracy even with poor initializations. Polyak and Juditsky [137] suggest to use an asynchronous mini-batch gradient descent algorithm that use the moving average of the last n gradients as an estimation of the weights update.

Other variations introduce heuristics to adjust the learning rate α during training. Indeed, nothing requires that α stays constant in the gradient desent algorithm. For example, one could manually adjust α during the training phase, for example by dividing it by a constant $\gamma < 1$ after some number of iterations. Bottou [16] recommends to use an averaged stochastic gradient descent with an evolutive learning α that follows $\alpha_{i+1} = \alpha_0(1+\gamma \cdot i)^{-1}$, while Loshchilov and Hutter [105] use a derivative of the simulated annealing algorithm. Howerver this introduces yet another manual hyperparameter to configure before training. Several works have therefore looked into adaptive moment methods in which α is automatically adjusted during training based on various heuristics [42, 166, 185, 81].

Independently from the gradient descent flavor used, the weights initialization is a critical step for the network optimization. The convergence properties and the performance of the optimal weights found depend at least partially on the initialization. If unsupervised layerwise pretrained was once common [66, 10], deep networks are nowadays mostly trained end-to-end in a supervised fashion. A good initialization strategy should assign random

values to the weights while avoiding vanishing and exploding gradients. Glorot and Bengio [55] and He et al. [63] introduced an initialization producing activations that follow a normal distribution which eases the training and generates reasonable gradients. Saxe, McClelland, and Ganguli [147] initialize convolutional kernels using random orthogonal matrices to keep the activation norm constant from one layer to the other and to decorrelate the initial filters.

Because of the stochastic nature of the neural network optimization, the deep learning community agregated several best practices for practical training [93, 9, 16]. As for shallow model training, it is common to center and normalize the input data. Image are often centered by substracting the mean pixel values computed on the whole dataset. In some cases, especially for images that have a common layout, the mean image is substracted. Normalizing the standard deviation is less common altough it can happen in some papers.

During training, it is recommended to shuffle the data after each epoch on the dataset to avoid cycles in the gradient descent [93]. The mini-batch size also impacts the back-propagation algorithm. Larger batches smooth the gradient estimation and makes for a more stable descent, yet smaller batches introduce a stochastic noise in the algorithm which can be beneficial for generalization. Finally, it is often recommended to start training the network with a large learning rate and to reduce it later when close to optimum to refine the weights [9]. Hyperparameters for the optimization are often difficult to tune, however it is possible to validate the parameters on a small subset [16]. Stopping the training is often done when the validation error stopped increasing (early stopping) or when the training error does not decreases, altough the latter encourages overfitting.

Most deep learning software libraries implement these best practices and other regulizations, initializations, learning rate policies and gradient descent flavors. This greatly simplifies the experimental work and reduces the uncertainty due to diverging practices inside the community. However tweaking the hyperparameters impact the final performances. Robust statistical evaluation by repeating training and averaging resutls, and allocating the same hyperparameters optimization time to all models, including the baseline, avoids false conclusions on relative accuracies [124].

Let us remind that altough the gradient descent minimizes the training error, what we are really interested in is the error on real data. Training the model is done on an empirical risk that does not necessarily correspond to the actual risk, but the latter is not available since the real dataset is unlabeled and potentially infinite. We work with the empirical risk that we can measure, which can result in some level of overfitting. The model can learn some biased knowledge due to sampling bias in the training set. For example, learning a cats/dogs classifier where dogs picture were taken during the day and cats pictures were taken at night would not really generalize: it will overfit on brightness and not learn to recognize animals.

To fight this overfitting, the literature as introduced many *regularization* techniques that try to alleviate to emperical nature of the objective function and reduce the impact of the dataset bias. A first classical regularization is a penalty applied on the network weights. This method is called *weight decay* and adds an auxiliary loss related to the euclidean norm of the weights. The total loss function becomes:

$$\mathcal{L}_{total} = \mathcal{L}_{cost}(\mathrm{W}, \Omega) + \lambda \sum_{w \in \mathrm{W}} w^2 \quad .$$

Krogh and Hertz [85] showed that the weight decay helped reduce the generalization error of the model.

More recently *Dropout* [159] was introduced as a regularization technique that could alleviate overfitting in deep networks. As neural networks are parametrized by millions of weights, Dropout tries to solve this by randomly deactivating a fraction of the neurons during the learning phase. At each training iteration, every neuron might be shut down with a probability p. Its connections to other neurons are severed and its activation is set to zero. This effectively produces a reduced network and only the weights of the active neurons

are updated during the backpropagation. During inference all activations are weighted by p so that the full signal remains constant compared to training. Each node in the network therefore sees only a part of the dataset and, since connections have been randomly dropped, each layer has to learn some redundancy to preserve its discrimination power. Weak signals linked to the sampling bias of the dataset have a lower probability to be used as an informative signal in the network, which reduces overfitting. Another way to frame Dropout is to consider it as a stochastic ensembling approach. Dropout will produce many subnetworks that are concurrently trained. If we suppress neurons at each step with a probability $p = 0,5$, then it is equivalent to randomly train a subset of the 2^n possible networks where n is the number of parameters on which Dropout is applied. The final model used for inference is an average of the reduced networks ensemble. Other regularization strategies are been derived from Dropout such as *DropConnect* [177], which removes connections instead of neurons, and the stochastic pooling from Zeiler and Fergus [186].

An alternative policy to alleviate overfitting is called *data augmentation*. This consists in generating fake synthetic samples that are added in the training set. By artificially increasing the number of training samples, the model can learn from a larger variety of examples, therefore reducing the sampling bias of the dataset. When dealing with images, data augmentation is especially easy to perform using geometrical transformations for which invariance or robustness is expected, such as mirroring or flipping the image, random rotations or rescaling.

Finally, [77] introduced the Batch Normalization (BN) that has, on some occasions, been presented as a regularization strategy. Since BN estimates the statistical moments stochastically, this adds a small noise to the internal activations between two layers that might reduce the influence of weak signals and therefore alleviate overfitting.

2.1.4 Deep convolutional neural networks

Altough the idea of sharing weights between neurons to perform the same operation everywhere on the image in a efficient manner is due to the *Neocognitron* [52], it is LeCun et al. [94] who introduce the convolutional layer. The convolution of two functions f and g is a commutative bilinear operator, generally noted $f * g$, which is given by the formula:

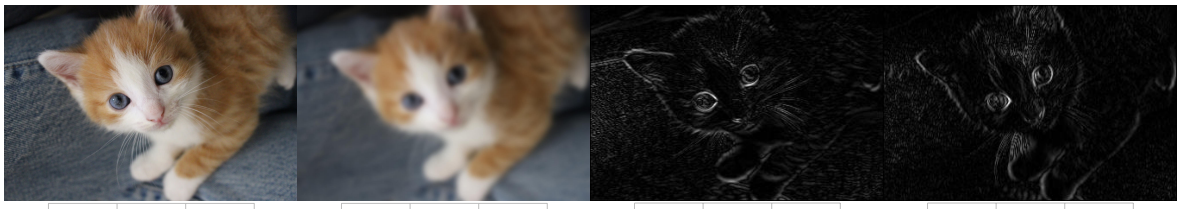
$$(f * g)(x) = \int_{-\infty}^{+\infty} f(t)g(x-t)\mathrm{d}t \,. \tag{2.5}$$

Convolutions are extremly popular in signal processing since it relates to the ubiquitous Fourier transform for spectral analysis [47]. Indeed the Fourier transform $\mathcal{F}$ transforms convolutions in the real domain into multiplications in the spectral domain and conversely:

$$\mathcal{F}(f * g) = \mathcal{F}(f)\mathcal{F}(g) \,. \tag{2.6}$$

Discrete convolution is omnipresent in image processing as it is involved in the computation of gradients for the SIFT [106] and HOG [35] features. Convolutional filters are also fundemental in wavelet theory [109] and its applications to image compression (e.g. the JPEG [37] format) and face detection Viola and Jones [175] thanks to the pseudo-Haar features [130]. Neurosciences have found that the Gabor filter model – often used as image features [132] – and the neural activations in the mammal visual cortex exhibited very similar properties [110, 80]. Some classical filtering operations are pictured in Fig. 2.5, such as discrete gradient computation using Sobel filters [157] or blurring using a Gaussian kernel. It is worth to note that, altough convolutional filters are very common, there are many non-linear filters that cannot be expressed as convolutions, such as median filter denoising [49].

The core idea of LeCun et al. [94] is to replace the first layers of a multi-layer perceptron by a set of learnable convolutions. Neurons are locally grouped and each clique compute a convolution on a part of the image. To simplify the model, the convolution weights (i.e.



0	0	0
0	1	0
0	0	0

Original image (identity)

0,25	0,5	0,25
0,5	1	0,5
0,25	0,5	0,25

Gaussian filter (blur)

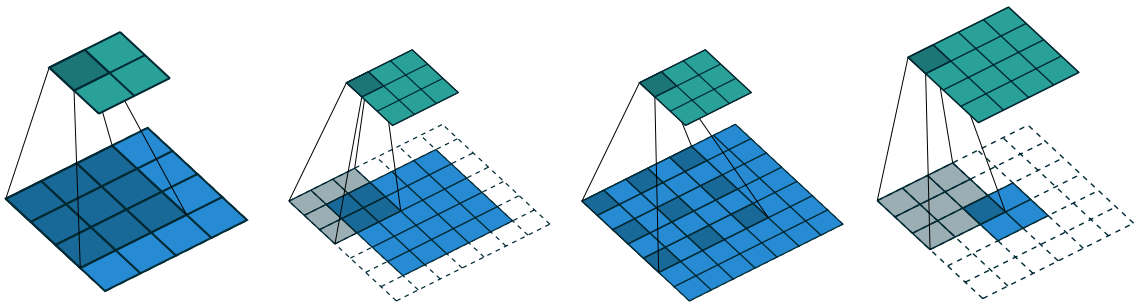
+1	+2	+1
0	0	0
-1	-2	-1

Sobel filter [157] (horizontal edges)

+1	0	-1
+2	0	-2
+1	0	-1

Sobel filter [157] (vertical edges)

Figure 2.5: Various convolutional filters applied on the same image.



Convolution

Strided convolution with padding

Dilated convolution

Transposed convolution

Figure 2.6: Convolution operation and variations on an image (figures from [43]).

the kernel) are shared along the image. This means that all groups compute the same convolution over the whole image. Several convolutions can be computed in parallel to extract various features. As the convolution kernels are optimized during training, there is a natural hierarchical representation learning process that occurs. This is especially interesting since the operators used are well-known and well-suited to image processing. Yosinski et al. [183] noted that the first convolutional layer of modern deep networks systemtically tend to resemble to a set of Gabor filters.

Convolution

The convolution operator on discrete functions can be rewritten as:

$$(f * g)[n] = \sum_{k=-\infty}^{+\infty} f[n]g[k-n]\,. \tag{2.7}$$

However this formula is written for one-dimensional signales while images are two-dimensional arrays. Luckily it is straightforward to extend the convolution operator to multivariate functions. In 2D, let $\mathrm{I} : [\![1;w]\!] \times [\![1;h]\!] \to \mathbb{R}$ denote an image of shape $w \times h$ and $\mathrm{K} : [\![1;k_w]\!] \times [\![1;k_h]\!] \to \mathbb{R}$ a convolutional *kernel* of shape $k_w \times k_h$. We define the filter $\mathcal{K}$ such as:

$$\mathcal{K}(\mathrm{I})[m,n] = \mathrm{K} * \mathrm{I}[m,n] = \sum_{i=-p}^{+p} \sum_{j=-q}^{+q} \mathrm{I}[m-i, n-j] \cdot \mathrm{K}[i,j]\,, \tag{2.8}$$

where $p = \frac{k_w - 1}{2}$ and $q = \frac{k_h - 1}{2}$. This operation is illustrated by the Fig. 2.6.

A drawback of this operation is that the product goes through the convolutional kernel K and the image I in opposite direction with increasing indices on the one hand and decreasing indices on the other hand. If is often more practical to implement this in software using the *crossed-correlation* operator:

$$\mathcal{K}(\mathrm{I})[m,n] = \mathrm{K} \star \mathrm{I}[m,n] = \sum_{i=-p}^{+p} \sum_{j=-q}^{+q} \mathrm{I}[m+i, n+j] \cdot \mathrm{K}[i,j] \,. \tag{2.9}$$

This operator is simpler to progarm altough it is not commutative anymore. Since K has learnable parameters, there is no practical difference between a convolution or cross-correlation operator since the matrices will be identical up to a symetrical transform. Other operations involved in CNNs are not commutative anyway so losing this property does not have a significant impact overall. We will use the formulas for the cross-correlation operator in the rest of this chapter.

Cross-correlation and convolution both are unknown when the operator is computed along the image borders since the values of I outside of the image are undefined. Generally these values are never computer and we restrict the convolution product on rows and columns for which it is well-defined. This results in an slightly smaller image and this is named the *valid* cross-correlation. Another solution is to fill the missing values of I using zeroes – a process called *zero-padding* – (cf. Fig. 2.6) for as many rows and columns as half the kernel size in every direction. This is often called the *same* crossed-correlation as it results in a filtered image that the exact same dimensions as the input. Finally a last operation consists in padding as many missing values as necessary so that each element of I is seen by every element of K. This is called the *full* cross-correlation[4].

Practically speaking, a n-dimensional convolutional layer from a neural network is parametrized by:

- The shapes $(k_1, \ldots, k_n)$ of its convolutional kernels, nearly always the same in all dimensions,
- The number C of filters, i.e. the number of concurrent convolutions, which defines how many feature maps are computed by the layer,
- The convolution stride s,
- The *padding* size p.

In most cases a convolutional layer therefore possess $k_1 \times \cdots \times k_n \times \mathrm{C}$ trainable parameters. In the 2-dimensional scenario, the convolutional kernels are generally square, i.e. the layer contains $\mathrm{C}k^2$ parameters.

There are several advantages to using convolution in deep neural networks [58]:

- Convolution operations are sparse since the convolutional kernels are very small compared to the image size,
- Features extracted by a convolutional layer are equivariant to translation, i.e. a translation of the input image also translates the output feature maps.
- Convolution weights are shared for the whole image, which makes it possible to detect the same features at any location in the image with a low memory cost.

[4] It is worth noting that more sophisticated padding methods exist. Instead of padding with zeroes, one can pad with arbitrary values, pad using the nearest neighbour or pad using a reflection of the existing data.

Compared to a fully connected layer, the convolutional layer is not invariant to pixel permutations because of its dependence on spatial structure. This prior is linked to the equivariance sought in image processing to various geometrical transforms. We must not forget that this structural prior does not always hold true. For example anomalies in time series do not necessarily have the same meaning based on the time at which they occur, while a 1D convolution would be activated in exactly the same way by the anomaly without any regard for its position in time. This strong prior is well-suited to image – and especially aerial and satellite images which present specific regularities. The hierachical CNN architecture seems tailored for image processing as it allows to decompose images in spaces that are invariant or equivariant to many transformations [170].

In 2D image processing, neural activations or *feature maps* are represented as 3-dimensional tensors $(\mathrm{C}, \mathrm{W}, \mathrm{H})$ with C the number of channels (sometimes referred to as convolutional planes), W the width and H the height.

A convolutional layers combines n_{in} input feature maps with the j^{th}convolutional kernel K_j:

$$\forall j \in [\![1; n_{out}]\!], \quad o_j = b_j + \sum_{i=1}^{n_{in}} \mathrm{K}(z_i) \,, \tag{2.10}$$

or:

$$\forall j \in [\![1; n_{out}]\!], \quad o_j(m,n) = b_j + \sum_{i=1}^{n_{in}} \sum_{p=-\frac{k-1}{2}}^{+\frac{k-1}{2}} \sum_{q=-\frac{k-1}{2}}^{+\frac{k-1}{2}} z_i(m-p, n-q) \cdot k_j(p,q) \,. \tag{2.11}$$

A convolution operator transforms a $(\mathrm{C}_{in}, \mathrm{W}_{in}, \mathrm{H}_{in})$ tensor into a $(\mathrm{C}_{out}, \mathrm{W}_{out}, \mathrm{H}_{out})$ tensor with the relationship[5]:

$$out = in - kernel + 2 \cdot padding + 1 \,. \tag{2.12}$$

Strided convolution A first variation of the convolution product consists in virtually pooling the feature maps spatial dimension by a factor s. To do so we only visit elements of I with indices that are a multiple of s:

$$\mathcal{K}_s(\mathrm{I})(m,n) = \mathrm{K}_s \star \mathrm{I} = \sum_{i=-p}^{+p} \sum_{i=-q}^{+q} \mathrm{I}[s \cdot m + i, s \cdot n + j] \cdot \mathrm{K}[i,j] \,. \tag{2.13}$$

A strided convolution transforms a $(\mathrm{C}_{in}, \mathrm{W}_{in}, \mathrm{H}_{in})$ tensor into a $(\mathrm{C}_{out}, \mathrm{W}_{out}, \mathrm{H}_{out})$ tensor with the relationship:

$$out = \left\lfloor \frac{in - kernel + 2 \cdot padding}{s} \right\rfloor + 1 \,. \tag{2.14}$$

Dilated convolution The dilated convolution [184], or "*convolution à trous*"[6], consists in computing a convolution at a lower resolution by skipping some values of I. To do so, the convolution kernel is virtually dilated by a factor d with missing values replaced by zeroes. The dilated convolution is computed using the formula:

$$\mathcal{K}^{(d)}(\mathrm{I})(m,n) = \mathrm{K}_s \star \mathrm{I} = \sum_{i=-p}^{+p} \sum_{i=-q}^{+q} \mathrm{I}[m + d \cdot i, n + \cdot j] \cdot \mathrm{K}[i,j] \,. \tag{2.15}$$

Output activations after a dilated convolution have a shape:

$$out = \left\lfloor \frac{in - kernel - (kernel-1)(dilation-1) + 2 \cdot padding}{s} \right\rfloor + 1 \,. \tag{2.16}$$

[5] Convolution arithmetic equations are from Dumoulin and Visin [43].

[6] The "à trous" algorithm [152] applies the same filter at multiple scales using dilated convolutions. The difference between the two is quite small and will not be debated here.

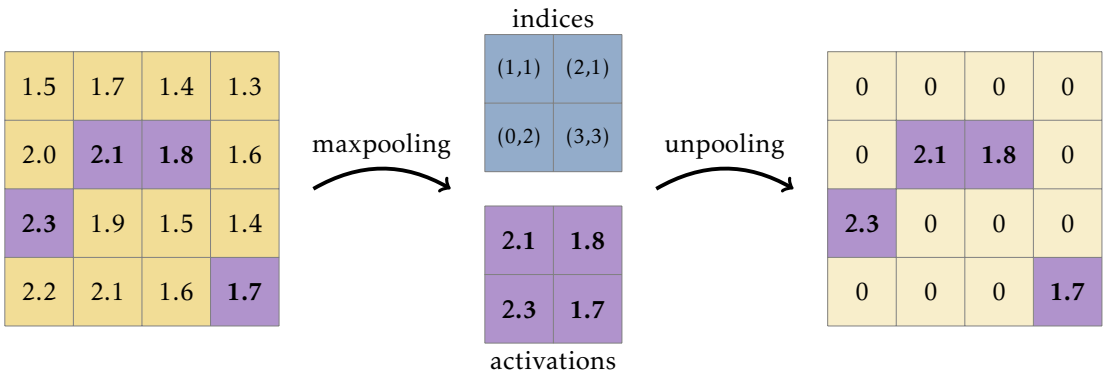


Figure 2.7: Max-pooling and max-unpooling in the 2D case.

Transposed convolution The transposed convolution is the inverse of the convolution operation in the sense that it corresponds to its gradient with respect to its inputs. For a given convolution kernel k, the tranposed convolution can reconstruct the image I based on the feature maps Z, with dimensions:

$$out = (in - 1) \cdot s + kernel - 2 \cdot padding \,. \tag{2.17}$$

A simpler reasoning is to see the transposed convolution as a standard convolution with a fractioned stride, i.e. a strided convolution with a stride $s = \frac{1}{s'}$ where $s' \in \mathbb{N}^*$.

This convolution is sometimes (wrongly) named "deconvolution" in literature. There is an actual mathematical deconvolution operator which is the inverse of the convolution. The tranposed convolution is useful to visualize the effects of a convolutional layer, for example in the decoder of convolutional autoencoders [189] or for image super-resolution [41].

Pooling

Pooling Pooling operations are useful to reduce the dimensions of the activation maps inside the network. It consists in a non-linear filter applied by a non-overlapping sliding window on the tensor. This filter is generally either the max or the average operator on a fixed-size window. These are respectively called the *max pooling* and the *average pooling* [192] layers. An of example of such a max-pooling is pictured in Fig. 2.7. In some cases the pooling window size is not fixed but only the output dimensions are. In this case this is called an adaptive pooling. It is used in some neural networks to reduce the feature maps dimensions to an arbitrary shape, often when input dimensions can vary (e.g. for object detection and semantic segmentation), since the shape of the fully connected layers does not allow size variations. In addition to the dimension reduction, the pooling operators also introduce an invariance to local perturbations.

The dimensions of the feature maps are pooling are:

$$out = \left\lfloor \frac{in - kernel}{s} \right\rfloor + 1 \,.$$

The pooling layers do not have any trainable parameter.

Unpooling Unpooling is the inverse of the pooling operator and tries to reconstruct the input based on the output. Since pooling discards some information, unpooling approximates the input and gives one possible solution. For average unpooling, the same value will be copied at several locations in the unpooled image. For maximum unpooling, the maximum value will be replaced at its original location and the remaining values will be filled with zeroes, as pictured in Fig. 2.7. The output shape of the unpooled feature maps are:

$$out = (in - 1) \cdot s + kernel \,.$$

As for pooling, unpooling layers do not have any trainable parameters.

Normalization

Altough data normalization and centering to approximate a Gaussian disitrbution has been used for a long time ine machine learning, normalization layers that operate directly on the feature maps inside the networks is relatively recent. Normalization transforms the neural activations to impose specific statistical properties that are assumed to be beneficial for the model optimization.

Jarrett et al. [78] suggest to use a local contrast normalization of the feature maps after the convolutional layers that is inspired by biological models [135]. For a tensor of N feature maps $a_1, \ldots, a_N$, the normalized maps z_i are given by substracting the local mean value on a gaussian window:

$$b_i[x,y] = a_i[x,y] - \sum_{p,q} w_{p,q} \cdot a_i[x+p, j+q]$$

where $w_{p,q}$ is a gaussian window such as $\sum_{p,q} w_{p,q} = 1$ and the normalizing the amplitudes by the weighted standard deviation of the features on the window,:

$$z_i[x,y] = \frac{b_i[x,y]}{\max(\text{moy}(\sigma[x,y]), \sigma[x,y])}$$

where $\sigma[x,y] = \left(\sum_{p,q} w_{p,q} \cdot b_i^2[x+p, y+q]\right)^{\frac{1}{2}}$.

The seminal work of Krizhevsky, Sutskever, and Hinton [84] introduces another local normalization layer, Local Response Normalization (LRN), that inhibates the features maps which are in the neighbourhood of a strongly excited neuron. This process is inspired by the lateral inhibition of biological neuron and resulted in a greater generalization ability. Contrary to the contrast normalization of Jarrett et al. [78], this layer does not apply on a local neighbourhood but along the feature maps dimension on adjacent neurons. The normalization is written as a kernel applied to the 2D activation maps. For a given tensor of N activation maps $a^1, \ldots, a^N$, the normalized features z^i are given by:

$$z^i[x,y] = \frac{a^i[x,y]}{\left(k + \alpha \sum_{j=max(0,i-n/2)}^{min(N-1,i+n/2)} (a^j[x,y])^2\right)^{\beta}} \ .$$

where n denotes the size of the neighbourhood to consider (in the channel dimension). This operation normalizes the full activation vector $(a^1[x,y], a^2[x,y], \ldots, a^N[x,y])$ at every location x, y in the feature maps.

The most common normalization layer in the literature is the BN [77]. This process consists in a normalizing the first and second order statistical moments of the activations for each plane independently. The moments are estimated batch-wise and therefore this layer requires a stochastic gradient descent-based optimizer. For a N-sized batch, the network computes at each step on the (N, C, W, H) tensor as follows:

Definition 5. *Batch normalization algorithm:*

Let $a_i^{(n)}, i \in [\![1, C]\!]$ denote the n^{th}output activation plane of a given layer. During the training phase, the mean μ and the variance σ^2 are computed on-the-fly based on the mini-batch statistics:

$$\mu_i = \frac{1}{N} \sum_{n=1}^{N} a_i^{(n)} \quad \textit{and} \quad \sigma_i^2 = \frac{1}{N} \sum_{n=1}^{N} (a_i^{(n)} - \mu_i)^2$$

The sliding averages of μ and σ^2 on the full dataset are stored in-memory and reused during the inference phase so that the network becomes batch-size independent.

In both situations, the normalized activations $\hat{a}$ are computed with the relationship:

$$\hat{a}_i = \frac{a_i - \mu_i}{\sqrt{\sigma_i^2 + \epsilon}}$$

.

The batch normalization is generally followed by an affine transform $z_i = \alpha\hat{a}_i + \beta$.[7] The original batch normalization was introduced for 2D feature maps but can be extended to 1D and 3D activations.

In practice the BN makes the gradient flow during backpropagation independent from the standard deviation of each layer weights. This allows to the optimization of deeper network since activations and gradients are less sensitive to vanishing or explosive behaviours. Practically speaking, BN often improves the overall model accuracy and significantly speeds up the convergence by smoothing the loss surface [146]. This normalization is used in most of the modern deep network architectures.

The non-linear activation function is applied after the convolutional layer either before or after the normalization depending on the model.

Fully connected layer

A fully connected layer is a bipartite complete graph in which all input neurons are connected by synapses to all output neurons. This is actually the same as the Perceptron from Rosenblatt [141]. The non-linearity is applied as an activation function on the output vector. A fully connected layer can be visualized as a simple matrix multiplication which projects a vector of shape 1×N into a vector of shape 1×M through a learnable weight matrix N×M. *Dropout* [159] is most the time used on the fully connected layers. Indeed, the weight matrix can contain a lot of parameters and therefore are the most sensitive to overfitting. On the opposite, convolutional layers are rarely used in conjunction to Dropout as the stochastic removal of multiple neurons could have adverse effects on the activation map spatial structure.

2.2 Deep learning for semantic segmentation

2.2.1 From classification to segmentation

Image segmentation is one of the first task considered for artificial vision. The myth says that Minsky ask to his then student Gerald Sussman to "*spend the summer linking a camera to a computer and getting the computer to describe what it saw*" in 1964[8]. Minsky and Papert envisioned for their *Summer Vision Project* (cf. Fig. 2.1) to "construct a system of programs which will divide a vidisector picture[9] into regions such as: likely objects, likely background areas, chaos. [...] The final goal is "object identification which will actually name objects by matching them with a vocabulary of known objects" [131]. Since the beginning, pattern recognition has been interested in dividing images based on its semantic content for visual understanding (cf. Fig. 2.8). Altough this task may seem trivial for a human, it is a significant challenge for the computer. Minsky's team was quickly confronted to Moravec's paradox [117]: "it is comparatively easy to make computers exhibit adult level performance on intelligence tests or playing checkers, and difficult or impossible to give them the skills of a one-year-old when it comes to perception and mobility".

[7] If this is the case, then the BN layer has 2C trainable parameters.

[8] As reported by Szeliski [165], citing Boden [12] who quotes Crevier [32].

[9] The *vidisector* was an image dissector invented by Philo Farnsworth in 1927. It uses the same principle as a cathode ray tube. The device receives light which excites a photocathode and emits electrons. The resulting electrical signal can be used to encode the image. It is the opposite of the old TV screen.

Figure 2.8: Classification and segmentation results on the same imagee. Classification is focused on recognizing objects in the image while segmentation is performed on every pixel. Image credits: PIRO4D (CC0).

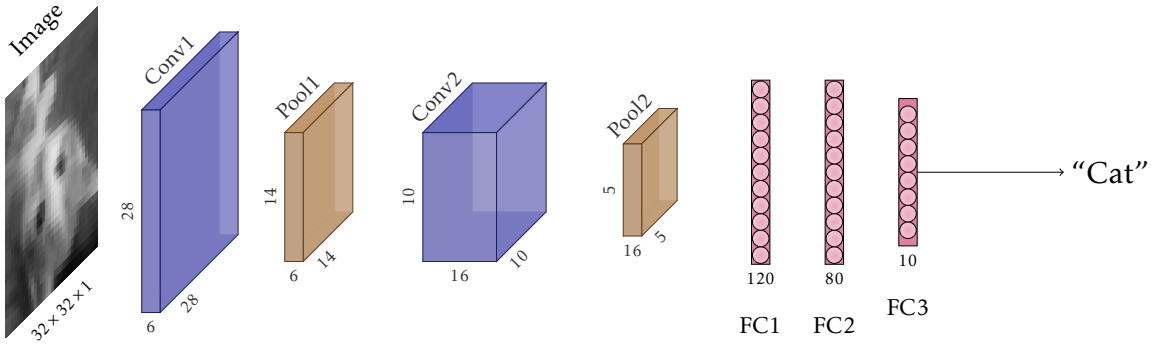


Figure 2.9: LeNet-5 architecture [94].

Many studies have been dedicated to object recognition, i.e. identifying the objects pictured in an image. We will call this task, which consists in mapping an image to one or more objects based on a vocabulary, "object classification". This has been the focus of the computer vision community for many years, from *ad hoc* features [169] to learning-based ones [174] and probabilistic modeling [148]. Several large-scale labeled datasets have been published, such as CIFAR-10 and CIFAR-100 [83], followed by ImageNet [40, 144] made possible the recent successes of deep convolutional neural networks. Many other specialized datasets played a significant role, such as the handwritten digits database MNIST [94] or traffic sign recognition [160] and chinese characters [99] datasets.

The LeNet-5 architecture was designed by LeCun et al. [94] and defines the standard structure of a CNN. The network is comprised of two convolutional layers followed by three fully connected layers as pictured in Fig. 2.9. The convolutional part performs the feature extraction in the image domain while the fully connected layers act like as multilayer perceptron computing the final classification based on the feature vector. LeNet-5 was originally designed for handwritten grayscale digits in 32×32 images. In comparison, the convolution kernels are quite large: 5×5. The first layer C1 has 6 kernels, i.e. six feature maps are produced. They are pooled with a stride 2 and fed to the next convolutional layer C2. Each of the 16 convolution kernels from C2 generates a map corresponding to the sum of all planes from C1 filtered using this kernel. The maps are once again pooled which results in a tensor of shape $(16, 5, 5)$. It is flattened in a vector 1×400 used as an input to a first fully

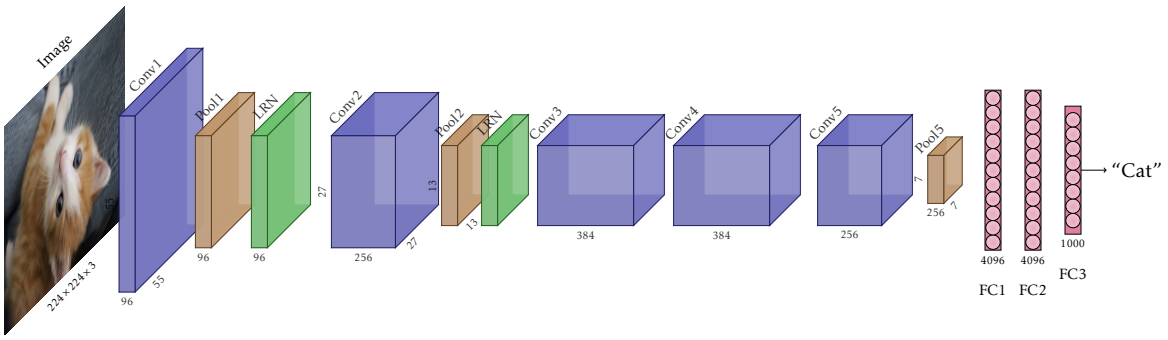


Figure 2.10: AlexNet architecture [84].

connected layer of size 120 and a second one of size 84. Finally, this feature vector is used as an input to the last fully connected layer of size 10, each activation corresponding to a digit. The output is transformed by a *softmax* activation to minimize to cross-entropy using the backpropagation algorithm.

Due to the fully connected layers, the number of neurons is fixed, which requires the feature maps extracted by the convolutional part to have specific dimensions. This requirement in turn constrains the size of the input image to be always the same. Using image with other shapes would require to retrain the fully connected layers with the right number of neurons. This drawback is shared by most CNNs.

The AlexNet AlexNet [84] network uses a similar approach altough it works on color images. This model is detailed in Fig. 2.10 and consists in 8 layers, 5 convolutional and 3 fully connected. AlexNet processess 224×224 red-green-blue (RGB) images. The first layer applies a large convolutional 11×11 convolutional kernel on the image followed a max-pooling to reduce the image dimensions. The feature maps are $96 \times 55 \times 55$ and $256 \times 27 \times 27$ after respectively the first and second convolutional layers. An LRN layer is applied after the first two convolutions to improve the model generalization capacity. The convolutional part of AlexNet is desgined so that the number of convolution planes increases while the spatial dimensions decrease. This maintains the number of activations in the same ballpark across the model and enhances the representational expressiveness of the model. The next three convolutional layers produce $384 \times 13 \times 13$ and $256 \times 13 \times 13$ tensors which are finally pooled into 256 feature maps of size 7×7. This representation is flattened into a unidimensional vector of length 12 544. The fully connected layer compress it into 2048 and project it into the classification space of dimension 1000, one for each of ImageNet's class of interest [144]. This model won the ILSVRC [144] competition in 2012 with a 15.3% top-5 error rate.

Zeiler and Fergus [187] worked on the AlexNet model to understand what were its strengths and weaknesses. They introduced a deconvolutional part[10] to inverse the convolutional layers and map the internal feature maps to the original pixel locations in the input image. The resulting *Deconvnet*, using unpooling layers and transposed convolutions, allow them to "see" the features learnt by AlexNet. Moreover they inspect the convolutional filters trained in the earlier layers and find several intriguing properties. First, the features learnt in the deeper levels are more invariant to geometric and color transforms than the first layers, meaning that they represent more abstract concepts. Second, it appears that the first convolutions consist in high and low-frequency filters but drop most of the middle frequency information. Therefore they replace the first layer and its 11×11 kernels by smaller 7×7 convolutions with a stride 2 instead of 4 to preserve more information. They show that the filters learnt this way are more diverse and that there are less "dead" kernels with a negligible amplitude. Finally, visualizing the internal feature maps show that they are far from random, but that neurons activate on image parts such as wheels, faces, etc.

[10] Actually a decoder based on transposed convolutions.

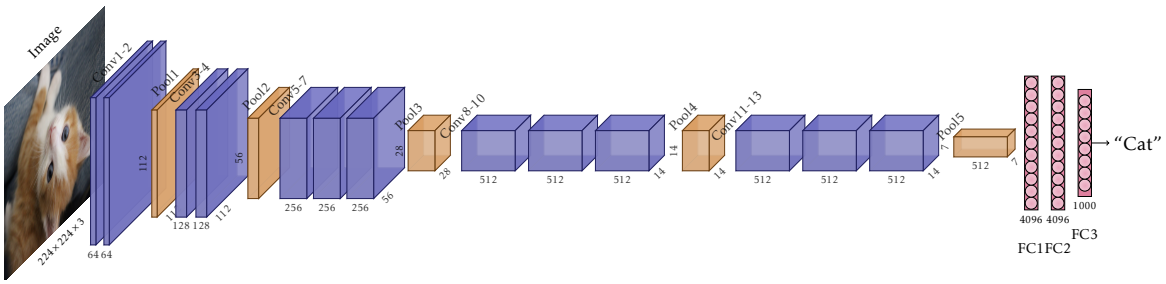


Figure 2.11: VGG-16 architecture [156].

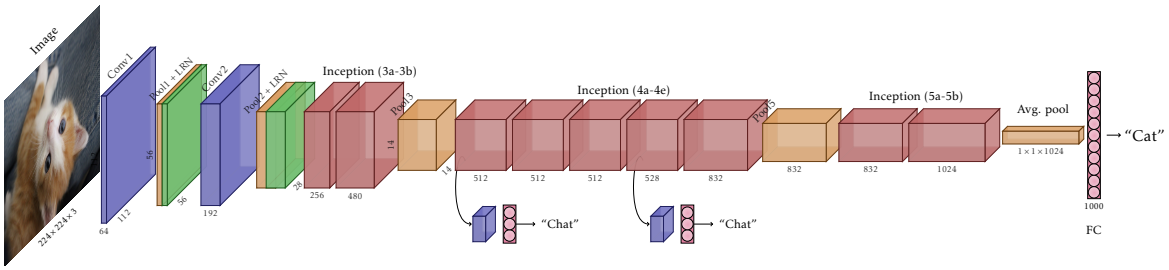


Figure 2.12: GoogLeNet architecture [162].

The VGG-16 model refines further the CNN architecture and introduce the "small kernel" paradigm. Indeed Chatfield et al. [21] and Simonyan and Zisserman [156] suggest that it is easier to train a stack of several 3×3 convolutions than one 11×11 convolutional layer. Moreover the presence of additional non-linearities might help increase the network expressiveness regarding composed functions. The VGG-16 replaces every large standard convolution by a stack of 2 or 3 convolutional layers with a 3×3 kernel as described in Fig. 2.11. The reference model – VGG-16 – consists of 16 layers, 13 of which are convolutional with the last 3 fully connected, following the standard set by LeNet and AlexNet. The network is divided in 5 consecutive blocks followed by a max-pooling of stride 2. The first two blocks contain 2 convolutional layers and the next contain 3 convolutional layers. The final activation maps are $512 \times 7 \times 7$, i.e. VGG-16 reduce the image spatial dimensions by a factor 32. This 25 088-long vector is then reduced to 4096 and classified into the 1000 classes of ImageNet. Dropout is applied on the fully connected layers to prevent overfitting. These enhancements on the usual CNN model improved in 2014 the error rate by 7.4% on object recognition during the ILSVRC.

Independently, Szegedy et al. [162] introduce the 22-layers deep GoogLeNet model. This architecture is designed around the new *Inception* module that applies several convolutions concurrently on the same feature maps. The principle is to apply various kernels – with different sizes – on the same activation map to perform multi-scale feature extraction, either using 1×1 convolutions – i.e. pixel-wise linear combination followed by an activation, pooling, 3×3 or 5×5 convolutions. This couples various features that possess the translation invariance due to pooling and the equivariance due to convolutions, allowing the representation to model a larger diversity of situations. The *Inception* module is illustrated by Fig. 2.13 while the full GoogLeNet network is detailed in Fig. 2.12. Since the network is relatively deep (22 layers), its authors suggest to ease the optimization of the lower layers by adding an intermediate classifier based on the middle feature maps after the *Inception* module (4a) and (4d). This deep supervision already showed to help fight vanishing gradients [95] and using it in deep networks makes sense. GoogLeNet makes the error rate on the ILSVRC drop to 6.4% for object recognition. It is subsequently improved [163] by replacing the 5×5 convolutional layer of the *Inception* module by two 3×3 convolutions as suggested for the VGG models [156]. It is also one of the first networks to use the *Batch Normalization*

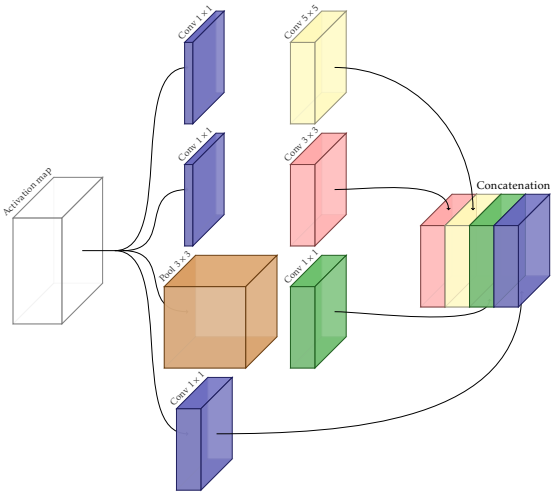


Figure 2.13: *Inception* module [162].

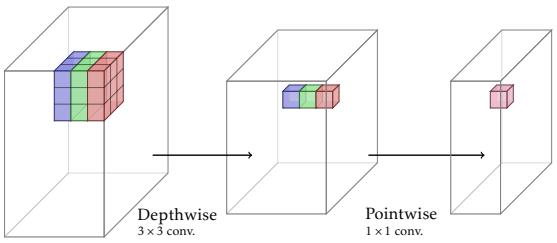


Figure 2.14: Depthwise separable convolutions [25].

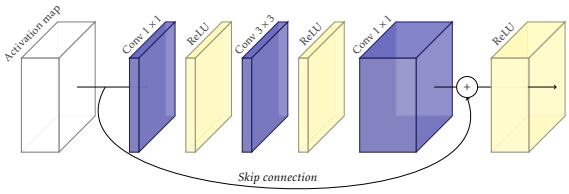


Figure 2.15: Residual convolutional block [64].

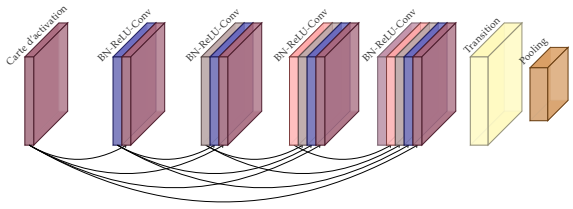


Figure 2.16: Dense convolutional block [74].

layer [77].

In 2015 He et al. [64] reduce the top-5 error rate on ImageNet to only 3.5% during the ILSVRC. Their approach involve a very deep network comprised of more than 100 convolutional layers. The optimization is possible using BN on the one hand and residual learning on the other hand. The latter is a new contribution that breaks the acyclic nature of feed-forward deep networks by introducing shortcut connections – or *skip connections* – that bypass some layers. These residual connections are an identity operation that allow both activations and gradients to freely flow in the whole network without exploding or vanishing. Instead of approximating $f : x \to f(x)$, the residual block learns an estimate of $\hat{f} : x \to f(x) - x$ which should have a smaller norm. The convolutional residual block is pictured in Fig. 2.15 and an example of ResNet model with 34 layers is detailed in Fig. 2.17. The introduction of the residual learning paradigm significantly changes the usual CNN design and switches the convolutional block by its residaul counterpart. ResNet models contain many layers but comparatively few parameters since only the last one is fully connected. As explained before, the fully connected layer generally concentrate most of a network weights and are also the most sensitive to overfitting, requiring regularization techniques such as *Dropout* [159]. ResNet only 3×3 convolutions, except for the first layer which is a 7×7 convolution with a stride of 2 for dimension reduction purposes. It is interesting to see that the final flattening of the feature maps is done using an adaptive pooling. No matter the size of the input image: the adaptive pooling will average the activations globally on the feature maps to generate the expected feature vector for the final fully connected layer. Howerver the large number of activations – and gradients – to compute make ResNet costly both in memory and computation time. ResNet models are rarely practical on large images. The *Inception* module has also been improved usnig residual connection in yet another variation [164].

The reuse of intermediate feature maps propagated to deeper layers improves the model performances as multiple abstraction levels can be combined for the final decision. Several studies have argued that these techniques are actually a way to ensemble several models

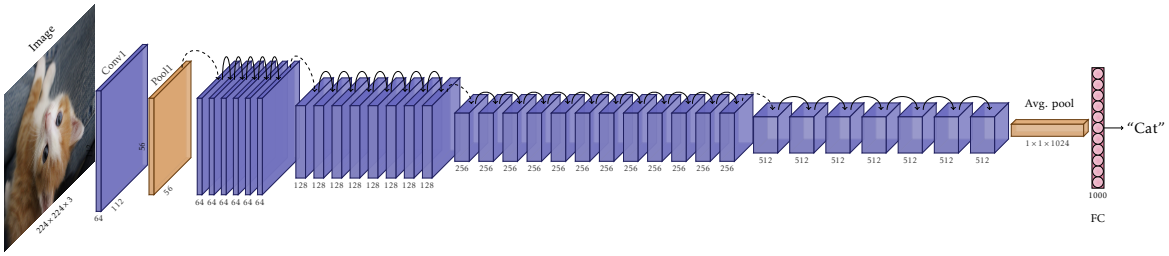


Figure 2.17: ResNet-34 architecture [64].

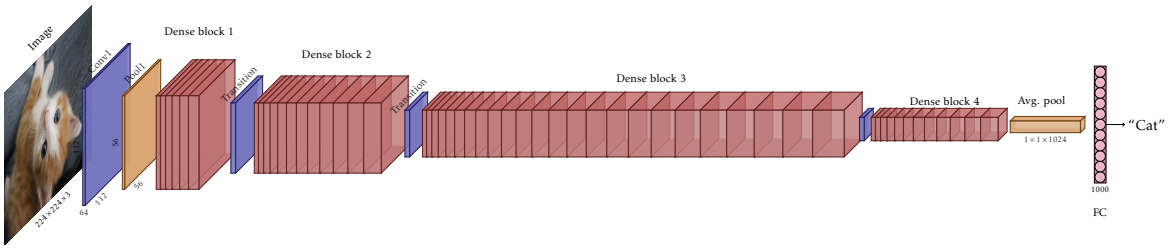


Figure 2.18: DenseNet-121 architecture [74].

in one as the activations can follow multiple paths in the network graph [173, 73]. Yet, the residual learning paradigm introduces shortcut connections that are restricted to the feature maps from the previous layer. Huang et al. [74] itroduce the *DenseNet* architecture comprised of dense shortcut connections, in which the activation maps from the earlier layers are propagated to all subsequent layers. To bound the exponential progression of the number of parameters, the model is divided in dense blocks with internal dense connections, as pictured in Fig. 2.18. A dense block is illustrated in Fig. 2.16. The skip connections allow to gradient to immediately reach shallow layers from the deep ones, which acts as a form of deep supervision [95]. A convolutional layer is used a transition between two blocks to reduce the number of planes, followed by a max-pooling layer to reduce the spatial dimensions of the activations. This architecture improved the state of the art top-5 accuracy on the ILSVRC 2012 validation test compared to the ResNet models. Yet, once again, altough DenseNet are less wasteful in parameters than traditional CNNs thanks to the absence of fully connected layers, the skip connections induce a large memory overhead to store activations and gradients.

Another architecture enhancement recently introduced is *the depthwise separable convolution* from Chollet [25]. This convolution applies one filter per plane in the activation tensor. The features are then recombined using a pixel-wise 1×1 convolution. It is a special use case of the usual convolution in which each tensor plane is filtered through one – and only one – kernel, as illustrated by the Fig. 2.14. The depthwise separable convolutions are suggested to replace the costly *Inception* and improved the GoogLeNet accuracy on the ImageNet and JFT (an internal dataset from Google) datasets. One advantage of using depthwise separable convolutions is that they achieve the same operation than traditional convolutions with less parameters. Indeed a $k_1 \times k_2$ convolution applied on N_{in} activation maps and producing N_{out} feature maps require $k_1 \times k_2 \times \mathrm{N}_{in} \times \mathrm{N}_{out}$ parameters. The depthwise separable convolution needs $\mathrm{N}_{in} \times \mathrm{N}_{in} \times k_1 \times k_2$ parameters for the first layer and $\mathrm{N}_{in} \times \mathrm{N}_{out}$ for the next, i.e. a total of $\mathrm{N}_{in} \times (\mathrm{N}_{in} \cdot k_1 \cdot k_2 + \mathrm{N}_{out})$ parameters which is interesting in the common $\mathrm{N}_{out} \geq \mathrm{N}_{in}$ scenario. These convolutions are effecient to compute and are popular for real-time and embedded applications [70].

If these achievements are promising, we must not forget that image classification is a somewhat limited task regarding scene understanding. Especially, object recognition only gives a binary information about the absence or presence of an object and gives no clue about its location. The first object localization sought to find objects by extracting dense

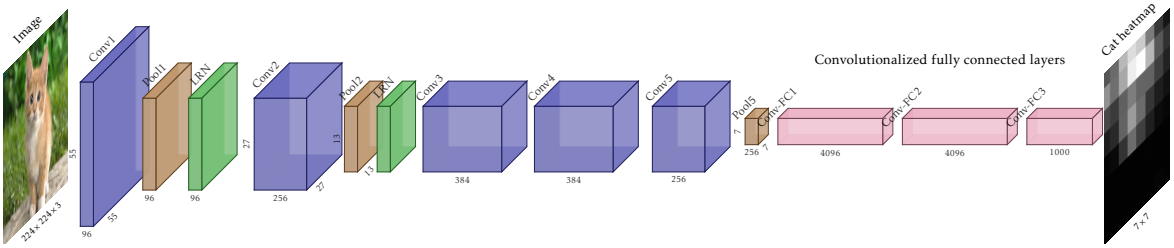


Figure 2.19: Fully convolutional AlexNet introduced by Long, Shelhamer, and Darrell [104].

features over the image, on top of which cascaded multi-scale classifiers could detect objects in various sub-regions. This is a standard approach that was used with SIFT [106] features, the pseudo-Haar from the face detector of Viola and Jones [175] or the HOG features [35]. Deep network-based approaches for object detection using a similar sub-region classification strategy quickly appear after 2012 [53, 102, 54], overthrowing most traditional localization techniques based on candidate window search (e.g. selective search) and expert features [60, 168]. The principle is however quite similar between these two families. A dense feature extraction is performed on the image to identify which regions could contain an object of interest. Yet these localization techniques still do not solve the problem described by Papert in 1966. It is not only a matter of finding the object in the image, but more a matter of determining its shape and separating it from the background, as shown in Fig. 2.8. This task, called *semantic segmentation*, is the mapping between a label and every pixel of an image.

Several semantic segmentation datasets have been built by the computer vision community to benchmark various methods, most of them consisting in every images such as PASCAL VOC [44] and Microsoft COCO [98], or autonomous driving scenarios for databases such as CamVid [19], Cityscapes [30] or Mapillary Vistas [121]. The first semantic segmentation algorithms used classifiers applied on dense features computed on the whole and grouped in homogeneous areas using a post-processing [154, 155]. Deep convolutional networks have been used to tackle semantic segmentation since the convolutions are efficient to compute a dense pixel-wise classification [59, 28] or simply as a standard image classifier applied on candidate regions [45, 149]. Indeed the feature maps computed by the convolutional layers preserve the visual structure of the image. It is generally feasible to map each feature to one or more pixels. Therefore this feature extractor is very efficient for object localization and has been widely used by the community [194]. We will detail further these approaches in the Chapter 3. In the meantime, we will focus on fully convolutional networks designed for dense pixel-wise classification. These models are a natural evolution of dense feature extraction approaches that allow an end-to-end training for semantic segmentation.

2.2.2 Fully convolutional models

The modern Fully Convolutional Network (FCN) architecture for semantic segmentation was popularized by Long, Shelhamer, and Darrell [104]. The essence of these models is to only work with convolutional layers, or rather to exclude completely the fully connected ones (cf. Fig. 2.19). This allows the activation maps to preserve the 2D image dimensions and its spatial structure, so that features can be replaced on the input image grid using a simple upsampling such as a bilinear interpolation. Long, Shelhamer, and Darrell [104] chose to transform the first fully connected layer into a convolution with a large kernel that coves the full activation maps. Indeed these two operations are mathematically the same but expressing it as a convolution removes the constraint on the input size. The first fully connected layer from AlexNet is replaced by a 7×7 convolution and the next ones by 1×1 kernels. This transformation preserves the network weights which is very interesting since AlexNet has already been pretrained for image classification on ImageNet. This modification is applied on VGG-16 where the pretrained Imagnet weights are reused in its fully convolutional version

thanks to this "convolutionalization" process. Instead a probability vector, the model now predicts a dense classification at resolution 1 : 32. This model is used to initialize a depper network producing a finer segmentation at resolution 1 : 16 and then yet another network at resolution 1 : 8 using an upsampling decoder. This bootstrapping approach based on pretraining signifcantly improved the state of the art on various semantic segmentation datasets at the time and more prominently on the famous PASCAL VOC [44].

Several improvements to the FCN architecture have been introduced in the literature for semantic segmentation of natural images. Some works focused on replacing the standard convolutional layers by dilated ones in the VGG-16 architecture [156]. For example Chen et al. [23] suggest to remove the maxpooling layers to avoid downsampling the image and loosing useful spatial information, while replacing the convolutions by dilated ones to enlarge the network receptive field. They also apply a Conditional Random Field (CRF) post-processing as a spatial regularizer on the resulting maps. In the same vein, Yu and Koltun [184] adopted the dilated convolution to agregate activation maps at multiple scales, which combines a larger receptive field with the parallel convolutional kernels from the *Inception* module [162]. Both models aim to learn a dense feature extraction and classification on the whole image and tweak the original CNN architecture to be more suited to the semantic segmentation task. Yet the final segmentation remains downsampled by a factor 4 or 8 compared to the input image.

Concurrently to these works, several derivatives of the FCN model have appeared, inspired by the convolutional autoencodeur archietcture [189]. The FCN from Long, Shelhamer, and Darrell [104] use a deep encoder similar to the convolutional part of a CNN, they use a shallow decoder based on deconvolutions to perform the upsampling. While this decoding strategy is efficient it also loses the benefit of high resolution images. Therefore symetrical encoder-decoder architectuers have been designed so that the low resolution feature maps coming out of the encoder can be projected into the classification space with a high spatial resolution, either using transposed convolutions/deconvolutions [119, 123] or using a sparse max-unpooling [4]. The U-Net architecture [140] uses skip connections to forward the encoded feature maps directly to the decoder so that the transposed convolutional layers can leverage both highly-abstract and low-level features to reconstruct the spatial resolution. These approaches remain pretrained using the CNN filters from the VGG-16 classifier. The symetrical architecture is useful to produce semantic maps at the *same resolution* as the input image thanks to the decoder that will iteratively upsample the features maps generated by the encoder.

As for image classification, ResNet and DenseNet models have also been successfully applie dto semantic segmentation. The main obstacle to using these models for full image segmentation was their significant computational cost, especially regarding the memory required to store the large number of intermediate activation maps and gradients. Thanks to always more powerful GPUs, Wu, Shen, and Van Den Hengel [181] were able to introduce a first ResNet-based semantic segmentation model which has also been adopted by the DeepLab model [23]. The memory consumption is reduced by downsampling the final semantic maps by a factor 1 : 4 as originally done by Long, Shelhamer, and Darrell [104]. More recently a fully convolutional DenseNet architecture has been introduced [79] for semantic segmentation using a symetrical encoder-decoder with skip connections backbone similar to U-Net [140]. Also relying on the skip connection principle, the GridNet model [48] looked into unconventional network structurse and introduced a model based on an ensemble of interconnected parallel ResNet. Activations can flow forward in the ResNet – layer by layer or skipping through using the identity connection – but also from laterally from one model to the other. This allow the feature maps to follow multiple paths in a "grid" of layers. This idea is also core to the work of Liu et al. [100] who use a decoder with multiplte paths in a convolutional network.

Some works investigated improvements to the semantic segmentation pipeline that did

not directly deal with the network architecture. Refining the semantic maps has been a quite successful topic of interesting, especially using structured graphical models as a post-processing for regularization. CRF have been in several ways to be jointly optimized with an FCN, e.g. as a learnable reccurrent neural network [190] or as post-processing [2].

Finally several multi-scale approaches for semantic segmentation have been introduced. Chen et al. [23], for one, integrated multi-scale predictions in the DeepLab model, all predicted maps being interpolated and averaged to produce the final classification. Others have used multiple convolutional kernel with variable sizes, either by using the dilation factor [184] or introducing *Inception*-like modules in the network [119, 188]. Peng et al. [133] proposed a global deconvolutional module that observes the whole image to model long-distance spatial relationships between objects to facilitate the global scene parsing. They combine this trick with residual learning to refine the object edges. Overall multi-scale inference for semantic segmentation mostly rely on parallel convolutions that produce a pyramid of activations at multiple resolutions.

To summarize, semantic segmentation of multimedia images is a task that has been frequently studied in the computer vision literature. FCN models have pushed further the state of the art on various datasets such as Microsoft COCO [98], PASCAL VOC [44], Cityscapes [30] and ADE20k [191]. However most works have dealt with everyday scene understanding: indoor images or autonomous driving scenarios where multiples objects can occur from various point of views, sometimes with occlusions, and with acquisitions based on digital cameras that are often consumer-grade. A major contribution of this thesis consists in providing a better understanding of how Earth Observation images can benefit from architectures that have been mainly designed for such multimedia data.

(a) Lidar image of North Carolina (U.S.A.). Color corresponds to height.
Image credits: Cintos (public domain, Wikimedia Commons)

(b) Composite RGB rendering of a multispectral Sentinel-2 image of the Viti Levu island (Fiji).
Image credits: Copernicus Sentinel data processed by ESA (CC BY-SA 3.0 IGO)

(c) SAR image acquired by Sentinel-1 over the Dotson ice shelf (Antarctica).
Image credits: Copernicus Sentinel data processed by A. Hogg/CPOM

Figure 2.20: Earth Observation is achieved through a large battery of sensors, each with its specificities.

2.3 Machine learning for Earth Observation image interpretaion

Remote sensing image interpretation mobilizes similar cognitive functions to those used to parse everyday images. It is no surprise that image processing and computer vision algorithms are ubiquituous in photointerpretation. However Earth Observation is not the same as traditional photography. Both the sensors and the viewpoint are peculiar. Automating cartography based on aerial and satellite image is not simply a matter of computer vision. Remote sensing for Earth Observation relies both on artificial perception using machine learning and signal processing tuned to the spatial sensors that have nothing in common with consumer-grade digital cameras.

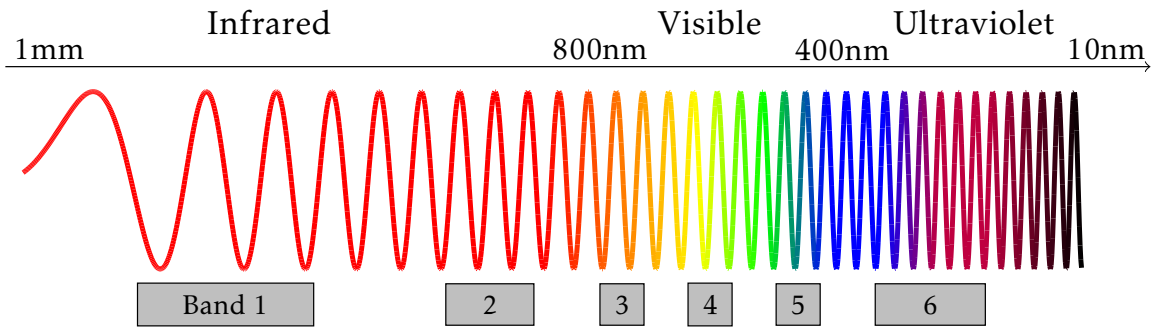


Figure 2.21: A multispectral sensor acquires simultaneously light intensities in several bands distributed on the infrared, visible and sometimes ultraviolet domains.

2.3.1 The many sensor types

Various images acquired by different Earth Observation sensors are pictured in Fig. 2.20. If aerial images are often acquired using standard RGB color cameras, satellite images use sophisticated sensors that have been designed for the pecularities of an elevated viewpoint. For example, some sensors can see a rich spectral information outside the visible wavelengths while others have been chosen for their ability to see through the cloud cover or because they can measure heat, elevation or other physical properties.

A common example is that of the infrared sensor which is often used in conjunction to color acquisitions. Infrared cameras are popular – included for aerial imaging – since they can see between 780 nm and 2500 nm. In the near-infrared domain, it helps detect vegetation due to high reflectance of chlorophyll for these wavelengths. In the short infrared domain, it can be used to estimate temperature thanks to Wien's law. Those thermal cameras are especially useful in space where Earth residual heat is weaker.

The Fig. 2.21 describe a multispectral camera or superspectral that, based on the same principle, can take pictures of a scene in several wavelength bands more or less wide which can be indifferently in the visible spectrum or in infrared/ultraviolet. Such a sensor produces multichannel images – generally ten or so channels – that the human eye cannot directly understand. One can reconstruct a natural color RGB image by compositing the intensities from the channels corresponding to the red, green and blue wavelengths. Multispectral acquisitions can exhibit different spatial resolution depending on the channel. For example satellites Sentinel-2A and Sentinel-2B produce images at ground sampling resolution 10 m/px in the visible domain, but some channels – especially in the infrared spectrum – have a resolution of 20 m/px or even as low as 60 m/px. Color images from satellite sensors with the highest spatial resolution reach about 30 cm/px, while aerial images can go up to 5 cm/px and sometimes even better using flying Unmanned Aerial Vehicles (UAVs). For this reason it is common to use both multispectral and panchromatic acquisitions at the same time for satellite sensors. The panchromatic image does not distinguish color and produces a grayscale image but with a higher ground sampling resolution. In France, the the Pléaides satellite constellation constellation performs a simultaneous panchromatic acquisition at 70 cm/px Ground Sample Distance (GSD), resampled at 50 cm/px, and at multispectral acquisition at 2.8 m/px GSD resampled at 2 m/px.

An extreme case of multispectral imaging is hyperspectral imagery, which consists in performing acquisitions on tens or hundreds of identical narrow spectral bands regularly distributed along the desired range. The camera sweeps the full light spectrum and returns a discrete estimation of what has been reflected, as pictured in the Fig. 2.22. Depending on the spectral resolution – often around 10 nm – and the width of spectral domain, the number of bands can go from a few dozens to several hundreds of bands. This kind of camera is useful to reconstruct the full reflected light intensity with respect to the wavelength for each pixel. As every material reflects sunlight differently depending on its albedo, this information can

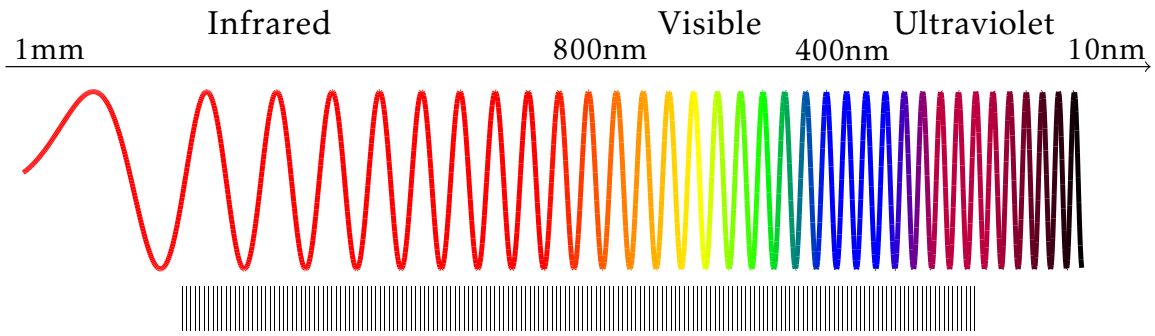


Figure 2.22: A hyperspectral sensor acquires simultaneously multiple narrow spectral bands regularly distributed along its spectral domain.

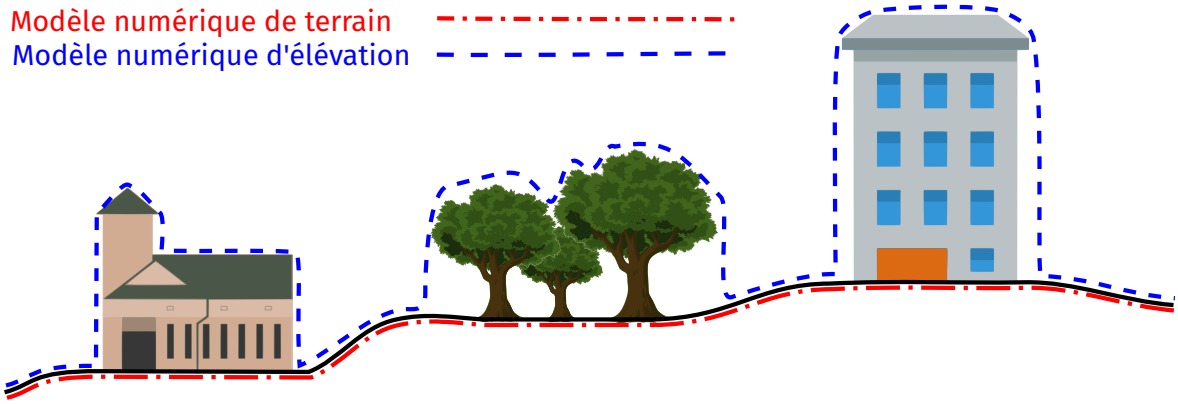


Figure 2.23: Difference between DTM and DSM.

be leveraged to characterize precisely the composition of the observed objects. The drawback of these hyperspectral cameras is their low spatial resolution which is significantly worse than those of other optical sensors. Indeed hyperspectral images have a GSD of about 1 m/px from an airplane and about 30 m/px from space.

Note that all optical sensors that we discussed here are passive; they only receive light that has been reflected or emitted by the observed area. These sensors are therefore sensitive to environmental brightness conditions and meteorological perturbations such as clouds which can occlude partially or totally objects of interest. Lots of satellites use active sensors that emit a signal and measure the response. The most common satellites of the sort are radar satellites and more specifically Synthetic Aperture Radar (SAR) ones which send one or more electromagnetic waves of which they measure the reflection to extract physical parameters from the observed area. SAR can pierce through the cloud cover, however it does not produce images stricly speaking.

Another active sensor is the Light Detection And Ranging (Lidar), that emits a laser pulse and measures its echo. Finding the location of the echo's maximum amplitude makes it possible to estimate the time needed by the photons to reach the target and come back, i.e. it gives an estimation of the distance the particles covered. These sensors are very common in remote sensing and in robotics for topographic surveys and 3D reconstruction. Yet laser measurements cover only one point at a time and the sensor only produces sparse point clouds. Satellite Lidar sensors measure about one point every 20 m, while air-based ones can do one measurment every 10 cm. Once the point cloud has been generated, one can generate a volumic mesh which is a topographic model of the area. In remote sensing it is common to rasterize this mesh, i.e. to project it on a 2D plane, to obtain either a Digital Terrain Model (DTM)) or a Digital Surface Model (DSM) depending on its accuracy. The DSM differs from the Digital Terrain Model (DTM)) by taking into account above-ground objects that elevate themselves on the top of the underlying terrain as pictured in Fig. 2.23. The

difference between these to values is called the normalized Digital Surface Model (nDSM): it is the normalized height of above-ground points with respect to the terrain.

This manuscript presents works that focus on optical acquisitions for cartography. However we will also use ancillary data, derivated from Lidar point clouds or froming geographic information systems (GIS).

2.3.2 Machine learning and remote sensing

As we have seen remote sensing data come in multiple favors: multispectral cameras, hyperspectral imagerys, SAR, Lidar... There are various machine learning techniques that can be applied to extract information from Earth Observation data without human intervention.

Feature extraction

Once data has been acquired and preprocessed, it must encoded in a way that is suitable for classification. More specifically we generally choose to project the data into a representation space in which classification will be easier. We detail below some commonly used approaches from the state of the art.

The simplest representation is based on the raw data itself, sometimes after normalization. The classifier can directly be applied on luminance or reflectance pixel values. However a 128×128 RGB image would be represented as a vector of length $128 \times 128 \times 3 = 49152$, which makes the computation intractable for most usual statistical models. This enforces a limit on the number of pixels that can used as features and generally, this kind of classification can only deal with individual pixels or very small regions. These approaches are very common for hyperspectral image processing [46, 62], but can also be found in multispectral and color image processing [39].

The raw data can be combined to hand-crafted expert features such as statistical moments or the first and second order derivatives of the signal. For example the Lidar signal is often complemented by the local deviation to the average height, which is a discriminant feature to separate various objects [61, 97, 90]. The local entropy is another feature commonly found in SAR image processing for remote sensing [5].

Moreover expert knowledge about the physics of multispectral and SAR sensors is often beneficial for the classifier. Reflectance ratios between various wavelengths can be used to characterize specific surfaces and materials. Two examples of such indices are the very popular Normalized Difference Vegetation Index (NDVI) [142] for vegetation and Normalized Difference Water Index (NDWI) [182] for water. These values are easy to interpret but determining the right combination to use for a given dataset is challenging: it requires expert knowledge of the phenomenon (it becomes unfeasible to detect something that we cannot at least partially characterize using spectral properties) and a systematic feature engineering effort for each new problem.

As for multimedia images it can be interesting to seek universal features. Color histograms can be used for multispectral and hyperspectral images in the same way as for RGB images. The histograms are invariant to rotations, local translationss and scaling. However they are sensitive to slight radiometric changes due to environment such as illumination changes. Moreover the value quantization in the histogram can add some robustness to noise but also severely reduce the digital accuracy of the intensities and therefore make some subtle differences between spectra disappear. For hyperspectral imaging, two similar materials can have nearly the same spectral profile with local difference in one or two wavelengths presenting different absorption peaks. A strong quantization can make these peaks dsappear, losing the discriminant feature. Other histograms such as HOG [35] also apply to remote sensing images with the same limitations.

A last group of features frequently used for remote sensing image classification is the morphological attribute profile group. Benediktsson, Pesaresi, and Árnason [6] introduced

these features, obtained by an iterative combination of morphological operations (dilation and erosion). Not only these image features give information about the spatial structures to which a pixel belong, they do so at multiple scales thanks to attribute profiles [36].

It is often practical to combine several features at multiple scales or on multiple sensors to obtain a richer information. Once the features have been generated many statistical models can used for classification or regression.

Usual statistical models

The feature vectors extracted from all the data samples can be fed to a classifier, a statistical decision model that can come in various forms. This subsection only deals with shallow classifiers that do not perform representation learning, excluding the deep neural network which will be discussed at a later point.

Literature on machine learning for remote sensing has long been supportive of decision trees – more specifically random forests [17] – and SVM [13, 31].

Deicision trees [18] are a group of statistical model that represent variables as inside nodes, each edge corresponding to a set of possible values for the variable. The set of edges leaving from a given node cover all its possibles values. Choosing an edge depends on mutually exclusive tests: the first one covers the case $a < 0$, the second one $0 < a < 0.1$, etc. The tree is optimized during the learning phase using a recursive splitting, dividing the dataset based on its first feature, then its second, its third and so on, until adding a new variable does not improve the accuracy anymore (or when all subsets converge to the same results).

While it is possible to use one decision tree, most of the time they are ensembled in so-called random forests [17]. A Random Forest (RF) is actually an ensemble of decision trees optimized on various random subsets of the input features. Each tree computes its own prediction independently from the others and the forest prediction is the class that received the most votes (majority rule). Learning an ensemble of trees produces a classifier with a smaller variance than every single decision trees. One significant strength of decision trees is that they can be linearized as decision rules easy to understand: the path taken by a sample in the tree is determined by explicit tests on its features (for example, this pixel has been predicted this way because the height was less than 5 m and the red intensity greater than 128)[11]. Random forests have been very popular in semantic mapping of remote sensing data for various applications, ranging from land cover prediction using Landsat images [127] to weather prediction [88]. Ensemble of decision trees can also be generated using the *gradient boosting* principle that leverages many weak classifiers and combine them to produce one strong classifier [50]. The gradient boosted trees are somewhat less common in remote sensing but can occasionally be found in the literature [89].

The SVM [13, 31] are classifiers that work by dividing the feature space so that the distance between the class-border and the nearest sample (the margin) is as large as possible. The border is actually an hyperplane in the input feature space (linear kernel) or an hyperplane of a space with a large dimensionality (possibly infinite) when using the kernel trick [13].

If the dimension of input data is large, computing exact hyperplanes that maximize the margin can untractable. In this case there are two workarounds: reducing the data dimension using dimension reduction techniques or using approximate optimization algorithms such as the online gradient descent for SVM training [15].

SVMs have found many applications in remote sensing and are especially popular for land cover classification based on multispectral [128] and hyperspectral images [113].

Finally the multilayer perceptron also found itself used as a classifier of remote sensing data for multispectral [7] and hyperspectral [57] image processing.

[11] Although it is harder to interpret random forests with hundreds or thousands of trees.

Spectral and spatial features

As we have seen, expert features in remote sensing generally focus on radiometric information. For a long time most classification approaches operated pixel-wise, i.e. the classifiers only predicted one pixel at a time. Even when the spatial neighbourhood was considered, the inferred class was valid only the center pixel.

This approach ensures that the predicted map has the highest possible resolution – the same as the input data – but it prevents the classifer from learning spatial relationships between objects. Since technology has continuously improved the sensor sensitivity and resolution, object of interests now cover several pixels and geometrical features such as connectivity and convexity can be observed. A given pixel submitted to extreme noise (either due to a sensor deficiency or an odd material) would be wrongly classified if taken in isolation. However a model that learns from spatial features could work around this mistake by taking into account that neighbouring pixels probably have the same class (i.e. an homogeneity criterion). Pixel-wise classification often results in noisy salt-and-pepper maps which requires post-processing using graphical models such as CRF.

In comparison patch-wise classification approaches have been designed to leverage the spatial context around objects. These techniques slide a window over the image to classify the center pixel of each square area. Patch-wise classification became increasingly popular thanks to the new deep CNN from the state of the art. Originally the remote sensing community relied on a mix of spatial and spectral expert features [46], which were superseded by the spectral-spatial representations automatically learnt by deep networks. The latter significantly outperformed the former and became the *de facto* new state of the art for many tasks [122, 24]. Several works have investigated the use of CNN applied using a sliding window for various applications, such as building detection [171] and land cover classification [129].

Yet these methods rapidly reach a limit as the number of pixels exhibits a quadratic growth with respect to the image size. One prediction per pixel does not scale when dealing with the high resolution (HR), very high resolution (VHR) and extremely high resolution (EHR) data that the new satellites deliver. It would require millions – even billions – of predictions to process one image. The only acceptable solution is to reduce the number of inferences needed using an efficient grouping scheme: one prediction should cover an area of several pixels.

This motivated the introduction of region-based classification methods. The principle is that similar pixels can be grouped in homogeneous regions that share the same class. The similarity criterion used to merge pixels depends on both their position and their values. The classifier can then perform a unique prediction for all pixels that belong to the same region, based on the hypothesis that neighbouring spectrally similar pixels share the same semantics (i.e. the same label). In that case feature extraction can be done on the whole region once. As an example, an image of shape 1500×1500 can be reasonably in 20 000 regions, which would entail only 20 000 predictions to map it entirely instead of 2 250 000 for pixel-wise techniques.

Many segmentation algorithms have been introduced both for remote sensing and natural images. These algorithms divide the set of pixels in an unsupervised fashion. Once the segmentation has been done, one can generate the features for every region and then train a classifier using the same pipeline as usual.

Region-based classification significantly reduces the computational burden of semantic mapping. Increasing the sensor spatial resolution does not alter the regions' homogeneity which can be kept as is. Therefore increasing the size of the image does not entail a quadratic growth of the number of regions. Since the seminal work of Mnih [116] using CNNs for patch-based road and building extraction in aerial images, these approaches have been successfully transfered on many VHR datasets [86, 172]. Note that mapping an image by a sliding window or even pixel-wise using the pixel grid are particular cases of the general

region-based classification strategy.

The image presegmentation is also very interesting when using morphological profiles since these operations are costly. A common variant is to first build a multiscale tree-like hierarchical segmentation of the image. The morphological operator can then be interpreted as tree pruning which are fast and efficient. Many approaches for tree-based segmentation exist [14] that can be used to compute attribute profiles, i.e. multi-scale features that extend the properties of morphological features. These methods have been the state of the art for semantic mapping for some time [134] and dedicated statistical models have been designed especially to leverage morphological attribute profiles [33].

Concurrently to this thesis, approaches based on fully convolutional networks (FCN) for semantic segmentation of remote sensing have become more and more popular. Indeed FCNs perform a dense inference on the whole image in only one forward pass. This solves the problem of the high computional cost of the patch-based classification, drastically reducing the computation time without unsupervised presegmentation. The first published paper using an FCN on optical aerial images appear in 2015 [126, 108, 153], based on the standard architectures from Long, Shelhamer, and Darrell [104]. Symetrical autoencoder-like architectures quickly followed [176, 3] and many derivative works have been proposed, such as a CRF [103] designed for data fusion or explicit regularization for edge smoothing [111]. Aerial EHR images were a natural starting point, but FCNs have also been applied to satellite data [51].

New datasets such as the *Inria Aerial Image Labeling Dataset* constitute an excellent playground for FCNs to excel. These models trust the first ranks of the leaderboards [71]. Overall deep learning approaches using FCNs are now well-established as the new state of the art for most remote sensing image processing tasks [101]. As more and more datasets are published in this area, some of which are detailed in Appendix A, it has never been easier to train deep models for various applications.

References

[1] David H. Ackley, Geoffrey E. Hinton, and Terrence J. Sejnowski. "A Learning Algorithm for Boltzmann Machines". In: *Cognitive Science* 9.1 (Jan. 1, 1985), pp. 147–169. ISSN: 0364-0213. DOI: 10.1016/S0364-0213(85)80012-4. URL: http://www.sciencedirect.com/science/article/pii/S0364021385800124 (cit. on p. 12).

[2] Anurag Arnab et al. "Higher Order Conditional Random Fields in Deep Neural Networks". In: *Computer Vision – ECCV 2016*. Ed. by Bastian Leibe et al. Lecture Notes in Computer Science. Springer International Publishing, 2016, pp. 524–540. ISBN: 978-3-319-46475-6 (cit. on p. 34).

[3] Nicolas Audebert, Bertrand Le Saux, and Sébastien Lefèvre. "Semantic Segmentation of Earth Observation Data Using Multimodal and Multi-Scale Deep Networks". In: *Computer Vision – ACCV 2016*. Springer, Cham, Nov. 20, 2016, pp. 180–196. DOI: 10.1007/978-3-319-54181-5_12 (cit. on p. 40).

[4] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Scene Segmentation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.12 (Dec. 2017), pp. 2481–2495. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2016.2644615 (cit. on p. 33).

[5] David G. Barber and Ellsworth LeDrew. "SAR Sea Ice Discrimination Using Texture Statistics: A Multivariate Approach". In: *Photogrammetric Engineering and Remote Sensing* 57 (Apr. 1, 1991) (cit. on p. 37).

[6] Jón Atli Benediktsson, Martino Pesaresi, and Kolbeinn Árnason. "Classification and Feature Extraction for Remote Sensing Images from Urban Areas Based on Morphological Transformations". In: *IEEE Transactions on Geoscience and Remote Sensing* 41.9 (Sept. 2003), pp. 1940–1949. ISSN: 0196-2892. DOI: 10.1109/TGRS.2003.814625 (cit. on p. 37).

[7] Jón Atli Benediktsson, Philip H. Swain, and Okan K. Ersoy. "Neural Network Approaches Versus Statistical Methods In Classification Of Multisource Remote Sensing Data". In: *IEEE Transactions on Geoscience and Remote Sensing* 28.4 (July 1990), pp. 540–552. ISSN: 0196-2892. DOI: 10.1109/TGRS.1990.572944 (cit. on p. 38).

[8] Yoshua Bengio. "Learning Deep Architectures for AI". In: *Foundations and trends® in Machine Learning* 2.1 (2009), pp. 1–127 (cit. on p. 12).

[9] Yoshua Bengio. "Practical Recommendations for Gradient-Based Training of Deep Architectures". In: *Neural Networks: Tricks of the Trade*. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, 2012, pp. 437–478. ISBN: 978-3-642-35288-1 978-3-642-35289-8. DOI: 10.1007/978-3-642-35289-8_26. URL: https://link.springer.com/chapter/10.1007/978-3-642-35289-8_26 (cit. on p. 19).

[10] Yoshua Bengio et al. "Greedy Layer-Wise Training of Deep Networks". In: *Advances in Neural Information Processing Systems 19*. Ed. by B. Schölkopf, J. C. Platt, and T. Hoffman. MIT Press, 2007, pp. 153–160. URL: http://papers.nips.cc/paper/3048-greedy-layer-wise-training-of-deep-networks.pdf (cit. on pp. 12, 18).

[11] Monica Bianchini and Franco Scarselli. "On the Complexity of Neural Network Classifiers: A Comparison Between Shallow and Deep Architectures". In: *IEEE Transactions on Neural Networks and Learning Systems* 25.8 (Aug. 2014), pp. 1553–1565. ISSN: 2162-237X. DOI: 10.1109/TNNLS.2013.2293637 (cit. on p. 15).

[12] Margaret Boden. *Mind as Machine: A History of Cognitive Science*. Oxford; New York: OUP Oxford, June 26, 2008. 1756 pp. ISBN: 978-0-19-954316-8 (cit. on p. 26).

[13] Bernhard E. Boser, Isabelle M. Guyon, and Vladimir Vapnik. "A Training Algorithm for Optimal Margin Classifiers". In: *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*. COLT '92. New York, NY, USA: ACM, 1992, pp. 144–152. ISBN: 978-0-89791-497-0. DOI: 10.1145/130385.130401. URL: http://doi.acm.org/10.1145/130385.130401 (cit. on p. 38).

[14] Petra Bosilj et al. "Partition and Inclusion Hierarchies of Images: A Comprehensive Survey". In: *Journal of Imaging* 4.2 (Feb. 1, 2018), p. 33. DOI: 10.3390/jimaging4020033. URL: http://www.mdpi.com/2313-433X/4/2/33 (cit. on p. 40).

[15] Léon Bottou. "Large-Scale Machine Learning with Stochastic Gradient Descent". In: *In COMPSTAT*. 2010 (cit. on p. 38).

[16] Léon Bottou. "Stochastic Gradient Descent Tricks". In: *Neural Networks: Tricks of the Trade*. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, 2012, pp. 421–436. ISBN: 978-3-642-35288-1 978-3-642-35289-8. DOI: 10.1007/978-3-642-35289-8_25. URL: https://link.springer.com/chapter/10.1007/978-3-642-35289-8_25 (cit. on pp. 18, 19).

[17] Leo Breiman. "Random Forests". In: *Machine Learning* 45.1 (Oct. 1, 2001), pp. 5–32. ISSN: 0885-6125, 1573-0565. DOI: 10.1023/A:1010933404324. URL: https://link.springer.com/article/10.1023/A:1010933404324 (cit. on p. 38).

[18] Leo Breiman. *Classification and Regression Trees*. Routledge, 2017 (cit. on p. 38).

[19] Gabriel J. Brostow, Julien Fauqueur, and Roberto Cipolla. "Semantic Object Classes in Video: A High-Definition Ground Truth Database". In: *Pattern Recognition Letters*. Video-based Object and Event Analysis 30.2 (Jan. 15, 2009), pp. 88–97. ISSN: 0167-8655. DOI: 10.1016/j.patrec.2008.04.005. URL: http://www.sciencedirect.com/science/article/pii/S0167865508001220 (cit. on p. 32).

[20] Augustin Louis Cauchy. *Comptes rendus hebdomadaires des séances de l'Académie des sciences*. ark:/12148/bpt6k2982c. Paris: Gauthier-Villars, July 1847. URL: http://gallica.bnf.fr/ark:/12148/bpt6k2982c (cit. on p. 15).

[21] Ken Chatfield et al. "Return of the Devil in the Details: Delving Deep into Convolutional Nets". In: *Proceedings of the British Machine Vision Conference*. British Machine Vision Conference (BMVC). British Machine Vision Association, 2014, pp. 6.1–6.12. ISBN: 978-1-901725-52-0. DOI: 10.5244/C.28.6. URL: http://www.bmva.org/bmvc/2014/papers/paper054/index.html (cit. on p. 29).

[22] Kumar Chellapilla, Sidd Puri, and Patrice Simard. "High Performance Convolutional Neural Networks for Document Processing". In: *Tenth International Workshop on Frontiers in Handwriting Recognition*. Suvisoft, 2006 (cit. on p. 12).

[23] Liang-Chieh Chen et al. "DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.4 (Apr. 2018), pp. 834–848. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2017.2699184 (cit. on pp. 33, 34).

[24] Yushi Chen et al. "Deep Feature Extraction and Classification of Hyperspectral Images Based on Convolutional Neural Networks". In: *IEEE Transactions on Geoscience and Remote Sensing* 54.10 (Oct. 2016), pp. 6232–6251. ISSN: 0196-2892. DOI: 10.1109/TGRS.2016.2584107 (cit. on p. 39).

[25] François Chollet. "Xception: Deep Learning with Depthwise Separable Convolutions". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Honolulu, United States, July 2017, pp. 1800–1807. DOI: 10.1109/CVPR.2017.195 (cit. on pp. 30, 31).

[26] Dan C. Cireşan, Ueli Meier, and Jürgen Schmidhuber. "Multi-Column Deep Neural Networks for Image Classification". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Washington, DC, United States, 2012, pp. 3642–3649. ISBN: 978-1-4673-1226-4. URL: http://dl.acm.org/citation.cfm?id=2354409.2354694 (cit. on p. 12).

[27] Dan C. Cireşan et al. "Deep Big Simple Neural Nets Excel on Handwritten Digit Recognition". In: *Neural Computation* 22.12 (Dec. 2010), pp. 3207–3220. ISSN: 0899-7667, 1530-888X. DOI: 10.1162/NECO_a_00052. arXiv: 1003.0358 (cit. on p. 12).

[28] Dan C. Cireşan et al. "Deep Neural Networks Segment Neuronal Membranes in Electron Microscopy Images". In: *Advances in Neural Information Processing Systems*. 2012, pp. 2843–2851 (cit. on p. 32).

[29] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. "Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)". In: *Proceedings of the International Conference on Learning Representations (ICLR)*. Nov. 23, 2015. URL: http://arxiv.org/abs/1511.07289 (cit. on p. 15).

[30] Marius Cordts et al. "The Cityscapes Dataset for Semantic Urban Scene Understanding". In: *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Las Vegas, United States, June 2016, pp. 3213–3223. DOI: 10.1109/CVPR.2016.350 (cit. on pp. 32, 34).

[31] Corinna Cortes and Vladimir Vapnik. "Support-Vector Networks". In: *Machine Learning* 20.3 (Sept. 1, 1995), pp. 273–297. ISSN: 0885-6125, 1573-0565. DOI: 10.1007/BF00994018. URL: https://link.springer.com/article/10.1007/BF00994018 (cit. on p. 38).

[32] Daniel Crevier. *AI: The Tumultuous History of the Search for Artificial Intelligence*. New York, NY, USA: Basic Books, Inc., 1993. ISBN: 978-0-465-02997-6 (cit. on p. 26).

[33] Yanwei Cui, Laetitia Chapel, and Sébastien Lefèvre. "Scalable Bag of Subpaths Kernel for Learning on Hierarchical Image Representations and Multi-Source Remote Sensing Data Classification". In: *Remote Sensing* 9.3 (Feb. 24, 2017), p. 196. DOI: 10.3390/rs9030196. URL: http://www.mdpi.com/2072-4292/9/3/196 (cit. on p. 40).

[34] George Cybenko. "Approximation by Superpositions of a Sigmoidal Function". In: *Mathematics of Control, Signals and Systems* 2.4 (Dec. 1, 1989), pp. 303–314. ISSN: 0932-4194, 1435-568X. DOI: 10.1007/BF02551274. URL: https://link.springer.com/article/10.1007/BF02551274 (cit. on pp. 11, 15).

[35] Navneet Dalal and Bill Triggs. "Histograms of Oriented Gradients for Human Detection". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2005, pp. 886–893. DOI: 10.1109/CVPR.2005.177 (cit. on pp. 12, 20, 32, 37).

[36] Mauro Dalla Mura et al. "Morphological Attribute Profiles for the Analysis of Very High Resolution Images". In: *IEEE Transactions on Geoscience and Remote Sensing* 48.10 (Oct. 2010), pp. 3747–3762. ISSN: 0196-2892. DOI: 10.1109/TGRS.2010.2048116 (cit. on p. 38).

[37] Ingrid Daubechies. *Ten Lectures on Wavelets*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 1992. ISBN: 978-0-89871-274-2 (cit. on p. 20).

[38] Guillaume de L'Hôpital. *Analyse des infiniment petits, pour l'intelligence des lignes courbes*. Paris : Montalant, 1716. 227 pp. URL: http://archive.org/details/infinimentpetits1716lhos00uoft (cit. on p. 17).

[39] Clément Dechesne et al. "Semantic Segmentation of Forest Stands of Pure Species Combining Airborne Lidar Data and Very High Resolution Multispectral Imagery". In: *ISPRS Journal of Photogrammetry and Remote Sensing* 126 (Apr. 1, 2017), pp. 129–145. ISSN: 0924-2716. DOI: 10.1016/j.isprsjprs.2017.02.011. URL: http://www.sciencedirect.com/science/article/pii/S0924271616302763 (cit. on p. 37).

[40] Jia Deng et al. "ImageNet: A Large-Scale Hierarchical Image Database". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2009, pp. 248–255. DOI: 10.1109/CVPR.2009.5206848 (cit. on pp. 12, 27).

[41] Chao Dong, Chen Change Loy, and Xiaoou Tang. "Accelerating the Super-Resolution Convolutional Neural Network". In: *Computer Vision – ECCV 2016*. European Conference on Computer Vision. Lecture Notes in Computer Science. Springer, Cham, Oct. 8, 2016, pp. 391–407. ISBN: 978-3-319-46474-9 978-3-319-46475-6. DOI: 10.1007/978-3-319-46475-6_25. URL: https://link.springer.com/chapter/10.1007/978-3-319-46475-6_25 (cit. on p. 24).

[42] John Duchi, Elad Hazan, and Yoram Singer. "Adaptive Subgradient Methods for Online Learning and Stochastic Optimization". In: *Journal of Machine Learning Research* 12 (Jul 2011), pp. 2121–2159. ISSN: ISSN 1533-7928. URL: http://jmlr.org//papers/v12/duchi11a.html (cit. on p. 18).

[43] Vincent Dumoulin and Francesco Visin. "A Guide to Convolution Arithmetic for Deep Learning". In: (Mar. 23, 2016). arXiv: 1603.07285 [cs, stat]. URL: http://arxiv.org/abs/1603.07285 (cit. on pp. 21, 23).

[44] Mark Everingham et al. "The Pascal Visual Object Classes Challenge: A Retrospective". In: *International Journal of Computer Vision* 111.1 (June 25, 2014), pp. 98–136. ISSN: 0920-5691, 1573-1405. DOI: 10.1007/s11263-014-0733-5. URL: http://link.springer.com/article/10.1007/s11263-014-0733-5 (cit. on pp. 32–34).

[45] Clément Farabet. "Towards Real-Time Image Understanding with Convolutional Networks". Université Paris-Est, 2013 (cit. on p. 32).

[46] Mathieu Fauvel et al. "Advances in Spectral-Spatial Classification of Hyperspectral Images". In: *Proceedings of the IEEE* 101.3 (Mar. 2013), pp. 652–675. ISSN: 0018-9219. DOI: 10.1109/JPROC.2012.2197589 (cit. on pp. 37, 39).

[47] Joseph Fourier. "Propagation de la chaleur dans un solide rectangulaire infini". In: *Théorie analytique de la chaleur*. F. Didot père et fils, 1822, pp. 159–177. URL: https://www.bibnum.education.fr/mathematiques/analyse/theorie-analytique-de-la-chaleur (cit. on p. 20).

[48] Damien Fourure et al. "Residual Conv-Deconv Grid Network for Semantic Segmentation". In: *BMVC 2017*. Londre, France, Sept. 2017. URL: https://hal.archives-ouvertes.fr/hal-01567725 (cit. on p. 33).

[49] Bernard Roy Frieden. "A New Restoring Algorithm for the Preferential Enhancement of Edge Gradients". In: *JOSA* 66.3 (Mar. 1, 1976), pp. 280–283. DOI: 10.1364/JOSA.66.000280. URL: https://www.osapublishing.org/josa/abstract.cfm?uri=josa-66-3-280 (cit. on p. 20).

[50] Jerome H. Friedman. "Greedy Function Approximation: A Gradient Boosting Machine." In: *The Annals of Statistics* 29.5 (Oct. 2001), pp. 1189–1232. ISSN: 0090-5364, 2168-8966. DOI: 10.1214/aos/1013203451. URL: https://projecteuclid.org/euclid.aos/1013203451 (cit. on p. 38).

[51] Gang Fu et al. "Classification for High Resolution Remote Sensing Imagery Using a Fully Convolutional Network". In: *Remote Sensing* 9.5 (May 18, 2017), p. 498. DOI: 10.3390/rs9050498. URL: http://www.mdpi.com/2072-4292/9/5/498 (cit. on p. 40).

[52] Kunihiko Fukushima. "Neocognitron: A Self-Organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position". In: *Biological Cybernetics* 36.4 (Apr. 1, 1980), pp. 193–202. ISSN: 0340-1200, 1432-0770. DOI: 10.1007/BF00344251. URL: https://link.springer.com/article/10.1007/BF00344251 (cit. on pp. 11, 20).

[53] Ross Girshick et al. "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2014, pp. 580–587. DOI: 10.1109/CVPR.2014.81 (cit. on p. 32).

[54] Ross Girshick et al. "Region-Based Convolutional Networks for Accurate Object Detection and Segmentation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38.1 (Jan. 2016), pp. 142–158. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2015.2437384 (cit. on p. 32).

[55] Xavier Glorot and Yoshua Bengio. "Understanding the Difficulty of Training Deep Feedforward Neural Networks". In: *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics. Mar. 31, 2010, pp. 249–256. URL: http://proceedings.mlr.press/v9/glorot10a.html (cit. on p. 19).

[56] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. "Deep Sparse Rectifier Neural Networks". In: *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics. June 14, 2011, pp. 315–323. URL: http://proceedings.mlr.press/v15/glorot11a.html (cit. on pp. 12, 14).

[57] Pradeep Goel et al. "Classification of Hyperspectral Data by Decision Trees and Artificial Neural Networks to Identify Weed Stress and Nitrogen Status of Corn". In: *Computers and Electronics in Agriculture* 39.2 (May 2003), pp. 67–93. ISSN: 0168-1699. DOI: 10.1016/S0168-1699(03)00020-6. URL: http://www.sciencedirect.com/science/article/pii/S0168169903000206 (cit. on p. 38).

[58] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. URL: http://www.deeplearningbook.org (cit. on p. 22).

[59] David Grangier, Léon Bottou, and Ronan Collobert. "Deep Convolutional Networks for Scene Parsing". In: *ICML 2009 Deep Learning Workshop*. Vol. 3. Citeseer, 2009 (cit. on p. 32).

[60] Chunhui Gu et al. "Recognition Using Regions". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2009, pp. 1030–1037. DOI: 10.1109/CVPR.2009.5206727 (cit. on p. 32).

[61] Li Guo et al. "Relevance of Airborne Lidar and Multispectral Image Data for Urban Scene Classification Using Random Forests". In: *ISPRS Journal of Photogrammetry and Remote Sensing* 66.1 (Jan. 1, 2011), pp. 56–66. ISSN: 0924-2716. DOI: 10.1016/j.isprsjprs.2010.08.007 (cit. on p. 37).

[62] JiSoo Ham et al. "Investigation of the Random Forest Framework for Classification of Hyperspectral Data". In: *IEEE Transactions on Geoscience and Remote Sensing* 43.3 (Mar. 2005), pp. 492–501. ISSN: 0196-2892. DOI: 10.1109/TGRS.2004.842481 (cit. on p. 37).

[63] Kaiming He et al. "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification". In: *Proceedings of the IEEE International Conference on Computer Vision*. IEEE International Conference on Computer Vision (ICCV). Dec. 2015, pp. 1026–1034. DOI: 10.1109/ICCV.2015.123 (cit. on pp. 15, 19).

[64] Kaiming He et al. "Deep Residual Learning for Image Recognition". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Las Vegas, United States, June 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90 (cit. on pp. 30, 31).

[65] Donald O. Hebb. *The Organization of Behavior*. 1949. pmid: 10643472 (cit. on pp. 10, 11).

[66] Geoffrey E. Hinton, Simon Osindero, and Yee-Whye Teh. "A Fast Learning Algorithm for Deep Belief Nets". In: *Neural Computation* 18.7 (July 2006), pp. 1527–1554. ISSN: 0899-7667. DOI: 10.1162/neco.2006.18.7.1527. URL: http://dx.doi.org/10.1162/neco.2006.18.7.1527 (cit. on pp. 12, 18).

[67] Geoffrey E. Hinton and Ruslan Salakhutdinov. "Reducing the Dimensionality of Data with Neural Networks". In: *Science* 313.5786 (July 28, 2006), pp. 504–507. ISSN: 1095-9203. DOI: 10.1126/science.1127647. pmid: 16873662 (cit. on p. 12).

[68] Sepp Hochreiter et al. "Gradient Flow in Recurrent Nets: The Difficulty of Learning Long-Term Dependencies". In: *Filed Guide to Dynamical Recurrent Networks*. IEEE Press, 2001 (cit. on p. 14).

[69] Kurt Hornik. "Approximation Capabilities of Multilayer Feedforward Networks". In: *Neural Networks* 4.2 (Jan. 1, 1991), pp. 251–257. ISSN: 0893-6080. DOI: 10.1016/0893-6080(91)90009-T. URL: http://www.sciencedirect.com/science/article/pii/089360809190009T (cit. on pp. 11, 15).

[70] Andrew G. Howard et al. "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications". In: (Apr. 16, 2017). arXiv: 1704.04861 [cs]. URL: http://arxiv.org/abs/1704.04861 (cit. on p. 31).

[71] Bohao Huang et al. "Large-Scale Semantic Classification: Outcome of the First Year of Inria Aerial Image Labeling Benchmark". In: *2018 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. July 22, 2018. URL: https://hal.inria.fr/hal-01767807/document (cit. on p. 40).

[72] Fu Jie Huang and Yann LeCun. "Large-Scale Learning with SVM and Convolutional for Generic Object Categorization". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2006, pp. 284–291. DOI: 10.1109/CVPR.2006.164 (cit. on p. 12).

[73] Gao Huang et al. "Deep Networks with Stochastic Depth". In: *Computer Vision – ECCV 2016*. Lecture Notes in Computer Science. Springer, Cham, Oct. 8, 2016, pp. 646–661. ISBN: 978-3-319-46492-3 978-3-319-46493-0. DOI: 10.1007/978-3-319-46493-0_39. URL: https://link.springer.com/chapter/10.1007/978-3-319-46493-0_39 (cit. on p. 31).

[74] Gao Huang et al. "Densely Connected Convolutional Networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Vol. 1. 2017, p. 3 (cit. on pp. 30, 31).

[75] David H. Hubel and Torsten N. Wiesel. "Receptive Fields of Single Neurones in the Cat's Striate Cortex". In: *The Journal of Physiology* 148.3 (Oct. 1959), pp. 574–591. ISSN: 0022-3751. pmid: 14403679. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1363130/ (cit. on p. 11).

[76] David H. Hubel and Torsten N. Wiesel. "Receptive Fields and Functional Architecture of Monkey Striate Cortex". In: *The Journal of Physiology* 195.1 (Mar. 1968), pp. 215–243. ISSN: 0022-3751. pmid: 4966457 (cit. on p. 11).

[77] Sergey Ioffe and Christian Szegedy. "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift". In: *Proceedings of the 32nd International Conference on Machine Learning*. International Conference on Machine Learning (ICML). 2015, pp. 448–456. URL: http://jmlr.org/proceedings/papers/v37/ioffe15.html (cit. on pp. 20, 25, 30).

[78] Kevin Jarrett et al. "What Is the Best Multi-Stage Architecture for Object Recognition?" In: *Proceedings of the 2009 IEEE 12th International Conference on Computer Vision (ICCV)*. Sept. 2009, pp. 2146–2153. DOI: 10.1109/ICCV.2009.5459469 (cit. on p. 25).

[79] Simon Jégou et al. "The One Hundred Layers Tiramisu: Fully Convolutional DenseNets for Semantic Segmentation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. Honolulu, United States, July 2017, pp. 1175–1183. DOI: 10.1109/CVPRW.2017.156 (cit. on p. 33).

[80] Judson P. Jones and Larry A. Palmer. "An Evaluation of the Two-Dimensional Gabor Filter Model of Simple Receptive Fields in Cat Striate Cortex". In: *Journal of Neurophysiology* 58.6 (Dec. 1987), pp. 1233–1258. ISSN: 0022-3077. DOI: 10.1152/jn.1987.58.6.1233. pmid: 3437332 (cit. on p. 20).

[81] Diederik Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization". In: *Proceedings of the International Conference on Learning Representations (ICLR)*. 2015. arXiv: 1412.6980. URL: http://arxiv.org/abs/1412.6980 (cit. on p. 18).

[82] Stephen Cole Kleene. "Representation of Events in Nerve Nets and Finite Automata". In: *Automata Studies*. Princeton University Press (1956), pp. 3–42 (cit. on p. 10).

[83] Alex Krizhevsky. *Learning Multiple Layers of Features from Tiny Images*. CIFAR, 2009 (cit. on p. 27).

[84] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Proceedings of the Neural Information Processing Systems (NIPS)*. NIPS. 2012, pp. 1097–1105. URL: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf (cit. on pp. 12, 25, 28).

[85] Anders Krogh and John A. Hertz. "A Simple Weight Decay Can Improve Generalization". In: *Proceedings of the 4th International Conference on Neural Information Processing Systems*. NIPS'91. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1991, pp. 950–957. ISBN: 978-1-55860-222-9. URL: http://dl.acm.org/citation.cfm?id=2986916.2987033 (cit. on p. 19).

[86] Adrien Lagrange et al. "Benchmarking Classification of Earth-Observation Data: From Learning Explicit Features to Convolutional Networks". In: *2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. 2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS). July 2015, pp. 4173–4176. DOI: 10.1109/IGARSS.2015.7326745 (cit. on p. 39).

[87] Joseph-Louis Lagrange. *Théorie des fonctions analytiques, contenant les principes du calcul différentiel , dégagés de toute considération d'infiniment petits ou d'évanouissans, de limites ou de fluxions, et réduits a l'analyse algébrique des quantités finies*. À Paris, de l'Imprimerie de la République. Prairial an V., 1797. URL: http://gallica.bnf.fr/ark:/12148/bpt6k86263h (cit. on p. 17).

[88] David J. Lary et al. "Machine Learning in Geosciences and Remote Sensing". In: *Geoscience Frontiers*. Special Issue: Progress of Machine Learning in Geosciences 7.1 (Jan. 1, 2016), pp. 3–10. ISSN: 1674-9871. DOI: 10.1016/j.gsf.2015.07.003. URL: http://www.sciencedirect.com/science/article/pii/S1674987115000821 (cit. on p. 38).

[89] Rick Lawrence et al. "Classification of Remotely Sensed Imagery Using Stochastic Gradient Boosting as a Refinement of Classification Tree Analysis". In: *Remote Sensing of Environment* 90.3 (Apr. 15, 2004), pp. 331–336. ISSN: 0034-4257. DOI: 10.1016/j.rse.2004.01.007. URL: http://www.sciencedirect.com/science/article/pii/S0034425704000148 (cit. on p. 38).

[90] Arthur Le Guennec et al. "Classification de données LiDAR bi-spectral topo-bathymétriques par une approche multi-échelle : Application en milieu fluvial". In: *Conférence Annuelle Française de Photogrammétrie et Télédétection (CFPT)*. Marne-la-Vallée, France, June 27, 2018, p. 8 (cit. on p. 37).

[91] Yann LeCun. "Learning Process in an Asymmetric Threshold Network". In: *Disordered Systems and Biological Organization*. NATO ASI Series. Springer, Berlin, Heidelberg, 1986, pp. 233–240. ISBN: 978-3-642-82659-7 978-3-642-82657-3. DOI: 10.1007/978-3-642-82657-3_24. URL: https://link.springer.com/chapter/10.1007/978-3-642-82657-3_24 (cit. on pp. 11, 15).

[92] Yann LeCun et al. "Backpropagation Applied to Handwritten Zip Code Recognition". In: *Neural Computation* 1.4 (Dec. 1989), pp. 541–551. ISSN: 0899-7667. DOI: 10.1162/neco.1989.1.4.541 (cit. on p. 12).

[93] Yann LeCun et al. "Efficient BackProp". In: *Neural Networks: Tricks of the Trade*. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, 1998, pp. 9–50. ISBN: 978-3-540-65311-0 978-3-540-49430-0. DOI: 10.1007/3-540-49430-8_2. URL: https://link.springer.com/chapter/10.1007/3-540-49430-8_2 (cit. on pp. 14, 16, 19).

[94] Yann LeCun et al. "Gradient-Based Learning Applied to Document Recognition". In: *Proceedings of the IEEE* 86.11 (Nov. 1998), pp. 2278–2324. ISSN: 0018-9219. DOI: 10.1109/5.726791 (cit. on pp. 12, 20, 27).

[95] Chen-Yu Lee et al. "Deeply-Supervised Nets". In: *Artificial Intelligence and Statistics*. Artificial Intelligence and Statistics. Feb. 21, 2015, pp. 562–570. URL: http://proceedings.mlr.press/v38/lee15a.html (cit. on pp. 29, 31).

[96] J. Y. Lettvin et al. "What the Frog's Eye Tells the Frog's Brain". In: *Proceedings of the IRE* 47.11 (Nov. 1959), pp. 1940–1951. ISSN: 0096-8390. DOI: 10.1109/JRPROC.1959.287207 (cit. on p. 12).

[97] Aihua Li et al. "Lidar Aboveground Vegetation Biomass Estimates in Shrublands: Prediction, Uncertainties and Application to Coarser Scales". In: *Remote Sensing* 9.9 (Aug. 31, 2017), p. 903. DOI: 10.3390/rs9090903. URL: http://www.mdpi.com/2072-4292/9/9/903 (cit. on p. 37).

[98] Tsung-Yi Lin et al. "Microsoft COCO: Common Objects in Context". In: *Computer Vision – ECCV 2014*. Ed. by David Fleet et al. Lecture Notes in Computer Science 8693. Springer International Publishing, Sept. 6, 2014, pp. 740–755. ISBN: 978-3-319-10601-4 978-3-319-10602-1. DOI: 10.1007/978-3-319-10602-1_48. URL: http://link.springer.com/chapter/10.1007/978-3-319-10602-1_48 (cit. on pp. 32, 34).

[99] Cheng-Lin Liu et al. "ICDAR 2011 Chinese Handwriting Recognition Competition". In: *2011 International Conference on Document Analysis and Recognition*. 2011 International Conference on Document Analysis and Recognition. Sept. 2011, pp. 1464–1469. DOI: 10.1109/ICDAR.2011.291 (cit. on pp. 12, 27).

[100] Shu Liu et al. "Path Aggregation Network for Instance Segmentation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Salt Lake City, United States, 2018. arXiv: 1803.01534. URL: http://arxiv.org/abs/1803.01534 (cit. on p. 33).

[101] Tao Liu et al. "Comparing Fully Convolutional Networks, Random Forest, Support Vector Machine, and Patch-Based Deep Convolutional Neural Networks for Object-Based Wetland Mapping Using Images from Small Unmanned Aircraft System". In: *GIScience & Remote Sensing* 55.2 (Mar. 4, 2018), pp. 243–264. ISSN: 1548-1603. DOI: 10.1080/15481603.2018.1426091. URL: https://doi.org/10.1080/15481603.2018.1426091 (cit. on p. 40).

[102] Wei Liu et al. "SSD: Single Shot MultiBox Detector". In: *Computer Vision – ECCV 2016*. European Conference on Computer Vision. Lecture Notes in Computer Science. Springer, Cham, Oct. 8, 2016, pp. 21–37. ISBN: 978-3-319-46447-3 978-3-319-46448-0. DOI: 10.1007/978-3-319-46448-0_2. URL: https://link.springer.com/chapter/10.1007/978-3-319-46448-0_2 (cit. on p. 32).

[103] Yansong Liu et al. "Dense Semantic Labeling of Very-High-Resolution Aerial Imagery and LiDAR with Fully-Convolutional Neural Networks and Higher-Order CRFs". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. Honolulu, United States, July 2017, pp. 1561–1570. DOI: 10.1109/CVPRW.2017.200 (cit. on p. 40).

[104] Jonathan Long, Evan Shelhamer, and Trevor Darrell. "Fully Convolutional Networks for Semantic Segmentation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2015, pp. 3431–3440. DOI: 10.1109/CVPR.2015.7298965 (cit. on pp. 32, 33, 40).

[105] Ilya Loshchilov and Frank Hutter. "SGDR: Stochastic Gradient Descent with Warm Restarts". In: *Proceedings of the International Conference on Learning Representations (ICLR)*. 2017 (cit. on p. 18).

[106] David G. Lowe. "Object Recognition from Local Scale-Invariant Features". In: *Proceedings of the Seventh IEEE International Conference on Computer Vision*. Proceedings of the Seventh IEEE International Conference on Computer Vision. Vol. 2. 1999, 1150–1157 vol.2. DOI: 10.1109/ICCV.1999.790410 (cit. on pp. 12, 20, 32).

[107] Andrew L. Maas, Awni Y. Hannun, and Andrew Y. Ng. "Rectifier Nonlinearities Improve Neural Network Acoustic Models". In: *ICML Workshop on Deep Learning for Audio, Speech and Language Processing*. 2013 (cit. on p. 15).

[108] E. Maggiori et al. "Fully Convolutional Neural Networks for Remote Sensing Image Classification". In: *2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. 2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS). July 2016, pp. 5071–5074. DOI: 10.1109/IGARSS.2016.7730322 (cit. on p. 40).

[109] Stéphane Mallat. *Une exploration des signaux en ondelettes*. Palaiseau: Éditions de l'École polytechnique, Sept. 12, 2001. 637 pp. ISBN: 978-2-7302-0733-1 (cit. on p. 20).

[110] Stjepan Marčelja. "Mathematical Description of the Responses of Simple Cortical Cells". In: *Journal of the Optical Society of America* 70.11 (Nov. 1, 1980), pp. 1297–1300. DOI: 10.1364/JOSA.70.001297. URL: https://www.osapublishing.org/josa/abstract.cfm?uri=josa-70-11-1297 (cit. on p. 20).

[111] Dimitrios Marmanis et al. "Classification With an Edge: Improving Semantic Image Segmentation with Boundary Detection". In: *ISPRS Journal of Photogrammetry and Remote Sensing* (2017). DOI: 10.1016/j.isprsjprs.2017.11.009. arXiv: 1612.01337 (cit. on p. 40).

[112] Warren S. McCulloch and Walter H. Pitts. "A Logical Calculus of the Ideas Immanent in Nervous Activity". In: *Bulletin of Mathematical Biophysics* 5 (1943), pp. 115–133. URL: http://www.cse.chalmers.se/~coquand/AUTOMATA/mcp.pdf (cit. on pp. 10, 11).

[113] Farid Melgani and Lorenzo Bruzzone. "Classification of Hyperspectral Remote Sensing Images with Support Vector Machines". In: *IEEE Transactions on Geoscience and Remote Sensing* 42.8 (Aug. 2004), pp. 1778–1790. ISSN: 0196-2892. DOI: 10.1109/TGRS.2004.831865 (cit. on p. 38).

[114] Hrushikesh Mhaskar, Qianli Liao, and Tomaso A. Poggio. "When and Why Are Deep Networks Better than Shallow Ones?" In: *AAAI*. 2017, pp. 2343–2349 (cit. on p. 15).

[115] Marvin Minsky and Seymour A. Papert. *Perceptrons*. MIT Press, 1969. URL: https://mitpress.mit.edu/books/perceptrons (cit. on p. 11).

[116] Volodymyr Mnih. "Machine Learning for Aerial Image Labeling". University of Toronto, 2013 (cit. on p. 39).

[117] Hans Moravec. *Mind Children – The Future of Robot & Human Intelligence*. Cambridge: Harvard University Press, 1988. 224 pp. ISBN: 978-0-674-57618-6 (cit. on p. 26).

[118] Vinod Nair and Geoffrey E. Hinton. "Rectified Linear Units Improve Restricted Boltzmann Machines". In: *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*. 2010, pp. 807–814 (cit. on p. 14).

[119] Vladimir Nekrasov, Janghoon Ju, and Jaesik Choi. "Global Deconvolutional Networks for Semantic Segmentation". In: *British Machine Vision Conference*. 2016. arXiv: 1602.03930. URL: http://arxiv.org/abs/1602.03930 (cit. on pp. 33, 34).

[120] Yurii Nesterov. "A Method of Solving a Convex Programming Problem with Convergence Rate O (1/K2)". In: *Soviet Mathematics Doklady*. Vol. 27. 1983, pp. 372–376 (cit. on p. 18).

[121] Gerhard Neuhold et al. "The Mapillary Vistas Dataset for Semantic Understanding of Street Scenes". In: *Proceedings of the International Conference on Computer Vision (ICCV), Venice, Italy*. 2017, pp. 22–29 (cit. on p. 32).

[122] Keiller Nogueira et al. "Learning to Semantically Segment High-Resolution Remote Sensing Images". In: *2016 23rd International Conference on Pattern Recognition (ICPR)*. 2016 23rd International Conference on Pattern Recognition (ICPR). Dec. 2016, pp. 3566–3571. DOI: 10.1109/ICPR.2016.7900187 (cit. on p. 39).

[123] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. "Learning Deconvolution Network for Semantic Segmentation". In: *2015 IEEE International Conference on Computer Vision (ICCV)*. 2015 IEEE International Conference on Computer Vision (ICCV). Dec. 2015, pp. 1520–1528. DOI: 10.1109/ICCV.2015.178 (cit. on p. 33).

[124] Avital Oliver et al. "Realistic Evaluation of Semi-Supervised Learning Algorithms". In: *Proceedings of the International Conference on Learning Representations Workshops (ICLR)*. 2018 (cit. on p. 19).

[125] Edouard Oyallon. "Building a Regular Decision Boundary with Deep Networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Honolulu, United States, July 2017, pp. 1886–1894. DOI: 10.1109/CVPR.2017.204 (cit. on p. 14).

[126] Sakrapee Paisitkriangkrai et al. "Effective Semantic Pixel Labelling with Convolutional Networks and Conditional Random Fields". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. June 2015, pp. 36–43. DOI: 10.1109/CVPRW.2015.7301381 (cit. on p. 40).

[127] Mahesh Pal. "Random Forest Classifier for Remote Sensing Classification". In: *International Journal of Remote Sensing* 26.1 (Jan. 1, 2005), pp. 217–222. ISSN: 0143-1161. DOI: 10.1080/01431160412331269698. URL: https://doi.org/10.1080/01431160412331269698 (cit. on p. 38).

[128] Mahesh Pal and Paul M. Mather. "Support Vector Machines for Classification in Remote Sensing". In: *International Journal of Remote Sensing* 26.5 (Mar. 1, 2005), pp. 1007–1011. ISSN: 0143-1161. DOI: 10.1080/01431160512331314083. URL: https://doi.org/10.1080/01431160512331314083 (cit. on p. 38).

[129] Maria Papadomanolaki, Maria Vakalopoulou, and Konstantinos Karantzalos. "Patch-Based Deep Learning Architectures for Sparse Annotated Very High Resolution Datasets". In: *2017 Joint Urban Remote Sensing Event (JURSE)*. 2017 Joint Urban Remote Sensing Event (JURSE). Mar. 2017, pp. 1–4. DOI: 10.1109/JURSE.2017.7924538 (cit. on p. 39).

[130] Constantine P. Papageorgiou, Michael Oren, and Tomaso Poggio. "A General Framework for Object Detection". In: *Proceedings of the Sixth International Conference on Computer Vision*. ICCV '98. Washington, DC, USA: IEEE Computer Society, 1998, pp. 555–. ISBN: 978-81-7319-221-0. URL: http://dl.acm.org/citation.cfm?id=938978.939174 (cit. on p. 20).

[131] Seymour Papert. *The Summer Vision Project*. 1966 (cit. on pp. 10, 26).

[132] Peeta Basa Pati and A. G. Ramakrishnan. "Word Level Multi-Script Identification". In: *Pattern Recognition Letters* 29.9 (July 1, 2008), pp. 1218–1229. ISSN: 0167-8655. DOI: 10.1016/j.patrec.2008.01.027. URL: http://www.sciencedirect.com/science/article/pii/S0167865508000354 (cit. on p. 20).

[133] Chao Peng et al. "Large Kernel Matters – Improve Semantic Segmentation by Global Convolutional Network". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. July 2017, pp. 4353–4361. URL: http://openaccess.thecvf.com/content_cvpr_2017/html/Peng_Large_Kernel_Matters_CVPR_2017_paper.html (cit. on p. 34).

[134] M. T. Pham, E. Aptoula, and S. Lefèvre. "Feature Profiles from Attribute Filtering for Classification of Remote Sensing Images". In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 11.1 (Jan. 2018), pp. 249–256. ISSN: 1939-1404. DOI: 10.1109/JSTARS.2017.2773367 (cit. on p. 40).

[135] Nicolas Pinto, David D. Cox, and James J. DiCarlo. "Why Is Real-World Visual Object Recognition Hard?" In: *PLOS Computational Biology* 4.1 (Jan. 25, 2008), e27. ISSN: 1553-7358. DOI: 10.1371/journal.pcbi.0040027. URL: http://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.0040027 (cit. on p. 25).

[136] Tomaso Poggio et al. "Why and When Can Deep-but Not Shallow-Networks Avoid the Curse of Dimensionality: A Review". In: *International Journal of Automation and Computing* 14.5 (Oct. 1, 2017), pp. 503–519. ISSN: 1476-8186, 1751-8520. DOI: 10.1007/s11633-017-1054-2. URL: https://link.springer.com/article/10.1007/s11633-017-1054-2 (cit. on p. 15).

[137] Boris Polyak and Anatoli Juditsky. "Acceleration of Stochastic Approximation by Averaging". In: *SIAM Journal on Control and Optimization* 30.4 (July 1992), pp. 838–855. ISSN: 0363-0129. DOI: 10.1137/0330046. URL: http://dx.doi.org/10.1137/0330046 (cit. on p. 18).

[138] Ning Qian. "On the Momentum Term in Gradient Descent Learning Algorithms". In: *Neural Networks* 12.1 (Jan. 1, 1999), pp. 145–151. ISSN: 0893-6080. DOI: 10.1016/S0893-6080(98)00116-6. URL: https://www.sciencedirect.com/science/article/pii/S0893608098001166 (cit. on p. 18).

[139] Rajat Raina, Anand Madhavan, and Andrew Y. Ng. "Large-Scale Deep Unsupervised Learning Using Graphics Processors". In: *Proceedings of the 26th Annual International Conference on Machine Learning*. ICML '09. New York, NY, USA: ACM, 2009, pp. 873–880. ISBN: 978-1-60558-516-1. DOI: 10.1145/1553374.1553486. URL: http://doi.acm.org/10.1145/1553374.1553486 (cit. on p. 12).

[140] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III*. Ed. by Nassir Navab et al. Cham: Springer International Publishing, 2015, pp. 234–241. ISBN: 978-3-319-24574-4. DOI: 10.1007/978-3-319-24574-4_28. URL: https://doi.org/10.1007/978-3-319-24574-4_28 (cit. on p. 33).

[141] Frank Rosenblatt. *The Perceptron: A Probabilistic Model for Information Storage and Organization In The Brain*. 1957 (cit. on pp. 11, 26).

[142] J. W. Rouse. "Monitoring Vegetation Systems in the Great Plains with ERTS". In: Jan. 1, 1974. URL: https://ntrs.nasa.gov/search.jsp?R=19740022614 (cit. on p. 37).

[143] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. "Learning Internal Representations by Error Propagation". In: ed. by David E. Rumelhart, James L. McClelland, and CORPORATE PDP Research Group. Cambridge, MA, USA: MIT Press, 1986, pp. 318–362. ISBN: 978-0-262-68053-0. URL: http://dl.acm.org/citation.cfm?id=104279.104293 (cit. on pp. 11, 15, 16).

[144] Olga Russakovsky et al. "ImageNet Large Scale Visual Recognition Challenge". In: *International Journal of Computer Vision* 115.3 (Apr. 11, 2015), pp. 211–252. ISSN: 0920-5691, 1573-1405. DOI: 10.1007/s11263-015-0816-y. URL: http://link.springer.com/article/10.1007/s11263-015-0816-y (cit. on pp. 27, 28).

[145] Ruslan Salakhutdinov and Geoffrey Hinton. "Deep Boltzmann Machines". In: *Artificial Intelligence and Statistics*. Artificial Intelligence and Statistics. Apr. 15, 2009, pp. 448–455. URL: http://proceedings.mlr.press/v5/salakhutdinov09a.html (cit. on p. 12).

[146] Shibani Santurkar et al. "How Does Batch Normalization Help Optimization? (No, It Is Not About Internal Covariate Shift)". In: (May 29, 2018). arXiv: 1805.11604 [cs, stat]. URL: http://arxiv.org/abs/1805.11604 (cit. on p. 26).

[147] Andrew M. Saxe, James L. McClelland, and Surya Ganguli. "Exact Solutions to the Nonlinear Dynamics of Learning in Deep Linear Neural Networks". In: *Proceedings of the International Conference on Learning Representations (ICLR)*. 2014. arXiv: 1312.6120 (cit. on pp. 17, 19).

[148] Henry Schneiderman and Takeo Kanade. "Probabilistic Modeling of Local Appearance and Spatial Relationships for Object Recognition". In: *Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference On*. IEEE, 1998, pp. 45–51 (cit. on p. 27).

[149] Pierre Sermanet et al. "OverFeat: Integrated Recognition, Localization and Detection Using Convolutional Networks". In: *Proceedings of the International Conference on Learning Representations (ICLR)*. 2014. arXiv: 1312.6229. URL: http://arxiv.org/abs/1312.6229 (cit. on p. 32).

[150] T. Serre, L. Wolf, and T. Poggio. "Object Recognition with Features Inspired by Visual Cortex". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Vol. 2. June 2005, 994–1000 vol. 2. DOI: 10.1109/CVPR.2005.254 (cit. on p. 12).

[151] Thomas Serre et al. "A Quantitative Theory of Immediate Visual Recognition". In: *Progress in Brain Research* 165 (2007), pp. 33–56. ISSN: 0079-6123. DOI: 10.1016/S0079-6123(06)65004-8. pmid: 17925239 (cit. on p. 12).

[152] M. J. Shensa. "The Discrete Wavelet Transform: Wedding the a Trous and Mallat Algorithms". In: *IEEE Transactions on Signal Processing* 40.10 (Oct. 1992), pp. 2464–2482. ISSN: 1053-587X. DOI: 10.1109/78.157290 (cit. on p. 23).

[153] Jamie Sherrah. "Fully Convolutional Networks for Dense Semantic Labelling of High-Resolution Aerial Imagery". In: (June 8, 2016). arXiv: 1606.02585 [cs]. URL: http://arxiv.org/abs/1606.02585 (cit. on p. 40).

[154] Jamie Shotton, Matthew Johnson, and Roberto Cipolla. "Semantic Texton Forests for Image Categorization and Segmentation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2008, pp. 1–8. DOI: 10.1109/CVPR.2008.4587503 (cit. on p. 32).

[155] Jamie Shotton et al. "Real-Time Human Pose Recognition in Parts from Single Depth Images". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2011, pp. 1297–1304. DOI: 10.1109/CVPR.2011.5995316 (cit. on p. 32).

[156] Karen Simonyan and Andrew Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition". In: *Proceedings of the International Conference on Learning Representations (ICLR)*. May 2015. URL: http://arxiv.org/abs/1409.1556 (cit. on pp. 29, 33).

[157] Irwin Sobel. "An Isotropic 3x3 Image Gradient Operator". In: *Presentation at Stanford A.I. Project 1968* (Feb. 8, 2014) (cit. on pp. 20, 21).

[158] Sho Sonoda and Noboru Murata. "Neural Network with Unbounded Activation Functions Is Universal Approximator". In: *Applied and Computational Harmonic Analysis* 43.2 (Sept. 1, 2017), pp. 233–268. ISSN: 1063-5203. DOI: 10.1016/j.acha.2015.12.005. URL: http://www.sciencedirect.com/science/article/pii/S1063520315001748 (cit. on p. 15).

[159] Nitish Srivastava et al. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting". In: *Journal of Machine Learning Research* 15 (2014), pp. 1929–1958. URL: http://jmlr.org/papers/v15/srivastava14a.html (cit. on pp. 19, 26, 30).

[160] J. Stallkamp et al. "The German Traffic Sign Recognition Benchmark: A Multi-Class Classification Competition". In: *The 2011 International Joint Conference on Neural Networks*. The 2011 International Joint Conference on Neural Networks. July 2011, pp. 1453–1460. DOI: 10.1109/IJCNN.2011.6033395 (cit. on pp. 12, 27).

[161] Ilya Sutskever et al. "On the Importance of Initialization and Momentum in Deep Learning". In: Proceedings of The 30th International Conference on Machine Learning. 2013, pp. 1139–1147. URL: http://jmlr.org/proceedings/papers/v28/sutskever13.html (cit. on p. 18).

[162] Christian Szegedy et al. "Going Deeper with Convolutions". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2015, pp. 1–9. DOI: 10.1109/CVPR.2015.7298594. URL: http://www.cv-foundation.org/openaccess/content_cvpr_2015/html/Szegedy_Going_Deeper_With_2015_CVPR_paper.html (cit. on pp. 29, 30, 33).

[163] Christian Szegedy et al. "Rethinking the Inception Architecture for Computer Vision". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2016, pp. 2818–2826. DOI: 10.1109/CVPR.2016.308 (cit. on p. 29).

[164] Christian Szegedy et al. "Inception-v4, Inception-Resnet and the Impact of Residual Connections on Learning." In: *AAAI Conference on Artificial Intelligence*. AAAI Conference on Artificial Intelligence. Vol. 4. 2017, p. 12 (cit. on p. 30).

[165] Richard Szeliski. *Computer Vision: Algorithms and Applications*. Texts in Computer Science. London: Springer-Verlag, 2011. ISBN: 978-1-84882-934-3. URL: //www.springer.com/us/book/9781848829343 (cit. on p. 26).

[166] Tijmen Tielman and Geoffrey Hinton. *Lecture 6.5—RmsProp: Divide the Gradient by a Running Average of Its Recent Magnitude*. 2012 (cit. on p. 18).

[167] A. M. Turing. *Computing Machinery and Intelligence*. 1950 (cit. on p. 10).

[168] J. R. R. Uijlings et al. "Selective Search for Object Recognition". In: *International Journal of Computer Vision* 104.2 (Sept. 1, 2013), pp. 154–171. ISSN: 0920-5691, 1573-1405. DOI: 10.1007/s11263-013-0620-5. URL: https://link.springer.com/article/10.1007/s11263-013-0620-5 (cit. on p. 32).

[169] Shimon Ullman. "Aligning Pictorial Descriptions: An Approach to Object Recognition". In: *Cognition* 32.3 (Aug. 1, 1989), pp. 193–254. ISSN: 0010-0277. DOI: 10.1016/0010-0277(89)90036-X. URL: http://www.sciencedirect.com/science/article/pii/001002778990036X (cit. on p. 27).

[170] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. "Deep Image Prior". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Salt Lake City, United States, June 2018. arXiv: 1711.10925. URL: http://arxiv.org/abs/1711.10925 (cit. on p. 23).

[171] M. Vakalopoulou et al. "Building Detection in Very High Resolution Multispectral Data with Deep Learning Features". In: *2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. 2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS). July 2015, pp. 1873–1876. DOI: 10.1109/IGARSS.2015.7326158 (cit. on p. 39).

[172] John E. Vargas et al. "Superpixel-Based Interactive Classification of Very High Resolution Images". In: *27th SIBGRAPI Conference on Graphics, Patterns and Images*. Aug. 2014, pp. 173–179. DOI: 10.1109/SIBGRAPI.2014.49 (cit. on p. 39).

[173] Andreas Veit, Michael Wilber, and Serge Belongie. "Residual Networks Behave Like Ensembles of Relatively Shallow Networks". In: *Proceedings of the 30th International Conference on Neural Information Processing Systems*. NIPS'16. USA: Curran Associates Inc., 2016, pp. 550–558. ISBN: 978-1-5108-3881-9. URL: http://dl.acm.org/citation.cfm?id=3157096.3157158 (cit. on p. 31).

[174] Michel Vidal-Naquet and Shimon Ullman. "Object Recognition with Informative Features and Linear Classification." In: *ICCV*. Vol. 3. 2003, p. 281 (cit. on p. 27).

[175] Paul Viola and Michael Jones. "Robust Real-Time Object Detection". In: *International Journal of Computer Vision*. 2001 (cit. on pp. 20, 32).

[176] Michele Volpi and Devis Tuia. "Dense Semantic Labeling of Subdecimeter Resolution Images With Convolutional Neural Networks". In: *IEEE Transactions on Geoscience and Remote Sensing* 55.2 (Feb. 2017), pp. 881–893. ISSN: 0196-2892. DOI: 10.1109/TGRS.2016.2616585 (cit. on p. 40).

[177] Li Wan et al. "Regularization of Neural Networks Using DropConnect". In: *International Conference on Machine Learning*. International Conference on Machine Learning. Feb. 13, 2013, pp. 1058–1066. URL: http://proceedings.mlr.press/v28/wan13.html (cit. on p. 20).

[178] Paul John Werbos. "Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences". Harvard University, 1975. 906 pp. (cit. on pp. 11, 15, 16).

[179] Bernard Widrow. *An Adaptive "ADALINE" Neuron Using Chemical "Memistors"*. Stanford University, Oct. 17, 1960. URL: http://www-isl.stanford.edu/~widrow/papers/t1960anadaptive.pdf (cit. on p. 11).

[180] Rodney Winter and Bernard Widrow. "MADALINE RULE II: A Training Algorithm for Neural Networks". In: *IEEE 1988 International Conference on Neural Networks*. IEEE 1988 International Conference on Neural Networks. July 1988, 401–408 vol.1. DOI: 10.1109/ICNN.1988.23872 (cit. on p. 11).

[181] Zifeng Wu, Chunhua Shen, and Anton Van Den Hengel. "High-Performance Semantic Segmentation Using Very Deep Fully Convolutional Networks". In: (Apr. 14, 2016). arXiv: 1604.04339 [cs]. URL: http://arxiv.org/abs/1604.04339 (cit. on p. 33).

[182] Huan Xie et al. "New Hyperspectral Difference Water Index for the Extraction of Urban Water Bodies by the Use of Airborne Hyperspectral Images". In: *Journal of Applied Remote Sensing* 8.1 (July 2014), p. 085098. ISSN: 1931-3195, 1931-3195. DOI: 10.1117/1.JRS.8.085098. URL: https://www.spiedigitallibrary.org/journals/Journal-of-Applied-Remote-Sensing/volume-8/issue-1/085098/New-hyperspectral-difference-water-index-for-the-extraction-of-urban/10.1117/1.JRS.8.085098.short (cit. on p. 37).

[183] Jason Yosinski et al. "How Transferable Are Features in Deep Neural Networks?" In: *Advances in Neural Information Processing Systems*. Neural Information Processing Systems (NIPS). 2014, pp. 3320–3328. URL: http://papers.nips.cc/paper/5347-how-transferable-are-features-in-deep-neural-networks (cit. on p. 21).

[184] Fisher Yu and Vladlen Koltun. "Multi-Scale Context Aggregation by Dilated Convolutions". In: *Proceedings of the International Conference on Learning Representations (ICLR)*. Nov. 23, 2015. URL: http://arxiv.org/abs/1511.07122 (cit. on pp. 23, 33, 34).

[185] Matthew D. Zeiler. "ADADELTA: An Adaptive Learning Rate Method". In: (Dec. 22, 2012). arXiv: 1212.5701 [cs]. URL: http://arxiv.org/abs/1212.5701 (cit. on p. 18).

[186] Matthew Zeiler and Rob Fergus. "Stochastic Pooling for Regularization of Deep Convolutional Neural Networks". In: *Proceedings of the International Conference on Learning Representations (ICLR)*. 2013. URL: https://openreview.net/forum?id=l_PClqDdLb5Bp (cit. on p. 20).

[187] Matthew Zeiler and Rob Fergus. "Visualizing and Understanding Convolutional Networks". In: *Computer Vision–ECCV 2014*. Springer, 2014, pp. 818–833. URL: http://link.springer.com/chapter/10.1007/978-3-319-10590-1_53 (cit. on p. 28).

[188] Hengshuang Zhao et al. "Pyramid Scene Parsing Network". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Honolulu, United States, July 2017, pp. 2881–2890. DOI: 10.1109/CVPR.2017.660. URL: http://openaccess.thecvf.com/content_cvpr_2017/html/Zhao_Pyramid_Scene_Parsing_CVPR_2017_paper.html (cit. on p. 34).

[189] Junbo Zhao et al. "Stacked What-Where Auto-Encoders". In: *Proceedings of the International Conference on Learning Representations (ICLR)*. June 8, 2015. URL: http://arxiv.org/abs/1506.02351 (cit. on pp. 24, 33).

[190] Shuai Zheng et al. "Conditional Random Fields as Recurrent Neural Networks". In: *Proceedings of the IEEE International Conference on Computer Vision*. IEEE International Conference on Computer Vision (ICCV). Dec. 2015, pp. 1529–1537. DOI: 10.1109/ICCV.2015.179 (cit. on p. 34).

[191] Bolei Zhou et al. "Scene Parsing through ADE20K Dataset". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Honolulu, United States, July 2017, pp. 5122–5130. DOI: 10.1109/CVPR.2017.544 (cit. on p. 34).

[192] Yi-Tong Zhou and Rama Chellappa. "Stereo Matching Using a Neural Network". In: *ICASSP-88., International Conference on Acoustics, Speech, and Signal Processing*. ICASSP-88., International Conference on Acoustics, Speech, and Signal Processing. Apr. 1988, 940–943 vol.2. DOI: 10.1109/ICASSP.1988.196745 (cit. on p. 24).

[193] Barret Zoph and Quoc Le. "Neural Architecture Search with Reinforcement Learning". In: *Proceedings of the International Conference on Learning Representations (ICLR)*. 2017 (cit. on p. 15).

[194] Will Zou et al. "Generic Object Detection with Dense Neural Patterns and Regionlets". In: *Proceedings of the British Machine Vision Conference*. BMVA Press, 2014, pp. 72.1–72.11. ISBN: 978-1-901725-52-0. DOI: 10.5244/C.28.72. URL: http://www.bmva.org/bmvc/2014/papers/paper050/index.html (cit. on p. 32).

Chapter

3 Automated semantic mapping of aerial images

Et la géographie, c'est exact, m'a beaucoup servi. Je savais reconnaître, du premier coup d'œil, la Chine de l'Arizona. C'est très utile, si l'on est égaré pendant la nuit.

— Antoine de Saint-Exupéry (Le Petit Prince, 1943)

Contents

Summary:

This chapter presents two semantic segmentation strategies for very high resolution aerial images: region-based classification and fully convolutional networks.

Region-based classification is rooted in unsupervised partitioning algorithms that divide the image in multiple homogeneous areas. A classifier is then applied on all regions independently based on deep features computed on each patch. We use pretrained CNN on ImageNet and show that the representation learnt on multimedia images can successfully be transfered for remote sensing data processing.

Moreover, we identify desirable properties of unsupervised segmentation algorithms that are used for region-based classification. Especially, we empirically show how undersegmentation impedes both feature extraction and segmentation and that reducing the region size to alleviate this does not scale on large images. We therefore introduce fully convolutional networks for remote sensing data to perform an end-to-end pixelwise feature extraction and classification in one forward pass.

We adapt state of the arts semantic segmentation models from the computer vision literature to aerial images and experimentally validate that they outperform significantly usual classification strategies. We also introduce multiscale convolutional layers to learn from various levels of spatial context.

3.1 Region-based classification of aerial images

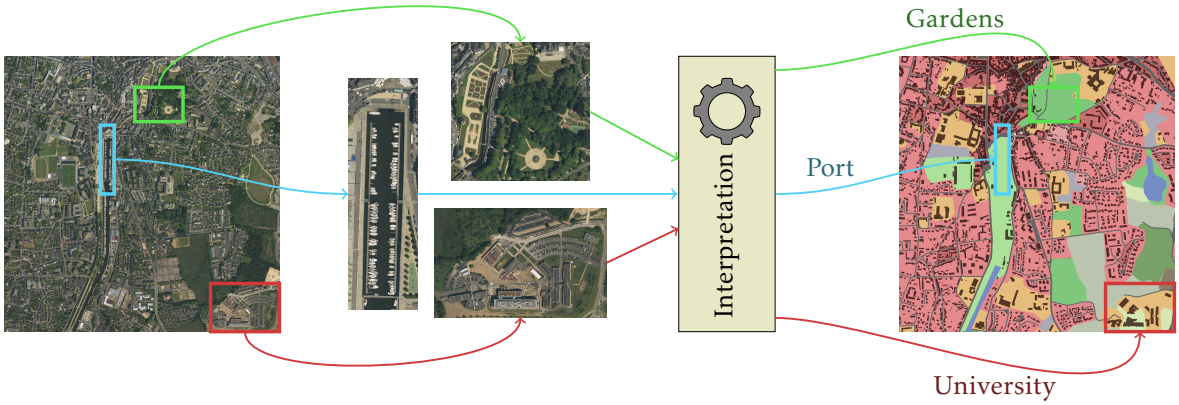


Figure 3.1: Semantic mapping of aerial images.

This chapter deals with semantic mapping of red-green-blue (RGB) or infrared-red-green (IRRG) three-channels aerial images, either at VHR (50cm) or EHR (10cm). Those images are acquired with digital cameras similar to consumer-grade ones and therefore present similar characteristics: high resolution, (pseudo-)RGB color space and well-known sensor. Therefore, they constitute a natural first step for applying modern computer vision to remote sensing images.

Our goal is to learn a semantic maps from images, i.e. an associative relationship between each element of the image and one of the classes of interest (Fig. 3.1). More precisely, given an image I with dimensions $\mathrm{M} \times \mathrm{N}$ and a set of labels from 1 to n we wish to map every pixel $\mathrm{I}_{i,j}$ to a class $k_{i,j} \in \{1,\dots,n\}$. To do so we approximate f defined as:

$$\forall (i,j) \in \{1 \dots \mathrm{M}\} \times \{1 \dots \mathrm{N}\} \quad f(\mathrm{I}[i,j]) = k_{i,j} \,. \tag{3.1}$$

Contrary to the object recognition problem in which we try to map one or more labels to the whole image, here we look for a *dense* classification. Because of spatial regularities and relationships between neighbouring pixels, a classified image can be seen as groups of neighbouring pixels all belonging to the same class. Generating such a map is called "semantic segmentation" in the literature.

To build a statistical model that approximates f, we can split the function in two steps. The first step, named feature extraction, projects the raw information into a predefined learning space. The second step consists in dividing the representation space in disjoint subspaces, i.e. to perform a classification.

Formally, let $\mathcal{E}$ denote the input space, $\mathcal{R}$ the representation space and $\{y_1,\dots,y_k\}$ the set of labels. f is the composotion of two functions:

$$f = c \circ p \tag{3.2}$$

where p is a projection from $\mathcal{E} \to \mathcal{R}$ and $c : \mathcal{R} \to \{y_1, \dots y_k\}$ such as:

$$\forall x \in \mathcal{E} \text{ a set of pixels}, f(x) = c(p(x)) = y \in \{y_1,\dots,y_k\} \,. \tag{3.3}$$

The choice of the representation space (and the projection p) is strongly tied to the choice of the classifier. For example, a linear SVM divide the representation space using hyperplanes that maximise the margin between different classses. Therefore it is preferable that the image of the input space $\mathcal{E}$ by p is linearily separable in $\mathcal{R}$ to match this assumption. Starting now, we will call "features" the elemetns of $\mathcal{R}$ and "feature extractor" the projection p.

This section details the principle behind region-based classification and the current state of the art in unsupervised image segmentation, before studying the properties of these algorithms when applied on aerial images.

3.1.1 Region-based classification

As we have seen in the previous chapter, image classification is at the center of a large body of works in the literature. Nonetheless let us remind that our interest is focused not on global matching between images and labels, but on dense pixel-wise mapping. As a first approach, it is feasible to deal indepently with image segmentation on the one hand and semantization on the other hand. This dichotomy allows us to split the image in many regions that will be independently classified: this process is called region-based classification. First, we apply a segmentation algorithm on the image and then a classifier assigns a label to each of the sub-region.

Definition 6. *Region-based classification of an image* I *consists in finding a partition* $\mathrm{P} = \mathrm{P}_1, \ldots, \mathrm{P}_n$ *such as:*

$$\bigcup_{i=1}^{n} \mathrm{P}_i = \mathrm{I} \quad \textit{(segmentation)}$$

and a classification function $\mathcal{C}$ *such as:*

$$\mathcal{C}(\mathrm{P}_i) = k_i \quad \textit{(classification)}$$

where k_i *is the label assigned to the* i^{th}*region*[1].

Many algorithms have been introduced for remote sensing image segmentation, for example using attribute profiles based on hierarchical tree-like segmentations [9], superpixel segmentations combined with visual bag-of-words techniques [33] or deep neural networks [22]. To begin, we will review unsupervised segmentation algorithms and study how choosing one over another impacts the classifier's accuracy in a region-based classification pipeline.

3.1.2 Image segmentation algorithms

There are many unsupervised image segmentation algorithms that can be applied to grayscale and color images in various color spaces such as RGB, hue-saturation-intensity or L*a*b* CIE 1976 (CIELAB).

A first group of segmentation algorithms treats the 2D image as graph: pixels are nodes and vertices are neighbourhood relationships such as pixel similarity. Building the regions is done by merging nodes based on the value of their vertices, often by propagating some constraint. This principles is at the core of the Felzenszwalb-Huttenlocher (FH) [19] algorithm that divides the image by computing the minimal spanning tree on its graph. Similarly, *Normalized Cuts* [56], Entropy Rate Superpixel (ERS) [38] and the algorithm from Grady [24] respectively use graph partition, entropy minimization through random walk and diffusion equations to segment the image.

Another family – which is more and more popular – finds its roots in iterative clustering algorithms. This principle grew into deux groups of "superpixel" segmentation algorithms. The first one originates from the Simple Linear Iterative Clustering (SLIC) algorithm [1]. SLIC projects every pixel in a 5-dimensional space (CIELAB color and (x, y) coordinates) and applies a variant of the k-means iterative clustering algorithm. SLIC initializes as many centers as specified by the user on a regular grid. These centers iteratively absorb their neighbours along the regions' borders. Several variants have been published such as the faster *Preemptive SLIC* [47] or Linear Spectral Clustering (LSC) [34] which includes

[1] This label is generally the most representend one inside the region.

Figure 3.2: Natural image segmentation. To capture finer details, some algorithms rely on irregular shapes. Image credits: Tom Frydenlund, CC0.

global constraints in addition to the local iterative update and the Superpixels with Contour Adherence using Linear Path (SCALP) algorithm [21]. SCALP forbids the generation of irregularly-shaped regions by scanning all pixels between the superpixel center and the one to be added. Moreover SCALP uses the result of an edge detection as an auxiliary input to strenghten its stickiness to object edges. The second group of superpixel-based methods derivates from the k-medoids algorithm. This works by projecting pixels into a non-euclidean space, generally RGB-(x, y) which has a dimensionality of 5, and performing a clustering by finding the main local mode in each neighbourhood, i.e. the medoid. This approach is used in *Mean Shift* [17] and *Quickshift* [63]. Other algorithms also use iterative clustering approaches. The Superpixels Extracted via Energy-Driven Sampling (SEEDS) algorithm [61] defines blocks of pixels that can trade elements along their border to maximize an energy function computed on color histograms. SEEDS uses a hill-climbing optimization algorithm to make the blocks converge to a stable configuration. Finally, some segmentation algorithms use level sets to converge, such as Chan-Vese algorithm [14] which is based on active contours or *TurboPixel* [32] which works on the local gradients and curvature.

For grayscale images, the morphological *watershed* algorithm [8] is particularly popular. Watershed considers an image as topological elevation map in which it simulates a rise of the

water level. At first, water flows from a specified number of sources positioned on markers which can be manually inserted or automatically generated[2]. The water fills the topological map as it rises. When two sources meet, watershed erects a virtual dam along the border, which defines one of the segmentation's edges. The algorithm stops when the whole image has been flooded. Watershed is sensitive to the markers' positions and choosing them is critical to obtain clean segmentation with smooth regions. A compact variant has been introduced [47] to make watershed robust to poor initialization by making it similar to SLIC. The morphological approach can also be applied to active contours, for example in a variant of Chan-Vese's algorithm [14] with morphological active contours [45].

Finally, algorithms specific to remote sensing data have been designed for radar and multispectral images. These algorithms use multiple scales of spatial context to perform object-oriented image processing. Multi-Resolution Segmentation (MRS) [5] is an example of a common remote sensing image segmentation algorithm, thanks to its implementation in the eCognition©software. MRS tries to detect salient objects and grows regions using a heuristic spectral homogeneity criterion. The segmentation is computed at multiple scale and an *ad hoc* criterion is applied to merge some of the smaller areas. Another popular remote sensing image segmentation algorithm, Hierarchial Segmentation (HSeg) [60] is a hierchical tree-like multi-scale segmentation algorithm. Regions from the larger scales are subdivided in smaller zones recursively. HSeg uses a region-growing strategy in which neighbouring pixels are fused if they do not meet a specific dissimilarity criterion. Close regions can be merged if they seem homogeneous in order to reduce the scale of the segmentation.

3.1.3 Choosing a segmentation algorithm

As many segmentation algorithms exist, we need to study if and how choosing one over another might impact our statistical model. There are two main points to evaluate: what preprocessing is required to segment our images and which segmentation best preserves the spatial properties of the image?

Image preprocessing

Most segmentation algorithms apply a light gaussian blur to the image to smooth borders and reduce noise, which eases the segmentation. On aerial images, such a gaussian blur does not harm the segmentation and can of course be dropped for classification.

In many cases the actual segmentation is done in the CIELAB color space. CIELAB has been designed to mimick human vision and the response of the human eye to color variations, which is logarithmic and not linear. However this becomes dubious when working on remote sensing images that are not originally RGB, but contains infrared for example. In practice this conversion does not seem harmful, yet those algorithms will not be straightforwardly extensible to multispectral images that one often encounters in remote sensing. Overall, only MRS and HSeg have been designed with such data in mind.

Region shape and size

Several benchmarks have been performed [46, 2, 58] to understand strengths and weaknesses of the main segmentation algorithms. Fig. 3.2 illustrates some of the more common segmentation algorithms.

The main source of variations between different segmentations stems from the regions' shapes. Geometrical properties can greatly vary between two algorithms. For example, the FH segmentation generates very heterogeneous regions in sizes and shapes since it walks freely on the graph. It can merge similar pixels that are far apart as well as producing small regions comprised of a handful of pixels, without any parameter to control this. On the contrary,

[2] For example at the local minima of the image gradients.

superpixel segmentations and more specifically those which are based on SLIC produce visually homogeneous regions. These algorithms take as an input a compacity parameter which controls the regions' stickyness to the underlying grid. Superpixels produced by SLIC are easily constrained in size while keeping a freedom of shape. Quickshift exhibits a similar behaviour although it often produces regions more irregular than its counterparts from SLIC. The compact watershed segmentation behaves nearly exactly as superpixels techniques and is a clear improvement on the original watershed since it does not need markers anymore. This first-order analysis is in line with the existing literature [46, 2]. Overall, most superpixel flavors exhibit similar properties and be swapped without care [58].

Since our work concentrates on remote sensing image classification, we have to investigate how these algorithms deal with aerial images acquired from above. Some segmentation examples are given in Fig. 3.3. The MRS is very effective on aerial and satellite images. Indeed, although visually chaotic, the regions stick very close to actual objects' edges up to the smallest details, which is often where superpixel approaches tend to fail. This is especially interesting since objects in remote sensing data can be quite small, as cars in the example.

To go further, we will then restrict our anlysis to segmentation algorithms designed for remote sensing (HSeg et MRS) and two classical segmentation algorithms for natural images: Quickshift et SLIC. *Watershed* approaches are dismissed since they are actually close to SLIC [47] and the FH algorithm is removed because of the too high variability of its regions [46]. Now that we have chosen our segmentation algorithms, we can move on to feature extraction and region classification.

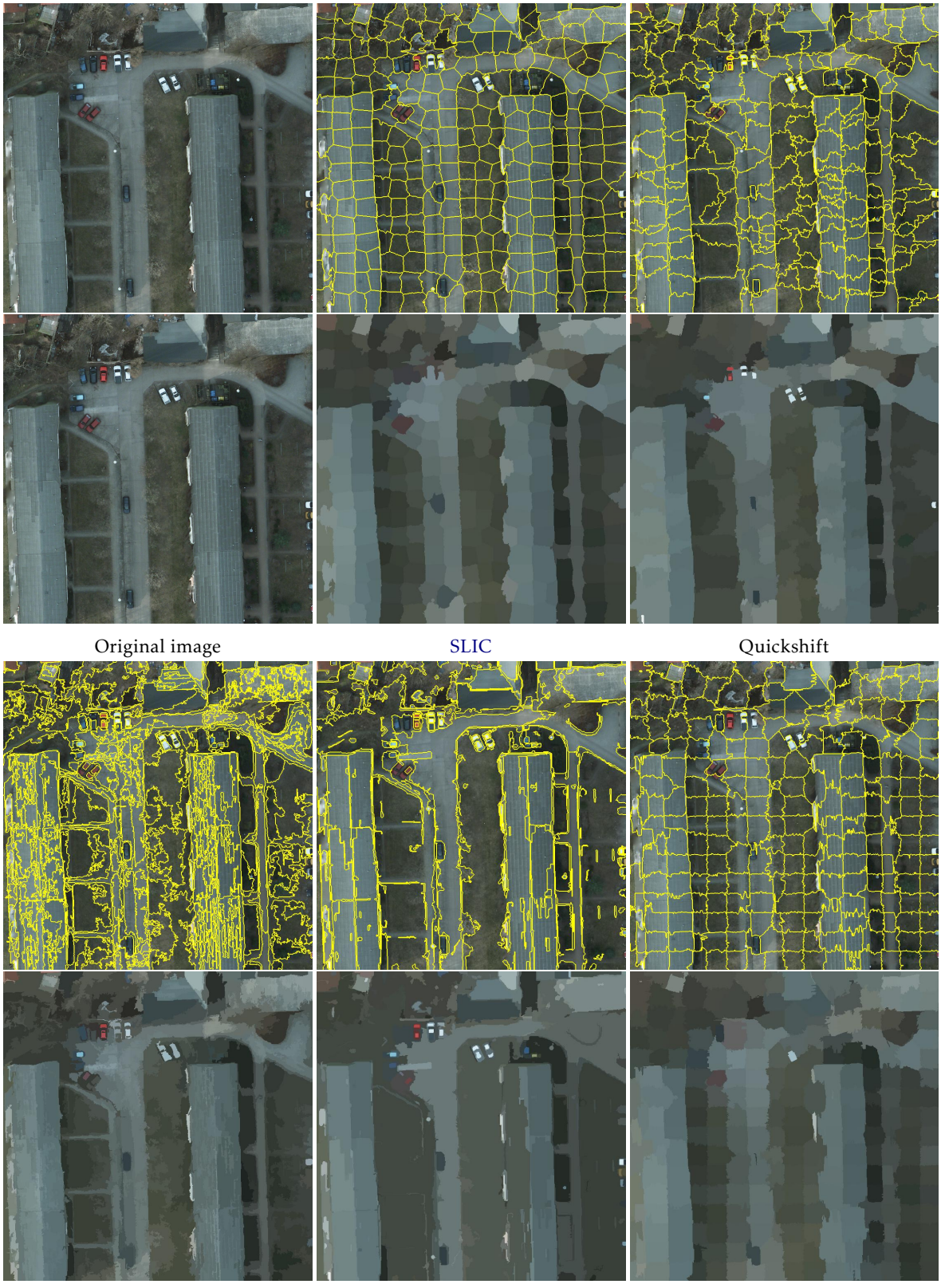


Figure 3.3: Aerial image segmentation (from ISPRS Potsdam). Cars are more or less well-segmented depending on the algorithm.

3.2 Deep neural networks

3.2.1 Convolutional neural networks as feature extractors

Deep learning's appeal resides in learning representations [7, 23]. As we saw in Chapter 2, convolutional neural networks perform a learnt feature extraction thanks to learnable filters in the first layers. As the network is trained end-to-end, the projection in the representation space is jointly optimized with the classification loss on the training samples.

By stopping the forward pass in a CNN after the softmax layer, i.e. before the last layer, we obtain activations that can be interpreted as an internal representation of the data, i.e. a deep feature vector.

Such features can be used to train usual statistical classifiers, either by retraining (*fine-tuning*) the last layer or with shallow models. It has been shown [54] that deep features computed by deep networks pretrained on ImageNet [18] fed to a linear SVM often led to state-of-the-art results compared to *ad hoc* features. Razavian et al. [54] demonstrated this on many tasks and showed that, despite its simplicity, deep features from the off-the-shelf pretrained models most of the time outperforms expert features such as HOG and SIFT. Interestingly, pretrained weights are also often better initializations than random weights when training new models, even when the two tasks vastly differ [65].

Various publications extended these findings to aerial image classification [51, 42, 30] with the same results on remote sensing datasets such as *UC Merced* and *Brazilian Coffe*. Marmanis et al. [42] and Penatti, Nogueira, and dos Santos [51] showed that deep features from ImageNet-pretrained networks significantly outperformed expert remote sensing features on aerial and satellite images, a result that Lagrange et al. [30] extended to semantic segmentation. This is counterintuitive since ImageNet is comprised of everyday images including dogs, cats, cars, people and landscapes. Yet, the sheer size of the dataset suffice to learn generic convolutional filters than can be suitable for many domains, even remote sensing.

Application to semantic cartography

Building upon segmentation and classification techniques previously described, we are able to design a complete semantic segmentation pipeline for aerial images, illustrated by Fig. 3.4:

1. Divide the image in homogeneous sub-regions using a segmentation algorithm.
2. For each region, extract an image pyramid (32×32, 64×64 and 128×128 patches) centered its barycenter to include spatial context.
3. Extract features from each patch.
4. Concatenate the features.

The training samples this process produces can be used either to fit the classifier in the learning phase or for inference during the evaluation. One benefit of the concatenation step (4) is the ability to inject arbitrary expert or multimodal features [30] to enrich the classifier.

When dealing with features, standard CNNs expect a fixed-sized input due to the presence of fully connected layers, which determine the output size of the convolutional layers and *de facto* the input dimensions. In this case, we resize the small patches to the dimensions expected by the network, e.g. 228×228 for AlexNet.

Moreover, deep features tend to exhibit a large dimensionality. As an example, AlexNet's last layer computes a vector of size 1000 for each image. Our pipeline will therefore produce a vector of size 3000 for each region. Fitting an SVM exactly in such a large space takes a long time, even on modern computers. To alleviate this burden, we use an online stochastic gradient descent optimization approximation from Bottou [10]. Detailed results are reported in Section 3.3.3.

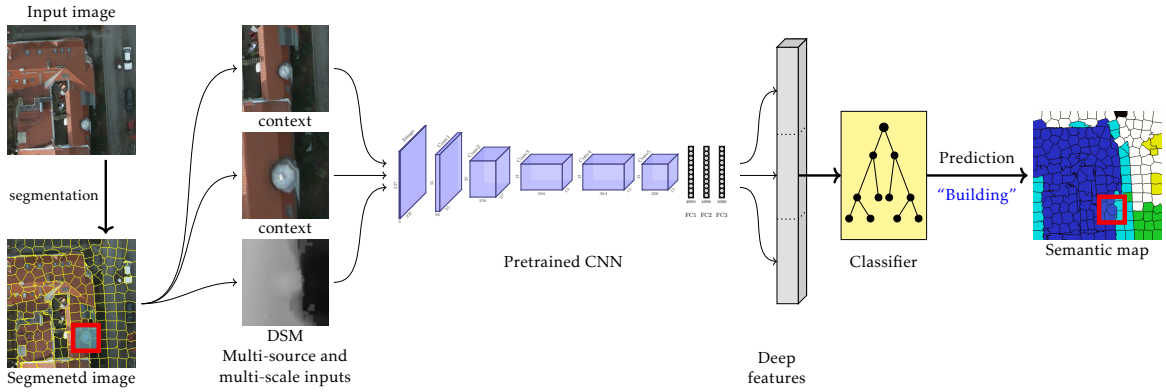


Figure 3.4: Region-based semantic segmentation of an aerial image. Every area of the segmented image is classified based on deep features from a pretrained convolutional network.

3.2.2 Fully convolutional networks

Region-based classification algorithms suffer from two drawbacks. First, the segmentation puts an upper bound to the level of detail that can be obtained in the final map. Indeed, coarse regions after the segmentaton are not refined by the classifier as the whole area will be labeled with the same class. Enhance the resolution of the semantic map requires decreasing the size of the segments which means increasing their number, proportionaly increasing the computation time to process the whole image. In extreme cases, objects can reach a subpixel size which means that an optimal classifier would operate pixel-wise, on 1 px-regions. However, computation becomes prohibitive on remote sensing images where the shortest size is easiliy counted in thousands of pixels.

Fully convolutional neural networks are one way to tackle this problem. As detailed in Chapter 2, Fully Convolutional Networks (FCNs) are neural networks consisting in convolutional layers designed for dense classification. One forward pass maps every input pixel to one of the classes of interest.

FCNs have several advantages:

- Prediction for a specific pixel automatically leverage the spatial context as far as the network's receptive field allow.
- Input images can now have variable rectangular shapes.
- Dense classification can be done at the same resolution as the input image.

FCNs can be applied on large images in a single pass without pre-segmentation. Deep feature extraction is automatically performed in a dense fashion and optimized jointly with the pixel-wise classification. As a result, the internal representations learnt for semantic segmentation include spectral and color information from the pixel, but also spatial cues from the network's field of view.

Application to semantic mapping

Many FCN architectures have been introduced for semantic segmentation. In this work, we focus on the SegNet model from SegNet Badrinarayanan, Kendall, and Cipolla [6] (cf. Fig. 3.5) as it is balanced between accuracy and compute load[3]. SegNet has a symetrical design using skip connections that replace precisely abstract high-level features at their geometrically-salient localizations based on lower-level features. Preliminary experiments with the seminal FCNs from Long, Shelhamer, and Darrell [41] and DeepLab [16] did not result in significantly better models. Nonetheless, our contributions are not specific to

[3]This includes both the memory required to run the model and the computation time.

SegNet can be be applied to any other architecture. To provide a comparison basis with newer models, we also experiment with the ResNet-34 model from He et al. [27].

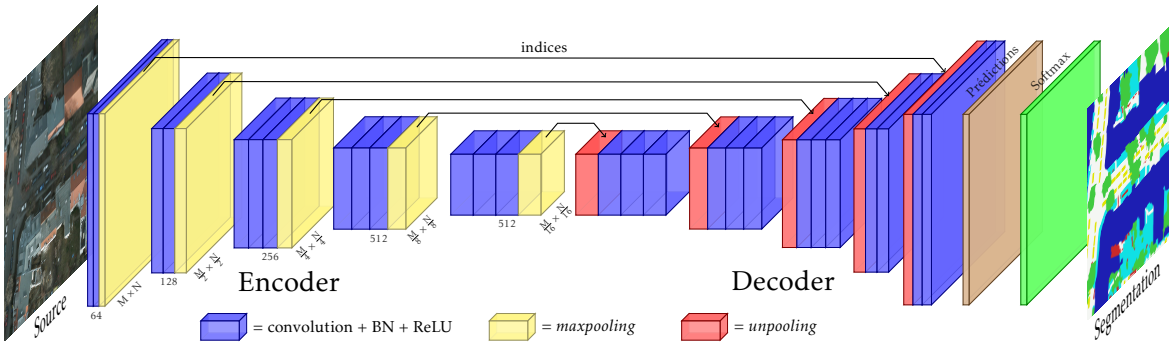


Figure 3.5: Fully convolutional network – SegNet [6].

SegNet is an encoder-decoder architecture built upon the convolutional layers of VGG-16 [15, 57]. The encoder is a sequence of convolutional layers comprised of a 3×3 convolution, BN and a ReLU non-linear activation. Each block contains 2 or 3 of those convolutional layers and is followed by a max-pooling layer on a 2×2 window with a stride of 2. The full architecture is detailed in Fig. 3.5.

The decoder is symetrical to the encoder: it contains the same number of convolutional blocks and the same number of layers. Dimension reduction through max-pooling is replaced by unpooling operations that replace the intermediate activation to the indices (*"argmax"*) computed during pooling. For example, the first max-pooling layer outputs the maximum activation mask and transfers them to the last unpooling layer. The before last activations are replaced to the indices that have been transferred and the rest of the positions are filled with zeros. These sparse unpooled feature maps are densified by the subsequent convolutional layers. To remove any ambiguity in the definition of the unpooling operator, it is necessary that input tensors have even spatial dimensions.

As the encoder is taken from VGG-16 we can initialize it using its pretrained counterpart on ImageNet [18]. The decoder is randomly initialized using He's policy He et al. [26]. The network is optimized end-to-end to minimize the empirical classification error on the whole image, i.e. the average cross-entropy computed for each pixel (i, j) between its label $y^{(i,j)}$ and the activations $z^{(i,j)}$ normalized by a *softmax* layer. Let M and N denote the input image dimensions and k the number of classes, then we search for the weights minimizing:

$$\mathcal{L}(softmax(z), y) = -\frac{1}{\mathrm{M} \times \mathrm{N}} \sum_{i=1}^{\mathrm{M}} \sum_{j=1}^{\mathrm{N}} \sum_{p=1}^{k} y_p^{(i,j)} \log \left(\frac{\exp(z_p^{(i,j)})}{\sum_{q=1}^{k} \exp(z_q^{(i,j)})} \right). \tag{3.4}$$

Although fully convolutional models do not require fixed-size inputs, processing full aerial HR tiles is not feasible due to the large memory requirements it would entail. As a workaround, we process a tile using a sliding window approach.

During training we randomly extract small patches over all available tiles. We apply random flipping or mirroring as a data augmentation strategy to improve the model's generalization capacity.

At test time, we process the high resolution images using a sliding window. As this might induce side effects on along the edges of the window grid, we use a stride smaller than size of the window. This generates an overlap for which several predictions will be made for the same pixel. By averaging these multiple predictions, we can smooth the semantic map and improve the overall accuracy, albeit at the cost of a slight increase in processing time.

3.2.3 Multi-scale analysis

Multi-kernel convolutional layer

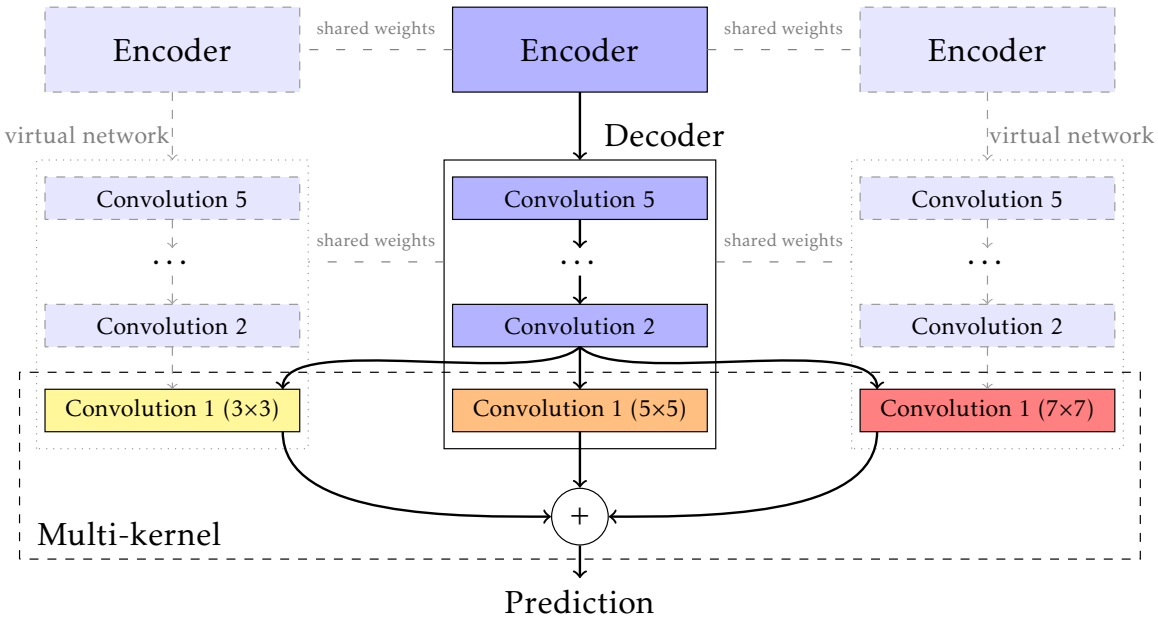


Figure 3.6: Multi-kernel convolutional layer. Replacing the last convolutional layer by a variant with 3 kernels working on several spatial contexts operates the same as averaging 3 models sharing weights.

Multi-scale convolutional layers have been shown useful for object recognition in the Inception model [59] and at multiple times for semantic segmentation [66], including on remote sensing data [67]. We suggest to alter the last convolutional layer from SegNet's decoder to extract features at various spatial context sizes. More specifically, we apply an ensemble of 3×3, 5x5 and 7x7 parallel convolutions instead of the usual single 3×3 kernel. In practice, this is the same as creating an ensemble of three models sharing architecture and weights up to the last layer, as illustrated by Fig. 3.6. Let X_{in} denote the input activations from the multi-kernel convolutional layer, $\mathrm{Z}_p^{(s)}$ the output feature maps for scale s ($s \in [\![1, \mathrm{S}]\!]$ with $\mathrm{S} = 3$ and $p \in [\![1, \mathrm{P}]\!]$ with P the number of filters of the before last convolutional layers, here 64), Z_q^* the final activations ($q \in [\![1, k]\!]$ where k is the number of classes) and $\mathrm{W}_{p,q}^{(s)}$ the q^{th}convolution kernel for the p^{th}activation map at scale s:

$$\mathrm{Z}_q^* = \frac{1}{\mathrm{S}} \sum_{s=1}^{\mathrm{S}} \mathrm{Z}_p^{(s)} = \frac{1}{\mathrm{S}} \sum_{s=1}^{\mathrm{S}} \sum_{p=1}^{\mathrm{P}} \mathrm{W}_{p,q}^{(s)} \mathrm{X}_p \,. \tag{3.5}$$

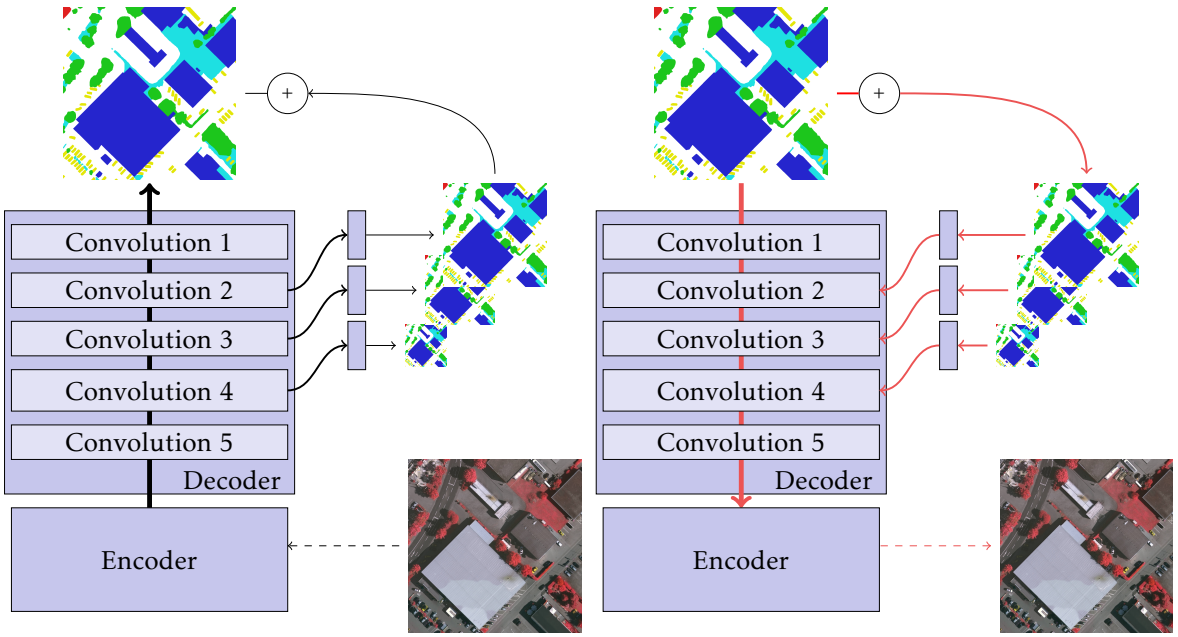
For a pixel located at (i, j) with an activation $z_k^{(s,i,j)}$ for class k and scale s, the cross-entropy after *softmax* is obtained using the following equation:

$$\mathcal{L}(softmax(z), y) = \sum_{l=1}^{k} y_l^{(i,j)} \log \left(\frac{\exp\left(\frac{1}{\mathrm{S}} \sum_{s=1}^{\mathrm{S}} z_l^{(s,i,j)} \right)}{\sum_{l'=1}^{k} \exp\left(\frac{1}{\mathrm{S}} \sum_{s=1}^{\mathrm{S}} z_{l'}^{(s,i,j)} \right)} \right) . \tag{3.6}$$

Although the network can trained end-to-end, it is more practical to add new kernels *a posteriori*. Initially, the network is trained on only one scale. After training, the last convolution kernels can be replaced by smaller or larger ones on which we perform a fine-tuning. The kernels can then be added to the parallel convolution.

This multi-kernel strategy is similar to Inception blocks from Szegedy et al. [59] and to the multiscale convolution from Liao and Carneiro [35]. Yet, in our case we only alter the last convolutional layer in flexible way so that it does not require retraining the whole model. This agregates spatial contexts similarly to dilated convolutions Yu and Koltun [66] to produce multi-scale features. Working on various sizes of local neighbourhoods allow us to avoid introducing a costly dilated convolutional kernel or an image pyramid [67]. It is a straightforward adaptation of an already trained classifier which improves its robustness by integrating multi-scale features at local level.

Deep supervision



(a) Multi-scale inference using SegNet.

(b) Multi-scale backpropagation in SegNet

Figure 3.7: Deeply supervised SegNet at three scale.

Multi-scale processing of remote sensing images most often rely on extracting image pyramids: several contexts at multiple resolutions are used as an input to one or more classifiers. This subsection introduces an alternative approach which processes only one image but generates a pyramid of predictions as the FCN's output, similarly to the DeepLab model [16]. Each output is a downscaled map predicted at a lower resolution on which it will be possible to backpropragate the gradients. This has two benefits; first it performs multi-scale inference which boosts accuracy through ensembling, second it allows a deep supervision of the networ [31].

In the SegNet architecture,the output pyramid naturally appears in the decoder. Each the decoder's p^{th}block, we add a convolutional layer that performs the dense classification at resolution $\frac{2^p \mathrm{M}}{32} \times \frac{2^p \mathrm{N}}{32}$ (where M, N are the image I dimensions), as shown in Fig. 3.7. These maps are interpolated to the full M×N resolution and averaged to produce the final semantic segmentation. Let $\mathrm{P}_{complète}$ denote the full-scale map, $\mathrm{P}_{réduite_d}$ the maps downscaled by a factor $1:d$ and $\mathcal{I}_d$ the bilinear interpolation with a factor d. The relationship tying these tensors is:

$$\mathrm{P}_{complète} = \sum_{d \in \{0,2,4,8\}} \mathcal{I}_d(\mathrm{P}_{réduite_d}) = \mathrm{P}_0 + \mathcal{I}_2(\mathrm{P}_2) + \mathcal{I}_4(\mathrm{P}_4) + \mathcal{I}_8(\mathrm{P}_8). \tag{3.7}$$

During backpropagation, every convolutional block receive two gradients:

- The gradient coming from the backpropagated error at full-scale,

- The gradient coming from the backpropagated error at small-scale.

Deeper layers learn to refine the maps at the smaller scale which simplify the overall network optimization [37].

3.3 Model evaluation

Quantitative metrics are required to compare the relative performances of the segmentation and classification models we introduced. This section details the metrics that we will subsequently use to benchmark various approaches.

3.3.1 Classification metrics

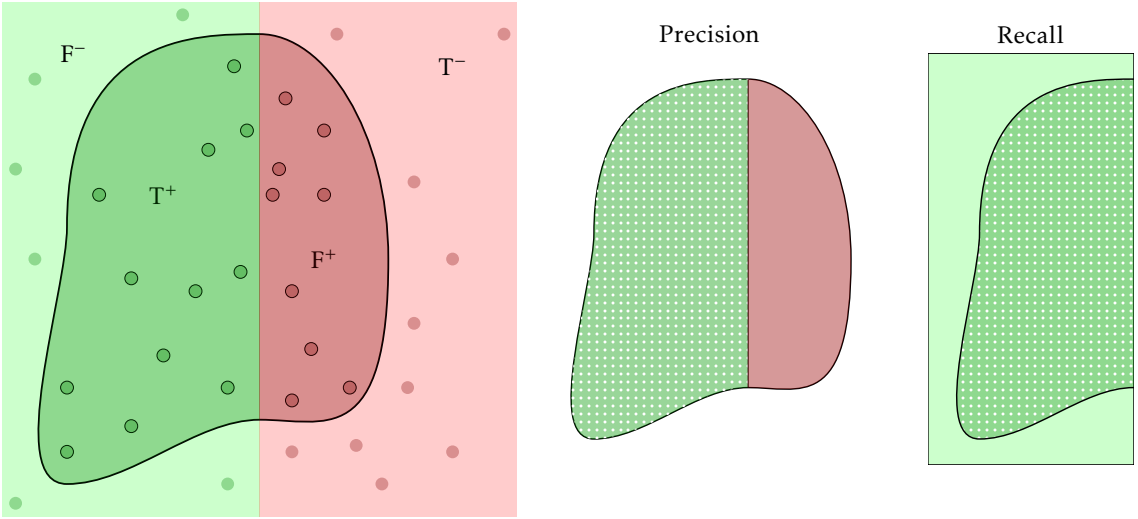


Figure 3.8: Distribution of true positives T^+, true negatives T^-, false positives F^+ and false negatives F^- in a binary classification setting with a 2D space.

For a given classifier and a class of interest i, we define T^+ as the set of true positives (samples belonging to class i that were correctly labeled), T^- the set of true negatives (samples beloging to class $j \neq i$ that were not labeled as i), F^+ the set of false positives (samples belonging to $j \neq i$ labeled as i) and F^- the set of false negatives (samples beloging to i labeled as $j \neq i$). This division is illustrated in Fig. 3.8.

We then define classification metrics for the classifier with respect to the class i:

- Precision is defined as the ratio between true positives and the total number of samples labeled as i by the classifier:
$$precision = \frac{T^+}{T^+ + F^+} .$$
- Recall is defined as the ratio between true positives and the total number of samples actually belonging to i:
$$recall = \frac{T^+}{T^+ + F^-} .$$
- The F_1 score, or Sorensen-Dice coefficient, is defined as the harmonic mean of precision and recall:
$$F_1 = 2 \cdot \frac{precision \times recall}{precision + recall} ,$$
or, differently written,:
$$F_1 = \frac{2T^+}{2T^+ + F^+ + F^-} .$$

- Accuracy is defined as the ratio between right predictions and the total number of samples:
$$exactitude = \frac{T^+ + T^-}{T^+ + F^+ + T^- + F^-} .$$
- L'intersection over union (IoU), or Jaccard's index, is defined as the ratio between right predictions and the set of samples beloging to the class i and samples labeled as i:
$$IsU = \frac{T^+}{T^+ + F^+ + F^-} .$$

The F_1 score and the IsU are not biased towards the main class in an unbalanced setting, contrary to the accuracy. For example, a naive classifier always predicting "background" would be accurate at 95% against a dataset containing 95% background and 5% object. However, its F_1 score would be 0.

Note that although the IsU is similar to the F_1 score, it grants a larger weight to true positives. Yet, both can be used to sort classifier since there is an monotonous relationship between the two: $IsU/F = 1/2 + IsU/2$. If model A is better with respect to the IsU than model B, it will also be the case with respect to the F_1 score and conversely.

Seeing that these metrics are defined in a binary setting, we will look for the overall accuracy and the average intersection over union (or F_1 score on all classes) in a multiclass setting. Additionnal metrics such as Cohen's Kappa can help estimate agreement between predictions and the actual label, even taking chance into account:

$$\kappa = \frac{P(agreement) - P(change)}{1 - P(change)}$$

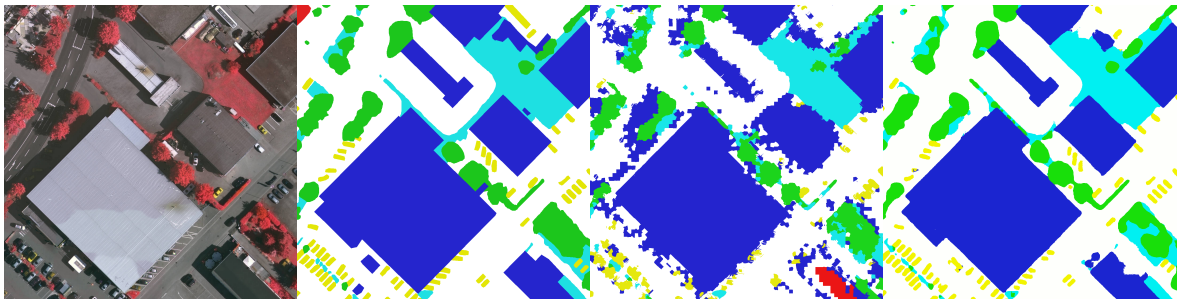
where $P(agremment)$ is the agreement ratio between predictions and actual labels and $P(chance)$ is the probability of a random agreement.

3.3.2 Segmentation metrics

To begin with, we wish to compare theorical capacities of several presegmentation algorithms. Indeed if the segmentation merge in the same region two pixels belonging to different classes, some error will necessarily be injected in the final classication since one region is labeled in one class only.

We rely on four metrics to benchmark the segmentation algorithms:

- Undersegmentation error (UE), defined as the percentage of pixels belonging to a region that overlaps two classes. Let $\mathcal{S}$ and $\mathcal{R}$ denote respectively the generated segmentation and the actual one, and N the total number of pixels in the image:
$$UE = \frac{1}{N} \sum_{R_i \in \mathcal{R}} \sum_{S_j \in \mathcal{S}/S_j \cap R_i \neq \emptyset} \min(|R_i \cap S_j|, |R_i \backslash R_i \cap S_j|)$$
- Border recall (BR), defined as the statistical recall of pixels at regions' borders that are in a 3-neighbourhood of an actual edge:
$$RB = \frac{T^+}{T^+ + F^-}$$
- Average purity (AP), defined as the mean of pixels belonging to the main class of their local cluster. Let maj the operation that maps a region S_i to its majority class:
$$PM = \frac{1}{|\mathcal{S}|} \sum_{S_i \in \mathcal{S}} \frac{|\{p \in S_i \text{ et } classe(p) = maj(S_i)\}|}{|S_i|}$$
- The oracle, defined as the pixel-wise overall accuracy achievable by a perfect classifier that would map every region to its majority class. This it the upper-bound of the accuracy that a classifier can achieve using the segmentation.



(a) IRRG image (b) Ground truth (c) SLIC+AlexNet+SVM (d) SegNet

Figure 3.9: Semantic maps inferred by region-based classification and a FCN. SegNet outputs dense predictions that are significantly more accurate and detailed that the result of a superpixel prediction and deep features region-based classification.
Colors: white: roads, blue: buildings, cyan: low vegetation, green: trees, yellow: vehicles, red: clutter.

3.3.3 Region-based classification

We choose to compare several unsupervised segmentation algorithms in a region-based classification setting on the ISPRS 2D *Semantic Labeling* Vaihingen dataset. This dataset is a EHR aerial acquisition of a medium-sized German town using IRRG channels. It has been labeled for 6 classes. More details regarding the dataset are given in Appendix A.1.1.

We evaluate the five segmentation algorithms commonly used in computer vision and remote sensing from Section 3.1.2: SLIC, LSC, Quickshift, MRS et HSeg. Each method's parameters are tuned to obtain roughly the same number of regions and the best results. These algorithms are representative from what is generally used in the literature.

Table 3.1: Benchmark of five segmentation algorithms on the ISPRS Vaihingen dataset. Best results are in **bold**, second best are in *italics*.

Algorithm	# regions	UE (%)	BR (%)	AP (%)	Oracle (%)
SLIC	≃**20 000**	**10.21**	84.07	75.10	*89.91*
LSC	≃22 800	*11.37*	91.13	71.54	85.83
Quickshift	≃*21 000*	11.66	88.34	72.90	83.61
MRS	≃23 500	13.12	**95.71**	**79.08**	**91.68**
HSeg	≃*21 000*	11.39	*94.83*	*78.66*	85.25

Table 3.2: Semantic segmentation metrics on the ISPRS Vaihingen validation set. Best results are in **bold**, second best are in *italics*.

Algorithm	# regions	Accuracy (%)	F_1 score (cars)	κ	Oracle (%)
SLIC	≃**20 000**	*82.20*	0.54	**0.76**	*89.91*
LSC	≃22 800	**82.45**	**0.58**	**0.76**	85.53
Quickshift	≃*21 000*	82.05	0.52	*0.75*	83.61
MRS	≃23 500	80.53	*0.56*	0.73	**91.68**
HSeg	≃*21 000*	79.56	0.54	0.72	85.25
Sliding window	≃23 800	81.22	0.53	0.74	92.56

We apply these segmentation algorithms on all tiles from the ISPRS Vaihingen dataset. We use the authors' implementation for LSC [34], the implementation from Guyet, Malinowski, and Benyounès [25] for MRS (adapted from the TerraLib [13] library) and the implementations from scikit-image [62] for SLIC and *Quickshift*. Results are reported in Table 3.1.

Best results for pure segmentation metrics are obtained by the remote sensing image segmentation algorithms. MRS and HSeg exhibit high border rappel and average purity, which indicates that region edges are close to the actual semantic borders from the ground truth. This is unsurprising since the similarity criterion used by both segmentations have been tuned for remote sensing data. However this segmentation comes at a price: regions are irregular which increases the undersegmentation error. In comparison, superpixel algorithms slightly underperform since their regions are more rigidly constrained, which results in a lower average purity and lower border recall. Yet, the multiplication of smaller regions uniformly distributed in the image decreases the undersegmentation error. Overall, the best theorical classification accuracy (oracle) vary from 83% to 91%. MRS and SLIC seem to be ahead of the pack based on this metric.

Region-based classification results are reported in Table 3.2. The AlexNet CNN for feature extraction is implemented using the Caffe deep learning library [28]. We use a linear SVM as a classifier, optimized by gradient descent as per the implementation from scikit-learn [50]. As the table shows, sorting by overall accuracy does not give the same ranking as the oracle; raw segmentation metrics do not suffice to compare segmentation algorithms for feature extraction.

Indeed, high classification results indicate that superpixel segmentations should be preferred. The classifier benefits from a stronger geometric regularity due to high compacity and strong convexity. When we extract the patch around a region, most pixels at the center of the image are relevant to this specific region. This means that the CNN will infer a richer feature vector. On the contrary, irregular segmentations are harder to integrate in this pipeline, since rectangular patches rarely overlap well with the regions. Discriminating between non-normalized samples becomes harder for the classifier, as shown in Fig. 3.3. In practice, these segmentations do not improve the overall accuracy compared to a simple sliding window with a fixed computational cost. An example of semantic map obtained by SLIC and deep features is given in Fig. 3.9. Large areas such as roads, buildings and vegetation are well-segmented, although borders are blurry and performance on vehicles is subpar.

One way to improve the accuracy when using MRS is to increase its compacity parameter, which results in regions similar to superpixels. Accuracy becomes on-par with SLIC altough the number of regions doubles: MRS needs twice as more segments than SLIC to reach the same accuracy. This impacts directly the computation since there are twice as many regions to process. Finally, we stress that segmentating small objects can be challenging and is sensitive to the choice of the presegmentation. F_1 score on cars can be significantly improved by using a suitable algorithm such as LSC.

3.3.4 Pixel-wise classification using fully convolutional networks

As we have seen, the unsupervised segmentation pre-processing limits the overall accuracy achievable in a region-based classification setting. Not only does an imperfect segmentation introduce errors that even the oracle can not recover, the regions shapes and sizes is rarely suited for deep feature extraction. Investigating FCNs that can learn both segmentation and classification in an end-to-end fashion is then quite promising.

We train SegNet and ResNet-34 on the ISPRS Vaihingen and ISPRS Potsdam datasets. Each tile is processed using a 128×128 sliding window with a variabel stride. Both models are trained during 50 000 iterations with a batch size of 10. The initial learning rate is set at 0.1 and divided by 10 after 350 000 and 450 000 iterations. The neural networks are implemented using the Caffe [28] and PyTorch [52] libraries.

At first, we validate our approach only on the tiles for which the ground truth is publicly available that we split in a training set and a validation set. To compare our methods with the state of the art, we then train on the whole dataset (train + validation) with the same hyper-parameters and we submit our results on the private test set to the ISPRS evaluation server,

which keeps the test labels hidden. As the Fig. 3.9 showed, dense pixel-wise predictions produce visually promising results.

Sliding window and overlap

Table 3.3: Semantic segmentation results on the ISPRS Vaihingen validation set with various sliding window strides.

Network/stride (px)	128	64	32
SegNet IRRG	87.8%	88.3%	88.8%
SegNet multi-kernel	88.2%	88.6%	89.1%

The use of a sliding window during inference forces to study how to deal with borders. Indeed if the stride of the sliding window is equal the window's size discontinuities along the edges can appear which will decrease the overall accuracy and be visually unpleasing. By reducing the stride, we can allow a partial overlap between two consecutive windows, i.e. that pixels along the borders will be observed twice or more. This increases the inference time but generally improves the overall accuracy as reported in Table 3.3. Dividing the stride by 2 multiply by 4 the windows to process. However averaging multiples predictions on the same area corrects classification artifacts along the edges where spatial context is missing (and filled by padding). Experiments show that a stride of 32 px (75% overlap) remains fast enough for offline processing and significantly boosts the overall accuracy (+1%). A full tile is processed in 4 minutes on a NVIDIA Tesla K20c with a stride of 32 px and less than 20 s with a stride of 128 px. We will the former parameters for the rest of this manuscript.

Overall, SegNet classifies correctly more than 87% of the pixels on the validation set. In comparison not one of the region-based classification techniques investigated in the previous section went further than 83%. Moreover SegNet overperforms the oracles for the HSeg, LSC and Quickshift segmentations. This proves that fully convolutional networks are definitely relevant for semantic segmentation. As they are trained on dense pixel-wise segmentation, SegNet models learn to jointly extract features and perform classification by taking the spatial context into account with no resolution loss.

Transfer learning

Table 3.4: Various initialization results on the ISPRS Vaihingen dataset.

Initialization	Random	VGG-16 (ImageNet)			
Encoder variability $\frac{\alpha_e}{\alpha_d}$	1	1	0.5	0.1	0
Accuracy	87.0%	87.2%	**87.8%**	86.9%	86.5%

Pretraining a deep network on a generic dataset is a commonly used technique to improve its generalization capacity. ImageNet is often used to pretrain networks for many visual tasks, including in remote sensing [51]. Nonetheless one could argue that features learnt on cats and dogs from ImageNet might not transfer very well for buildings and trees seen from above and that it is necessary to retrain the convolutional filters to learn the specificities of the data. We investigate these claims by training several SegNet models using various strategies, especially various learning rates for the encoder (α_e) compared to the decoder (α_d):

- same learning rate : $\alpha_d = \alpha_e$,

- reduced encoder variability: $\alpha_d = 2 \cdot \alpha_e$,
- low encoder variability: $\alpha_d = 10 \cdot \alpha_e$,
- frozen encoder (no gradient backpropagation): $\alpha_e = 0$.

We compare the results of these networks with a reference baseline obtained by a random initialization [26], i.e. SegNet trained from scratch without pretraining or transfer learning.

As reported in Table 3.4, the best accuracy is obtained with an encoder initialized by the pretrained ImageNet weights and optimized with a small learning rate. This strengthens the idea that the ideal is between the two paradigms: pretrained weights act as a powerful initialization but they need to be learnable to leverage specific knowledge. Letting the convolutional filters loose might on the opposite encourage overfitting and tweaking the encoder learning rate acte as a regularization. These findings are aligned with the previous works from Nogueira, Penatti, and dos Santos [48] and the broader conclusions of Yosinski et al. [65] regarding transfer learning. In the following, we will use the initialization from the pretrained VGG-16 when possible.

Choosing a model

Table 3.5: Semantic segmentation results on the ISPRS Vaihingen validation set.

Network	Roads	Buildings	Low veg.	Trees	Cars	Accuracy
SegNet	92.2± 2.1	95.6± 0.8	**82.6**± 4.2	**88.1**± 2.5	**88.2**± 0.6	90.2± 1.4
ResNet-34	**93.0**± 1.7	**96.0**± 0.6	82.3± 2.6	87.0± 3.7	87.0± 2.0	**90.3**± 1.0

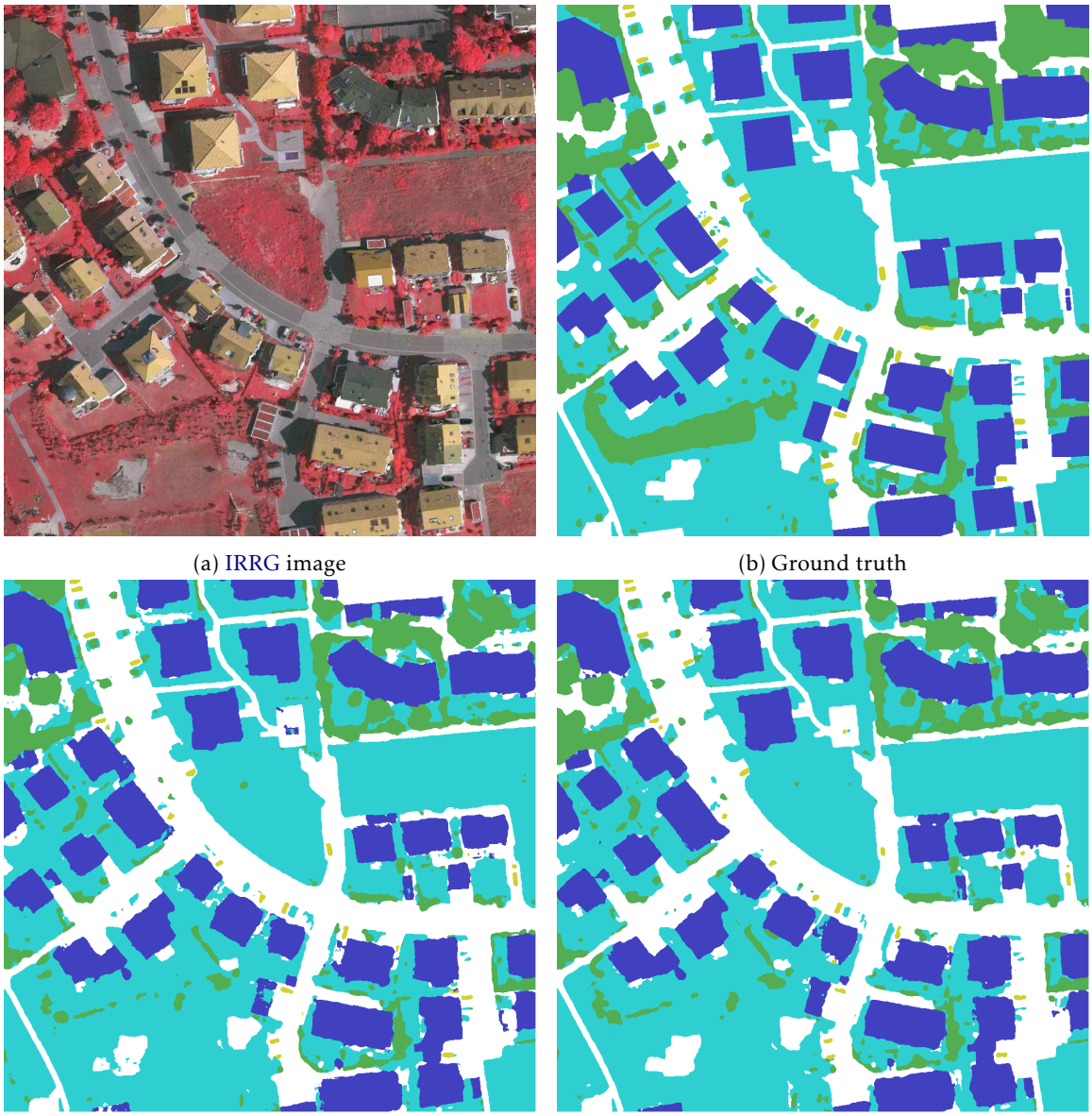
3-fold cross-validation results on the ISPRS Vaihingen validation set are reported in Table 3.5. Switching from SegNet to ResNet-34 gives a slight accuracy improvement of 0.1% and a more robust model with lower standard deviation. However, ResNet-34 requires 25% more GPU memory compared to SegNet which is not justified by the accuracy boost. Moreover ResNet-34 underperforms SegNet on small objects (i.e. vehicles) for which SegNet can leverage the unpooling layers for precise relocation. We assume that deeper ResNet models such as ResNet-101 could extract richer features than ResNet-34 and VGG-16, yet it would also involve a lot more computation than SegNet. Therefore we stick to the latter for most our experiments.

Impact of the multi-scale strategies

IRRG | Prediction | Prediction (multi) | Ground truth | IRRG | Prediction | Prediction (multi) | Ground truth

Figure 3.10: Impact of the multi-kernel convolutional layer on the ISPRS Vaihingen dataset. Colors: white: roads, blue: buildings, cyan: low vegetation, green: trees, yellow: vehicles, red: clutter.

Multi-kernel convolution As reported in Table 3.3, using a multi-kernel convolution as the last layer improves the overall accuracy by 0.4%. This improvement comes from a smoothing of the predicted maps that make salt-and-pepper classification noise disappear. This phenomenon is illustrated in Fig. 3.10. Brahimi et al. [12] obtained similar improvements by introducing multi-kernel convolutions in a DenseNet model for semantic segmentation in an autonomous driving context.



(a) IRRG image (b) Ground truth

(c) SegNet (d) SegNet multi-scale (3 branches)

Figure 3.11: Impact of the deep multi-scale supervision on the ISPRS Vaihingen dataset. Small objects and spectrally ambiguous surfaces benefit from the multi-scale combination.
Colors: white: roads, blue: buildings, cyan: low vegetation, green: trees, yellow: vehicles, red: clutter.

Deep supervision Table 3.6 show the small positive effect of the deep multi-scale supervision on SegNet and the associated metrics with a computational overhead nearly absent. As expected, large structures benefit the most from the downscaled predictions while cars – the smallest objects from the dataset – are harder to detect at lower resolution. Moreover, the absence of structure in vegetation seems to confuse the boundary between trees and low vegetation at the lower scales. Adding more branches after 3 only marginaly improve the model performances. This hints that deep supervision has a limited role compared to the multi-scale effect.

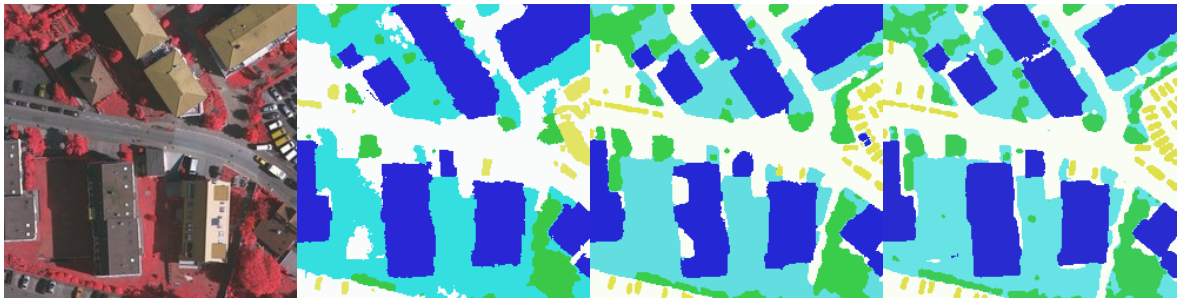
Despite the relatively low quantitative improvement, visual inspection of the predicted maps show a significant qualitative improvement. Multi-scale predictions from the Fig. 3.11 exhibit a lower noise and are generally smoother than the usual SegNet. This simplifies the maps post-processing, either their interpretation by an expert or an automated vectorization. These results are coherent with the later findings of Jiang et al. [29] for semantic segmentation of Red-Green-Blue + Depth (RGB-D) images.

Table 3.6: Semantic segmentation results of the multi-scale approach on the ISPRS Vaihingen validation set.

# branches	Roads	Buildings	Low veg.	Trees	Cars	Accuracy
No branch	92.2	95.5	**82.6**	**88.1**	88.2	90.2± 1.4
1 branch	92.4	95.7	82.3	87.9	**88.5**	90.3± 1.5
2 branches	92.5	**95.8**	82.4	87.8	87.6	90.3± 1.4
3 branches	**92.7**	**95.8**	**82.6**	**88.1**	88.1	**90.5**± 1.5

Finally, this study showed that intermediate feature maps from the decoder are nearly as accurate as the full scale final predictions. For example, the map computed by the 2nd convolutional block, i.e. with a downscale factor of 8, is only 0.5% less accurate than the full resolution one. The main differences are due to the "vehicles" class, which is understandable since cars cover approximately 30 px at 9 cm/px, i.e. 3-4 px at 1 : 8 resolution. Yet it seems that for larger classes of interest, it is feasible to keep the accuracy nearly the same while dropping the number of parameters and the computation time of SegNet by 30% only by stopping the inference earlier.

Final results



(a) IRRG image (b) RF + CRF [53] (c) "DLR" (FCN) [43] (d) **SegNet++**

Figure 3.12: Various segmentations on a sample from the ISPRS Vaihingen test set.
Colors: white: roads, blue: buildings, cyan: low vegetation, green: trees, yellow: vehicles, red: clutter.

IRRG image Ground truth Prediction IRRG image Ground truth Prediction

(a) Predicted maps are, on occasion, more accurate than the ground truth.

(b) SegNet might overfit on geometrical deformations due to orthorectification errors.

Figure 3.13: Edge cases of disagreements between SegNet and the ground truth.
Colors: white: roads, blue: buildings, cyan: low vegetation, green: trees, yellow: vehicles, red: clutter.

Our best model improves the state of the art on the ISPRS Vaihingen dataset (cf. Table 3.7) [4]. Fig. 3.12 gives a qualitative comparison between various methods. Metrics are computed after a 3-pixel erosion along the borders to take uncertainties during the labeling process into account. At the time of our submission, the best published method used a combination of a FCN and expert features, while ours does not rely on expert knowledge. The previous

[4] http://www2.isprs.org/commissions/comm2/wg4/vaihingen-2d-semantic-labeling-contest.html

Table 3.7: Results from the ISPRS 2D *Semantic Labeling Challenge* Vaihingen (chronological order).

Method	Roads	Buildings	Low veg.	Trees	Cars	Accuracy
Stair Vision Library ("SVL_3") [20]	86.6	91.0	77.0	85.0	55.6	84.8
RF + CRF ("HUST") [53]	86.9	92.0	78.3	86.9	29.0	85.9
Ensemble de CNN ("ONE_5") [11]	87.8	92.0	77.8	86.2	50.7	85.9
FCN ("UZ_1") [64]	89.2	92.5	81.6	86.9	57.3	87.3
FCN ("UOA") [36]	89.8	92.1	80.4	88.2	82.0	87.6
CNN + nDSM + RF + CRF ("ADL_3") [49]	89.5	93.2	82.3	88.2	63.3	88.0
FCN ("DLR_2") [43]	90.3	92.3	82.5	89.5	76.3	88.5
FCN + RF + CRF ("DST_2") [55]	90.5	93.7	83.4	89.2	72.6	89.1
SegNet++ (multi-kernel) [4]	91.5	94.3	82.7	89.3	85.7	89.4
FCN + CRF + edges + corrected nDSM ("DLR_9") [44]	92.4	95.2	83.9	89.9	81.2	90.3
ResNet-101 ("CASIA_2") [40]	93.2	96.0	84.7	89.9	86.7	91.1

Table 3.8: Results from the ISPRS 2D *Semantic Labeling Challenge* Potsdam (chronological order).

Method	Roads	Buildings	Low veg.	Trees	Cars	Accuracy
SVL [20]	83.5	91.7	72.2	63.2	62.2	77.8
FCN [55]	92.5	96.4	86.7	88.0	94.7	90.3
FCN + CRF + expert features [39]	91.2	94.6	85.1	85.1	92.8	88.4
FCN + CRF [64]	89.3	95.4	81.8	80.5	86.5	85.8
SegNet (IRRG)	92.4	95.8	86.7	87.4	95.1	90.0
ResNet-101 [40]	93.3	97.0	87.7	88.4	96.2	91.1

best model using only a FCN ("DLR_1") reaches 88.4% accuracy, which we surpass by 1%. Previous methods based on CNNs obtain 85.9% ("ONE_5"[11]) and 86.1% ("ADL_1"[49]). Our approach obtains better results without relying on expert features or post-processing such as CRF.

On the ISPRS Potsdam dataset (cf. Table 3.8)[5], our approach is competitive with the state of the art at the time of our submission. Especially we improve the best results for models based on the optical data only by 0.3% compared to a standard FCN Sherrah [55] and by 4.2% with respect to the FCN from Volpi and Tuia [64]. An example image is given in Fig. 3.14.

It is interesting to see that some models reach accuracies sufficiently high to saturate on errors that can be attributed to uncertainties in the labeling process. In Fig. 3.13a, we give an example where the ground truth coarsely circles the tree while the model sticks very closely to its actual edges. Moreover the orthorectification pre-processing of the image mosaic introduces some geometrical distorsions which are not taken into account – with reason – in the ground truth. Yet the model still picks up the deformation, which generates a disagreement between spectral values and labels as shown in Fig. 3.13b. This shows that it becomes harder and harder to significantly improve the results on these datasets since FCNs are close from what could be reasonably expected by the challenges' organizers. For this reason, the competition and the evaluation server have been closed in July, 2018.

To conclude, we showed that FCNs are particularly effective for semantic segmentation

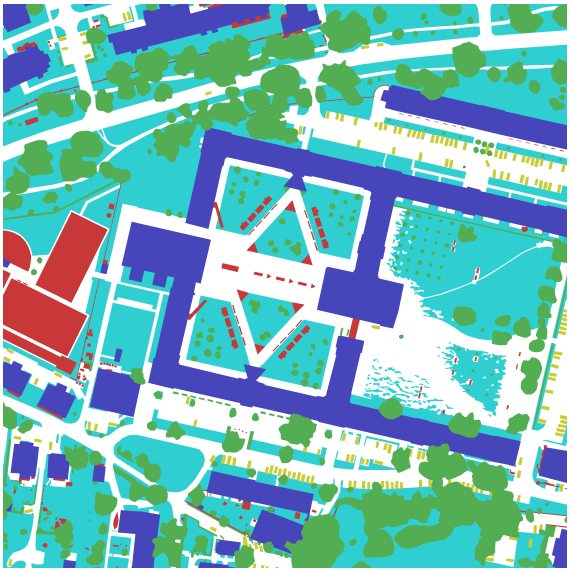
[5] http://www2.isprs.org/commissions/comm2/wg4/potsdam-2d-semantic-labeling.html

of aerial images. Specifically, we improved the state of the art on the ISPRS datasets using the SegNet fully convolutional network which outperformed significantly previous region-based classification approaches. We introduced several guidelines to initialize and pretrain such networks and to deal with large images using sliding windows. We proposed two segmentation techniques to learn from multiple scales and multiple contexts to boost further the maps' accuracies. However these achievements still are limited to aerial images comprised of 3 channels, either IRRG or RGB at very high resolution. The next chapter will therefore look into extending these results on other sensors commonly deployed for Earth Observation.

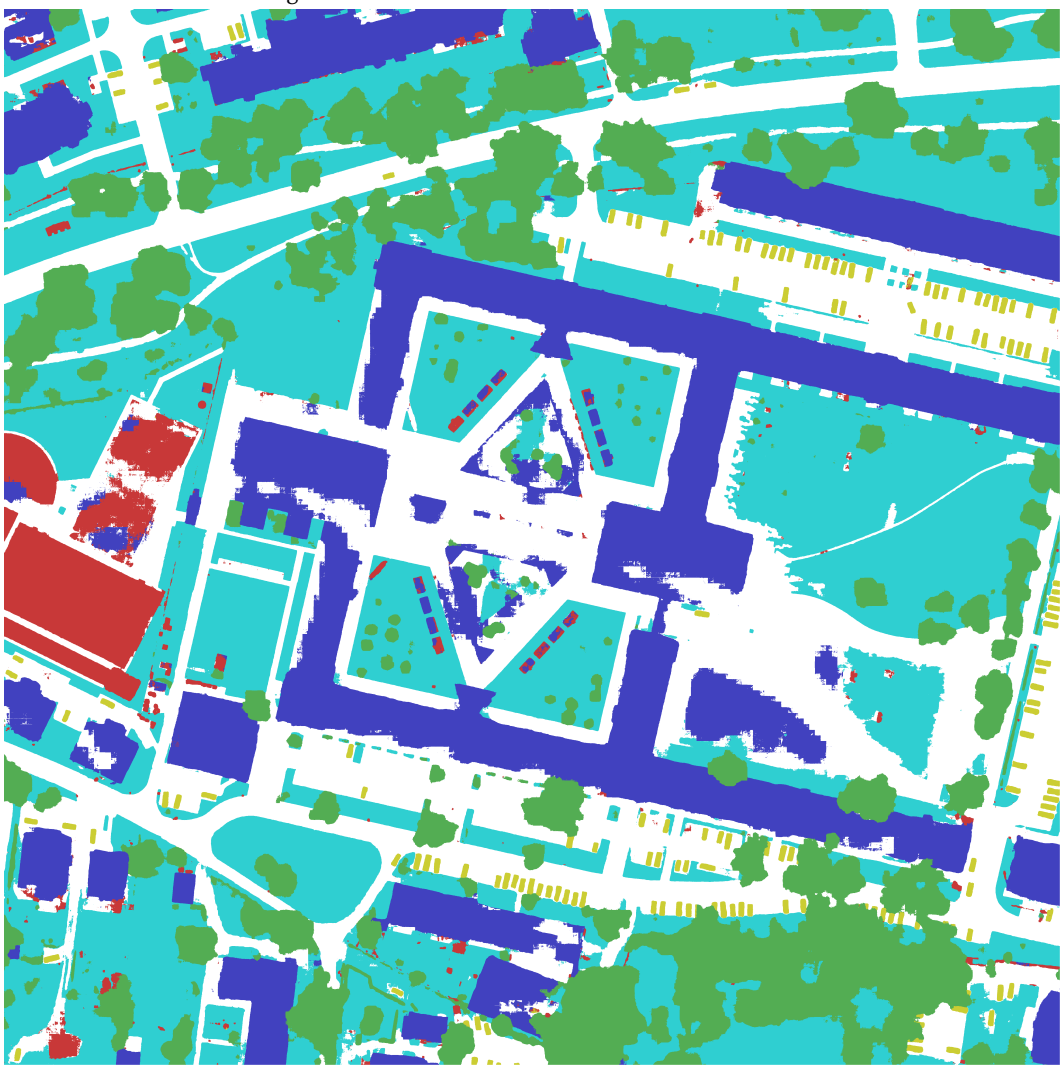
The works presented in this chapter were the topic of two conference publications:

- Nicolas Audebert, Bertrand Le Saux, and Sébastien Lefèvre. "How Useful Is Region-Based Classification of Remote Sensing Images in a Deep Learning Framework?" In: *2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. July 2016, pp. 5091–5094. DOI: 10.1109/IGARSS.2016.7730327
- Nicolas Audebert, Bertrand Le Saux, and Sébastien Lefèvre. "Semantic Segmentation of Earth Observation Data Using Multimodal and Multi-Scale Deep Networks". In: *Computer Vision – ACCV 2016*. Springer, Cham, Nov. 20, 2016, pp. 180–196. DOI: 10.1007/978-3-319-54181-5_12

RGB image



Ground truth



SegNet prediction

Figure 3.14: Semantic map predicted by SegNet on tile 3_11 from the ISPRS Potsdam dataset. Colors: white: roads, blue: buildings, cyan: low vegetation, green: trees, yellow: vehicles, red: clutter.

References

[1] Radhakrishna Achanta et al. *SLIC Superpixels*. 2010. URL: http://infoscience.epfl.ch/record/149300 (cit. on p. 59).

[2] Radhakrishna Achanta et al. "SLIC Superpixels Compared to State-of-the-Art Superpixel Methods". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34.11 (Nov. 2012), pp. 2274–2282. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2012.120 (cit. on pp. 61, 62).

[3] Nicolas Audebert, Bertrand Le Saux, and Sébastien Lefèvre. "How Useful Is Region-Based Classification of Remote Sensing Images in a Deep Learning Framework?" In: *2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. July 2016, pp. 5091–5094. DOI: 10.1109/IGARSS.2016.7730327 (cit. on p. 78).

[4] Nicolas Audebert, Bertrand Le Saux, and Sébastien Lefèvre. "Semantic Segmentation of Earth Observation Data Using Multimodal and Multi-Scale Deep Networks". In: *Computer Vision – ACCV 2016*. Springer, Cham, Nov. 20, 2016, pp. 180–196. DOI: 10.1007/978-3-319-54181-5_12 (cit. on pp. 77, 78).

[5] Martin Baatz and Arno Schäpe. "Multiresolution Segmentation: An Optimization Approach for High Quality Multi-Scale Image Segmentation". In: *Angewandte Geographische Informationsverarbeitung XII: Beiträge zum AGIT-Symposium Salzburg* (2000), pp. 12–23 (cit. on p. 61).

[6] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Scene Segmentation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.12 (Dec. 2017), pp. 2481–2495. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2016.2644615 (cit. on pp. 65, 66).

[7] Yoshua Bengio, Aaron Courville, and Pascal Vincent. "Representation Learning: A Review and New Perspectives". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.8 (Aug. 2013), pp. 1798–1828. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2013.50 (cit. on p. 64).

[8] Serge Beucher and Fernand Meyer. "The Morphological Approach to Segmentation: The Watershed Transformation. Mathematical Morphology in Image Processing." In: *Optical Engineering* 34 (1993), pp. 433–481 (cit. on p. 60).

[9] Petra Bosilj. "Indexation et recherche d'images par arbres des coupes". PhD thesis. Université de Bretagne Sud, Jan. 25, 2016. URL: https://tel.archives-ouvertes.fr/tel-01362165/document (cit. on p. 59).

[10] Léon Bottou. "Large-Scale Machine Learning with Stochastic Gradient Descent". In: *In COMPSTAT*. 2010 (cit. on p. 64).

[11] Alexandre Boulch. *DAG of Convolutional Networks for Semantic Labeling*. Office national d'études et de recherches aérospatiales, 2015. URL: https://www.itc.nl/external/ISPRS_WGIII4/ISPRSIII_4_Test_results/papers/onera_boulch.pdf (cit. on p. 77).

[12] Sourour Brahimi et al. "Multiscale Fully Convolutional DenseNet for Semantic Segmentation". In: *WSCG 2018, International Conference on Computer Graphics, Visualization and Computer Vision*. Pilsen, Czech Republic, May 2018. URL: https://hal.archives-ouvertes.fr/hal-01786688 (cit. on p. 74).

[13] Gilberto Câmara et al. "TerraLib: An Open Source GIS Library for Large-Scale Environmental and Socio-Economic Applications". In: *Open Source Approaches in Spatial Data Handling*. Advances in Geographic Information Science. Springer, Berlin, Heidelberg, 2008, pp. 247–270. ISBN: 978-3-540-74830-4 978-3-540-74831-1. DOI: 10.1007/978-3-540-74831-1_12. URL: https://link.springer.com/chapter/10.1007/978-3-540-74831-1_12 (cit. on p. 71).

[14] Tony Chan and Luminita Vese. "An Active Contour Model without Edges". In: *Scale-Space Theories in Computer Vision*. International Conference on Scale-Space Theories in Computer Vision. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, Sept. 26, 1999, pp. 141–151. ISBN: 978-3-540-66498-7 978-3-540-48236-9. DOI: 10.1007/3-540-48236-9_13. URL: https://link.springer.com/chapter/10.1007/3-540-48236-9_13 (cit. on pp. 60, 61).

[15] Ken Chatfield et al. "Return of the Devil in the Details: Delving Deep into Convolutional Nets". In: *Proceedings of the British Machine Vision Conference*. British Machine Vision Conference (BMVC). British Machine Vision Association, 2014, pp. 6.1–6.12. ISBN: 978-1-901725-52-0. DOI: 10.5244/C.28.6. URL: http://www.bmva.org/bmvc/2014/papers/paper054/index.html (cit. on p. 66).

[16] Liang-Chieh Chen et al. "DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.4 (Apr. 2018), pp. 834–848. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2017.2699184 (cit. on pp. 65, 68).

[17] Dorin Comaniciu and Peter Meer. "Mean Shift: A Robust Approach toward Feature Space Analysis". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24.5 (May 2002), pp. 603–619. ISSN: 0162-8828. DOI: 10.1109/34.1000236 (cit. on p. 60).

[18] Jia Deng et al. "ImageNet: A Large-Scale Hierarchical Image Database". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2009, pp. 248–255. DOI: 10.1109/CVPR.2009.5206848 (cit. on pp. 64, 66).

[19] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. "Efficient Graph-Based Image Segmentation". In: *International Journal of Computer Vision* 59.2 (Sept. 2004), pp. 167–181. ISSN: 0920-5691, 1573-1405. DOI: 10.1023/B:VISI.0000022288.19776.77 (cit. on p. 59).

[20] Markus Gerke. *Use of the Stair Vision Library within the ISPRS 2D Semantic Labeling Benchmark (Vaihingen)*. International Institute for Geo-Information Science and Earth Observation, 2015. URL: https://www.researchgate.net/profile/Markus_Gerke/publication/270104226_Use_of_the_Stair_Vision_Library_within_the_ISPRS_2D_Semantic_Labeling_Benchmark_(Vaihingen)/links/54ae59c50cf2828b29fcdf4b.pdf (cit. on p. 77).

[21] Rémi Giraud, Vinh-Thong Ta, and Nicolas Papadakis. "Robust Superpixels Using Color and Contour Features along Linear Path". In: *Computer Vision and Image Understanding* (Jan. 31, 2018). ISSN: 1077-3142. DOI: 10.1016/j.cviu.2018.01.006. URL: http://www.sciencedirect.com/science/article/pii/S1077314218300067 (cit. on p. 60).

[22] Maoguo Gong et al. "Superpixel-Based Difference Representation Learning for Change Detection in Multispectral Remote Sensing Images". In: *IEEE Transactions on Geoscience and Remote Sensing* 55.5 (May 2017), pp. 2658–2673. ISSN: 0196-2892. DOI: 10.1109/TGRS.2017.2650198 (cit. on p. 59).

[23] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. URL: http://www.deeplearningbook.org (cit. on p. 64).

[24] Leo Grady. "Random Walks for Image Segmentation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28.11 (2006), pp. 1768–1783 (cit. on p. 59).

[25] Thomas Guyet, Simon Malinowski, and Mohand Cherif Benyounès. "Extraction des zones cohérentes par l'analyse spatio-temporelle d'images de télédétection". In: *Revue Internationale de Géomatique* 25.4 (2015), pp. 473–494. ISSN: 1260-5875, 2116-7060. DOI: 10.3166/RIG.25.473-494. URL: https://rig.revuesonline.com/articles/1vrig/abs/2015/04/1vrig254p473/1vrig254p473.html (cit. on p. 71).

[26] Kaiming He et al. "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification". In: *Proceedings of the IEEE International Conference on Computer Vision*. IEEE International Conference on Computer Vision (ICCV). Dec. 2015, pp. 1026–1034. DOI: 10.1109/ICCV.2015.123 (cit. on pp. 66, 74).

[27] Kaiming He et al. "Deep Residual Learning for Image Recognition". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Las Vegas, United States, June 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90 (cit. on p. 66).

[28] Yangqing Jia et al. "Caffe: Convolutional Architecture for Fast Feature Embedding". In: *Proceedings of the 22Nd ACM International Conference on Multimedia*. MM '14. New York, NY, USA: ACM, 2014, pp. 675–678. ISBN: 978-1-4503-3063-3. DOI: 10.1145/2647868.2654889. URL: http://doi.acm.org/10.1145/2647868.2654889 (cit. on p. 72).

[29] Jindong Jiang et al. "RedNet: Residual Encoder-Decoder Network for Indoor RGB-D Semantic Segmentation". In: (June 4, 2018). arXiv: 1806.01054 [cs]. URL: http://arxiv.org/abs/1806.01054 (cit. on p. 75).

[30] Adrien Lagrange et al. "Benchmarking Classification of Earth-Observation Data: From Learning Explicit Features to Convolutional Networks". In: *2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. 2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS). July 2015, pp. 4173–4176. DOI: 10.1109/IGARSS.2015.7326745 (cit. on p. 64).

[31] Chen-Yu Lee et al. "Deeply-Supervised Nets". In: *Artificial Intelligence and Statistics*. Artificial Intelligence and Statistics. Feb. 21, 2015, pp. 562–570. URL: http://proceedings.mlr.press/v38/lee15a.html (cit. on p. 68).

[32] Alex Levinshtein et al. "TurboPixels: Fast Superpixels Using Geometric Flows". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31.12 (Dec. 2009), pp. 2290–2297. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2009.96 (cit. on p. 60).

[33] Hongguang Li et al. "Superpixel-Based Feature for Aerial Image Scene Recognition". In: *Sensors* 18.1 (Jan. 8, 2018), p. 156. DOI: 10.3390/s18010156. URL: http://www.mdpi.com/1424-8220/18/1/156 (cit. on p. 59).

[34] Zhengqin Li and Jiansheng Chen. "Superpixel Segmentation Using Linear Spectral Clustering". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 1356–1363. DOI: 10.1109/CVPR.2015.7298741. URL: http://www.cv-foundation.org/openaccess/content_cvpr_2015/html/Li_Superpixel_Segmentation_Using_2015_CVPR_paper.html (cit. on pp. 59, 71).

[35] Zhibin Liao and Gustavo Carneiro. "Competitive Multi-Scale Convolution". In: (Nov. 17, 2015). arXiv: 1511.05635 [cs]. URL: http://arxiv.org/abs/1511.05635 (cit. on p. 68).

[36] Guosheng Lin et al. "Efficient Piecewise Training of Deep Structured Models for Semantic Segmentation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Las Vegas, United States, 2016, pp. 3194–3203. DOI: 10.1109/CVPR.2016.348. URL: http://arxiv.org/abs/1504.01013 (cit. on p. 77).

[37] Guosheng Lin et al. "RefineNet: Multi-Path Refinement Networks with Identity Mappings for High-Resolution Semantic Segmentation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. July 2017, pp. 5168–5177. DOI: 10.1109/CVPR.2017.549 (cit. on p. 69).

[38] Ming-Yu Liu et al. "Entropy Rate Superpixel Segmentation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. CVPR 2011. June 2011, pp. 2097–2104. DOI: 10.1109/CVPR.2011.5995323 (cit. on p. 59).

[39] Yansong Liu et al. "Dense Semantic Labeling of Very-High-Resolution Aerial Imagery and LiDAR with Fully-Convolutional Neural Networks and Higher-Order CRFs". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. Honolulu, United States, July 2017, pp. 1561–1570. DOI: 10.1109/CVPRW.2017.200 (cit. on p. 77).

[40] Yongcheng Liu et al. "Semantic Labeling in Very High Resolution Images via a Self-Cascaded Convolutional Neural Network". In: *ISPRS Journal of Photogrammetry and Remote Sensing* (Dec. 21, 2017). ISSN: 0924-2716. DOI: 10.1016/j.isprsjprs.2017.12.007. URL: http://www.sciencedirect.com/science/article/pii/S0924271617303854 (cit. on p. 77).

[41] Jonathan Long, Evan Shelhamer, and Trevor Darrell. "Fully Convolutional Networks for Semantic Segmentation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2015, pp. 3431–3440. DOI: 10.1109/CVPR.2015.7298965 (cit. on p. 65).

[42] D. Marmanis et al. "Deep Learning Earth Observation Classification Using ImageNet Pretrained Networks". In: *IEEE Geoscience and Remote Sensing Letters* 13.1 (Jan. 2016), pp. 105–109. ISSN: 1545-598X. DOI: 10.1109/LGRS.2015.2499239 (cit. on p. 64).

[43] Dimitrios Marmanis et al. "Semantic Segmentation of Aerial Images with an Ensemble of CNNs". In: *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences* 3 (June 1, 2016), pp. 473–480. DOI: 10.5194/isprs-annals-III-3-473-2016. URL: http://adsabs.harvard.edu/abs/2016ISPAnIII3..473M (cit. on pp. 76, 77).

[44] Dimitrios Marmanis et al. "Classification With an Edge: Improving Semantic Image Segmentation with Boundary Detection". In: *ISPRS Journal of Photogrammetry and Remote Sensing* (2017). DOI: 10.1016/j.isprsjprs.2017.11.009. arXiv: 1612.01337 (cit. on p. 77).

[45] P. Márquez-Neila, L. Baumela, and L. Alvarez. "A Morphological Approach to Curvature-Based Evolution of Curves and Surfaces". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36.1 (Jan. 2014), pp. 2–17. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2013.106 (cit. on p. 61).

[46] Peer Neubert and Peter Protzel. "Superpixel Benchmark and Comparison". In: *Proc. Forum Bildverarbeitung*. 2012, pp. 1–12. URL: http://books.google.com/books?hl=en&lr=&id=39rFsOLJiRAC&oi=fnd&pg=PA205&dq=%22Liu+et+al.%22+%22algorithm+using+the+same+implementation,+data+set+and+error%22+%22image+edges+by+placing+them+inside+a+superpixel.+Depending%22+%22of+the+segmentation+compared+to+human+ground+truth%22+&ots=DmTz25PMw2&sig=TQoa_LmdyN4zyJIfJhugNHaSHEM (cit. on pp. 61, 62).

[47] Peer Neubert and Peter Protzel. "Compact Watershed and Preemptive SLIC: On Improving Trade-Offs of Superpixel Segmentation Algorithms." In: *ICPR*. 2014, pp. 996–1001. URL: https://www.tu-chemnitz.de/etit/proaut/forschung/rsrc/cws_pSLIC_ICPR.pdf (cit. on pp. 59, 61, 62).

[48] Keiller Nogueira, Otávio Penatti, and Jefersson A. dos Santos. "Towards Better Exploiting Convolutional Neural Networks for Remote Sensing Scene Classification". In: (Feb. 3, 2016). arXiv: 1602.01517 [cs]. URL: http://arxiv.org/abs/1602.01517 (cit. on p. 74).

[49] Sakrapee Paisitkriangkrai et al. "Effective Semantic Pixel Labelling with Convolutional Networks and Conditional Random Fields". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. June 2015, pp. 36–43. DOI: 10.1109/CVPRW.2015.7301381 (cit. on p. 77).

[50] Fabian Pedregosa et al. "Scikit-Learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (Oct 2011), pp. 2825–2830. ISSN: ISSN 1533-7928. URL: http://www.jmlr.org/papers/v12/pedregosa11a.html (cit. on p. 72).

[51] Otávio Penatti, Keiller Nogueira, and Jefersson A. dos Santos. "Do Deep Features Generalize from Everyday Objects to Remote Sensing and Aerial Scenes Domains?" In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). June 2015, pp. 44–51. DOI: 10.1109/CVPRW.2015.7301382 (cit. on pp. 64, 73).

[52] *PyTorch: Tensors and Dynamic Neural Networks in Python with Strong GPU Acceleration*. http://pytorch.org/. 2016–. URL: http://pytorch.org/ (cit. on p. 72).

[53] Nguyen Tien Quang et al. "An Efficient Framework for Pixel-Wise Building Segmentation from Aerial Images". In: *Proceedings of the Sixth International Symposium on Information and Communication Technology*. International Symposium on Information and Communication Technology (SoICT). ACM, 2015, p. 43. URL: http://dl.acm.org/citation.cfm?id=2833272 (cit. on pp. 76, 77).

[54] Ali Sharif Razavian et al. "CNN Features Off-the-Shelf: An Astounding Baseline for Recognition". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. June 2014, pp. 512–519. DOI: 10.1109/CVPRW.2014.131 (cit. on p. 64).

[55] Jamie Sherrah. "Fully Convolutional Networks for Dense Semantic Labelling of High-Resolution Aerial Imagery". In: (June 8, 2016). arXiv: 1606.02585 [cs]. URL: http://arxiv.org/abs/1606.02585 (cit. on p. 77).

[56] Jianbo Shi and Jitendra Malik. "Normalized Cuts and Image Segmentation". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 22.8 (Aug. 2000), pp. 888–905. ISSN: 0162-8828. DOI: 10.1109/34.868688. URL: http://dx.doi.org/10.1109/34.868688 (cit. on p. 59).

[57] Karen Simonyan and Andrew Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition". In: *Proceedings of the International Conference on Learning Representations (ICLR)*. May 2015. URL: http://arxiv.org/abs/1409.1556 (cit. on p. 66).

[58] David Stutz, Alexander Hermans, and Bastian Leibe. "Superpixels: An Evaluation of the State-of-the-Art". In: *Computer Vision and Image Understanding* 166 (Jan. 1, 2018), pp. 1–27. ISSN: 1077-3142. DOI: 10.1016/j.cviu.2017.03.007. URL: http://www.sciencedirect.com/science/article/pii/S1077314217300589 (cit. on pp. 61, 62).

[59] Christian Szegedy et al. "Going Deeper with Convolutions". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2015, pp. 1–9. DOI: 10.1109/CVPR.2015.7298594. URL: http://www.cv-foundation.org/openaccess/content_cvpr_2015/html/Szegedy_Going_Deeper_With_2015_CVPR_paper.html (cit. on pp. 67, 68).

[60] James C. Tilton et al. "Best Merge Region-Growing Segmentation With Integrated Nonadjacent Region Object Aggregation". In: *IEEE Transactions on Geoscience and Remote Sensing* 50.11 (Nov. 2012), pp. 4454–4467. ISSN: 0196-2892, 1558-0644. DOI: 10.1109/TGRS.2012.2190079. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6182584 (cit. on p. 61).

[61] Michael Van den Bergh et al. "SEEDS: Superpixels Extracted via Energy-Driven Sampling". In: *Computer Vision – ECCV 2012*. Ed. by Andrew Fitzgibbon et al. Lecture Notes in Computer Science 7578. Springer Berlin Heidelberg, Oct. 7, 2012, pp. 13–26. ISBN: 978-3-642-33785-7 978-3-642-33786-4. DOI: 10.1007/978-3-642-33786-4_2. URL: http://link.springer.com/chapter/10.1007/978-3-642-33786-4_2 (cit. on p. 60).

[62] Stéfan van der Walt et al. "Scikit-Image: Image Processing in Python". In: *PeerJ* 2 (June 19, 2014), e453. ISSN: 2167-8359. DOI: 10.7717/peerj.453. URL: https://peerj.com/articles/453 (cit. on p. 71).

[63] Andrea Vedaldi and Stefano Soatto. "Quick Shift and Kernel Methods for Mode Seeking". In: *Computer Vision – ECCV 2008*. Ed. by David Forsyth, Philip Torr, and Andrew Zisserman. Lecture Notes in Computer Science 5305. Springer Berlin Heidelberg, Oct. 12, 2008, pp. 705–718. ISBN: 978-3-540-88692-1 978-3-540-88693-8. DOI: 10.1007/978-3-540-88693-8_52. URL: http://link.springer.com/chapter/10.1007/978-3-540-88693-8_52 (cit. on p. 60).

[64] Michele Volpi and Devis Tuia. "Dense Semantic Labeling of Subdecimeter Resolution Images With Convolutional Neural Networks". In: *IEEE Transactions on Geoscience and Remote Sensing* 55.2 (Feb. 2017), pp. 881–893. ISSN: 0196-2892. DOI: 10.1109/TGRS.2016.2616585 (cit. on p. 77).

[65] Jason Yosinski et al. "How Transferable Are Features in Deep Neural Networks?" In: *Advances in Neural Information Processing Systems*. Neural Information Processing Systems (NIPS). 2014, pp. 3320–3328. URL: http://papers.nips.cc/paper/5347-how-transferable-are-features-in-deep-neural-networks (cit. on pp. 64, 74).

[66] Fisher Yu and Vladlen Koltun. "Multi-Scale Context Aggregation by Dilated Convolutions". In: *Proceedings of the International Conference on Learning Representations (ICLR)*. Nov. 23, 2015. URL: http://arxiv.org/abs/1511.07122 (cit. on pp. 67, 68).

[67] Wenzhi Zhao and Shihong Du. "Learning Multiscale and Deep Representations for Classifying Remotely Sensed Imagery". In: *ISPRS Journal of Photogrammetry and Remote Sensing* 113 (Mar. 2016), pp. 155–165. ISSN: 0924-2716. DOI: 10.1016/j.isprsjprs.2016.01.004. URL: http://www.sciencedirect.com/science/article/pii/S0924271616000137 (cit. on pp. 67, 68).

Chapter 4

Extension to unconventional sensors

I suppose it is tempting, if the only tool you have is a hammer, to treat everything as if it were a nail.

— Abraham Maslow

Contents

Summary:

Multispectral images are very common in remote sensing, yet their large number of channels prevents us from simply copy/pasting FCN architecture pretrained on RGB data. This chapter shows how to extend the state of the art segmentation results obtained on color images to multispectral acquisitions. We will start with aerial infrared-red-green-blue (IRRGB) images and move on to multispectral Sentinel-2 data, for which tuned FCN architectures can benefit from the spectral information outside the visible domain.

We will then study the extreme case of hyperspectral imagery, for which the very high number of spectral frequencies to learn from can be a problem. We will dedicate a section to review the state of the art in deep learning for hyperspectral image classification, which will show that 3D convolutional models can significantly improve the model accuracy on the hypercube despite the small size of the datasets.

Finally we will investigate how to extract semantic information from digital surface models using convolutional networks. These data contain a very rich height information that is very useful to discriminate vegetation and man-made objects, that is otherwise absent from orthorectified images. We will show that while grayscale FCN can successfully learn from the surface models, they are generally significantly less accuracte than those trained on color images, which will motivate the move to multi-modal architectures.

4.1 Multispectral images

In the previous chapter, we have seen that FCNs excelled for semantic segmentation of aerial images, both using RGB and IRRG channels. However most Earth Observation satellites, no matter if instutional (Landsat, SPOT...) or private (IKONOS, WorldView...), are equipped with multispectral sensors. These instruments can see light information invisible to the human eye but that we wish to use for automatic cartography.

4.1.1 Leveraging the near-infrared channel

The simplest multispectral sensors perform two simultaneous acquisitions – color and infrared – that result in a 4-channel IRRGB image. Most Earth Observation satellites embark this kind of sensor, like the French SPOT and Pléiades constellations. Moreover these satellites often perform a panchromatic acquisition with a significantly better spatial resolution. This combination is very common since super-resolution techniques – *pansharpening* – can then produce multispectral images with the resolution of the panchromatic acquisition. As a first step we consider the IRRGB images from the ISPRS Potsdam dataset as the simplest multispectral images, on which we can test some approaches before moving on to satellite data.

In the Chapter 3, we established that FCNs can indifferently process IRRG or RGB images. More specifically we were able to transfer the weights of a network pretrained on ImageNet to IRRG remote sensing data. However this transfer learning is not possible anymore with IRRGB multispectral images: since the number of channels changed, the network structure cannot stay the same. We worked around this problem before by dropping the infrared channels and by dealing only with the remaining 3-channels color image.

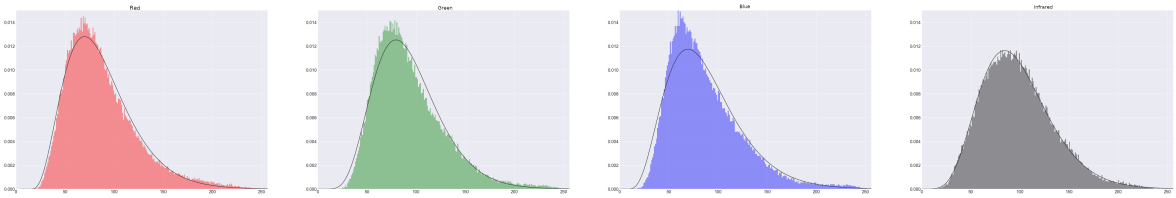


Figure 4.1: Intensity distribution for the red, green, blue and infrared channels from the ISPRS Potsdam dataset.

Before experimenting blindly we take a look at the statistical distrbution of the insentities for each channel in the dataset. The histograms plotted in the Fig. 4.1 reveal that the distributions can be modelized by gamma laws with similar parameters for the RGB channels. However the infrared channel deviates significantly from the visible channels, as showed by the inter-channel correlation maps plotted in Fig. 4.2. The visible channels are strongly correlated (Pearson coefficient > 0.87). On the contrary, the infrared channel is only moderately correlated with the other frequencies and the correlation decreases as the wavelength gap increases. The Pearson coefficient between red and infrared is at 0.80, drops at 0.69 between green and infrared and finally ata 0.57 between blue and infrared.

We perform a preliminary study on the ISPRS Potsdam dataset based on the same models and hyperparameters detailed in Chapter 3. In particular we generate several SegNet models, with and without the initialization scheme from pretrained VGG-16 weights on ImageNet, for various channel combinations. This allows us to separate the influence of the spectral bands from the gain provoked by transfer learning. We use tiles 2_10, 2_11, 2_12, 3_10, 3_11, 3_12, 4_10, 4_12, 5_11, 5_12, 6_7, 6_8, 6_9, 6_10, 6_11, 7_9, 7_10 and 7_12 for training and tiles 4_11, 5_10, 6_12, 7_7, 7_8 and 7_11 for validation. Results of this experiment are reported in the Table 4.1.

At the first glande, it seems that the 3-channel combination (RGB vs. IRRG) does not alter the model accuracy. However including the infrared channel by using the 4-channels

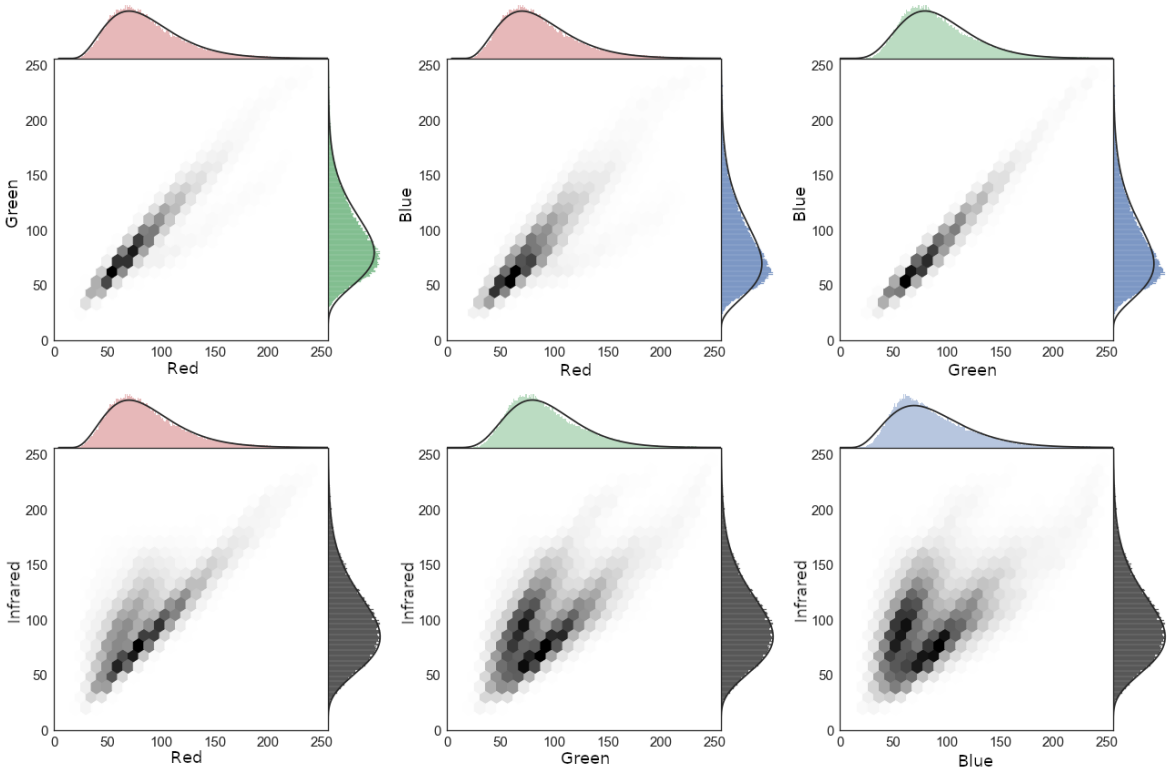


Figure 4.2: Inter-channel correlation maps on the ISPRS Potsdam dataset.

IRRGB image as input to the model significantly decreases the final classification accuracy. This holds true both with and without pretraining, i.e. when the convolutional filters are trained from scratch. Models trained on only 2 channels seem to indicate that the infrared information has an adversarial effect with the blue channel, the IR+B model leading to the worst accuracy of the tested models. This might point to a relationship between radiometric correlation between the channels stacked in the input and the overall accuracy. However this hypothesis is not easy to confirm.

4.1.2 Multispectral images

Most optical sensors embedded in Earth Observation satellites are multispectral. Indeed the most useful information is not always the one humans can see. For example chlorophyll reflects light in the near infrared, which makes thie wavelength around 706 nm a very strong

Table 4.1: Benchmark of SegNet variants for semantic segmentation on the ISPRS Potsdam dataset for several channels combinations. Transfer corresponds to the pretrained weights from ImageNet initialization.

Channels	Transfer	Roads	Buildings	Low veg.	Trees	Vehicles	Clutter	Accuracy
IR+B	✗	72.79	87.22	57.61	71.74	87.05	13.15	70.69
R+G	✗	89.92	95.80	81.49	84.30	95.21	40.42	88.48
IR+R	✗	90.75	95.89	82.77	84.97	95.50	42.47	89.25
RGB	✗	90.40	96.32	82.38	83.78	95.42	39.97	89.02
IRRG	✗	90.83	95.91	83.31	84.26	94.99	43.74	89.29
IRRGB	✗	89.67	95.57	82.35	83.82	95.17	42.89	88.51
RGB	✓	92.35	**97.62**	85.18	87.19	**96.11**	**52.15**	91.22
IRRG	✓	**92.51**	97.34	**85.68**	**87.54**	**96.11**	50.24	**91.28**

feature for vegetation. However this wavelength is outside the visible spectrum. Additionnal acquisition wavelengths in multispectral sensors help experts identify specific materials that could not be detected otherwise. Sentinel-2 is equipped to detect coastal aerosols (band 1 at 443 nm), the *red edge* from chlorophyll (bands 5 to 7 between 705 nm and 783 nm), water vapour (band 9 at 945 nm) and cirrus clouds (band 10 at 1.375 µm). The design of these sensors can be considered expert domain knowledge that should not be wasted.

This section therefore studies how FCNs can be sused for semantic segmentation of satellite multispectral images. To do so we leverage a dataset of Sentinel-2 images along with land cover labels from the *GlobeCover* 2009 project [1] agregated by Ben Hamida et al. [7]. The Sentinel-2 images we work with cover a large area at the border of France, Switzerland and Italy for a time period comprised between May and October 2016. We hypothesize that the land cover changes that occurred during the 7 years that separate the map and the data acquisition are scarce relative to the considered geographical area.

A first dataset is built based on images from May to October that do not present any cloud cover (opaque clouds or cirrus). A second dataset consists in all images, including those with clouds, but only during summer (June to August). In this second dataset, clouds are defined as separate semantic class in addition to the land cover ones, based on the Copernicus cloud mask. Indeed clouds are a major problem for satellite image processing since optical sensors cannot see through them. Clouds strongly attenuate the light going through them and detecting clouds is a major challenge that needs to be solved to deal with occlusions that naturally occur in most of the globe that do not have a desertic climate. Both datasets have multiple acquisitions on the same area at different times. All Sentinel-2 tiles are interpolated at a 20 m/px GSD for all bands. The *GlobeCover* are kept at their initial 300 m/px resolution. Details for both datasets are summarized in the Tables 4.2 and 4.3 alongside the list of classes.

Table 4.2: Descriptions of the two Sentinel-2 datasets. The first dataset covers a long time period but excludes images with clouds. The second dataset is restricted to a short time span but includes the cloud cover. Both datasets contain approximately 150 millions pixels each.

Datasets (time span)	# images training	validation	# classes
D1, large time period, no clouds (May–Oct. 2016)	140	54	16
D2, short time period, with clouds (June–August 2016)	158	39	17

We use a reduced SegNet architecture to perform the semantic segmentation of these datasets. We cut SegNet's decoder after the second convolutional block. Indeed the complete decoder is not useful since we do not aim for 1 : 1 resolution in the final maps. We use the multi-scale approach from Section 3.2.3 to generate maps at resolution 1 : 8 (160 m/px), 1 : 16 and 1 : 32. This reduces the computation time and the number of parameters to optimize. The maps are resampled by interpolation at 300 m/px, then averaged and fed to the softmax classifier. The final prediction is compared to the ground truth during training using the cross-entropy loss.

We compare two SegNet variants on this task. The first one, named SegNet RGB, only considers the bands 2, 3 and 4 from the Sentinel-2 data, i.e. the true color images. The second one, SegNet MSI, is trained using all of the 12 bands as an input[1]. The remaining hyperparameters are the same as in the Chapter 3. All networks are optimized until convergence usnig stochastic gradient descent with momentum, at learning rate of 0.001 and a momentum of 0.9 for 150 000 iterations. RGB and MSI models use respectively a batch size

[1] Band 8A is excluded in this experiment.

Table 4.3: List of classes in the D1 and D2 datasets, derived from the *GlobeCover* 2009 annotations. * The "clouds" class is added *a posteriori* using the mask from the Sentinel-2 Copernicus program.

Value	Color	*GlobeCover* 2009 name
1		Rainfeld croplands
2		Mosaic Cropland (50-70%) / Vegetation (grassland, shrubland, forest) (20-50%)
3		Mosaic Vegetation (grassland, shrubland, forest) (50-70%) / Cropland (20-50%)
4		Closed (>40%) broadleaved deciduous forest (>5m)
5		Closed (>40%) needleleaved evergreen forest (>5m)
6		Open (15-40%) needleleaved deciduous or evergreen forest
7		Closed to open (>15%) mixed broadleaved and needleleaved forest (>5m)
8		Mosaic Forest/Shrubland (50-70%) / Grassland (20-50%)
9		Mosaic Grassland (50-70%) / Forest/Shrubland (20-50%)
10		Closed to open (>15%) shrubland (<5m)
11		Closed to open (>15%) grassland
12		Sparse (>15%) vegetation (woody vegetation, shrubs, grassland)
13		Closed to open (>15%) vegetation (grassland, shrubland, woody vegetation) on regularly flooded or waterlogged soil - Fresh, brackish or saline water
14		Artificial surfaces and associated areas (urban areas >50%)
15		Bare areas
16		Water bodies
17		Clouds*

of 20 and 10, both occupying about 6 Gb of GPU memory. Training takes about 18 hours on a NVIDIA Titan X (Pascal) GPU using our PyTorch [47] implementation.

The model trained on 12 bands reaches 66.5% accuracy on the D1 dataset (no cloud) and 86.4% accuracy on D2 (with clouds). The large gap between the two datasets is due to two elements. On the one hand adding a "clouds" class increases the overall accuracy since clouds are easy to detect and plentiful in the dataset (F_1 *score* > 97%). On the other hand, as reported in Table 4.4, the scores on all classes from D1 are lesser than the same classes on D2. This is because images from D1 exhibit a low variability. Since this dataset exludes all images with a cloud cover, as thin it may be, it covers a smaller surface always with the same environmental conditions. Models trained on D1 generalize poorly on new acquisitions. On the contrary, D2 covers a shorter time span but the images are more diverse, with various weather and illumination perturbations. This allows the model to learn a form of invariance needed for a good generalization.

More importantly, learning on all 12 multispectral bands improves the overall accuracy by an absolute 2% on D1 and 2.5% on D2 compared to the RGB model. This is not specific to one class in particular, as most classes benefit from the additionnal spectral information (cf. Table 4.4). This strengthens our initial intuition: multispectral information is richer and more expressive than color image alone.

Let us underline that the labels used in this study are relatively old (2009) and coarse (300 m/px). Not only this explains the "pixel" and blocky look of the ground truth on the illustrations, it also introduces quite the approximation in the evaluation protocol. Indeed the areas imaged by Sentinel-2 might have changed (and their land cover too) since 2009. Moreover the spatial resolution of Sentinel-2 is enough to identify objects and structures that are mixed – and therefore invisible – in the *GlobeCover* annotations. Nonetheless there is a strong qualitative agreement between the predictions and the labels. It is plausible that some of the disagreements are actually due to the coarseness of the *GlobeCover* map or to changes

that occurred over time. This leads us to think that predictions inferred by SegNet might be actually more accurate on some classes – especially artificial surfaces – than the avilable "ground truth". Some qualitative examples of semantic maps are pictured in the Figs. 4.3 and 4.4.

Finally, this study leads us to two conclusions. First it shows that fully convolutional networks are also relevant for multispectral image processing. Indeed, the SegNet architecture requires minimal adaptation to work on Sentinel-2 data. Although some details remain to be polished, such as better resampling of the various bands and the reference ground truth resolution, there is no major obstacle to a large-scale implementation of FCNs for semantic segmentation of multispectral images. Second, we showed that learning from all bands including those outside the visible domain increase the model accuracy, which benefit from a richer and more expressive information. This notably increases the discrminative power of SegNet on classes that are ambiguous when looked at only using the RGB color channels.

Table 4.4: SegNet accuracy and F_1 scores on the D1 and D2 Sentinel-2 datasets (cf. Table 4.3 for detail on the classes).

Dataset	Model	Accuracy	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
D1	SegNet RGB	55.0	39.0	35.5	1.81	71.0	0.91	0.00	0.00	0.00	0.00	2.08	0.00	5.33	0.00	36.0	0.00	74.6	–
	SegNet MSI	66.5	36.2	38.1	1.45	85.1	6.33	0.00	1.39	0.00	0.00	2.18	0.00	3.36	0.00	34.3	0.00	97.8	–
D2	SegNet RGB	84.9	74.7	64.1	38.0	89.9	68.4	58.4	51.4	36.7	39.8	61.0	45.7	55.2	47.9	76.9	66.9	98.7	96.7
	SegNet MSI	86.4	76.0	66.8	43.4	92.4	72.3	51.0	59.5	22.9	50.0	67.4	48.0	53.5	41.0	77.0	66.7	98.8	97.7

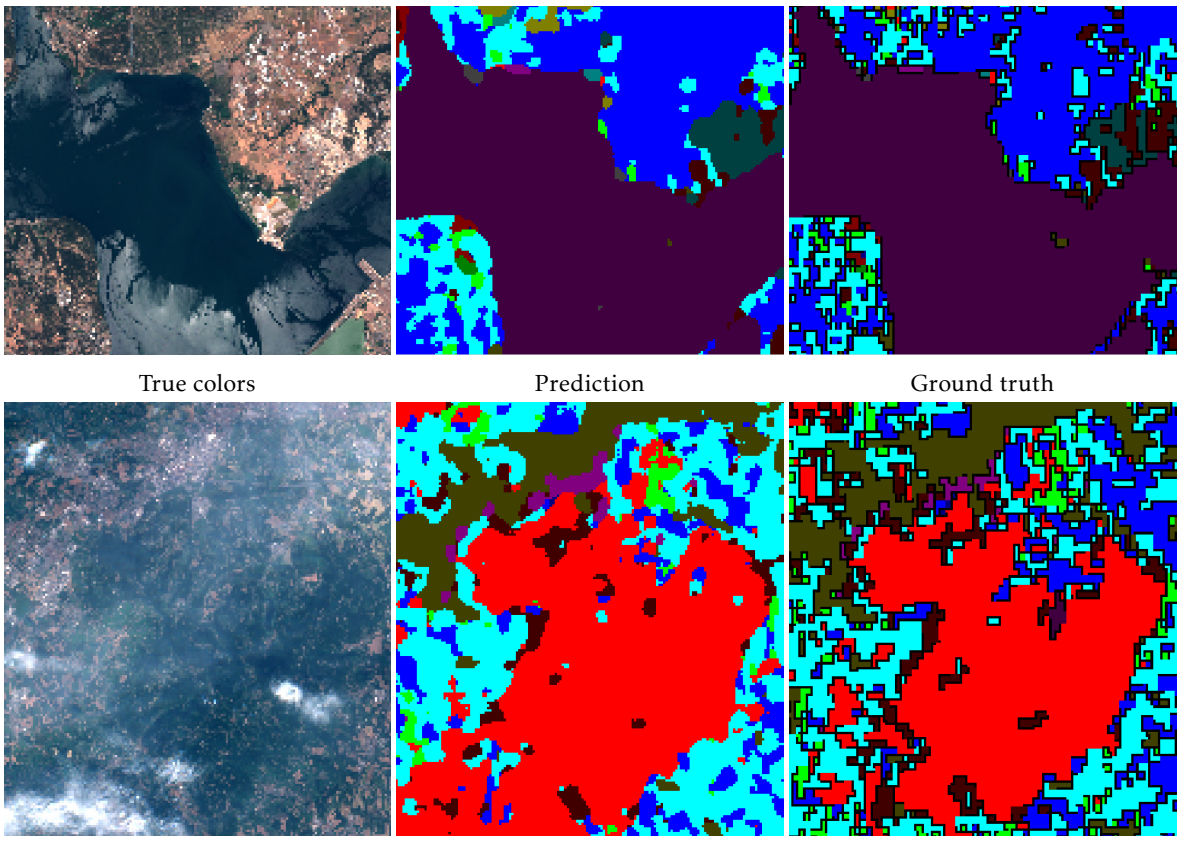
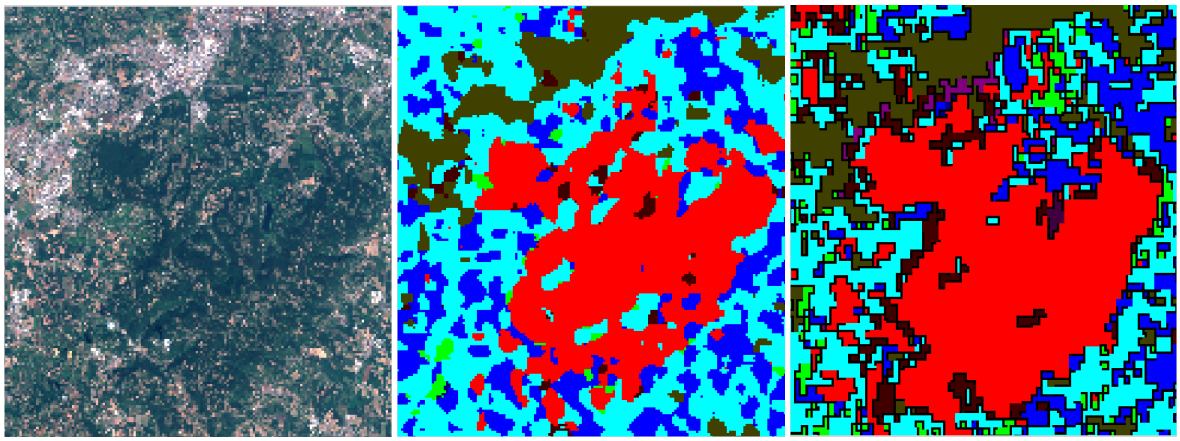


Figure 4.3: Prediction samples using SegNet MSI trained on D2 (with clouds).



True colors Prediction Ground truth

Figure 4.4: Prediction samples using SegNet MSI trained on D1 (no clouds).

4.2 Hyperspectral imaging

Up until this point we have noted that usual RGB and infrared color images strongly benefitted from the introduction of deep convolutional neural networks. Moreover we extended in Section 4.1.2 these CNNs to multispectral imaging, so that we were able to leverage the additional light wavelengths that can help detect objects that would be invisible otherwise. In the real world, every material has its own spectral signature, defined by the way it reflects light. If one could measure the complete spectrum of reflected light intensity, it would be possible to perform fine-grained analysis of ground occupation [16, 20].

This reasoning is core to the development of hyperspectral sensors. They are cameras with a relatively low spatial resolution but a high spectral resolution so they can measure the full light signature of an object for several hundreds of regularly distributed wavelengths.Deep learning techniques that have been designed for natural image and classical computer vision cannot be directly applied to this kind of data. Indeed the spectral dimension generally dominates the characteristic spatial dimension of the objects of interest. For example a standard aerial hyperspectral acquisition has a GSD of about 1 m/px and a spectral resolution of 10 nm/bande with 200 spectral bands between 0.4 µm and 2.5 µm. A standard house of $12\,\text{m} \times 10\,\text{m}$ [2], this object is described in the hyperspectral data by a $12 \times 10 \times 200$ tensor. As a comparison, EHR RGB aerial images would present a 10 cm/px GSD on 3 channels between 0.4 µm and 0.7 µm. The same house would be described by $120 \times 100 \times 3$ tensor. Although this represents the same amount of raw data (24 000 scalars versus 36 000[3]), the structures differ significantly. Hyperspectral images are generally named hyperpsectral cubes (or sometimes *hypercubes*) (cf. Fig. 4.5). Finally the low spatial resolution of hyperspectral sensors entails that a hypercube covers the same geographic area with less pixels than a color image. Labeled training samples often come in lesser number than we have been accustomed to in the previous chapter. These two factors are the main obstacles we will face to use deep learning on hyperspectral data.

In Section 4.2.1, we recall the fundamentals of hyperspectral imaging before detailing some commonly used public datasets in Section 4.2.2. We give an broad overview of the usual classification techniques used on hyperspectral images in Section 4.2.3. The Section 4.2.4 concludes this chapter with a comparative review, both theoretical and experimental of deep neural networks for automated cartography based on hyperspectral images. Readers familiar with hyperspectral imaging can skip the first sections and directly read the last one.

[2] Based on a survey from the ministry of ecology, the average surface of a house was France is 121 m^2 in 2015 (*Le prix des terrains à bâtir en 2015*).

[3] Considerations regarding integer and floating point representations are swept under the rug in this example.

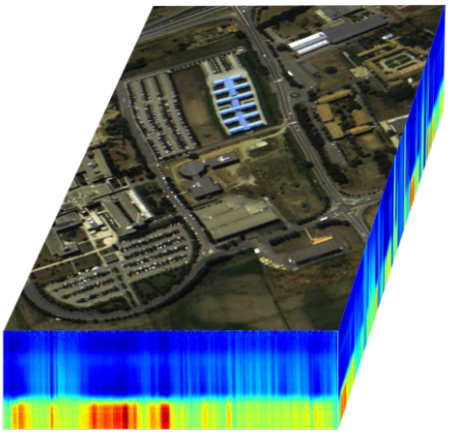


Figure 4.5: Hypercube of the *Pavia University* dataset.

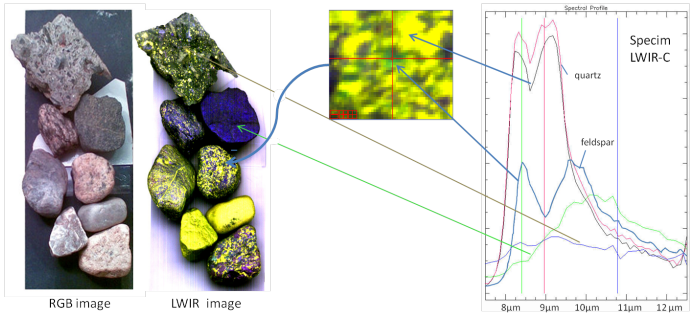


Figure 4.6: Sample reflectance for mineral identification. Image credits: Aappo Roos (Wikimedia Commons, CC-BY-SA 3.0)

4.2.1 Fundamentals of hyperspectral imaging

A hyperspectral camera measures[4] the light intensity (in spectral luminance units) of the luminous flux ϕ per unit surface per unit solid angle. This is a physical value expressed in $W{\cdot}m^{-2}{\cdot}sr^{-1}$. The sensors captures this luminous flux for a set of radiometric bands regularly distributed, generally with on 10 nm width. For each pixel, i.e. for each atomic surface unit, the sensor samples the surface spectral signature on tens – or even hundreds – of wavelengths. All these spectra can be pictured as reflectance curves, as illustrated by Fig. 4.6. The luminous flux is comprised of the light emitted and reflected by the object, but also the light diffused by the environment which is added to the measurement.

Earth Observation acquisitions are performed either from the top of the atmosphere (satellite images) or from the atmosphere itself (aerial images). However the atmosphere is not a neutral medium for light waves and changes the signal when it propagates. Satellite images can then be altered by clouds, fog and aerosols. Since the acquisitions bands are narrow, hyperspectral sensors can be very sensitive to these perturbations. In this work we focus on ground-level surfaces and materials. Therefore the relevant value is the reflectance of the ground, defined as the ratio between the flux it reflects and the incoming flux:

$$\rho = \frac{\phi_{reflected}}{\phi_{incoming}} \tag{4.1}$$

The reflectance ρ indicates the reflecting ability of an object for a given wavelength – and is also called the albedo. This a value with no unit between 0 (fully absorbing surface) and 1 (fully reflecting surface). Generally an object that reflects more than 80% of the white light appears white and an object that reflects less than 3% appears black. As for luminance measurements, we consider the radiometric reflectance ρ_λ that depends on the wavelength. The reflectance is more useful than luminance since it is an intrinsic property of the material and does not depend on environmental conditions. The reflectance curve of a material corresponds to its spectral signature and is a very discriminative feature (cf. Fig. 4.6). When possible we will prefer to work with reflectance values.

Environmental corrections Converting luminance values to reflectances require that environmental factors be suppressed to avoid polluting the measurement. Compensating

[4]Multispectral and hyperspectral cameras generally work on a *push broom* mode: they acquire an image line by line, which is different from usual color cameras. Technical details regarding the acquisition of the images and sensor calibration are out of the scope of this manuscript.

perturbations involve techniques called atmospheric corrections [19, 48, 11]. These are designed to reduce the influence of the atmosphere on the measurement [25] (diffusion and distortion) and transform luminance images into reflectance ones. To do so specialistes design atmospheric models they then invert to estimate and remove the impact of light duffision and radiative phenomenons. Generally these models require knowledge of experimental conditions, especially the brightness. Part of these informations can be obtained after the experiment thanks to ephemerids or *in situ* by embedding a sunlight sensor on the back of the plane. Portable hyperspectral sensors use an active illumination directly oriented on the target to work around this problem.

In additio, geometry is also involved in estimating the reflectance. Indeed it is often assumed that the ground is locally planar. Natural terrain relief and elevated objects can introduce undesired reflections and occlusions. The former result in over-illumination when several rays converge to the same point while the latter induce shadows that attenuate the signal. Some correction techniques can take the DSM into account to alleviate these problems, especially common in urban areas [10].

No matter the correction applied and the expertise involved, let us underline that all preprocessing of the sort is imperfect and might induce errors and uncertainties in the data.

Visualization Contrary to the human eye, a hyperspectral camera can see well beyond the visible spectrum. Most hyperspectral sensors cover wavelengths from ultraviolet (300 nm) to the limit of the medium infrared (3000 nm) using bands of about 10 nm. In comparison the visible light only covers wavelengths from 300 nm (purple) to $\simeq$ 700 nm (red). Everyday screens use the RGB color mode and agregate three intensity maps in red, green and blue to give the illusion of a natural image. This approaches mimick the three types of receptive cells present in the human eye. However a hyperspectral image is a data cube inside which every pixel contains a full spectral response. These spectral signature characterize surfaces and materials when they are completely pure. In practice the low spatial resolution induces some level of mixing of various materials inside the same pixel, especially when vegetation is involved.

As there is a significant spectral resolution gap between hyperspectral (very accurate compared to human eyes) and traditional RGB imaging, there is straightforward way to switch between the two. A hyperspectral image contains much more information than the color image with the spatial resolution. Moreover, if it is feasible to reconstruct a RGB image by compositing three well-chosen channels in the hypercube, the difference in resolution entails that is only a pseudo-image which would not have been this way by human eyes. Indeed a consumer-grade camera acquire separately red, green and blue light using filters to imitate the human eye, while a hyperspectral sensor generally perform a line-by-line acquisition of the whole spectrum separated by a prism (“*pushbroom*” sensor). These two different modes that are not equivalent.

4.2.2 Datasets

The community has made public several labeled hyperspectral images to study and compare machine learning techniques for cartography[5]. We detail here the most popular datasets.

As we will see, one of the main difficulties in machine learning for hyperspectral image processing is the scarcity of labeled data for supervised learning. Since different sensors exhibit different behaviours, calibrations and specificiations (number of bands, resolution…), it is hard to combine multiple datasets, especially since external factors also play a role (brightness, atmospheric correction etc.). Unluckily the few labeled hyperspectral images available to scientists are very small compared to the usual RGB image datasets. This make benchmarking and validating supervised machine learninge techniques more complex and

[5] http://www.ehu.eus/ccwintco/index.php?title=Hyperspectral_Remote_Sensing_Scenes

error-prone. There is one large-scale hyperspectral dataset[6] over the US, acquired using the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) camera, however the images are not labeled and therefore not directly useful to train supervised models.

Pavia Pavia is a hyperspectral dataset acquired using the ROSIS sensor with 1.3 m GSD over the city of Pavia, Italy. It consists in two scenes: Pavia University (103 bands, 610 px × 340 px) and Pavia Centre (102 bands, 1096 px × 715 px). The two images are labeled for 9 classes of interest, ranging from urban materials (bricks, asphalt, metals) to water and vegetation. Labeled pixels cover approximately 50% of the images.

It is one of the main dataset in the hyperspectral image processing community since the images are two of the biggest available that have been extensively annotated. Moreover the same camera has been used for both images which makes it possible to test transfer learning approaches on hyperspectral data.

Indian Pines The Indian Pines dataset is an acquisition of the American AVIRIS sensor. It represents an agricultural area using 224 spectral bands on 145 px × 145 px, with a ground resolution of 3.7 m/px. Most of the image consists in fields divided in about 10 types of crops, the remaining surface being mostly dense vegetation. 16 classes of interest have been labeled, some being very rare (less than 100 samples). Water absorption bands (108→112, 154→167 and 224) are generally removed from the data since they are extremely noisy.

Depite its small size, the Indian Pines dataset is very popular in the hyeprspectral literature. The scarce classes are sometimes ignored when evaluating classification algorithms.

Salinas The Salinas dataset is another image acquired using the AVIRIS sensor. The scene is comprised of 512 × 217 spectral samples at 3.7 m/px GSD. The water absorption bands (108→112, 154→167 and 224) are generally removed from the data. There are labels for 16 classes, mostly crops, vegetation and types of soil.

Kennedy Space Center (KSC) The KSC dataset is another AVIRIS image with a spatial resolution of 18 m/px. It covers the area around the Kennedy Space Center at Cape Canaveral (Florida, United States). The image is of dimension 512 × 614. Water absorption bands and bands with a low signal-to-noise ratio are removed to keep only the 176 most informative bands. 13 vegetation classes have been labeled around the center.

Botswana The Botswana dataset is a satellite acquisition over the Okavango river delta using the Hyperion sensor from NASA's EO-1 satellite. Its GSD is 30 m/px with 242 bands, resulting in a 1476 × 256 image. Only the 145 bands 10→55, 82→97, 102→119, 134→164 and 187→220 are generally kept, the other being either in the water absorption frequencies or wrongly calibrated. There are 14 classes of interest consisting in various vegetation and swamps types relevant to the local ecosystem.

DFC 2018 The DFC 2018 hyperspectral dataset is a large hyperspectral 2384 × 1202 image over Houston downtown (Texas, United-States) using an aerial hyperspectral camera. Its spectral range covers 380–1050 nm with 48 bands at a 1 m/px GSD. There are 20 labeled classes, including urban structures (buildings, different types of roads, rails, cars, trains…) and vegetation (healthy, stressed, evergreen and deciduous). The image is one of the data source from the Data Fusion Contest 2018, detailed in l'Appendix A.1.3. Half the labels are publicly available, the rest is kept hidden by the organizers to manage an independent evaluation server.

[6] *AVIRIS Data Portal:* https://aviris.jpl.nasa.gov/alt_locator/

Summary Characteristics of the presented public datasets are listed in the Table 4.5. The main conclusion we can draw from this report is that most datasets are small, with Indian Pines being tiny compared to other large remote sensing datasets. The AVIRIS sensor is used on multiple scenes but the labeled classes are not the same across acquisitions, which prevents model generalization on multiple datasets.

Table 4.5: Summary of the various publicly annotated hyperspectral datasets.

Dataset	# pixels	# bands	Spectral range	Resolution	# labels	# classes	Mode
Pavia	991 040	103	0.43–0.85 µm	1.3 m/px	50 232	9	Aerial
Indian Pines	21 025	224	0.4–2.5 µm	3.7 m/px	10 249	16	Aerial
Salinas	111 104	224	0.4–2.5 µm	3.7 m/px	54 129	16	Aerial
KSC	314 368	176	0.4–2.5 µm	18 m/px	5211	13	Aerial
Botswana	377 856	145	0.4–2.5 µm	30 m/px	3248	14	Satellite
DFC 2018	5 014 744	48	0.38–1.05 µm	1 m/px	547 807	20	Aerial

4.2.3 Usual approaches

This section gives a brief overview of commonly used machine learning techniques applied in hyperspectral image processing state of the art, with a focus on supervised methods.

Normalization and preprocessing

As we have seen raw hyperspectral data are difficult to interpret directly. In addition to atmospheric correction and geometric orthorectification to produce georeferenced reflectance maps, it is common to apply some sort of normalization on the hypercube.

First, it is not uncommon to remove some wavelengths that are difficult to learn from. Reflectance values in some bands might saturate and squash the spectral dynamics, depending on sensor calibration. On the opposite, atmospheric humidity attenuates the signal in the water absorption bands, adding noise to the data. Overall it is frequent to keep only bands with a decent signal-to-noise ratio, which reduces slightly the dimension of the data and facilitates the training by removing noisy information.

Second, a numerical normalization of the spectrum values are frequently applied. The normalization strategy depends on the spectral properties that one wish to highlight:

- If the shape of the spectra is more imoprtant the actual intensity values, practicionners often switch to the spectral angle representation which is obtained using a spectral normalization:
$$X^* := \frac{X}{\|X\|} \ ,$$
- Normalizing the statistical moments of first and second order (zero-mean and unit-variance) can be done globally or for each band to make outlier more prominent and easier to remove (using the $\pm 5\sigma$ rule for example):
$$I^* := \frac{I - m_I}{\sigma_I^2} \text{ where } m_I \text{ is the mean ofI and } \sigma_I \text{ its standard deviation,}$$
- The global normalization in $[0,1]$ is often used to simplify the handling of numerical values. It can also be applied band-wise to give the same importante to all wavelengths:
$$I^* := \frac{I - \min(I)}{\max(I) - \min(I)} \ .$$

Outliers, e.g. reflectances over the 98$^{\text{th}}$ percentile or over $m_I + 5\sqrt{\sigma_I}$ can either be truncated or ignored to prevent them from degrading the classification. This can be critical to avoid learning on anomalous pixels provoked by correction errors, multiple reflections or strongly reflective materials (such as metals).

Let us stress that one pixel from a hyperspectral would correspond to the spectral signature of the observed material on the unit surface – in perfect experimental conditions. In the real world the spatial resolution of the hyperspectral sensors is quite low and one pixel corresponds to a mixture of several materials. Let $\varphi_1, \ldots, \varphi_n$ denote the pure spectra of the set of unique materials appearing in the observation. Then, to a pixel (i, j) corresponds a local observed spectrum $\phi_{i,j}$ that is related to φ_i by a function f:

$$\phi_{i,j} \quad = \quad f_{i,j}(\varphi_1, \ldots, \varphi_n) \quad \simeq \quad \sum_{k=1}^{n} \lambda_k \varphi_k \ . \tag{4.2}$$

If the terrain is plane, we can hypothetize that f is a linear combination in wich the coefficient λ_k represents the proportion of material k in the mixture corresponding to the observed area[7].

A large body of work exists in the literature to solve the "unmixing" problem, i.e. inverting the mixtures [44]. The simplest classification possible consists in finding all the raw materials that are present in the observation and then compute abundancy maps. The reference spectra of the pure materials are called *endmembers*[8] and can be used to decompose the mixed spectra. Abundancy maps consist in the various proportions of the materials in every pixel. Generally, if the pure spectra S_k are known for an image I, it is possible to solve the inverse linear system to obtain the λ_k mixture coefficients on all pixels. These methods mostly rely on linear algebra techniques and numerical methods for problem inversion, e.g. signal decomposition. Learning-based techniques also exist for unmixing, for example using clustering to find the endmembers when they are unknown. Identifying endmembers and unmixing in its broadest sense is out of the scope of this work.

Spectrum classification

The simplest hyperspectral data classification approaches are piwel-wise operations that process every spectrum independently from the others. We present in the following subsection some of these 1-dimensional methods. We deliberately ignore techniques relying on expert feature engineering to focus on statistical learning.

A first step in many classification pipelines consists in dimension reduction to alleviate the curse of dimensionality. Because of the high spectral resolution of hyperspectral imaging sensors, neighbouring bands tend to be located in very close wavelengths and therefore be strongly correlated. Close bands therefore express redundant information. Compressing the hypercube can then be interesting both for reducing the size of dataset and keeping only discriminative information. For example, Le Bris et al. [30] and Bevilacqua and Berthoumieu [8] use band selection strategies to preserve only the bands relevant for classification. Rodarmel and Shan [50] applies a compression strategy by transforming the spectra with a Principal Component Analysis (PCA) before classification. Some expert features such as the NDVI and NDWI can also be interpreted as a way to reduce the dimensionality of the hypercube. Once reduced the data is classified using standard statistical models: decision trees and random forests, SVM, etc. Dimension reduction simplifies the representation space and facilitates the optimization of the classifiers.

[7] However it should be noted that some materials have non-linear interactions and these cases have to be dealt with separately.

[8] A reference to mineralogy, where *endmembers* are minerals at the end of a purity chain. Most minerals are solid solution, i.e. mixtures of these *endmembers*).

Nonetheless the spectral-only strategy is often not enough since it does not leverage the spatial structure of the objects of interest. Indeed as imaging technology improves, so does the sensor spatial resolution and the number of pixels observed for the same area. Neighbouring pixels will share many spectral properties while also exhibiting specific spatial structures. For example buildings are generally built with the same material but have polygonal shapes while vegetation is chaotic and fractal. Learning from these characteristics robustify the models.

The simplest strategy consists in performing a spectrum-wise classification using a 1D model and then regularize the inferences using a post-processing. Graphical models and especially CRF are very well-suited to this end [62] as they can model priors regarding class transitions. The spatial regularization is decorrelated from the spectral prediction as it comes only in a second step.

On the contrary, some methods integrate the spatial features as soon as possible using the region-based classification pipeline – presented in Section 3.1. Tarabalka, Chanussot, and Benediktsson [57] and Fauvel et al. [22] compare several two-step pipelines: first a segmentation of the hyperspectral image and then pixel-wise predictions agregated and merged for each region to enforce a local spatial consistency.

Finally there are some classifiers based on spatial-spectral features. This is the approach originally introduced to leverage correlation between spatially close pixels to detect end-members [46, 18] using a mixture of spatial and spectral classifiers. More recent classifiers use kernels specifically tailored to work on local spectral neighbourhoods, either with fixed or adaptative shapes, to extract spatial-spectral features. Notably Camps-Valls et al. [9] introduced SVMs with spatial-spectral kernels for hyperspectral images which became quickly popular in the literature [56, 21]. More recently Cui, Chapel, and Lefèvre [17] proposed SVMs with kernels tailored to work on morphological attribute profiles while Tuia, Flamary, and Courty [59] designed an adaptive kernel selection on a set of random convolutional filters that reduces the gap with deep representation learning methods.

4.2.4 Deep learning and hyperspectral imaging

Models detailed up to this point are shallow classifiers with no representation learning. Yet the hyperspectral image processing community started to look into deep neural networks in 2013 and many papers have ssince been published to adapt these methods to hypercubes.

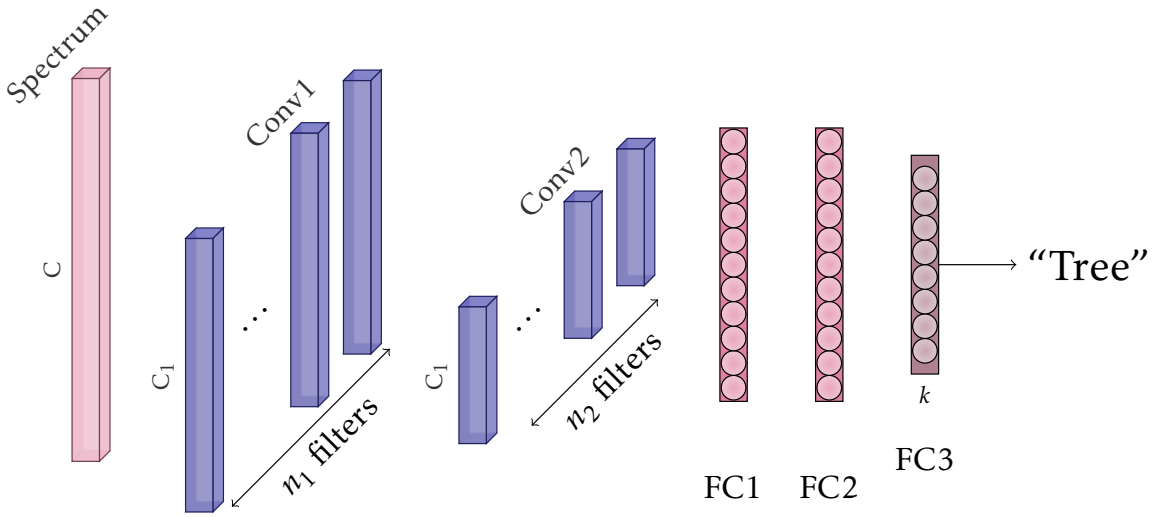


Figure 4.7: 1D CNN for spectra classification Hu et al. [29].

An initial improvement on the standard shallow classifiers (SVMs and random forests) consists in replacing them by multilayer perceptron. The pipeline remains exactly the same,

yet if the neural network is deep enough, it will probably be more expressive than the shallow classifiers and *ergo* able to learn more discriminative features. This approach is actually not that recent since there were already some works based on shallow neural networks with one or two hidden layers in the 2000s [26, 49]. The deep learning tsunami needed a few years to reach the field of hyperspectral image processing. In 2015, Hu et al. [29] used 1D-CNNs on individual spectra (cf. Fig. 4.7) to classify them using automatically learnt features Mou, Ghamisi, and Zhu [41] introduced an alternative take on spectrum classification by considering them as sequences of reflectances from which a Recurrent Neural Network (RNN) can learn patterns.

Once deep neural networks are applied on hyperspectral data, few authors spend a lot of time looking into band selection, outlier rejection, saturation compensation or extensive analysis of the physics involved. Deep models indeed shine in representation learning and they are most often used on the – normalized – raw data, even when it includes noisy or saturated spectral bands. They are robust enough to not care too much about these problems in practice as the model is supposed to naturally ignore non-relevant information from the data.

A popular group of models that contributed to this trend are the autoencoders. As signal compression algorithms, they have been used to train dimension models with a minimal information loss. Since autoencoders are tailored on a specific dataset, they learn compressed embeddings that are significantly more efficient than standard unsupervised approaches such as PCA, with many applications in denoising [63]. The unsupervisedly learnt embeddings can finally be used as features for any sort of statistical classifier [24].

As explained before spectral only approaches are rarely enough and there exists a plethora of spatial-spectral techniques based on a mixture of spatial and spectral features, combining one pixel and its neighbours. A classical feature consists in concatenating the spectrum of the considered pixel with the K main components of an PCA applied on its local $w \times h$ neighbourhood (in most cases, $w = h \simeq 8$ and $K = 3$). This vector is used as an input to deep classifiers, either supervised or unsupervised: DBN [33, 12], RBM [35, 40] or stacked autoencoders [13, 38, 55, 61].

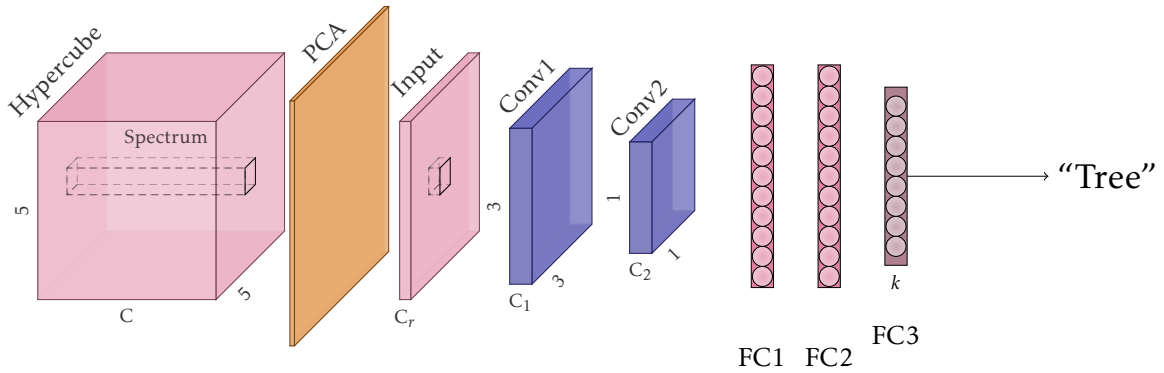


Figure 4.8: Hybrid PCA+CNN architecture for hypercube classification Makantasis et al. [39].

The comeback of CNNs after 2010 also impacted the hyperspectral community. These networks were designed to mimick the human eye and process RGB or grayscale images using 2D convolutional filters. Makantasis et al. [39] and Slavkovikj et al. [53] designed a hybrid architecture that alternates between spatial convolutions and spectral dimension reduction (using a PCA for Makantasis et al. [39] and by downsampling for Slavkovikj et al. [53]). The features produced this way can be flattened and fed to a multilayer perceptron that performs the final classification, as pictured in Fig. 4.8. The main advantage of this approach is to automatically learn features tailored for the classification task at hand. Zhao et al. [68] extend this technique in the semi-supervised setting by introducing multi-scale convolutional autoencoders. In the unsupervised setting, Romero, Gatta, and Camps-Valls

[51] proposed a CNN for feature extraction that learnt a sparse dimension reduction based on a spectrum and its neighbours. Finally Zhao and Du [67] and Yue et al. [66] suggest a hybrid approach combining a 2D CNN as a spatial feature extractor combined to a 1D-CNN to learn spectral features.

Although effective these architectures lack a form of elegance as they separate spectral and spatial aspects of the hyeprspectral data. However more classical approaches have shown that spatial-spectral kernels often outperform the classifiers based one type of feature only. Several works have simultaneously proposed 3-dimensional convolutions to learn kernels that directly operate on the data cube. Ben Hamida et al. [6] and Chen et al. [14] suggested CNN architectures based on a combination of 3D convolutions for feature learning and 1D convolution for spectral compression. Luo et al. [37] introduced a variant of the PCA-CNN of Makantasis et al. [39] by replacing the PCA by a 3D convolutional layer that achieves the same dimension reduction, followed by a classical 2D CNN. As usual, classification is achived pixel-wise using two fully convolutional layers at the top of the network. Lee and Kwoon [32] extended this structure to the more efficient FCN design with a first layer that learns spatial-spectral features with two parallel convolutions in 1D and 3D, inspired by the *Inception* module. The rest of the network has a fully convolutional design based on 1D convolutions inside residual blocks.

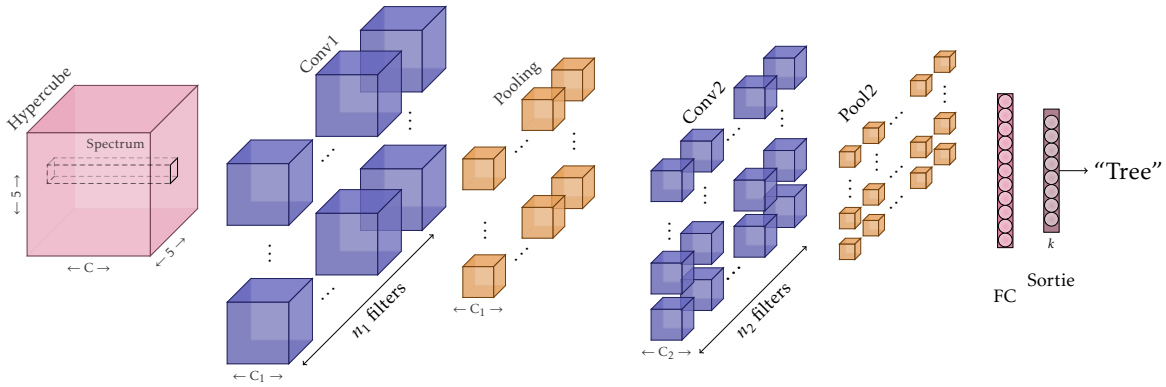


Figure 4.9: 3D CNN for hypercube classification Chen et al. [14].

Finally the most recent approaches converged to full 3D convolutional networks derived from the canonical CNN architectured, extended from RGB color images to hypercubes in the third dimension. Many variations of this principle have been introduced [34] integrating more and more bells and whistles from the computer vision literature such as multi-scale feature extraction [28] and semi-supervised training [36]. Overall this a natural extension of the CNN model from LeCun et al. [31] to 3D data volumes, as schematized in Fig. 4.9.

Despite the large number of publications on hyperspectral image classification, there is no standardized benchmark to validate models and pit several methods in competition. Worringly, each author has their own strategy and chooses one or more datasets from those presented in Section 4.2.2 with specific train/validation/test splits. This makes it difficult to robustly compare methods across papers since authors rarely prepare data the same way. Moreover there are virtually no open source implementations of deep hyperspectral classifiers officially sponsored by their authors, while this is now relatively common in comptuer vision. For these reasons we developed a modular deep learning toolbox for semantic mapping of hyperspectral images, named *DeepHyperX*[9]. It encompass several supervised models, from linear SVMs to the state of the art 3D CNNs. These models can trained and evaluated on various public datasets such as Pavia Center and University, Indian Pines, Kennedy Space Center or Data Fusion Contest (DFC) 2018. The most common hyperparameters can be tuned to study how the size of the spatial neighbourhood, the number of training samples or the

[9] https://github.com/nshaud/DeepHyperX

optimizer impact the classification accuracy. This software allows us to provide an unified benchmarking setting to compare the performance of various state of the art models.

On the technical side, this toolbox is written in Python [23] and is an interface wrapping the PyTorch [47] and scikit-learn [45] libraries. Deep neural networks are implemented on PyTorch so they can be run either on Central Processing Unit (CPU) or GPU while the SVMs use the scikit-learn implementation. Several public datasets are preconfigured to facilitates experiments. The modular architecture of the toolbox is designed so that programmers can easily add new custom datasets or new deep networks to test new ideas or experiment state of the art models on private datasets.

In what follows we evaluate several deep models from the state of the art in hyperspectral image classification. To the best of our knowledge it is the first principled benchmark of the convolutional networks from the literature. Most publications perform experiments that slightly differ one from another, either because they ignore some classes or because the train/validation splits are not consistent across papers. Moreover the most frequent approach to select training samples consists in training the models on a set of pixels randomly sampled uniformly on the image, and then validate on the test set which is the remaining pixels of the image. We argue that this method is a best unrealistic and sometimes even plain wrong. Indeed, close pixels will be strongly correlated and therefore the validation set will be very similar to the validation set, so the classification metrics will not represent a reasonable estimate of the model's generalization ability. Instead, accuracy will be significantly overestimated. as this reward overfitting. These practices are not standard in the machine learning community and even discouraged. The preference genrally goes to a clear split between train and test. In our case will evaluate the models using *spatially disjoint* train/test split; We perform k-fold cross-validation on multiple splits to ensure the robustness of the results. More critically, for 2D and 3D CNNs that looks not only at a pixel but also its neighbours, it ensures that no pixel from the validation set can be accidentally seen during training.

In the rest of this section, we will use the splits defined by the Institute of Electrical and Electronics Engineers (IEEE) Geoscience & Remote Sensing Society (GRSS) on the DASE evaluation server[10] for the Indian Pines, Pavia University and DFC 2018 datasets. Hyperparameters are tuned using a small validation set containing 5% of the training set.

We use our toolbox to reimplement several models from the state of the art. We tried our best to faithfully reproduce the models. Some modifications have been made and are listed below:

- CNN 1D de Hu et al. [29]. As the optimizer was not specified in the paper, we use stochastic gradient descent with momentum in its stead.
- RNN 1D de Mou, Ghamisi, and Zhu [41]. We used the usual *tanh* activation instead of the parametrized version introduced in the paper.
- CNN 3D+1D de Ben Hamida et al. [6]. No modification.
- CNN 3D de Li, Zhang, and Shen [34]. We increased the number of filters from 16 to 32 in the convolutional layers for better convergence.

The 3D CNNs are trained on 5×5 neighbourhoods. We use two simple models as baselines: an SVM with hyperparameters tuned by grid search and a multilayer perceptron with three layers using the ReLU [43] activation with *Dropout* [54] regularization. The unbalance between classes is corrected at the loss function level by using the inverse median frequency weighting. Data augmentation is applied in the form of flipping and mirroring. Detailed results are reported in the Table 4.6, incudling the overall accuracy et Cohen's κ on the three datasets. Experiments have been repeated 5 times on Pavia University and Indian Pines, though only once on the DFC 2018 dataset since it is significantly larger.

[10] *GRSS Data and Algorithm Standard Evaluation website* : http://dase.ticinumaerospace.com/

Table 4.6: Classification results of several models from the *DeepHyperX* toolbox on the Indian Pines, Pavia University and DFC 2018 datasets. The best results are in **bold**, the second best in *italics*.

Model	Indian Pines		Pavia University		DFC 2018	
	Accuracy	κ	Accuracy	κ	Accuracy	κ
SVM	81.43	0.788	69.56	0.592	42.51	0.39
1D NN	**83.13**± 0.84	**0.807**± 0.009	76.9± 0.86	0.711± 0.010	41.08	0.37
1D CNN [29]	*82.99*± 0.93	*0.806*± 0.011	81.18± 1.96	0.759± 0.023	*47.01*	*0.44*
RNN [41]	79.70± 0.91	0.769± 0.011	67.71± 1.25	0.599± 0.014	41.53	0.38
3D+1D CNN [6]	74.31± 0.73	0.707± 0.008	*83.80*± 1.29	*0.792*± 0.016	46.28	0.43
3D CNN [34]	75.47± 0.85	0.719± 0.010	**84.32**± 0.72	**0.799**± 0.009	**49.26**	**0.46**

Unsurprisingly we obtain results significantly lower than those reported in the original publications as we use a spatially disjoint train and validation sets. Our results highlight a singular behaviour of the Indian Pines dataset compared to other images. In practice, it seems that spatial models underperform pure spectral models on this dataset. We suggest that the low spatial resolution of Indian Pines (20 m/px) might entail already significanty mixtures of various endmembers in crop patches of 400 m^2. Neighbouring pixels might not bring more information. On higher resolution Pavia University and DFC 2018, 3D CNNs significantly outperform 1D models, increasing the overall accuracy by respectively 3% and 2%. In particular, the DFC 2018 dataset is quite hard due to the large number of similar classes with low inter-class variability. In our experiments 1D fully connected networks suffer from a strong overfitting and generally performs worse than a simple linear SVM. This overfitting is especially consequent on the DFC 2018 as the test set is completely disjoint from the initial training image, whereas train and test objects can be nearby in Indian Pines and Pavia University splits from DASE. Finally the 3D CNN 3D from Ben Hamida et al. [6] is not able to learn discriminative spatial information from the DFC 2018 dataset. In practice the first two 3D convolutional layers have too few parameters and the overall receptive field of the network is too small to be effective in modeling the spatial relationships between pixels at such as high resolution.

A major obstacle that we identify in this study is the difficulty of training models that do not suffer from overfitting on hyperspectral datasetS. The small number of labeled samples available for training are rarely enough to train the large deep networks introduced in the literature without falling into the trivial memorization solution. Increasing the training set size is not easy either as sensors have different specifications and calibrations that prevent practictioners from relying on transfer learning. Even though domain adaptation techniques can alleviate problems regarding the generalization of trained models on new images [60], they do not solve the problem of the initial training. One workaround consists in generating fake synthetic data that is realistic enough to improve the generalization ability of the model. This approach will be studied in Chapter 6.

Let us also remind that most current neural networks for hyperspectral images perform pixel-wise classification using one inference per pixel. As disccused in the Chapter 3, the ever-increasing resolution of the hyperspectral sensors will require that new models switch to the more efficient fully convolutional architectures adapted to 3D kernels.

Finally, we hope that new annotated hyperspectral datasets – larger and more complex than the existing ones – will appear in the next few years. The current public datasets have reached saturation and the incremental improvements from the state of the art are of dubious statistical robustness. Moreover it is unclear what the benefit is from replacing a model with a 99,5% accuracy with another one with 99,8%. It is more plausible that current models are overfitting on the datasets: are human annotators even precise at more than 99%? A standard benchmark based on a new large-scale dataset would make it possible to fairly benchmark various methods on complex hyperspectral classification tasks, as has been done for natural

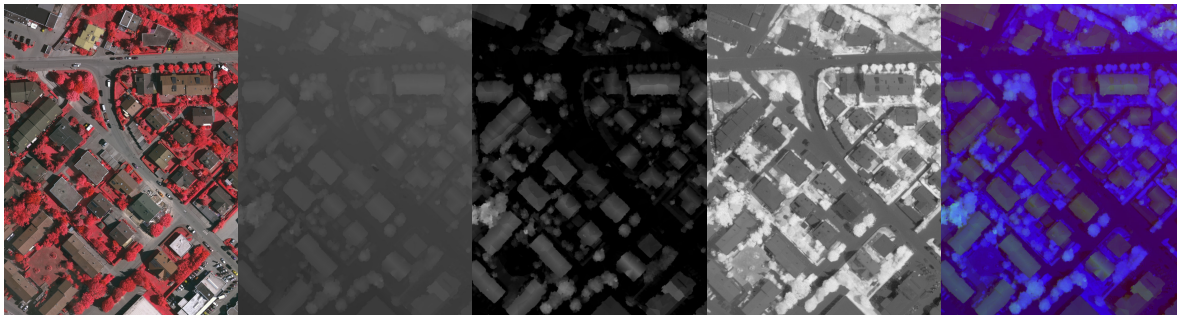
images. The efforts from the IEEE GRSS in this direction are laudable, especially thanks to the introduction of the DFC 2018.

Overall we were able to highlight the problems that practioners will face when looking to apply deep learning on hyperspectral images. Although there is an apparent consensus praising spatial-spectral approaches, especially 3D CNNs, as superior to traditional classification techniques, we showed that actual accuracies are significantly lower than reported in the literature when using more strict evaluation strategies. We developed and published a software toolbox called *DeepHyperX* for deep learning on hyperspectral images that makes it possible to easily compare various models on multiple public datasets with a standard protocol. This will allow hyperspectral specialists to apply deep networks from the state of the art on their semantic mapping tasks, but also machine learning experts to validate their models robustly. We hope this tool will help reinforce the progresses that have been made in combining deep learning and hyperspectral imaging.

4.3 Lidar imaging and digital surface models

One of the popular Earth Observation sensor except optical sensors is the Light Detection And Ranging (Lidar). It is laser sensor that can be used, among other applications, to measure the height of any point at the surface of the Earth. Optical acquisitions from an airplane are often complemented by a Lidar sensor to estimate the ground topology. This section studies how FCNs can be used to leverage this information and compare this approach to the results obtained with image-based models in the Chapter 3.

The Section 4.3.1 first details a mapping strategy based on digital surface models as the sole input while the Section 4.3.2 investigates the use of composite fake color images agregating height maps and NDVI.



IRRG orthoimage | DSM | nDSM | NDVI | Composite

Figure 4.10: Multiple modalities of the tile #30 of the ISPRS Vaihingen dataset.

4.3.1 Digital surface model

Point clouds acquired by Lidar imagery produce digital models of the terrain, its relief and topology: DTM, DSM and nDSM. These points are not regularly distributed on the ground, instead they form a cloud with a variable density that never corresponds to a well-defined regular grid. Obtaining digital surface models can be done through rasterization (i.e. projection in a 2D plane and interpolation) of the Lidar point clouds [15] or by stereo matching in an image pair [58]. We will focus on the former. Terrain models are interesting since they contain local elevation at the ground level (DTM) or at the object level (nDSM). Most remote sensing data, both aerial and satellite, are acquired from a nadir position and are orthorectified: they do not contain any perspective information so there is no geometrical clue that could help find heights or distances (a pecularity that is not shared with natural images). This has its advantages (very few occludings elements, a unique scale factor) but

also a drawback: it is very hard to estimate the height of an object based on an orthophoto alone. Projected shadows can give a partial information regarding objects' heights but it is not reliable since it depends on the ground relief and environmental illumination conditions (acquisition azimuth, Sun's location, weather).

Yet urban environments contain many elevated structures that can be confused with artificial soils: bridges, parking garages, concrete or vegetal roofs, dense tree-like vegetation... Vision-based interpretation using deep networks as done in the Chapter 3 produces semantic maps where these situations result in erroneous predictions. The digital surface models would provide an ancillary information that would complement optical images and improve the model accuracies.

A digital surface model can be understood as an image in which every pixel has an intensity proportional to its height, based on its geographical coordinates. The reference level can vary and is not necessarily the same for all pixels (for example in the nDSM). In practice the heights are normalized between $[0, 255]$ resulting in grayscale images. A first question that one could ask is "how do semantic segmentation deep networks such as SegNet perform on these images?". For completeness' sake, let us recall that there are techniques to directly process the raw Lidar signal. For example Yan, Shaker, and El-Ashmawy [64] looked into land cover mapping based on classification of Lidar echos while Yang et al. [65] used a CNN for semantic segmentation of Lidar point clouds. These methods are however out of the scope of this work.

Focusing on the grayscale rasters corresponding to DSM and nDSM, we use the ISPRS Vaihingen dataset. We keep the hyperparameters tuned in the Chapter 3. The goal here is to estimate how much semantic information deep models can extract from the digital surface models. In this case the models are derived from Lidar data although they could be computed by stereo matching with no practical difference.

We train a SegNet model with only one input channel on the DSM and the nDSM. We use the tiles 1, 3, 7, 11, 13, 17, 23, 26, 28, 32, 34 and 37 for training and the tiles 5, 15, 21 and 30 for validation. Network weights are initialized randomly using the policy from He et al. [27]. Results are reported in Table 4.7, including F_1 for the five classes of interest of the ISPRS Vaihingen dataset and the overall accuracy. These results can be compared to those reported in the Table 3.5 from Section 3.3.4.

Table 4.7: Validation results on the ISPRS Vaihingen using SegNet trained on the DSM and nDSM (F_1 scores and global accuracy).

Input	Imp. surfaces	Buildings	Low veg.	Trees	Vehicles	Accuracy
nDSM	78.57	93.16	55.86	83.80	32.29	80.53
DSM	77.94	92.69	56.57	84.15	60.60	80.29
nDSM + DSM	77.67	93.47	55.93	84.01	28.39	80.30

It seems that learning from a digital surface model alone can result in high F_1 scores on roads, buildings and trees. These classes are indeed the simplest to separate based on the height information given by the Lidar data: buildings are large homogeneously elevated surfaces, trees are moderately elevated objects with a high local entropy and the ground is a large plane with a low slope. It is interesting to see that the model learns a spatial prior regarding how low vegetation is distributed in the city. It is randomly placed around trees in order to create "green" areas that correspond to the small parks and vegetalized decor of a small town. In addition, vehicles can be predicted with a moderate accuracy from the DSM. However there are harder to predict based on the nDSM since the normalization process flattens the ground and makes most cars disappear.

To conclude, although there is some information to be extracted from the digital surface models, the overall accuracy is significantly lower than the one obtained using the IRRG images.

4.3.2 Building a composite image

As previously detailed, digital surface models alone are not enough to map the diversity of objects we are interested in over urban areas. The information is not rich enough to discriminate low vegetation from impervious surfaces or smaller vehicles that disappear in the nDSM.

To encompass all classes we need not only the height but also some kind of color information. The NDVI is a vegetation index defined as the normalized ratio between the near-infrared and the red intensities:

$$NDVI = \frac{IR - \mathrm{R}}{IR + \mathrm{R}} \ . \tag{4.3}$$

The NDVI is bounded in $[-1, +1]$, -1 means that there is no vegetation while $+1$ means a high vegetation density. NDVI is effective as it models that reflection peak of the vegetation in the near-infrared and its absorption peak in the red wavelength due to chlorophyll in the leafs. Therefore NDVI characterize the presence and the density of vegetation in the observed area [42]. NDVI also can used to detect artificial structures when it is close to -1 [52].

We design a composite 3-channels image agregating the DSM, nDSM and NDVI as pictured in the Fig. 4.10.

Table 4.8: Validation results on the ISPRS Vaihingen dataset using a SegNet model trained on composite images (with and without ImageNet pretraining).

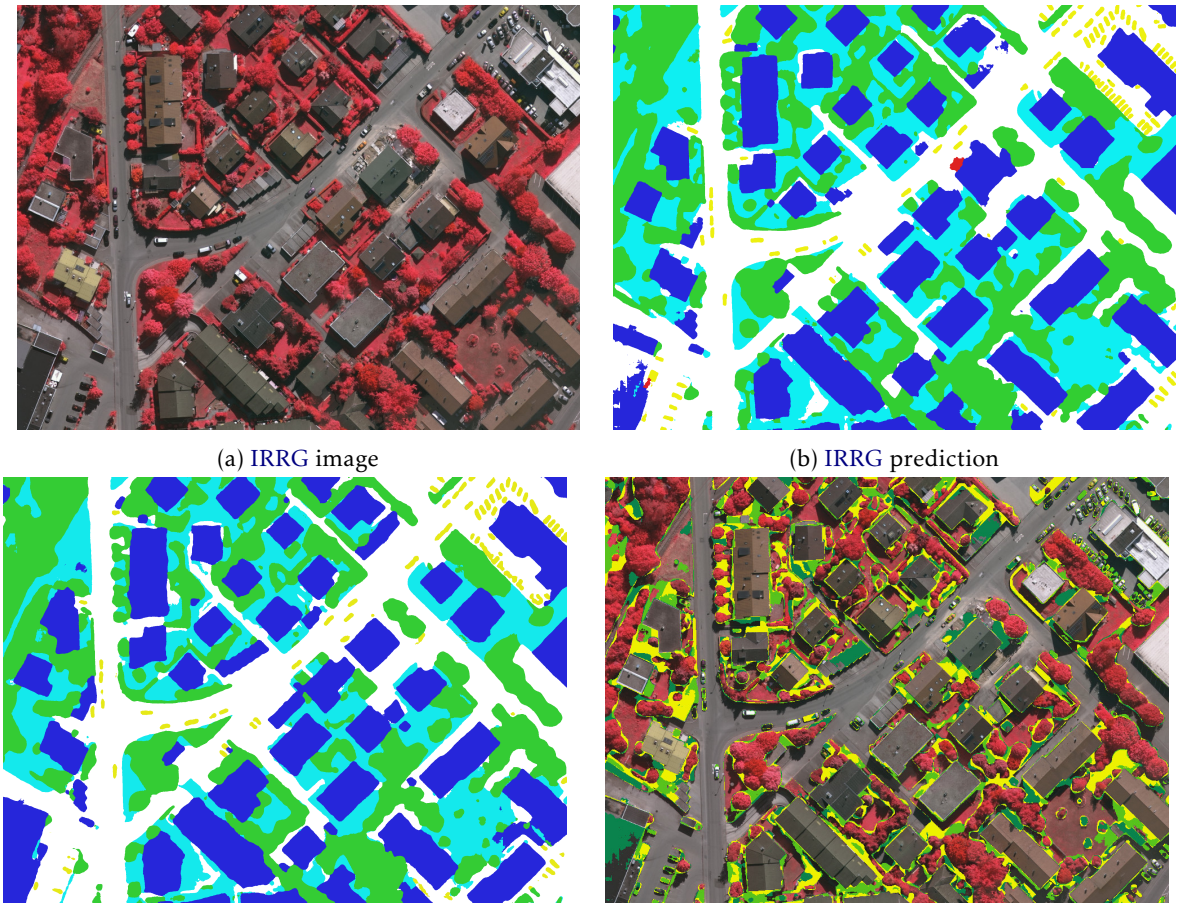
Input	Transfer	Imp. surfaces	Buildings	Low veg.	Trees	Vehicles	Accuracy
Composite	✗	91.39	95.02	75.68	88.66	61.86	89.07
Composite	✓	91.34	95.48	76.47	89.39	73.47	89.61
IRRG	✓	91.43	95.37	79.97	90.53	90.41	90.47

Table 4.9: Validation results on the ISPRS Potsdam dataset using a SegNet model trained on composite images (with and without ImageNet pretraining).

Input	Transfer	Imp. surfaces	Buildings	Low veg.	Trees	Vehicles	Accuracy
Composite	✗	89.81	96.72	79.04	80.55	87.60	87.87
Composite	✓	90.81	97.23	80.89	81.17	92.47	89.20
RGB	✓	92.35	97.62	85.18	87.19	96.11	91.22

Results obtained using a SegNet trainined on the composite DSM/nDSM/NDVI images on the ISPRS Vaihingen and Potsdam datasets are reported in the Tables 4.8 and 4.9 respectively (cf. Table 4.7). In addition we tried models with and without ImageNet pretrained weights from VGG-16. Although pretraining outperforms initializing the weights from scratch on all classes, the overall accuracy never reaches the one we obtained when training SegNet directly on the IRRG images.

We show in Fig. 4.11 some predictions obtained by SegNet when trained respectively on IRRG and composite tiles. The first mask contains only 12% wrong pixels while the second contains about 13% wrong pixels. While comparable, the two error masks are complementary: errors do not occur on the same pixels. Each of the two inputs gives information about different parts of the image. Were we able to perfectly combine the two maps so that only pixels that are wrongly classified by both models are wrong, the error rate would drop to 7% on this example (yellow mask). This motivates the study of multimodal learning for data fusion in deep neural networks, which is the topic of the next chapter.



(a) IRRG image (b) IRRG prediction

(c) Composite prediction (d) Error mask

Figure 4.11: Differences between the predictions from the IRRG and composite models. (b),(c) Colors: white: roads, blue: buildings, cyan: low vegetation, green: trees, yellow: vehicles, red: clutter. (d) In lime green we show the errors of the model trained on the composite data, in dark green the errors of the model trained on the IRRG data and the intersection of both masks in yellow.

The works presented in this chapter have been published in international conferences:

- Amina Ben Hamida et al. "Deep Learning for Semantic Segmentation of Remote Sensing Images with Rich Spectral Content". In: *2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. July 2017, pp. 2569–2572. DOI: 10.1109/IGARSS.2017.8127520
- Nicolas Audebert, Bertrand Le Saux, and Sébastien Lefèvre. "Semantic Segmentation of Earth Observation Data Using Multimodal and Multi-Scale Deep Networks". In: *Computer Vision – ACCV 2016*. Springer, Cham, Nov. 20, 2016, pp. 180–196. DOI: 10.1007/978-3-319-54181-5_12
- Nicolas Audebert et al. "A Real-World Hyperspectral Image Processing Pipeline for Vegetation and Hydrocarbon Characterization". In: *Proceedings of the 9th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*. Sept. 2018

Two of these conference publications have been extended into journal articles:

- Nicolas Audebert, Bertrand Le Saux, and Sébastien Lefèvre. "Beyond RGB: Very High Resolution Urban Remote Sensing with Multimodal Deep Networks". In:

ISPRS Journal of Photogrammetry and Remote Sensing (Nov. 23, 2017). ISSN: 0924-2716. DOI: 10.1016/j.isprsjprs.2017.11.011

- Nicolas Audebert, Bertrand Le Saux, and Sébastien Lefèvre. "Deep Learning for Classification of Hyperspectral Data: A Comparative Review". In: *IEEE Geoscience and Remote Sensing Magazine* in press (Mar. 2019)

References

[1] Olivier Arino et al. *Global Land Cover Map for 2009 (GlobCover 2009)*. Aug. 23, 2012. DOI: https://doi.org/10.1594/PANGAEA.787668. URL: https://doi.pangaea.de/10.1594/PANGAEA.787668 (cit. on p. 90).

[2] Nicolas Audebert, Bertrand Le Saux, and Sébastien Lefèvre. "Semantic Segmentation of Earth Observation Data Using Multimodal and Multi-Scale Deep Networks". In: *Computer Vision – ACCV 2016*. Springer, Cham, Nov. 20, 2016, pp. 180–196. DOI: 10.1007/978-3-319-54181-5_12 (cit. on p. 107).

[3] Nicolas Audebert, Bertrand Le Saux, and Sébastien Lefèvre. "Beyond RGB: Very High Resolution Urban Remote Sensing with Multimodal Deep Networks". In: *ISPRS Journal of Photogrammetry and Remote Sensing* (Nov. 23, 2017). ISSN: 0924-2716. DOI: 10.1016/j.isprsjprs.2017.11.011 (cit. on p. 107).

[4] Nicolas Audebert, Bertrand Le Saux, and Sébastien Lefèvre. "Deep Learning for Classification of Hyperspectral Data: A Comparative Review". In: *IEEE Geoscience and Remote Sensing Magazine* in press (Mar. 2019) (cit. on p. 108).

[5] Nicolas Audebert et al. "A Real-World Hyperspectral Image Processing Pipeline for Vegetation and Hydrocarbon Characterization". In: *Proceedings of the 9th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*. Sept. 2018 (cit. on p. 107).

[6] Amina Ben Hamida et al. "Deep Learning Approach for Remote Sensing Image Analysis". In: *Big Data from Space (BiDS'16)*. Ed. by SOILLE Pierre MARCHETTI Pier Giorgio. Santa Cruz de Tenerife, Spain: Publications Office of the European Union, Mar. 2016, p. 133. DOI: 10.2788/854791. URL: https://hal.archives-ouvertes.fr/hal-01370161 (cit. on pp. 101–103).

[7] Amina Ben Hamida et al. "Deep Learning for Semantic Segmentation of Remote Sensing Images with Rich Spectral Content". In: *2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. July 2017, pp. 2569–2572. DOI: 10.1109/IGARSS.2017.8127520 (cit. on pp. 90, 107).

[8] Marco Bevilacqua and Yannick Berthoumieu. "Unsupervised Hyperspectral Band Selection via Multi-Feature Information-Maximization Clustering". In: *2017 IEEE International Conference on Image Processing (ICIP)*. Pékin, China: IEEE, Sept. 2017. DOI: 10.1109/ICIP.2017.8296339. URL: https://hal.archives-ouvertes.fr/hal-01717011 (cit. on p. 98).

[9] Gustavo Camps-Valls et al. "Composite Kernels for Hyperspectral Image Classification". In: *IEEE Geoscience and Remote Sensing Letters* 3.1 (2006), pp. 93–97. URL: http://ieeexplore.ieee.org/abstract/document/1576697/ (cit. on p. 99).

[10] Xavier Ceamanos et al. "Using 3D Information for Atmospheric Correction of Airborne Hyperspectral Images of Urban Areas". In: *2017 Joint Urban Remote Sensing Event (JURSE)*. Mar. 2017, pp. 1–4. DOI: 10.1109/JURSE.2017.7924563 (cit. on p. 95).

[11] Pat S. Chavez. "Image-Based Atmospheric Corrections: Revisited and Improved". In: *Photogrammetric engineering and remote sensing* 62.9 (1996), pp. 1025–1036. URL: http://cat.inist.fr/?aModele=afficheN&cpsidt=3201162 (cit. on p. 95).

[12] Yushi Chen, Xing Zhao, and Xiuping Jia. "Spectral-Spatial Classification of Hyperspectral Data Based on Deep Belief Network". In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 8.6 (June 2015), pp. 2381–2392. ISSN: 1939-1404, 2151-1535. DOI: 10.1109/JSTARS.2015.2388577. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7018910 (cit. on p. 100).

[13] Yushi Chen et al. "Deep Learning-Based Classification of Hyperspectral Data". In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 7.6 (June 2014), pp. 2094–2107. ISSN: 1939-1404. DOI: 10.1109/JSTARS.2014.2329330 (cit. on p. 100).

[14] Yushi Chen et al. "Deep Feature Extraction and Classification of Hyperspectral Images Based on Convolutional Neural Networks". In: *IEEE Transactions on Geoscience and Remote Sensing* 54.10 (Oct. 2016), pp. 6232–6251. ISSN: 0196-2892. DOI: 10.1109/TGRS.2016.2584107 (cit. on p. 101).

[15] Ziyue Chen, Bingbo Gao, and Bernard Devereux. "State-of-the-Art: DTM Generation Using Airborne LIDAR Data". In: *Sensors* 17.1 (Jan. 14, 2017), p. 150. DOI: 10.3390/s17010150. URL: http://www.mdpi.com/1424-8220/17/1/150 (cit. on p. 104).

[16] Manuel Cubero-Castan et al. "A Physics-Based Unmixing Method to Estimate Subpixel Temperatures on Mixed Pixels". In: *IEEE Transactions on Geoscience and Remote Sensing* 53.4 (Apr. 2015), pp. 1894–1906. ISSN: 0196-2892. DOI: 10.1109/TGRS.2014.2350771 (cit. on p. 93).

[17] Yanwei Cui, Laetitia Chapel, and Sébastien Lefèvre. "Scalable Bag of Subpaths Kernel for Learning on Hierarchical Image Representations and Multi-Source Remote Sensing Data Classification". In: *Remote Sensing* 9.3 (Feb. 24, 2017), p. 196. DOI: 10.3390/rs9030196. URL: http://www.mdpi.com/2072-4292/9/3/196 (cit. on p. 99).

[18] Fabio Dell'Acqua et al. "Exploiting Spectral and Spatial Information in Hyperspectral Urban Data with High Resolution". In: *IEEE Geoscience and Remote Sensing Letters* 1.4 (Oct. 2004), pp. 322–326. ISSN: 1545-598X. DOI: 10.1109/LGRS.2004.837009 (cit. on p. 99).

[19] P. Y. Deschamps and T. Phulpin. "Atmospheric Correction of Infrared Measurements of Sea Surface Temperature Using Channels at 3.7, 11 and 12 μm". In: *Boundary-Layer Meteorology* 18.2 (Mar. 1, 1980), pp. 131–143. ISSN: 0006-8314, 1573-1472. DOI: 10.1007/BF00121320. URL: https://link.springer.com/article/10.1007/BF00121320 (cit. on p. 95).

[20] Sophie Fabre, Xavier Briottet, and Audrey Lesaignoux. "Estimation of Soil Moisture Content from the Spectral Reflectance of Bare Soils in the 0.4–2.5 μm Domain". In: *Sensors* 15.2 (Feb. 2, 2015), pp. 3262–3281. DOI: 10.3390/s150203262. URL: http://www.mdpi.com/1424-8220/15/2/3262 (cit. on p. 93).

[21] Mathieu Fauvel, Jocelyn Chanussot, and Jón Atli Benediktsson. "A Spatial-Spectral Kernel-Based Approach for the Classification of Remote-Sensing Images". In: *Pattern Recogn.* 45.1 (Jan. 2012), pp. 381–392. ISSN: 0031-3203. DOI: 10.1016/j.patcog.2011.03.035. URL: http://dx.doi.org/10.1016/j.patcog.2011.03.035 (cit. on p. 99).

[22] Mathieu Fauvel et al. "Advances in Spectral-Spatial Classification of Hyperspectral Images". In: *Proceedings of the IEEE* 101.3 (Mar. 2013), pp. 652–675. ISSN: 0018-9219. DOI: 10.1109/JPROC.2012.2197589 (cit. on p. 99).

[23] Python Software Foundation. *Python Language Reference*. https://www.python.org/. URL: https://www.python.org/ (cit. on p. 102).

[24] Qiongying Fu et al. "Semi-Supervised Classification of Hyperspectral Imagery Based on Stacked Autoencoders". In: *Proceedings of the 8th Interational Conference on Digital Image Processing (ICDIP)*. Vol. 10033. 2016. DOI: 10.1117/12.2245011. URL: http://dx.doi.org/10.1117/12.2245011 (cit. on p. 100).

[25] Bo-Cai Gao et al. "Atmospheric Correction Algorithms for Hyperspectral Remote Sensing Data of Land and Ocean". In: *Remote Sensing of Environment* 113 (2009), S17–S24. URL: http://www.sciencedirect.com/science/article/pii/S0034425709000741 (cit. on p. 95).

[26] Pradeep Goel et al. "Classification of Hyperspectral Data by Decision Trees and Artificial Neural Networks to Identify Weed Stress and Nitrogen Status of Corn". In: *Computers and Electronics in Agriculture* 39.2 (May 2003), pp. 67–93. ISSN: 0168-1699. DOI: 10.1016/S0168-1699(03)00020-6. URL: http://www.sciencedirect.com/science/article/pii/S0168169903000206 (cit. on p. 100).

[27] Kaiming He et al. "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification". In: *Proceedings of the IEEE International Conference on Computer Vision*. IEEE International Conference on Computer Vision (ICCV). Dec. 2015, pp. 1026–1034. DOI: 10.1109/ICCV.2015.123 (cit. on p. 105).

[28] Mingyi He, Bo Li, and Huahui Chen. "Multi-Scale 3D Deep Convolutional Neural Network for Hyperspectral Image Classification". In: *2017 IEEE International Conference on Image Processing (ICIP)*. 2017 IEEE International Conference on Image Processing (ICIP). Sept. 2017, pp. 3904–3908. DOI: 10.1109/ICIP.2017.8297014 (cit. on p. 101).

[29] Wei Hu et al. "Deep Convolutional Neural Networks for Hyperspectral Image Classification". In: *Journal of Sensors* 2015 (2015). DOI: 10.1155/2015/258619. URL: https://www.hindawi.com/journals/js/2015/258619/ (cit. on pp. 99, 100, 102, 103).

[30] Arnaud Le Bris et al. "Extraction of Optimal Spectral Bands Using Hierarchical Band Merging out of Hyperspectral Data". In: *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* XL-3/W3 (Aug. 20, 2015), pp. 459–465. ISSN: 2194-9034. DOI: 10.5194/isprsarchives-XL-3-W3-459-2015. URL: http://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XL-3-W3/459/2015/ (cit. on p. 98).

[31] Yann LeCun et al. "Gradient-Based Learning Applied to Document Recognition". In: *Proceedings of the IEEE* 86.11 (Nov. 1998), pp. 2278–2324. ISSN: 0018-9219. DOI: 10.1109/5.726791 (cit. on p. 101).

[32] Hyungtae Lee and Heesung Kwoon. "Contextual Deep CNN Based Hyperspectral Classification". In: *2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. IGARSS. Beijing, July 2016, pp. 3322–3325. DOI: 10.1109/IGARSS.2016.7729859 (cit. on p. 101).

[33] Tong Li, Junping Zhang, and Ye Zhang. "Classification of Hyperspectral Image Based on Deep Belief Networks". In: *Image Processing (ICIP), 2014 IEEE International Conference On*. IEEE, 2014, pp. 5132–5136. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=7026039 (cit. on p. 100).

[34] Ying Li, Haokui Zhang, and Qiang Shen. "Spectral–Spatial Classification of Hyperspectral Imagery with 3D Convolutional Neural Network". In: *Remote Sensing* 9.1 (Jan. 13, 2017), p. 67. DOI: 10.3390/rs9010067. URL: http://www.mdpi.com/2072-4292/9/1/67 (cit. on pp. 101–103).

[35] Zhouhan Lin et al. "Spectral-Spatial Classification of Hyperspectral Image Using Autoencoders". In: *Information, Communications and Signal Processing (ICICS) 2013 9th International Conference On*. IEEE, 2013, pp. 1–5. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6782778 (cit. on p. 100).

[36] Bing Liu et al. "A Semi-Supervised Convolutional Neural Network for Hyperspectral Image Classification". In: *Remote Sensing Letters* 8.9 (Sept. 2, 2017), pp. 839–848. ISSN: 2150-704X. DOI: 10.1080/2150704X.2017.1331053. URL: https://doi.org/10.1080/2150704X.2017.1331053 (cit. on p. 101).

[37] Yanan Luo et al. "HSI-CNN: A Novel Convolution Neural Network for Hyperspectral Image". In: (Feb. 28, 2018). arXiv: 1802.10478 [cs]. URL: http://arxiv.org/abs/1802.10478 (cit. on p. 101).

[38] Xiaorui Ma, Hongyu Wang, and Jie Geng. "Spectral-Spatial Classification of Hyperspectral Image Based on Deep Auto-Encoder". In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 9.9 (Sept. 2016), pp. 4073–4085. ISSN: 1939-1404. DOI: 10.1109/JSTARS.2016.2517204 (cit. on p. 100).

[39] Konstantinos Makantasis et al. "Deep Supervised Learning for Hyperspectral Data Classification through Convolutional Neural Networks". In: *Geoscience and Remote Sensing Symposium (IGARSS), 2015 IEEE International*. Geoscience and Remote Sensing Symposium (IGARSS), 2015 IEEE International. July 2015, pp. 4959–4962. DOI: 10.1109/IGARSS.2015.7326945 (cit. on pp. 100, 101).

[40] Elamkulam Midhun et al. "Deep Model for Classification of Hyperspectral Image Using Restricted Boltzmann Machine". In: *Proceedings of the 2014 International Conference on Interdisciplinary Advances in Applied Computing*. ICONIAAC '14. New York, NY, USA: ACM, 2014, 35:1–35:7. ISBN: 978-1-4503-2908-8. DOI: 10.1145/2660859.2660946. URL: http://doi.acm.org/10.1145/2660859.2660946 (cit. on p. 100).

[41] Lichao Mou, Pedram Ghamisi, and Xiao Xiang Zhu. "Deep Recurrent Neural Networks for Hyperspectral Image Classification". In: *IEEE Transactions on Geoscience and Remote Sensing* 55.7 (July 2017), pp. 3639–3655. ISSN: 0196-2892. DOI: 10.1109/TGRS.2016.2636241 (cit. on pp. 100, 102, 103).

[42] Ranga B. Myneni et al. "The Interpretation of Spectral Vegetation Indexes". In: *IEEE Transactions on Geoscience and Remote Sensing* 33.2 (Mar. 1995), pp. 481–486. ISSN: 0196-2892. DOI: 10.1109/36.377948 (cit. on p. 106).

[43] Vinod Nair and Geoffrey E. Hinton. "Rectified Linear Units Improve Restricted Boltzmann Machines". In: *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*. 2010, pp. 807–814 (cit. on p. 102).

[44] Lucas Parra et al. "Unmixing Hyperspectral Data". In: *Proceedings of the 12th International Conference on Neural Information Processing Systems*. MIT Press, 1999, pp. 942–948. URL: http://dl.acm.org/citation.cfm?id=3009790 (cit. on p. 98).

[45] Fabian Pedregosa et al. "Scikit-Learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (Oct 2011), pp. 2825–2830. ISSN: ISSN 1533-7928. URL: http://www.jmlr.org/papers/v12/pedregosa11a.html (cit. on p. 102).

[46] Antonio Plaza et al. "Spatial/Spectral Endmember Extraction by Multidimensional Morphological Operations". In: *IEEE Transactions on Geoscience and Remote Sensing* 40.9 (Sept. 2002), pp. 2025–2041. ISSN: 0196-2892. DOI: 10.1109/TGRS.2002.802494 (cit. on p. 99).

[47] *PyTorch: Tensors and Dynamic Neural Networks in Python with Strong GPU Acceleration*. http://pytorch.org/. 2016–. URL: http://pytorch.org/ (cit. on pp. 91, 102).

[48] Hafizur Rahman and Gérard Dedieu. "SMAC: A Simplified Method for the Atmospheric Correction of Satellite Measurements in the Solar Spectrum". In: *International Journal of Remote Sensing* 15.1 (Jan. 1, 1994), pp. 123–143. ISSN: 0143-1161. DOI: 10.1080/01431169408954055. URL: http://dx.doi.org/10.1080/01431169408954055 (cit. on p. 95).

[49] Frédéric Ratle, Gustau Camps-Valls, and Jason Weston. "Semisupervised Neural Networks for Efficient Hyperspectral Image Classification". In: *IEEE Transactions on Geoscience and Remote Sensing* 48.5 (May 2010), pp. 2271–2282. ISSN: 0196-2892. DOI: 10.1109/TGRS.2009.2037898 (cit. on p. 100).

[50] Craig Rodarmel and Jie Shan. "Principal Component Analysis for Hyperspectral Image Classification". In: *Surveying and Land Information Science* 62.2 (2002), p. 115. URL: http://search.proquest.com/openview/621b3a7187ca7f1dff4769113d396b20/1?pq-origsite=gscholar&cbl=27246 (cit. on p. 98).

[51] Adriana Romero, Carlo Gatta, and Gustau Camps-Valls. "Unsupervised Deep Feature Extraction for Remote Sensing Image Classification". In: *IEEE Transactions on Geoscience and Remote Sensing* PP.99 (2015), pp. 1–14. ISSN: 0196-2892. DOI: 10.1109/TGRS.2015.2478379 (cit. on p. 100).

[52] Mitsuteru Sakamoto et al. "Automatic Detection of Damaged Area of Iran Earthquake by High-Resolution Satellite Imagery". In: *IGARSS 2004. 2004 IEEE International Geoscience and Remote Sensing Symposium*. IGARSS 2004. 2004 IEEE International Geoscience and Remote Sensing Symposium. Vol. 2. Sept. 2004, 1418–1421 vol.2. DOI: 10.1109/IGARSS.2004.1368685 (cit. on p. 106).

[53] Viktor Slavkovikj et al. "Hyperspectral Image Classification with Convolutional Neural Networks". In: *Proceedings of the 23rd ACM International Conference on Multimedia*. ACM Press, 2015, pp. 1159–1162. ISBN: 978-1-4503-3459-4. DOI: 10.1145/2733373.2806306. URL: http://dl.acm.org/citation.cfm?doid=2733373.2806306 (cit. on p. 100).

[54] Nitish Srivastava et al. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting". In: *Journal of Machine Learning Research* 15 (2014), pp. 1929–1958. URL: http://jmlr.org/papers/v15/srivastava14a.html (cit. on p. 102).

[55] Chao Tao et al. "Unsupervised Spectral-Spatial Feature Learning With Stacked Sparse Autoencoder for Hyperspectral Imagery Classification". In: *IEEE Geoscience and Remote Sensing Letters* 12.12 (Dec. 2015), pp. 2438–2442. ISSN: 1545-598X. DOI: 10.1109/LGRS.2015.2482520 (cit. on p. 100).

[56] Yuliya Tarabalka, Jón Atli Benediktsson, and Jocelyn Chanussot. "Spectral–Spatial Classification of Hyperspectral Imagery Based on Partitional Clustering Techniques". In: *IEEE Transactions on Geoscience and Remote Sensing* 47.8 (2009), pp. 2973–2987. URL: http://ieeexplore.ieee.org/abstract/document/4840429/ (cit. on p. 99).

[57] Yuliya Tarabalka, Jocelyn Chanussot, and Jon Atli Benediktsson. "Segmentation and Classification of Hyperspectral Images Using Watershed Transformation". In: *Pattern Recognition* 43.7 (2010), pp. 2367–2379. URL: http://www.sciencedirect.com/science/article/pii/S003132031000049X (cit. on p. 99).

[58] Thierry Toutin. "Comparison of Stereo-Extracted DTM from Different High-Resolution Sensors: SPOT-5, EROS-a, IKONOS-II, and QuickBird". In: *IEEE Transactions on Geoscience and Remote Sensing* 42.10 (Oct. 2004), pp. 2121–2129. ISSN: 0196-2892. DOI: 10.1109/TGRS.2004.834641 (cit. on p. 104).

[59] Devis Tuia, Rémi Flamary, and Nicolas Courty. "Multiclass Feature Learning for Hyperspectral Image Classification: Sparse and Hierarchical Solutions". In: *ISPRS Journal of Photogrammetry and Remote Sensing* 105 (July 2015), pp. 272–285. ISSN: 0924-2716. DOI: 10.1016/j.isprsjprs.2015.01.006. URL: http://www.sciencedirect.com/science/article/pii/S0924271615000234 (cit. on p. 99).

[60] Devis Tuia, Claudio Persello, and Lorenzo Bruzzone. "Domain Adaptation for the Classification of Remote Sensing Data: An Overview of Recent Advances". In: *IEEE Geoscience and Remote Sensing Magazine* 4.2 (June 2016), pp. 41–57. ISSN: 2168-6831. DOI: 10.1109/MGRS.2016.2548504. URL: http://ieeexplore.ieee.org/document/7486184/ (cit. on p. 103).

[61] Lizhe Wang et al. "Spectral–Spatial Multi-Feature-Based Deep Learning for Hyperspectral Remote Sensing Image Classification". In: *Soft Computing* 21.1 (Jan. 1, 2017), pp. 213–221. ISSN: 1432-7643, 1433-7479. DOI: 10.1007/s00500-016-2246-3. URL: https://link.springer.com/article/10.1007/s00500-016-2246-3 (cit. on p. 100).

[62] Junfeng Wu et al. "Semi-Supervised Conditional Random Field for Hyperspectral Remote Sensing Image Classification". In: *2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. 2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS). July 2016, pp. 2614–2617. DOI: 10.1109/IGARSS.2016.7729675 (cit. on p. 99).

[63] Chen Xing, Li Ma, and Xiaoquan Yang. "Stacked Denoise Autoencoder Based Feature Extraction and Classification for Hyperspectral Images". In: *Journal of Sensors* 2016 (Nov. 30, 2015), e3632943. ISSN: 1687-725X. DOI: 10.1155/2016/3632943. URL: https://www.hindawi.com/journals/js/2016/3632943/abs/ (cit. on p. 100).

[64] Wai Yeung Yan, Ahmed Shaker, and Nagwa El-Ashmawy. "Urban Land Cover Classification Using Airborne LiDAR Data: A Review". In: *Remote Sensing of Environment* 158 (Mar. 1, 2015), pp. 295–310. ISSN: 0034-4257. DOI: 10.1016/j.rse.2014.11.001. URL: http://www.sciencedirect.com/science/article/pii/S0034425714004374 (cit. on p. 105).

[65] Zhishuang Yang et al. "A Convolutional Neural Network-Based 3D Semantic Labeling Method for ALS Point Clouds". In: *Remote Sensing* 9.9 (Sept. 10, 2017), p. 936. DOI: 10.3390/rs9090936. URL: http://www.mdpi.com/2072-4292/9/9/936 (cit. on p. 105).

[66] Jun Yue et al. "Spectral-Spatial Classification of Hyperspectral Images Using Deep Convolutional Neural Networks". In: *Remote Sensing Letters* 6.6 (June 3, 2015), pp. 468–477. ISSN: 2150-704X. DOI: 10.1080/2150704X.2015.1047045. URL: http://dx.doi.org/10.1080/2150704X.2015.1047045 (cit. on p. 101).

[67] Wenzhi Zhao and Shihong Du. "Spectral-Spatial Feature Extraction for Hyperspectral Image Classification: A Dimension Reduction and Deep Learning Approach". In: *IEEE Transactions on Geoscience and Remote Sensing* 54.8 (Aug. 2016), pp. 4544–4554. ISSN: 0196-2892. DOI: 10.1109/TGRS.2016.2543748 (cit. on p. 101).

[68] Wenzhi Zhao et al. "On Combining Multiscale Deep Learning Features for the Classification of Hyperspectral Remote Sensing Imagery". In: *International Journal of Remote Sensing* 36.13 (July 3, 2015), pp. 3368–3379. ISSN: 0143-1161. DOI: 10.1080/2150704X.2015.1062157. URL: http://dx.doi.org/10.1080/2150704X.2015.1062157 (cit. on p. 100).

Chapter

5 Multi-modal semantic segmentation

"I can see nothing," said I, handing it back to my friend.
"On the contrary, Watson, you can see everything. You fail, however, to reason from what you see. You are too timid in drawing your inferences."

— Arthur Conan Doyle (The Adventure of the Blue Carbuncle, 1892)

Contents

Summary:

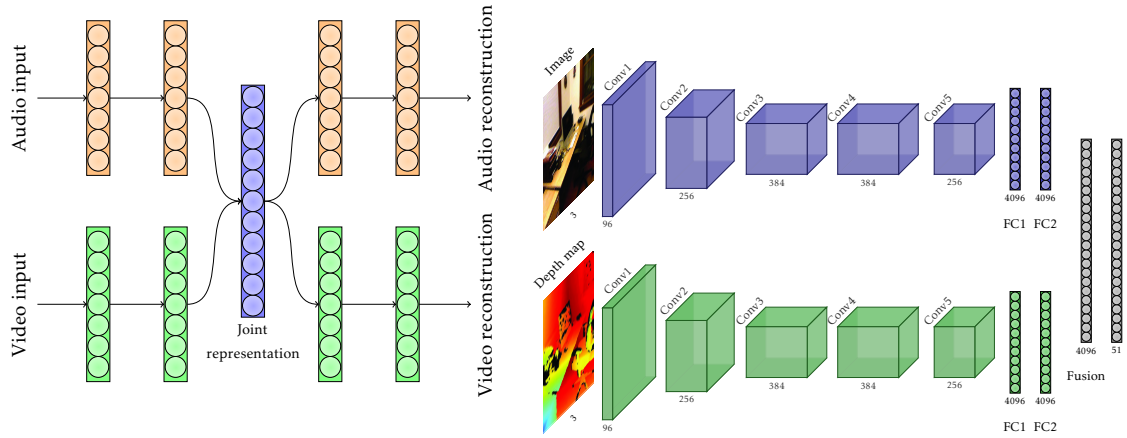
In the previous chapters, we pushed further the state of the art in semantic segmentation of remote sensing images based on the most common optical sensors deployed for Earth Observation. However we also noted that digital surface models derived from Lidar data were not enough to perform a comprehensive semantic mapping.

In this chapter, we investigate data fusion techniques to combine information extracted from heterogeneous sensors. More specifically, we introduce several multi-modal deep learning architectures that are able to learn jointly from optical images and digital surface models. We validate succesfully these models on several public datasets and show that it is indeed possible to leverage the specific strengths of multiple sensors inside one model.

We then extend these works to non-physical data inputs. Especially we apply the same data fusion architectures to existing geographical knowledge coming from online crowdsourced databases to reinforce the semantic maps generated by the model. We process this ancillary information as a virtual sensor that we feed to our previously developed multimodal architectures to inject prior knowledge in the learning and inference processes of our neural networks.

5.1 Multi-modal learning

5.1.1 Neural networks for multi-modal learning



(a) Multi-modal autoencoder [29]

(b) Multi-modal CNN [8]

Figure 5.1: Examples of deep networks with a multi-modal architectures.

The deep networks that we have worked with until now were mono-modal and processed a unique input. However the human sensory perception leverages multiple modalities, notably sound and image, that interact and complement one another. This questions the ability of the machine to mimick to behaviour by learning from multiple data sources that can exhibit heteregeneous shapes and proprties, with partial redondancies.

This topic relates to multi-modal learning and is far from specific to remote sensing. Representation learning from heterogeneous signals is a large research field. Baltrušaitis, Ahuja, and Morency [5] introduced the following taxonomy that defines joint representations (one representation for multiple modalities) or synchronized (every modality has its own representation, orchestrated and combined at a higher level):

- Repreentation: generating features that leverage the complementarity and redondancy of multiple modalities.
- Translation: converting from one modality to another.
- Alignement: finding correspondancies between the representations of several modalities.
- Fusion: including multiple modalities in the decision process.

A model able to summarize visual and sound information to label video sequences could be implemented in various ways. A first implementation would rely on extracting separate features for sound and image and then use a classifier that processes the concatenation of both features (fusion). Another implementation could add an explicit constraint on the audio and visual features to enforce similarity of audio and image frames at the same timestamp (alignment) using a distance penalty. On the opposite, one modality can be used as reference for domain adaptation algorithms that would project the audio features into the image latent space, or conversely (translation). Finally, instead of working with separate representations, it is possible to directly learn on the full video data without separating its two modalities to generate a unique multi-modal feature (representation).

Many tasks are related to multi-modal learning: automatic image captioning [18], video classification [19] or activaty recognition based on wearable trackers [32]. Many multi-modal datasets have been published for a large diversity of tasks: image captioning [14],

medical prognosis based on multiple scanners [27], action recognition in images and 3D data [31], emotion detection and recognition in videos [35], even alongside heartbeat and neural measurements [34].

One of the first multi-modal method designed for data fusion looked into the so-called "late" fusion which intervenes only at the classification stage. In its simplest expression, one statistical model is learnt for each modality and specific combination algorithms are used to create a multi-modal ensemble of models. A simple ensemble can, through a linear combination or a majority vote, take into accounts all data, as heterogeneous they may be. This is the approach from Yuhas, Goldstein, and Sejnowski [44] and Meier, Hürst, and Duchnowski [26] for automatic syllable recognition in videos. These techniques are also found in recent literature on video processing, e.g. [30] use a hidden Markov chain for automatic speech recognition.

However the true strength of deep learning is the expressiveness of learnt representations. Therefore [29] investigated the design of an autoencoder bi-modal DBN for sound and image data. Two parallel encoders process each channel separately and converge into a shared representation. Two decoders are tasked to reconstruct each modality based on the same representation as illustrated in the Fig. 5.1a. An intersting idea in this design is that the shared representation can be used to alleviate the loss of an input. For example their model can be used to infer a phonem based on the image only, or the sound only. Srivastava and Salakhutdinov [41] introduced a similar architecture built using deep Botlzmann machines that can be applied to videos, but also to very heterogeneous data such as images (raw data) and descriptive labels (symbolic language). This model allows them to infer one modality from the other when it is missing. More recently, Simonyan and Zisserman [38] introduced dual-stream networks for action recognition in videos based on sound and image modalities.

The democratization of robust, reliable and moderately cheap RGB-D sensors (such as the Kinect camera) motivated the computer vision community to look into the RGB color image and depth map data fusion, i.e. to 2.5D image processing. Although one could simply concatenate features from pretrained models applied on both color and grayscale images to generate artificial multimodal features [36, 20], it seems more effective to find a way to learn automatically an efficient joint representation that leverages complementarities between the inputs. Eitel et al. [8], Guo, Wang, and Chen [11], and Song, Jiang, and Herranz [40] have been inspired by the multi-modal autoencoder from Ngiam et al. [29] and introduce two parallel CNN that extract features fused in a joint vector by the last layers. This allows them to directly classify RGB-D images using an end-to-end model that does not separate depth and color data. This architecture is illustrated by the Fig. 5.1b. In practice there are two AlexNet models trained in parallel to extract features from the RGB image and the depth map encoded in a 3-channels image. The features extracted by the convolutional part of both AlexNets converge using the same technique as Ngiam et al. [29]: they are merged in a fully connected layer and fed to a multilayer perceptron that performs the classification. This approach improves the accuracy of convolutional classifiers compared to working on the RGB image alone. Indeed the depth information introduces geometrical information useful to determine semantics but also reduces the impact of occlusions.

The FuseNet architecture introduced by Hazirbas et al. [12] is the logical extension of this architecture to semantic segmentation. Applied on RGB-D images, FuseNet is a variant of SegNet [4] that we already detailed in the Chapter 3. Two encoders perform a dense feature extraction on the color image and the depth map encoded on 3 channels. A unique decoder performs simultaneously both the upsampling and the dense pixel-wise classification. This architecture improved the state of the art on the SUN RGB-D datase dataset, dedicated to semantic segmentation of 2.5D indoor images. Guerry, Le Saux, and Filliat [10] also obtained state of the art performance for person detection in RGB-D images using bi-modal learning strategies where both encoders exchange information from the two modalities. Finally, Lee, Park, and Hong [21] suggested an enhancement to FuseNet by introducing residual learning

inside the network, once again increasing the accuracy on the SUN RGB-D dataset.

An interesting observation stemming for this short review of the state of the art is that RGB-D classification models all process separately color and depth information. Indeed, as we have seen for multispectral data, leveraging ImageNet (and therefore color) pretrained models generally outperforms training from scratch. Concatenating the depth map to the color image to form a 4-channels tensor that would be fed to the model seems to result in worse results than using this dual-stream approach[1].

5.1.2 Multi-modal learning in remote sensing

Working with digital surface models to improve classifier accuracy in remote sensing is not a novel research topic and has been investigated in the past. It is quite close conceptually to 2.5D RGB-D image processing, as DTM plays a role similar to the depth maps.

Nonetheless most works published before this thesis have employed *ad hoc* fusion strategies. For example, Lagrange et al. [20] simply concatenate features extracted by multiple deep networks to train an SVM classifier and so do Paisitkriangkrai et al. [33] with random forests and expert features. More recently, Liu et al. [23] used the same features but fed them to a CRF graphical model in order to combine semantic maps inferred by a FCN, digital surface model and NDVI.

In this manuscript we look into deep learning end-to-end approaches that do not rely on graphical models or expert features. We will begin optical/digital surface model fusion and later move on to the integration of prior geographical knowledge.

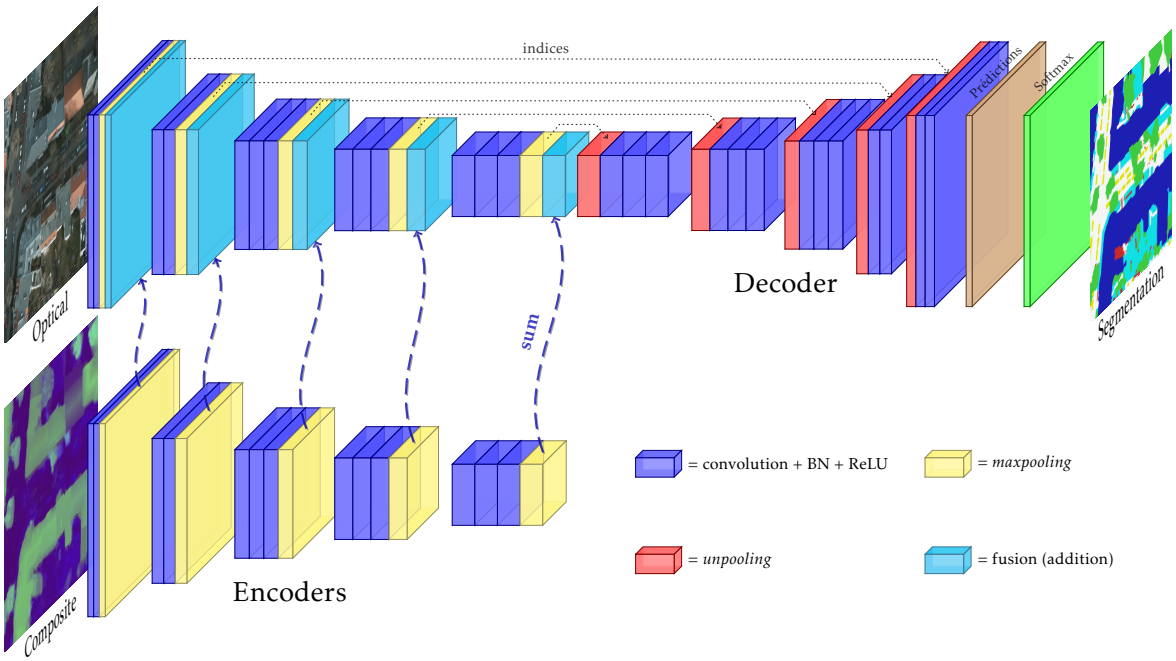


Figure 5.2: FuseNet architecture [12].

5.2 Model fusion

5.2.1 Learning-based fusion

The FuseNet [12] architecture is a multi-modal SegNet variant, as illustrated by Fig. 5.2. FuseNet encodes simultaneously the RGB image and the depth map using two identical

[1] Or at least it would seem in the state of the art. Preliminary experiments tend to confirm this, yet few papers have published negative results on this matter.

encoders. The intermediate feature maps from the depth encoder are added to the feature of the color encoder after each convolutional block. A unique decoder then performs the upsampling and classification. This approach can adapted to many other CNN, such as ResNet.

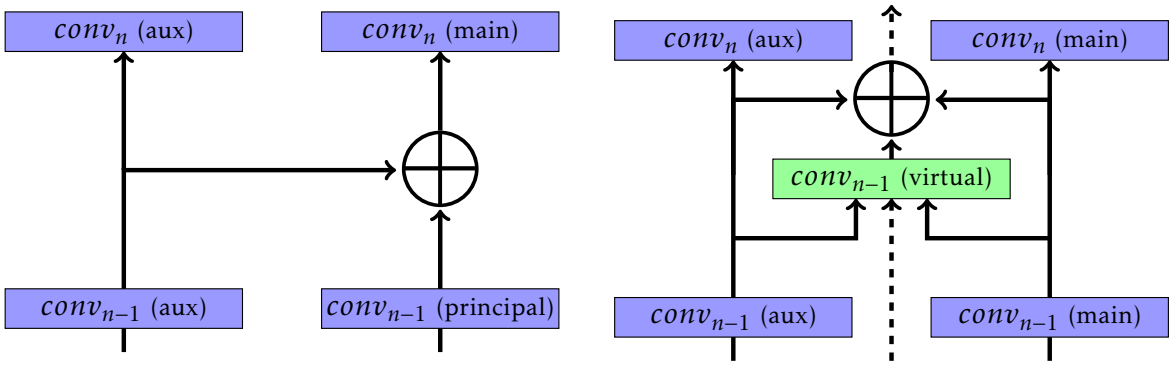
Formally, let $\hat{\mathrm{P}}$ denote the prediction function modeled by FuseNet applied to an image I and a depth Δ. Let $\mathcal{D}$ be the decoder and $\mathrm{E}_i^{\mathrm{I}}, \mathrm{E}_i^{\Delta}$ the outputs of the i^{th}encoding block for the image and depth, and $\mathcal{B}_i$ the operation corresponding to the i^{th}block. Then:

$$\hat{\mathrm{P}}(\mathrm{I}, \Delta) = \mathcal{D}\left(\mathrm{E}_5^{\mathrm{I}}(\mathrm{I}, \Delta)\right) \tag{5.1}$$

where

$$\begin{cases} \mathrm{E}_{i+1}^{\mathrm{I}}(\mathrm{I}, \Delta) = \mathcal{B}_i^{I}\left(\mathrm{E}_i^{\mathrm{I}} + \mathrm{E}_i^{\Delta}\right) \\ \mathrm{E}_{i+1}^{\Delta}(\Delta) = \mathcal{B}_i^{\Delta}(\mathrm{E}_i^{\Delta}) \end{cases} \tag{5.2}$$

In our case, we can modify FuseNet in the same way we altered SegNet in the previous chapter to process remote sensing images. Indeed a heigh map such as the DSM can be processed as a depth map associated to a RGB color image. Therefore we suggest to adapt FuseNet to multi-modal remote sensing image processing. In practice we will use as inputs RGB or IRRG optical images and the composite images built in Section 4.3.2.



(a) FuseNet: auxiliary activations are added to the main stream.

(b) V-FuseNet: activation from both streams are fused using a residual convolutional block.

Figure 5.3: Fusion strategies for the FuseNet architecture.

However the FuseNet architecture considers the depth data as auxiliary. Indeed the two streams in the encoder are not symetrical: the depth stream only process depth information while the color stream learns a joint RGB-D representation. Moreover the decoder upsamples the feature maps based on the indices from the main stream, i.e. the optical data. This means that we need to choose an input that will act as the main stream and the other will act as the auxiliary stream (cf. Fig. 5.3a). There is an unbalance between the way the two modalities are processed. We suggest here a symetrical alternative that balance the FuseNet model by introducing a virtual third source.

Insteaad of adding up activation maps, we use a learnable fusion block that encode a multi-modal representation of the features. We introduce a third encoder that does not match any actual modality but only serves as a proxy for the joint multi-modal representation. After the n^{th}encoding block, the virtual encoder concatenate the feature maps from both actual encoders and feeds them to a residual convolutional block that extracts the joint multi-modal features. These features are the ones that are finally upsampled and classified in the decoder. This process is detailed in the Fig. 5.3b. This strategy makes FuseNet symetrical and removes the need to choose a "main" data source. In the taxonomy defined by Baltrušaitis, Ahuja, and Morency [5], this matches a switch from an alignment method to a representation-based one. This architecture is denoted *V-FuseNet* in the rest of this chapter.

Another drawback of the FuseNet architecture is that it requires topologically compatible models, i.e. networks with similar computational graphs so that the activations can be summed in the encoder. This is not always true, especially if the inputs were to have very different natures, such as a 2D image and 3D point cloud. Here, the depth map can be resampled at the same resolution as the color image. These matching encoders might waste weights, especially if one data source is less rich in information than the other. Therefore, we also introduce a late fusion strategy that could learn from any kind of input using heterogeneous models. It is based on a late fusion paradigm instead of the multi-modal representation learning.

5.2.2 Residual correction

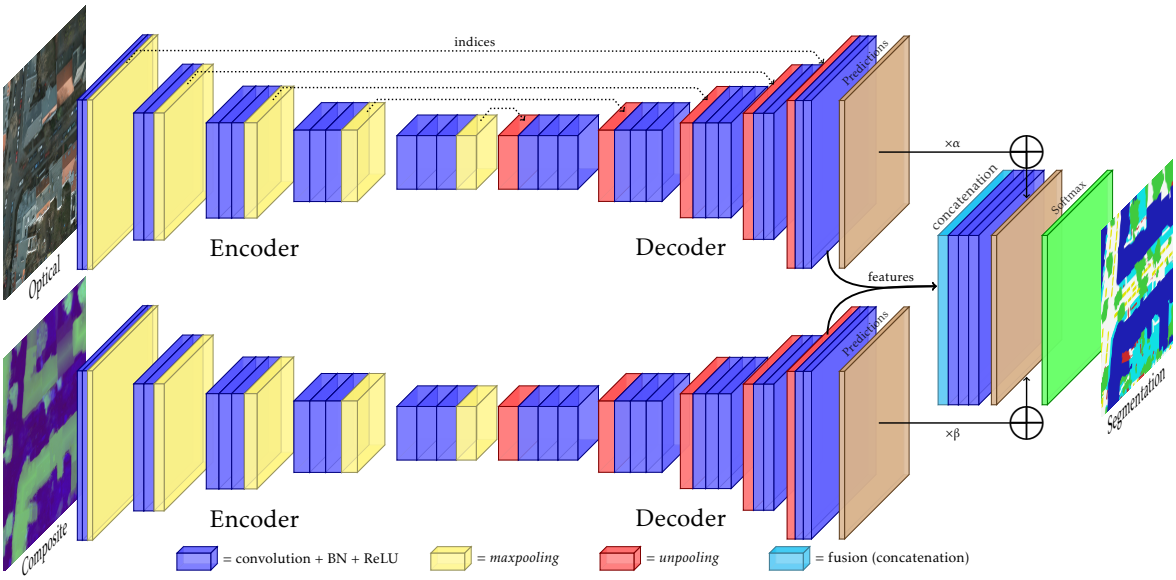


Figure 5.4: Residual correction applied to SegNet.

An alternative approach to the fusion problem consists in processing separately both modalities and combine the set of predictions inferred by the model ensemble. We can, for example, train one deep network per data source and then average the output maps. However this fusion scheme does not really take into account the specificities of the sensors. We introduce a trainable residual correction module that takes as inputs the last feature maps from the single-modality networks and learns to fuse the probability maps [1]. The residual correction module learns a correction ϵ that is applied to the average prediction to increase the overall accuracy of the model ensemble. This process is detailed in the Fig. 5.4 on the SegNet architecture.

This module performs a decision-level fusion using the residual learning principle [13]. It consists in three convolutional layers with 3×3 kernels and 1 px padding. The intermediate activation maps coming out from the two SegNet decoders are concanetated and fed as an input to the correction module (cf. Fig. 5.4). The model output is summed in a residual fashion with the average of the two predictions comig from the SegNet models, as shown in the Fig. 5.5. Residual learning is well-suited to this work since the averaged maps should be already quite close to the expected result. The additional module fuses the decisions using an adaptive weighting that depends on the activation maps and compute an corrective term to apply on the average prediction to shift it towards the ground truth. The fusion network is trainable using backpropagation and can be trained end-to-end with to the two networks, or more simply in our case by posterior fine-tuning with frozen models. The latter allows for a very fast training since gradients are computed only for the fusion module. Training

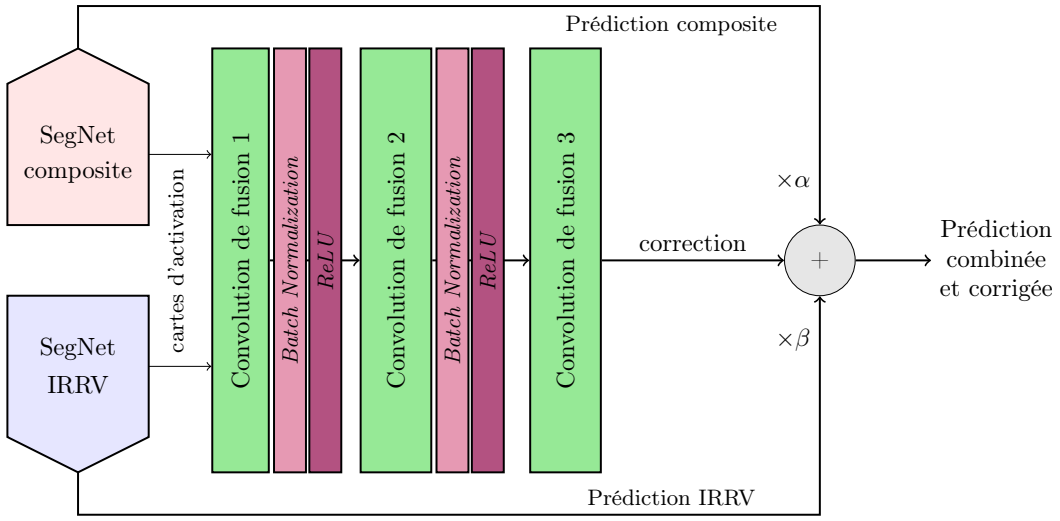


Figure 5.5: Residual correction module.

the whole model ensemble end-to-end can be costly as it requires to store SegNet twice plus the small residaul correction module, which is not always possible on all GPUs. This multi-modal SegNet using residual correction is denoted *SegNet-RC* in the rest of the section.

Let P_{gt} be the ground truth tensor and $\hat{\mathrm{P}}_i$ be the predictions of the i^{th}model. We define the error ϵ_i as:

$$\hat{\mathrm{P}}_i = \mathrm{P}_{gt} + \epsilon_i \quad \text{avec} \quad |\epsilon_i| \ll |\hat{\mathrm{P}}_i| \ . \tag{5.3}$$

If the prediction P_i is close to the ground truth, then ϵ_i has a low magnitude. The goal of the residual correction module is to approximate this error to correct it during inference.

Let n be the number of predictions to fuse using residaul correction. Then the module output, noted $\hat{\mathrm{P}}^*$, is the sum of the average predictions from the $\hat{\mathrm{P}}_i$ and a corrective term c:

$$\hat{\mathrm{P}}^* = \hat{\mathrm{P}}_{average} + c = \frac{1}{n}\sum_{i=1}^{n} \mathrm{P}_i + c = \mathrm{P}_{gt} + \frac{1}{n}\sum_{i=1}^{n} \epsilon_i + c \ . \tag{5.4}$$

Since the residual correction module is trained to minimize its cost function, it actually comes to:

$$\left\| \hat{\mathrm{P}}^* - \mathrm{P}_{gt} \right\| \to 0 \tag{5.5}$$

which can be interpreted as a constraint on c and ϵ_i :

$$\left\| \frac{1}{n}\sum_{i=1}^{n} \epsilon_i - c \right\| \to 0 \ . \tag{5.6}$$

Another way to frame this is to consider the residual correction module as a way to compensate for the average error of the model ensemble. During the training phase, the ground truth P_{gt} is known. The module weights are optimized by backpropagation so the correction c gets close to $\frac{1}{n}\sum_{i=1}^{n} \epsilon_i$. As we expect the average error to be small, the error correction actually matches the residual learning paradigm [13]. Indeed, c is an additive term of small magnitude added to the initial signal using a bypass. This approach is schematized in the Fig. 5.5.

5.2.3 Experimental results

As expected both fusion schemes improve the classification accuracies on the two datasets, as illustrated in the Figs. 5.6a and 5.7. Detailed quantitative results are reported in Tables 5.1 to 5.3. As in the Chapter 3, using ResNet-34 as a base model in place of SegNet does

Table 5.1: Multi-modal semantic segmentation results on the ISPRS Vaihingen validation set.

Model	Accuracy	F_1 score
SegNet (IRRG)	90.2± 1.4	89.3± 1.2
SegNet (composite)	88.3± 0.9	81.6± 0.8
SegNet-RC	90.6± 1.4	89.2± 1.2
FuseNet	90.8± 1.4	90.1± 1.2
V-FuseNet	**91.1**± 1.5	**90.3**± 1.2
ResNet-34 (IRRG)	90.3± 1.0	89.1± 0.7
ResNet-34 (composite)	88.8± 1.1	83.4± 1.3
ResNet-34-CR	90.8± 1.0	89.1± 1.1
FusResNet	90.6± 1.1	89.3± 0.7

Table 5.2: Multi-modal semantic segmentation results on the ISPRS Vaihingen test set (multi-modal approaches). Best results are in **bold** and second best are in *italics*.

Model	Roads	Buildings	Low veg.	Trees	Vehicles	Accuracy
FCN+CRF + contours + fixed nDSM [25]	92.4	95.2	83.9	89.9	81.2	90.3
SegNet (IRRG)	91.5	94.3	82.7	89.3	85.7	89.4
SegNet-RC	91.0	94.5	84.4	89.9	77.8	89.8
FuseNet	91.3	94.3	84.8	89.9	85.9	90.1
V-FuseNet	91.0	94.4	84.5	89.9	86.3	90.0

not significantly improve the performances and the gain is not justified compared to the additional computational burden. The Fig. 5.6 shows several objects that have been wrongly classified based on the optical image alone, for which the multi-modal learning including the composite data generates correct maps. For example, in the Figs. 5.6a and 5.6b, the SegNet model is not able to separate between the classes "impervious surface" and "building". Indeed the roof of the parking garage is used an open-air parking lot and is visually similar to usual parking lots (cars, white markings on the ground). FuseNet is able to leverage the nDSM to choose the "building" class but completely ignore the vehicles. In the meantime the residual correction preserves part of the spatial information of the "cars" class. This is similar to the Fig. 5.6c in which SegNet confuses roads with buildings and low vegetation with trees while both fusion strategies predict accurately the various objects, mainly thanks to the nDSM. Overall it seems that FuseNet multi-modal encoding scheme results in multi-modal internal representation that use less parameters than the residual correction of a model ensemble and converge to an overall better accuracy. On the opposite the late fusion using residual correction mostly increases the classification metrics on the "impervious surfaces" and "buildings" classes with a smaller overall impact.

Yet, a practical strength of the residual correction is that it learns to combine predictions based on the activation magnitude. In the Fig. 5.6b, we show an example of successful fusion in which the confusion of the model in IRRG around the cars is compensated for by the high confidence in the "building" class of the composite model.

The FuseNet architecture learns a joint representation of the two inputs but doing so, it becomes more sensitive to the overfitting that plagues SegNet and deep models in general. Rare occurrences such as cars on top of a buildings are ignored. The multi-modal representation from the dual stream encoders perform better overall but FuseNet might also incorporate more of the instrinsic dataset bias, where the residual correction was able to correct errors even on those edge cases. Late fusion is therefore more relevant when

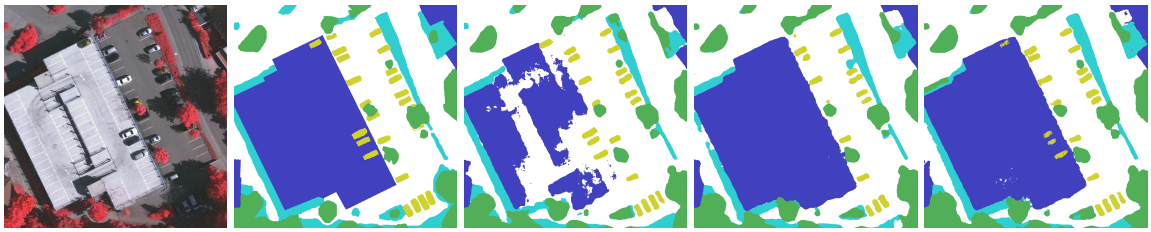
Table 5.3: Multi-modal semantic segmentation results on the ISPRS Potsdam test set (multi-modal approaches). Best results are in **bold** and second best are in *italics*.

Model	Roads	Buildings	Low veg.	Trees	Vehicles	Accuracy
FCN + CRF + expert features [23]	91.2	94.6	85.1	85.1	92.8	88.4
FCN [37]	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-RC	**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**

one needs to combine complementary predictions and acts more a stronger adapatively weighted average scheme. On the parking garage, the composite SegNet predicts a building with confidence because the nDSM is very reliable, while the RGB SegNet produces high confidence predictions on cars but mixed predictions around them because the spatial context is ambiguous. On the contrary, FuseNet overfits on the "cars are on roads" prior and vehicles disappear completely at test time, because this is an unique case in the dataset on which the model was not able to generalize to. To conclude, both fusion strategies can be applied but not for the same purpose. Late fusion using residual correction is more useful to combine complementary strong classifiers, while the FuseNet strategy is more suited to leverage ancillary information in the learning process. On the final test set from the Vaihingen dataset (cf. Table 5.2), the V-FuseNet strategy has marginally better accuracies compared to the original FuseNet. However let us stress that the F_1 scores are significantly higher on several classes, especially the "clutter" class (+1,7%) which is not taken into account in the overall accuracy. On the ISPRS Potsdam dataset, V-FuseNet slightly outperforms FuseNet both class-wise and overall.

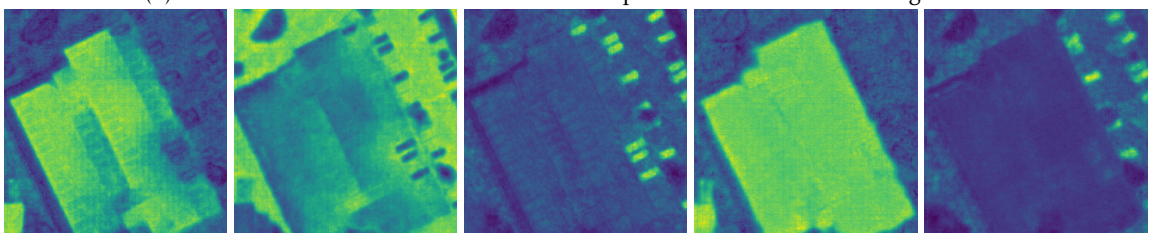
Robustness to missing data

The multi-modal architectures we introduced allow us to benefit from various georeferenced and co-registered images. However they also add some new constraints regarding data availability. Indded despite the high density of the Lidar point cloud published alongside the ISPRS dataset, the normalization used to generate the height map is not perfect and there are some remaining artifacts. More specifically, Marmanis et al. [25] identified that several buildings have disappeared innDSM, as the corresponding pixels have been asigned a height of 0 m. This provokes significant classification errors for both fusion methods as illustrated in the Fig. 5.8. One solution, suggested by [25], consists in manually fixing the nDSM but this does not scale on very large datasets. Robust multi-modal techniques to deal with noisy or even missing data could help solve problem, for example based on *hallucination networks* [15] to generate the missing information [17]. Alternatively, recent works on generative models could also help reduce overfitting and improve the overall robustness of the models by training them on noisy synthetic datasets [43].



IRRG image | Ground truth | SegNet | FuseNet | SegNet-RC

(a) Predictions from various models on a sample tile of the ISPRS Vaihingen dataset.



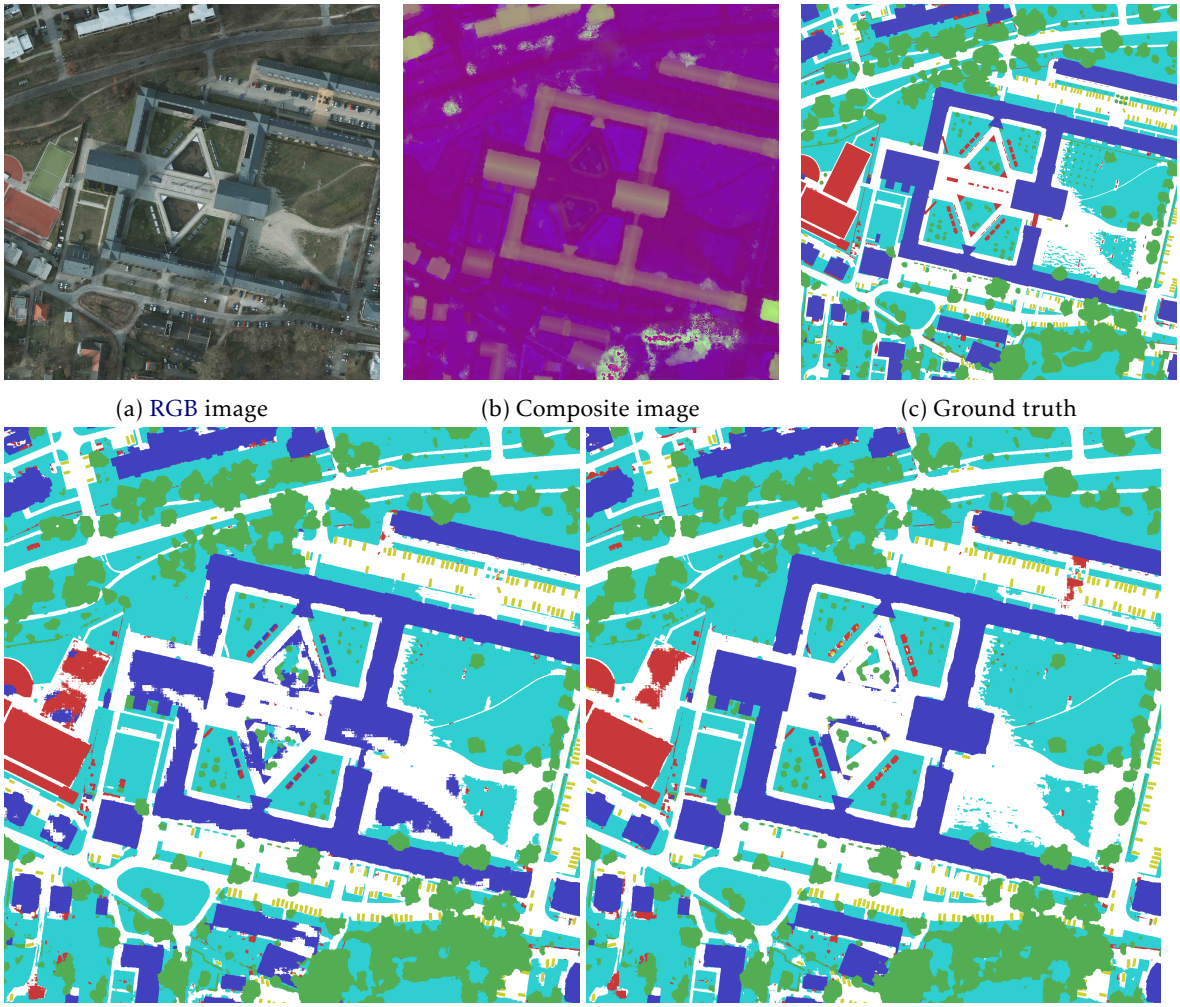
SegNet IRRG (confidence, buildings) | SegNet IRRG (confidence, roads) | SegNet IRRG (confidence, cars) | SegNet composite (confidence, buildings) | SegNet composite (confidence, cars)

(b) Confidences maps from SegNet for several classes based on the input source.

IRRG image | Ground truth | SegNet | FuseNet | SegNet-RC

(c) Predictions from various models on a sample tile of the ISPRS Vaihingen dataset.

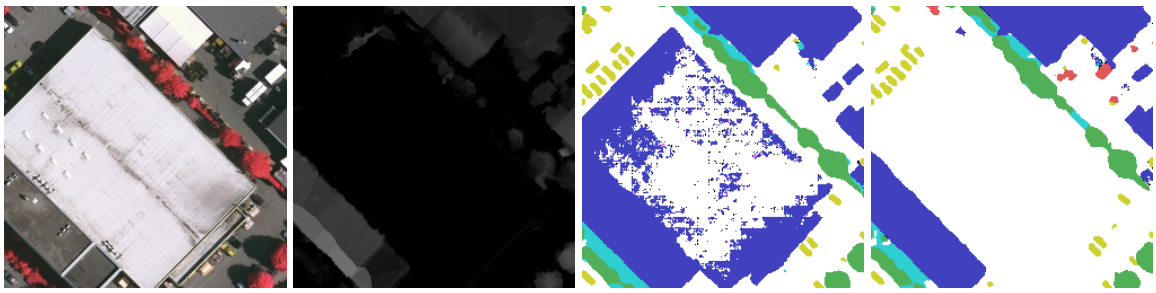
Figure 5.6: Samples of successful multi-modal predictions on Vaihingen.
Colors: white: roads, blue: buildings, cyan: low vegetation, green: trees, yellow: vehicles, red: clutter.



(a) RGB image (b) Composite image (c) Ground truth

(d) SegNet (e) V-FuseNet

Figure 5.7: Impact of the fusion strategy on a sample tile of the ISPRS Potsdam dataset. Confusion between roads and buildings is greatly reduced thanks to the digital surface models.
Colors: white: roads, blue: buildings, cyan: low vegetation, green: trees, yellow: vehicles, red: clutter.



IRRG nDSM SegNet-RC V-FuseNet

Figure 5.8: Errors in the nDSM are mishandled by both fusion methods on the ISPRS Vaihingen dataset. In this case an entire building is erased from the map.

5.3 Learning from prior knowledge

In the previous section we introduced deep multi-modal architectures that could be trained using heterogeneous sensors. However geospatial data are not always acquired by physical means. There are multiple semantic knowledge databases, backed by institutions, non-profit or companies, containing geographical information that could be helpful for automated cartography.

5.3.1 OpenStreetMap

OpenStreetMap (OSM) is a free crowdsourced GIS managed by volunteers who contribute their time to map locations that they are familiar with. Contributors to OpenStreetMap (OSM) can use an online editor to annotate georeferenced maps by adding or updating semantic nodes related to the road network, building footprints, parks, forests, rivers etc. In addition to these manuel edits, the OSM community pulls data from official sources such as the French "cadastre" to automatically update the boundaries of administrative geographical entities or to generate new up-to-date building footprints. OSM is the largest online geographic database under a free and open license. It regroups a large ontonlogy of geographic objects and entities from highways to leisure parks, churches, cimeteries and agricultural lands.

Surprsingly few works have looked into using OSM data for machine learning since the website opened in 2004. In most cases OSM is used as a target ground truth for roads and buildings detection [28, 24] in a supervised learning setting or sometimes for automatic registering of satellite images [42]. Isola et al. [16] investigated automatic generation of OSM tiles using satellite data but only for visualization purposes, without any accuracy assessment. Yet OSM contains extremly and diversified data that could help extracting abstract knowledge. Chen and Zipf [6] designed active learning strategies to automatically detect objects that were not annotated in OSM but existed in the images to suggest fixes to contributors. Danylo et al. [7] used random forest classifiers on various OSM data layers to predict local climactic zones, while Geiß et al. [9] used OSM data to detect areas prone to natural disasters.

In this section we suggest to use the semantic knowledge contaiend in OSM as an input for a semantic segmentation deep network. The idea is to leverage the semantic knowledge, even if noisy or partial, from OSM to extract richer information at a higher resolution by combining OSM and VHR optical data. Indeed this approach goes further than the usual mapping image $\rightarrow$ OSM. On the contrary we want here to combine multiple data sources in a multi-modal learning setting and whether data are images or GIS does not matter.

To do so we will use the ISPRS Potsdam dataset. In addition to the existing images tiles, we also collect the relevant OSM from 2017. We select the layers corresponding to the roads, the building footprints, water bodies and urban vegetation (mostly parks). Roads are defined in OSM as a collection of linear elements. During rasterization, we assign to each segment a fixed width depending on the road type ($\simeq$3.5 m for each lane in an urban area). Moreover buildings, greenways and water bodies do not necessarily correspond completely to the aerial images from the dataset, either because they have been built recently (images were acquired in 2014, OSM data in 2017) or because the OSM data is simply incomplete. We generate a 2D raster with the same resolution than the RGB images, with 4 binary channels: a road mask, a building mask, a water mask and a "green" (green infrastructure, greenways, forests) mask.

5.3.2 Prior knowledge as a virtual sensor

The core of our approach consists in processing OSM data as a virtual sensor, i.e. a new data input that complements the optical images we have at hand. The raster generated from OSM data is an incomplete information but we expect it to facilitate the training process since

RGB image | OSM tile | OSM raster | Ground truth

Figure 5.9: Tile #4_12 from the ISPRS Potsdam dataset and corresponding OSM data.
Colors: white: roads, blue: buildings, cyan: low vegetation, green: trees, yellow: vehicles, red: clutter.

there is a significant overlap with the ground truth. It should be beneficial to the network to learn how to find buildings and roads not on the optical image alone, but on both data sources to rely on the already existing OSM annotations. Less weights will be required to find buildings from scratch, and the newly freed parameters will be available for harder cases. Instead of performing of a full segmentation of the whole image, the model will be able to locally enrich the existing geopgrahical knowledge based on the color images, contrasting with the usual pipeline. Therefore we apply the FuseNet and residual correction multi-modal learning strategies on our two inputs: optical images and rasterized OSM layers.

If the classes of interest we are looking for in the segmentation ground truth are already annotated in the OSM data (e.g. buildings and roads), it is feasible to use those as a first approximation of the ground truth. It will then be possible to refine them to correct small inaccuracies due to OSM contributors or even predict missing classes and objects. This process is similar to residual learning, but first and foremost to the refining networks from [22], both known to improve the performance of CNNs and FCNs classifiers.

In our case we use a simple FCN using one convolutional block from the VGG-16 [39] model to convert the OSM raster into a semantic map approximating the expected ground truth. This model is denoted OSMNet. The color images are processed using an FCN based on the SegNet architecture [4] following the approach designed in Chapter 3. We use these two models to generate an average prediction map by combining both inputs. In that case, if I is the input colour image, $\mathcal{O}$ the OSM raster, $\hat{\mathrm{P}}_{image}$ the SegNet prediction function and $\hat{\mathrm{P}}_{OSM}$ the OSMNet prediction function, then the average prediction function $\hat{\mathrm{P}}$ is obtained by:

$$\hat{\mathrm{P}}(\mathrm{I},\mathcal{O}) = \frac{1}{\alpha+\beta}(\alpha \cdot \hat{\mathrm{P}}_{image}(\mathrm{I}) + \beta \cdot \hat{\mathrm{P}}_{OSM}(\mathcal{O})) \,. \tag{5.7}$$

Since OSM already contains a significant part of the expected information, we expect that $\hat{\mathrm{P}}_{OSM}(\mathcal{O})$ is a close approximation of the ground truth P_{gt}. $\hat{\mathrm{P}}_{image}$ can be written as a refining function [22]:

$$\left\| \hat{\mathrm{P}}_{image}(\mathrm{I}) \right\| \propto \left\| \mathrm{P}_{gt} - \hat{\mathrm{P}}_{OSM}(\mathcal{O}) \right\| \ll \left\| \mathrm{P}_{gt} \right\| \,. \tag{5.8}$$

Moreover this can also be rewritten as a residual correction module C. Indeed, given Eq. (5.4), the prediction $\hat{\mathrm{P}}^*$ after residual correction is:

$$\hat{\mathrm{P}}^*(\mathrm{I},\mathcal{O}) = \hat{\mathrm{P}}(\mathrm{I},\mathcal{O}) + \mathrm{C}\left(\mathrm{Z}_{image}, \mathrm{Z}_{OSM}\right) \,, \tag{5.9}$$

where Z_{image} and Z_{OSM} are the last feature maps from SegNet and OSMNet respectively.

In this pipeline the residual learning process is used to model a corrective signal to be added to the average prediction, as illustrated in Fig. 5.10. The refined OSM map is then refined again by the residual learning process in a two-steps iterative correction.

Similarly, we can apply the FuseNet architecture on I and $\mathcal{O}$, i.e. the colour image and the virtual OSM sensor. This requires that both encoders from SegNet and OSMNet have

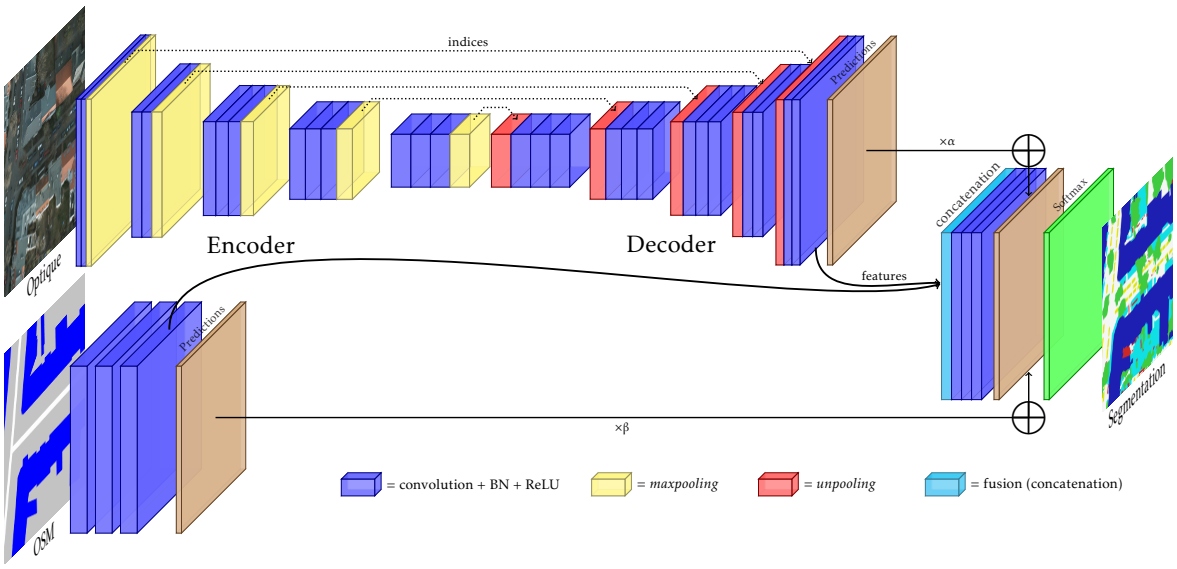


Figure 5.10: Residual correction applied to SegNet and OSMNet.

Table 5.4: Multi-modal semantic segmentation using the OSM prior on the ISPRS Potsdam dataset (class-wise F_1 scores and overall accuracy).

Method	Imp. surfaces	Buildings	Low veg.	Trees	Vehicles	Accuracy
RF IRRGB	77.0	79.7	73.1	59.4	58.8	74.2
SegNet RGB	93.0	92.9	85.0	85.1	95.1	89.7
RF IRRGB+OSM	85.6	92.4	73.8	59.5	67.6	80.9
RC RGB+OSM	93.9	92.8	85.1	**85.2**	95.8	90.6
FuseNet	**95.3**	**95.9**	**86.3**	85.1	**96.8**	**92.3**

identical graphs to ensure compatible shapes when summing the activation tensors during fusion, as explained in the Section 5.2.

5.3.3 Multi-modal architecture for geographic knowledge

We use the ISPRS Potsdam dataset for which we download the relevant OSM data (cf. Fig. 5.9). Seeing that the tiles embed geographic coordinates, we can generate the corresponding OSM rasters comprised of the roads, buildings, vegetalized areas and water bodies footprints using the Maperitive software[2]. We use a 3-fold cross-validation of the dataset to validate empirically our findings.

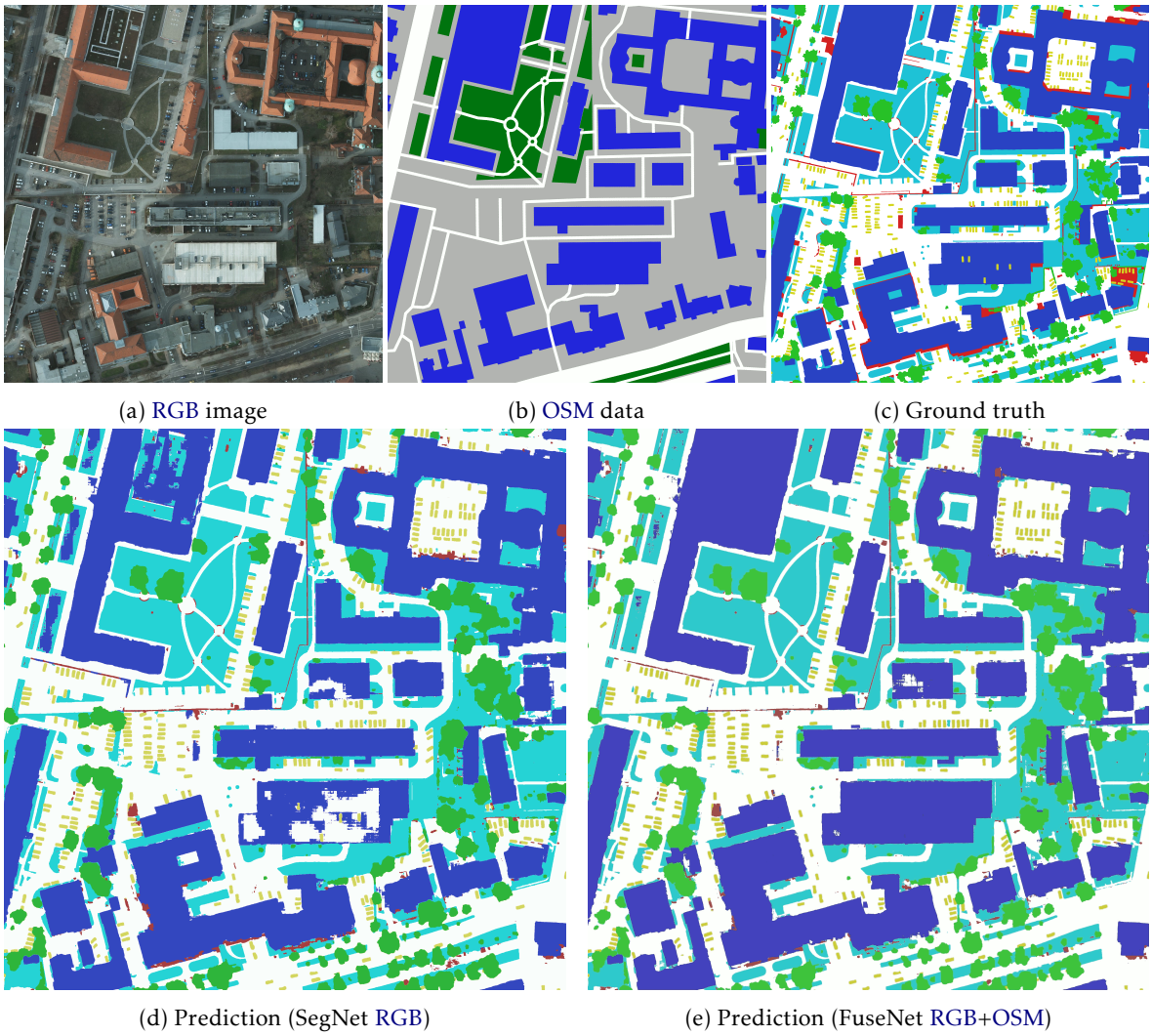
Experimental validation

Once again we reuse the hyperparameters described in the Chapter 3. Results from the multi-modal models are compared to a baseline using a random forest (FR) on the image after superpixel segmentation. The baseline use histograms of oriented gradients and histograms of colors as image features and the histogram of classes as OSM feature.

Results obtained by cross-validation on the ISPRS Potsdam dataset are reported in the Table 5.4. We report metrics defined in the Section 3.3, i.e. the overall accuracy and F_1 score for each class on the eroded ground truth.

As expected the inclusion of OSM data significantly improves the classification accuracy, especially on roads and buildings which benefit the most from the geographical information.

[2] http://maperitive.net/



(a) RGB image (b) OSM data (c) Ground truth

(d) Prediction (SegNet RGB) (e) Prediction (FuseNet RGB+OSM)

Figure 5.11: Segmentation sample on a tile from the ISPRS Potsdam dataset including the OSM information. Errors on buildings are significantly reduced.
Colors: white: roads, blue: buildings, cyan: low vegetation, green: trees, yellow: vehicles, red: clutter.

Indeed this additional information (whether or not an OpenStreetMap contributor annotated the area as a building) can remove ambiguities harder to understand based on the visual appearance alone. Moreover it is interesting to see that even classes that not directly represented in OSM such as the various vegetation types and the cars can also benefit from the additional contextual information.

In addition, including OSM data in the learning phase also speeds up the training. On the same dataset, SegNet trained using a refinement strategy from OSM requires approximately 25% less iterations to converge than the usual SegNet RGB. Moreover the local minima it reaches is better, with a loss function at 0.39 instead of 0.45 for the same accuracy. Finally, using the OSM data generally gives more consistent outputs that have a clearer spatial structure as pictured in Fig. 5.12.

To summarize, it seems that FCNs can be adapted to the multi-modal learning pardigm. Especially we showed in this chapter that it is possible to leverage multiple input data sources, either coming from heterogeneous sensors or from knowledge GIS databses. The multi-modal information enhance the inference capabilities of the models both on qualitative and quantitative aspects on the two datasets we worked with. Finally, the multi-modal learning and data fusion strategies that we introduced can address various obstacles that one can encounter when applying machine learning for Earth Observation.

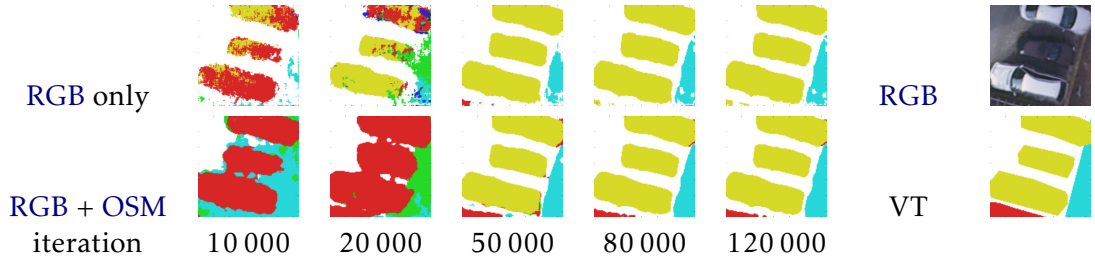


Figure 5.12: Evolution during training of the segmentation using SegNet RGB and RGB+OSM. Taking OSM into account makes the maps visually more structured.
Colors: white: roads, blue: buildings, cyan: low vegetation, green: trees, yellow: vehicles, red: clutter.

The works presented in the Section 5.1 have been published in an international journal:

- Nicolas Audebert, Bertrand Le Saux, and Sébastien Lefèvre. "Beyond RGB: Very High Resolution Urban Remote Sensing with Multimodal Deep Networks". In: *ISPRS Journal of Photogrammetry and Remote Sensing* (Nov. 23, 2017). ISSN: 0924-2716. DOI: 10.1016/j.isprsjprs.2017.11.011

The works presented in the Section 5.3 have been presented at an international conference:

- Nicolas Audebert, Bertrand Le Saux, and Sébastien Lefèvre. "Joint Learning from Earth Observation and OpenStreetMap Data to Get Faster Better Semantic Maps". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. Honolulu, United States, July 2017, pp. 1552–1560. DOI: 10.1109/CVPRW.2017.199

References

[1] Nicolas Audebert, Bertrand Le Saux, and Sébastien Lefèvre. "Semantic Segmentation of Earth Observation Data Using Multimodal and Multi-Scale Deep Networks". In: *Computer Vision – ACCV 2016*. Springer, Cham, Nov. 20, 2016, pp. 180–196. DOI: 10.1007/978-3-319-54181-5_12 (cit. on p. 120).

[2] Nicolas Audebert, Bertrand Le Saux, and Sébastien Lefèvre. "Beyond RGB: Very High Resolution Urban Remote Sensing with Multimodal Deep Networks". In: *ISPRS Journal of Photogrammetry and Remote Sensing* (Nov. 23, 2017). ISSN: 0924-2716. DOI: 10.1016/j.isprsjprs.2017.11.011 (cit. on p. 130).

[3] Nicolas Audebert, Bertrand Le Saux, and Sébastien Lefèvre. "Joint Learning from Earth Observation and OpenStreetMap Data to Get Faster Better Semantic Maps". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. Honolulu, United States, July 2017, pp. 1552–1560. DOI: 10.1109/CVPRW.2017.199 (cit. on p. 130).

[4] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Scene Segmentation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.12 (Dec. 2017), pp. 2481–2495. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2016.2644615 (cit. on pp. 117, 127).

[5] Tadas Baltrušaitis, Chaitanya Ahuja, and Louis-Philippe Morency. "Multimodal Machine Learning: A Survey and Taxonomy". In: (May 25, 2017). arXiv: 1705.09406 [cs]. URL: http://arxiv.org/abs/1705.09406 (cit. on pp. 116, 119).

[6] Jiaoyan Chen and Alexander Zipf. "DeepVGI: Deep Learning with Volunteered Geographic Information". In: *26th International World Wide Web Conference (Poster)*. ACM, 2017 (cit. on p. 126).

[7] Olha Danylo et al. "Contributing to WUDAPT: A Local Climate Zone Classification of Two Cities in Ukraine". In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 9.5 (May 2016), pp. 1841–1853. ISSN: 1939-1404, 2151-1535. DOI: 10.1109/JSTARS.2016.2539977. URL: http://ieeexplore.ieee.org/document/7447735/ (cit. on p. 126).

[8] Andreas Eitel et al. "Multimodal Deep Learning for Robust RGB-D Object Recognition". In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Sept. 2015, pp. 681–687. DOI: 10.1109/IROS.2015.7353446 (cit. on pp. 116, 117).

[9] Christian Geiß et al. "Joint Use of Remote Sensing Data and Volunteered Geographic Information for Exposure Estimation: Evidence from Valparaíso, Chile". In: *Natural Hazards* 86.1 (Mar. 1, 2017), pp. 81–105. ISSN: 0921-030X, 1573-0840. DOI: 10.1007/s11069-016-2663-8 (cit. on p. 126).

[10] Joris Guerry, Bertrand Le Saux, and David Filliat. ""Look at This One" Detection Sharing between Modality-Independent Classifiers for Robotic Discovery of People". In: *2017 European Conference on Mobile Robots (ECMR)*. 2017 European Conference on Mobile Robots (ECMR). Sept. 2017, pp. 1–6. DOI: 10.1109/ECMR.2017.8098679 (cit. on p. 117).

[11] Hengkai Guo, Guijin Wang, and Xinghao Chen. "Two-Stream Convolutional Neural Network for Accurate RGB-D Fingertip Detection Using Depth and Edge Information". In: *Proceedings of the IEEE International Conference on Image Processing (ICIP)*. 2016, pp. 2608–2612. DOI: 10.1109/ICIP.2016.7532831 (cit. on p. 117).

[12] Caner Hazirbas et al. "FuseNet: Incorporating Depth into Semantic Segmentation via Fusion-Based CNN Architecture". In: *Computer Vision – ACCV 2016*. Asian Conference on Computer Vision. Springer, Cham, Nov. 20, 2016, pp. 213–228. DOI: 10.1007/978-3-319-54181-5_14 (cit. on pp. 117, 118).

[13] Kaiming He et al. "Deep Residual Learning for Image Recognition". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Las Vegas, United States, June 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90 (cit. on pp. 120, 121).

[14] Micah Hodosh, Peter Young, and Julia Hockenmaier. "Framing Image Description As a Ranking Task: Data, Models and Evaluation Metrics". In: *Journal of Artificial Intelligence Research* 47.1 (May 2013), pp. 853–899. ISSN: 1076-9757. URL: http://dl.acm.org/citation.cfm?id=2566972.2566993 (cit. on p. 116).

[15] Judy Hoffman, Saurabh Gupta, and Trevor Darrell. "Learning with Side Information through Modality Hallucination". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 826–834. URL: http://www.cv-foundation.org/openaccess/content_cvpr_2016/html/Hoffman_Learning_With_Side_CVPR_2016_paper.html (cit. on p. 123).

[16] Phillip Isola et al. "Image-to-Image Translation with Conditional Adversarial Networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Honolulu, United States, July 2017, pp. 5967–5976. DOI: 10.1109/CVPR.2017.632 (cit. on p. 126).

[17] Michael Kampffmeyer, Arnt-Borre Salberg, and Robert Jenssen. "Semantic Segmentation of Small Objects and Modeling of Uncertainty in Urban Remote Sensing Images Using Deep Convolutional Neural Networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. Las Vegas, United States, 2016, pp. 1–9 (cit. on p. 123).

[18] Andrej Karpathy and Li Fei-Fei. "Deep Visual-Semantic Alignments for Generating Image Descriptions". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 3128–3137 (cit. on p. 116).

[19] Yelin Kim, Honglak Lee, and Emily Mower Provost. "Deep Learning for Robust Feature Generation in Audiovisual Emotion Recognition". In: *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. May 2013, pp. 3687–3691. DOI: 10.1109/ICASSP.2013.6638346 (cit. on p. 116).

[20] Adrien Lagrange et al. "Benchmarking Classification of Earth-Observation Data: From Learning Explicit Features to Convolutional Networks". In: *2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. 2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS). July 2015, pp. 4173–4176. DOI: 10.1109/IGARSS.2015.7326745 (cit. on pp. 117, 118).

[21] Seungyong Lee, Seong-Jin Park, and Ki-Sang Hong. "RDFNet: RGB-D Multi-Level Residual Feature Fusion for Indoor Semantic Segmentation". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. Oct. 2017, pp. 4990–4999. DOI: 10.1109/ICCV.2017.533 (cit. on p. 117).

[22] Guosheng Lin et al. "RefineNet: Multi-Path Refinement Networks with Identity Mappings for High-Resolution Semantic Segmentation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. July 2017, pp. 5168–5177. DOI: 10.1109/CVPR.2017.549 (cit. on p. 127).

[23] Yansong Liu et al. "Dense Semantic Labeling of Very-High-Resolution Aerial Imagery and LiDAR with Fully-Convolutional Neural Networks and Higher-Order CRFs". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. Honolulu, United States, July 2017, pp. 1561–1570. DOI: 10.1109/CVPRW.2017.200 (cit. on pp. 118, 123).

[24] Emmanuel Maggiori. "Learning Approaches for Large-Scale Remote Sensing Image Classification". PhD thesis. Université Côte d'Azur, June 22, 2017. URL: https://hal.inria.fr/tel-01589661/document (cit. on p. 126).

[25] Dimitrios Marmanis et al. "Classification With an Edge: Improving Semantic Image Segmentation with Boundary Detection". In: *ISPRS Journal of Photogrammetry and Remote Sensing* (2017). DOI: 10.1016/j.isprsjprs.2017.11.009. arXiv: 1612.01337 (cit. on pp. 122, 123).

[26] Uwe Meier, Wolfgang Hürst, and Paul Duchnowski. "Adaptive Bimodal Sensor Fusion for Automatic Speechreading". In: *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*. 1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings. Vol. 2. May 1996, 833–836 vol. 2. DOI: 10.1109/ICASSP.1996.543250 (cit. on p. 117).

[27] Bjoern H. Menze et al. "The Multimodal Brain Tumor Image Segmentation Benchmark (BRATS)". In: *IEEE Transactions on Medical Imaging* 34.10 (Oct. 2015), pp. 1993–2024. ISSN: 0278-0062. DOI: 10.1109/TMI.2014.2377694 (cit. on p. 117).

[28] Volodymyr Mnih. "Machine Learning for Aerial Image Labeling". University of Toronto, 2013 (cit. on p. 126).

[29] Jiquan Ngiam et al. "Multimodal Deep Learning". In: *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*. 2011, pp. 689–696. URL: http://machinelearning.wustl.edu/mlpapers/paper_files/ICML2011Ngiam_399.pdf (cit. on pp. 116, 117).

[30] Kuniaki Noda et al. "Audio-Visual Speech Recognition Using Deep Learning". In: *Applied Intelligence* 42.4 (June 1, 2015), pp. 722–737. ISSN: 0924-669X, 1573-7497. DOI: 10.1007/s10489-014-0629-7 (cit. on p. 117).

[31] Ferda Ofli et al. "Berkeley MHAD: A Comprehensive Multimodal Human Action Database". In: *2013 IEEE Workshop on Applications of Computer Vision (WACV)*. Jan. 2013, pp. 53–60. DOI: 10.1109/WACV.2013.6474999 (cit. on p. 117).

[32] Francisco Javier Ordóñez and Daniel Roggen. "Deep Convolutional and LSTM Recurrent Neural Networks for Multimodal Wearable Activity Recognition". In: *Sensors* 16.1 (Jan. 18, 2016), p. 115. DOI: 10.3390/s16010115. URL: http://www.mdpi.com/1424-8220/16/1/115 (cit. on p. 116).

[33] Sakrapee Paisitkriangkrai et al. "Effective Semantic Pixel Labelling with Convolutional Networks and Conditional Random Fields". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. June 2015, pp. 36–43. DOI: 10.1109/CVPRW.2015.7301381 (cit. on p. 118).

[34] Fabien Ringeval et al. "Introducing the RECOLA Multimodal Corpus of Remote Collaborative and Affective Interactions". In: *10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*. Apr. 2013, pp. 1–8. DOI: 10.1109/FG.2013.6553805 (cit. on p. 117).

[35] Björn Schuller et al. "AVEC 2011–The First International Audio/Visual Emotion Challenge". In: *Affective Computing and Intelligent Interaction*. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, 2011, pp. 415–424. ISBN: 978-3-642-24570-1 978-3-642-24571-8. DOI: 10.1007/978-3-642-24571-8_53 (cit. on p. 117).

[36] Max Schwarz, Hannes Schulz, and Sven Behnke. "RGB-D Object Recognition and Pose Estimation Based on Pre-Trained Convolutional Neural Network Features". In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*. May 2015, pp. 1329–1335. DOI: 10.1109/ICRA.2015.7139363 (cit. on p. 117).

[37] Jamie Sherrah. "Fully Convolutional Networks for Dense Semantic Labelling of High-Resolution Aerial Imagery". In: (June 8, 2016). arXiv: 1606.02585 [cs]. URL: http://arxiv.org/abs/1606.02585 (cit. on p. 123).

[38] Karen Simonyan and Andrew Zisserman. "Two-Stream Convolutional Networks for Action Recognition in Videos". In: *Advances in Neural Information Processing Systems 27*. Curran Associates, Inc., 2014, pp. 568–576. URL: http://papers.nips.cc/paper/5353-two-stream-convolutional-networks-for-action-recognition-in-videos.pdf (cit. on p. 117).

[39] Karen Simonyan and Andrew Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition". In: *Proceedings of the International Conference on Learning Representations (ICLR)*. May 2015. URL: http://arxiv.org/abs/1409.1556 (cit. on p. 127).

[40] Xinhang Song, Shuqiang Jiang, and Luis Herranz. "Combining Models from Multiple Sources for RGB-D Scene Recognition". In: *Proceedings of the 26th International Joint Conference on Artificial Intelligence*. IJCAI'17. Melbourne, Australia: AAAI Press, 2017, pp. 4523–4529. ISBN: 978-0-9992411-0-3 (cit. on p. 117).

[41] Nitish Srivastava and Ruslan Salakhutdinov. "Multimodal Learning with Deep Boltzmann Machines". In: *Journal of Machine Learning Research* 15 (2014), pp. 2949–2980. ISSN: 1532-4435 (cit. on p. 117).

[42] Maria Vakalopoulou et al. “Simultaneous Registration, Segmentation and Change Detection from Multisensor, Multitemporal Satellite Image Pairs.” In: International Geoscience and Remote Sensing Symposium (IGARSS). July 10, 2016. DOI: 10.1109/IGARSS.2016.7729469 (cit. on p. 126).

[43] Cihang Xie et al. “Adversarial Examples for Semantic Segmentation and Object Detection”. In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. Oct. 2017, pp. 1378–1387. ISBN: 978-1-5386-1032-9. DOI: 10.1109/ICCV.2017.153 (cit. on p. 123).

[44] Ben P. Yuhas, Moise H. Goldstein, and Terrence J. Sejnowski. “Integration of Acoustic and Visual Speech Signals Using Neural Networks”. In: *IEEE Communications Magazine* 27.11 (1989), pp. 65–71 (cit. on p. 117).

Chapter

6 Model generalization

I see no limit to the capabilities of machines. As microchips get smaller and faster, I can see them getting better than we are. I can visualize a time in the future when we will be to robots as dogs are to humans.

— Claude Shannon

Contents

Summary:

The generalization of deep models trained on a given ground truth to new acquisitions is the key for large-scale application of deep learning for Earth Observation. Indeed there exists a large diversity of ecosystems and environments, both in terms of time and space, that might limit the scope of the trained models on a handful of local scenes. Two problems arise from this statement.

On the one hand, we know that creating a ground truth on some sensors can be both difficult and expensive, for example on hyperspectral data that requires expert knowledge. Yet statistical models trained on small datasets are prone to overfitting that is at opposed to the desired goal of generalizability. For this reason we start by investigating how one can create new synthetic training samples that might alleviate the scarceness of real labeled data.

On the other hand, remote sensing data is abundant and extremely diverse. This rises questions whether the deep networks we worked with in the previous chapters can scale on such massive datasets. A simplified version of this problem consists on trying to transfer the knowledge from a model trained on a given scene to another. We perform some transfer learning experiments between the ISPRS Potsdam and Vaihingen datasets to evaluate how pretrained models can generalize on new acquisitions that are weakly labeled – or even not labeled at all. Finally we introduce a new large-scale labeled remote sensing data called *MiniFrance*, the largest based on public data as far as we know. It recoups aerial images on 16 conurbations in France along with land cover and building footprints annotations.

6.1 Synthetic data generation

As we have seen in the Chapter 4, the current labeled hyperspectral datasets are rare and of small sizes. Labeleing hyperspectral images is hard and because of the low spatial resolution of the sensor, acquisitions tend to be small which makes building a large-scale annotated hyeprspectral dataset pretty much impossible. There are laboratory spectrum measurements, such as the United States Geological Survey (USGS)[1] database but those are nearly impossible to use because the sensors used are different, not calibrated the same way and more importantly the acquisitions have not been performed in the same experimental conditions compared to remote sensing. Therefore as gathering more real data is not an option, one could be interested on augmenting the data at hand.

Data augmentation consists in introducing fake synthetic samples to enlarge the size of a training set [8]. This practice is very common when training deep neural networks and especially CNN since the seminal work of Krizhevsky, Sutskever, and Hinton [12] as it prevents overfitting. In a hyperspectral image classification framework, the scarcity of actual training labels make the perspective of data augmentation even more appeleaing. However most state of the art publications using 2D or 3D CNNs for hyperspectral image classification [6, 17, 21, 13] often limit their scope to small datasets that do not show the strength of representation learning.

Some works have been focused on the artificial augmentation of publicily available hyperspectral datasets. For example, Windrim et al. [23] proposed a physical model to simulate how a spectrum is distorted under illumination conditions that differ from those it was originally acquired under. This allows them to introduce some invariance to such environmental variations at inference time. This however involve the creation and the implementation of a sophisticated physical model based on expert prior knowledge that is not generic and that introduces uncertainty due to the illumination estimation that is not necessarily accurate on remote sensing images. A more straightforward technique has been suggested by Chen et al. [6] to augment the number of training samples by generating fake mixtures of existing spectra using linear combination and some gaussian noise. These alterations are considered plausible based on prior work and somehow simulate how mixtures can be observed by a hyperspectral camera. Finally Acquarelli et al. [1] introduced a label propagation scheme from a pixel to its neighbour using a clustering approach. The goal is to incorporate in the training set pixels that have been observed but not included in the training because unlabeled. This approach allows them to learn from more pixels but the total number of samples is still upper-bounded by the size of the acquisition.

In this work we ask the following question: how to augment the training datasets when no physical model or prior expert knowledge is available, while adding as many new samples as we want? A first idea is at the core of the work from Gemp et al. [9]. They implement variational autoencoders on hyeprspectral data they then use as generative models for unmixing to autoamtically find the endmembers and abundancy maps from an image. For classification, Davari et al. [7] use a Gaussian Mixture Model (GMM) to estimate the distrbution of spectral features based on attribute profiles. They then generate new synthetic attribute profiles based on the approximated distribution to augment the original training set.

The generative models are very interesting as they can approximate the latent statistical distribution to the set of observations in order to sample new observations. We suggest to use generative models to approximate the latent distrbution of the spectra in the hyperspectral image in order to synthetize new samples that could realistically belong to it. This is a data-driven approach that does not require any physical prior about the scene or the sensor.

More specifically, we build on the Generative Adversarial Network (GAN) [10] framework to estimate the latent distribution of the true spectra and then use it to sample new ones

[1] *USGS Spectral Library*: https://speclab.cr.usgs.gov/spectral-lib.html

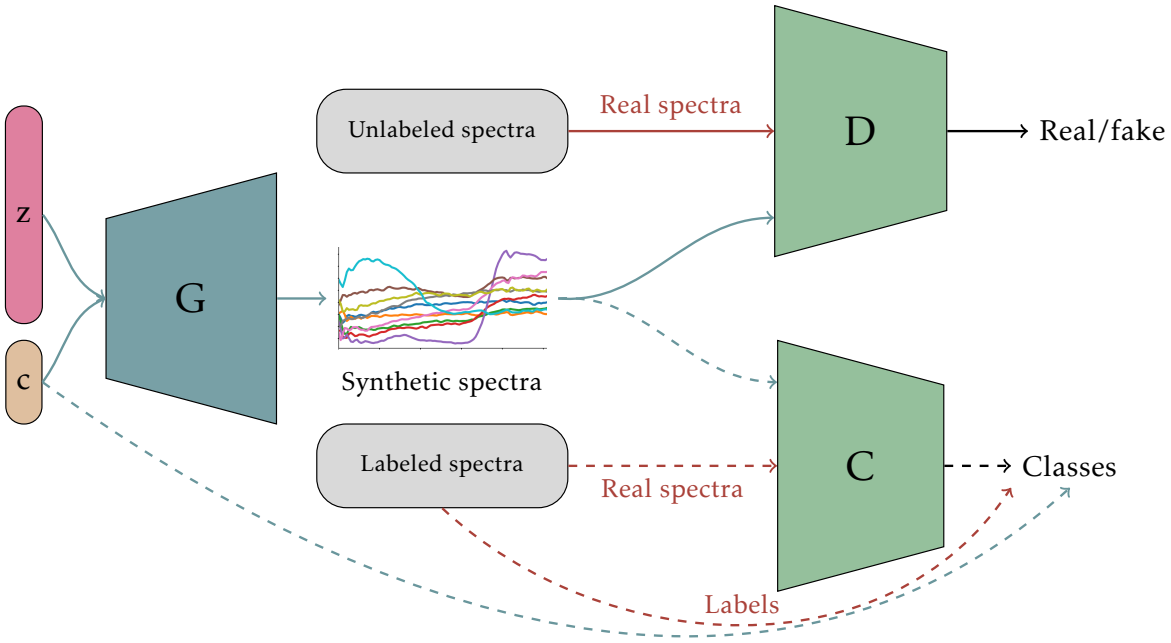


Figure 6.1: The GAN structure used to synthesize artificial spectra. Red arrows indicate the data flow when training the classifier and the discriminator while blue arrows indicate training of the generator. Dotted arrows indicate connections that occur only in the supervised setting (when the label is available).

which statistically should belong to this distribution. We aim for a semi-supervised method that can leverage both labeled and unlabeled data. We validate this data augmentation scheme using fake spectra on various public hyperspectral datasets using aerial and satellite images on various geographical areas.

6.1.1 Generative adversarial networks

The principle of GANs was introduced by Goodfellow et al. [10] in 2014. The core idea consists in using deep neural networks to model the statistical distribution underlying an empirical set of observations. A generator is trained to approximate the projection between a latent space of gaussian noise to the empirical observed distribution. However the distribution is observed only through some samples and we wish to use the generator to create new observations that could realistically belong to it. To do so, the generator is trained to approximate the distribution using an adversarial loss function. This loss function is implicitly defined through a second network, called the discriminator (or sometimes the critique). The discriminateur learns to estimate whether a sample comes from the true observed dataset or from the fake dataset (i.e. produced by the generator). At each optimization step, the discriminator is trained for a few iterations so that it learns to separate real and fake data. the generator is then optimized to that it *fools* the discriminator, i.e. that the second network misclassifies the synthetic samples as real and that both distribution are not separable anymore by the critique (cf. Definition 7)[2].

Many GANs flavors have been proposed since their introduction. In our case we use a generator G and a discriminator D based on the Wasserstein GAN model [2] using the

[2]A common analogy consists in describing the generator as a painting forger and the discriminator as a detective. The forger wants to imitate famous paintings (the real data). To do so, its mission is to trick the detective into thinking that the fake is actually a masterpiece, i.e. that both are indistinguishable.

gradient penalty from Gulrajani et al. [11]. The Wasserstein GAN is designed to minimize the Wasserstein distance between the real and fake distributions. G transforms a random noise vector z into a spectrum so that D predicts that it belongs to the true distribution. However this is a unsupervised behaviour, i.e. this can generate new samples that belong to the real distribution but we cannot know their label yet. One possibility is to train one generator for each class but this would be slow and expensive. In our case we want to be able to *condition* the generator with respect to the class of interest for which we want to generate new samples. We therefore use an auxiliary classifier C [18] that adds an additional penalty when optimizing the generator that enforces that the generated samples are classified in the chosen class. In this case, G takes as an input both the latent vector c and a condition vector c that is a one-hot encoding of the desired class. The spectar generated by G have to wrongly classified by D and rightly classified by C. The complete architecture is detailed in the Fig. 6.1. If G and D can trained without any label, i.e. unsupervisedly, C needs label information to understand how to separate the actual classes. The ensemble therefore constitues a semi-supervised model that can leverage both labeled and unlabeled samples from the whole hypercube.

Definition 7. *Training of the generative adversarial networks:*
Let n be the batch size, $\mathcal{Z}$ the latent distribution, Ω the set of samples (labeled and unlabeled), $\Omega^* \subset \Omega$ the subset of labeled samples and $\mathcal{L}$ the cross-entropy function. While the convergence criterion has not been reached:

1. *Optimize* D. *Repeat k_D times:*
   - *Draw a random vector $\mathbf{x}$ of n samples in Ω*
   - *Iterate the gradient descent on* D *to maximize* $D(\mathbf{x})$
   - *Draw a random vector $\mathbf{z}$ of n samples in $\mathcal{Z}$*
   - *Iterate the gradient descent on* D *to minimize* $D(G(\mathbf{z}))$
2. *(optional) Optimize* C. *Repeat k_C times:*
   - *Draw a random vector $\mathbf{x}^*$ of n samples in Ω^*, with y the class vector*
   - *Iterate the gradient descent on* C *to minimize* $\mathcal{L}(C(\mathbf{x}^*), y)$
3. *Optimize* G *once.*
   - *Draw a random vector $\mathbf{z}$ of n samples in $\mathcal{Z}$*
   - *(optional) Generate and concatenate to $\mathbf{z}$ a condition vector $\mathbf{c}$*
   - *Generate the fake samples* $\hat{\mathbf{x}} = G(\mathbf{z})$
   - *Compute the loss fonction* $\mathcal{L}_{totale}(\mathbf{z}) = -D(\hat{\mathbf{x}})$
   - *(optional) Add the classification error on* C: $\mathcal{L}_{totale}(\mathbf{z}) := \mathcal{L}_{totale}(\mathbf{z}) + \mathcal{L}(C(\hat{\mathbf{x}}), \mathbf{c}))$
   - *Iterate the gradient descent on* G *to minimize* $\mathcal{L}_{totale}$

6.1.2 Experimental setup

We trained the previously described GAN architecture on the Pavia University, Pavia Center, Indian Pines and Botswana datasets (cf. Section 4.2.2) on reflectances values – after atmospheric correction when available. Since we aim to generate individual spectra and not hypercubes, we use simple fully connected networks for G, D and C with four layers each and the *Leaky ReLU* [15] activation. This non-linearity is more popular with GANs than the usual ReLU since its gradient is non-zero everywhere so that gradients backpropagated from D to G are less sparse. The output from the generator G is followed by a sigmoid to constrain the synthetic normalized reflectances in the $[0, 1]$ range. D only has one continuous

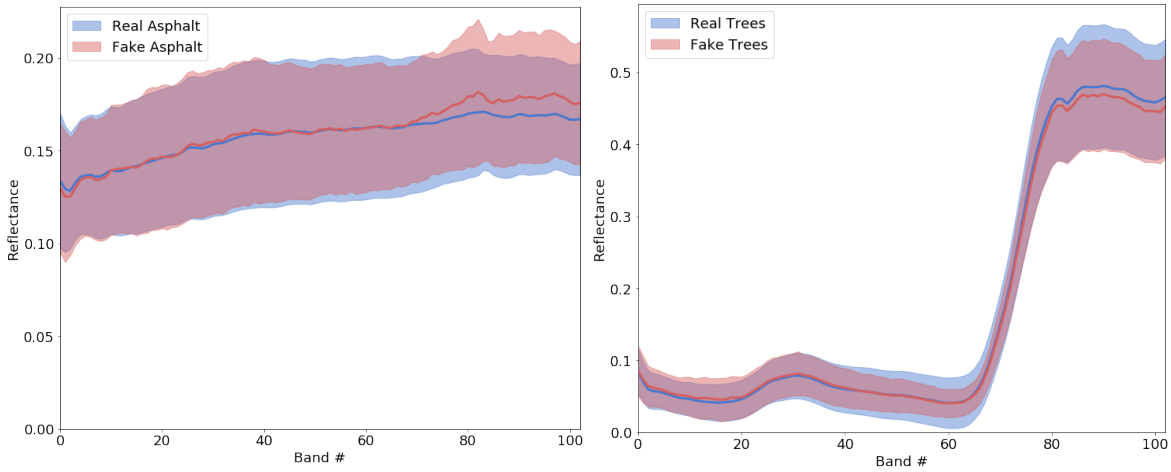


Figure 6.2: Average spectrum and standard deviation for to materials from the Pavia Center dataset. The average synthetic samples are noisier than the real ones and overfit on some local spectral properties.

output (binary classification with a sigmoid) and C has as many outputs as there are classes of interest in the dataset (softmax classifier).

The optimization of the three networks is done using a stochastic gradient descent flavor called *RMSProp* [22]. The ensemble is trained for 100 000 iterations with a batch size of 256. We apply two training iterations to C and D for one iteration of G. When backpropagating on G, the auxiliary classification loss function is weighted by 0.2. The global learning rate is set to 5×10^{-5}.

As a baseline for evaluating the performance of our GANs on spectrum generation, we also implement a gaussian mixture model using the scikit-learn library [19]. We reconstruct one mixture for each from the datasets using 10 components. We generate new spectra by sampling in the mixture of gaussian distribution.

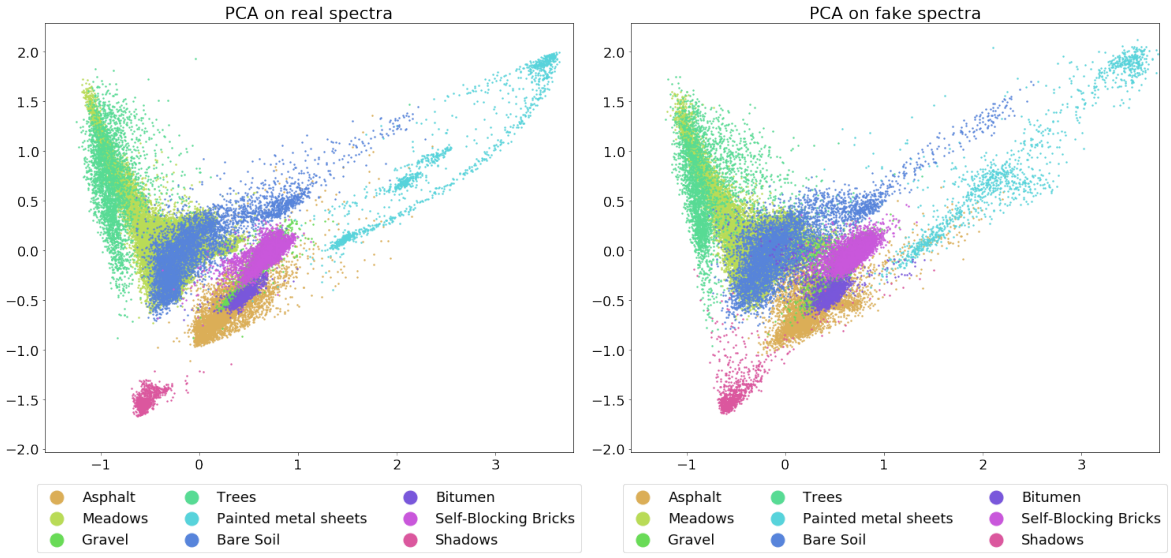
6.1.3 Spectrum analysis

To begin with the analysis of the synthetic spectra, we start by training deux GANs on Pavia University and Indian Pines. We compare the fake samples to the actual distribution based on various criteria.

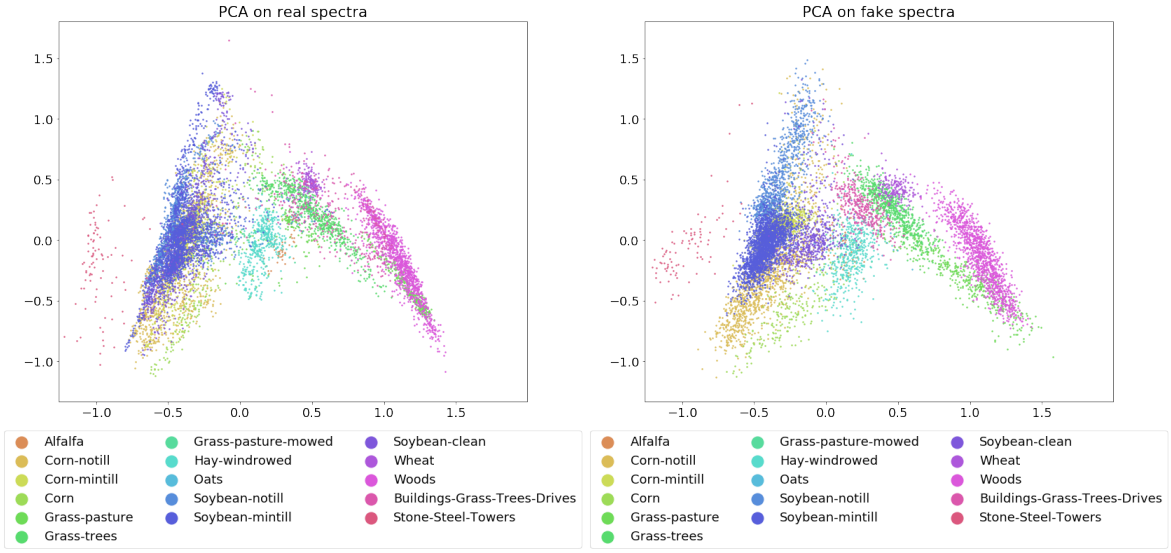
We can see in the Fig. 6.2 that the fake synthetic spectra exhibit similar statistical moments to the real samples. The general shapes of the spectra are correctly approximated for every class that we considered. However there are two limitations that we can identify. First synthetic spectra look "noisier" than their real counterparts. Indeed it appears that the GAN is overfitting some spectral characteristics existing in the training samples and that the classifier C probably use to separate between classes. Those features are exacerbated in the fake samples. Moreover the standard deviation in the synthetic distribution is lesser than the actual standard deviation, i.e. synthetic spectra are less diversified than the true samples. Once again this indicates that the generator is overfitting and falls into a phenomenon called *mode collapse* [20], where it learns only the main mode of the data while forgetting lesser represented groups of samples.

In order to understand better this overfitting, we apply a PCA on real and fake spectra to project the distributions into a 2-dimensional space for simpler visualization (cf. Fig. 6.3). We can observe that the clusters corresponding to the different classes are well-reproduced by the synthetic distribution. However there are some geometrical distorsions in the fake distribution that show that, altough the GAN was able to generally approximate the various types of spectra, it was not able to capture the full distribution of the spectral features.

We can now try to estimate how well the synthetic distribution fits the actual class



(a) Pavia University



(b) Indian Pines

Figure 6.3: PCA on real and fake spectra. The real distribution contains all labeled samples from the dataset. The two distributions (real and fake) contain the same number of samples.

boundaries from the real distribution. To do so we train a linear SVM on the real spectra and we evaluate it on the synthetic spectra. The linear SVM will compute the best separating hyperplanes for the true distribution. Ideally these hyperplanes should separate true and synthetic spectra equally well. If they are less accurate on fake spectra than on real ones, then it means that the generator has produced unrealistic samples that do not belong to the relevant class. If they are significantly more accurate on fake spectra, it means that the generator produce samples with a very low intra-class variance that are clustered around the class centroid and therefore exhibit a low diversity. The results of these experiments are reported in the Table 6.1. We consider two approaches: training on a subset of 3% of the spectra randomly sampled in the image or 50% of the image but spatially disjoint from the validation area. In the unsupervised mode, we also use unlabeled samples to train the GAN. As could have been expected based on the previous observations, the SVM is more accurate on fake spectra than on the real ones. However training the SVM on the synthetic spectra only still results in hyperplanes that separate between real spectra up to a certain point. Another way to frame this is that synthetic spectra are less diverse than real ones, but

Table 6.1: Accuracies of a linear SVM applied on real and fake spectra from the Pavia University dataset.

Split	Random (uniform) – 3% (r)		Disjoint – 50% (s)	
Train \ Test	Real	Synthetic	Real	Synthetic
Real	89.5	98.3	87.2	98.8
Synthetic	87.8	99.2	79.4	99.9

they are still relatively representative of the main features from each class, despite the fact that they have been generated *ex nihilo* from random noise.

Finally, since GAN establish a mapping between a latent representation space and the empirical signal distribution, it is possible to explore the diversity of spectra by interpolating continously between two points from latent space. Indeed, if z_1 and z_2 are two random vectors sampled from the latent gaussian distribution, then one can interpolate between the two along the unit hypersphere:

$$\begin{cases} \forall \alpha \in [0,1], \;\; z_\alpha = \frac{\sin((1-\alpha)\cdot\omega)}{\sin\omega} \cdot z_1 + \frac{\sin(\alpha\cdot\omega)}{\sin\omega} \cdot z_2 \\ \hat{x}_\alpha = \mathrm{G}(z_\alpha, c) \text{ where } x_0 = \mathrm{G}(z_1, c) \text{ and } x_1 = \mathrm{G}(z_2, c) \end{cases} \tag{6.1}$$

where ω is the angle between z_1 and z_2. Another possibility is to interpolate between two condition vectors c_1 et c_2 with a fixed noise:

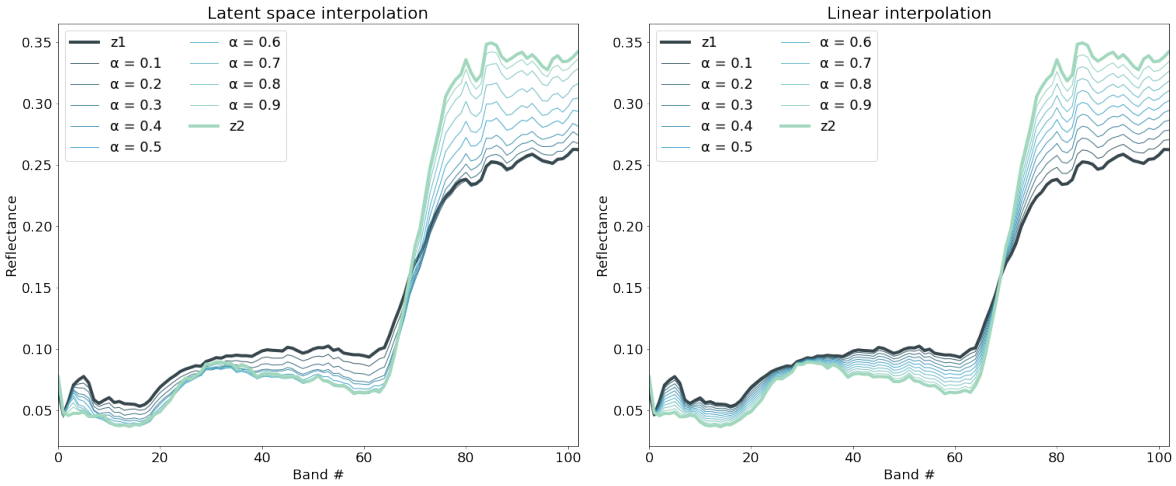
$$\begin{cases} \forall \alpha \in [0,1], \;\; c_\alpha = (1-\alpha)\cdot c_1 + \alpha \cdot c_2 \\ \hat{x}_\alpha = \mathrm{G}(z, c_\alpha) \text{ where } x_0 = \mathrm{G}(z, c_1) \text{ and } x_1 = \mathrm{G}(z, c_2) \end{cases} \tag{6.2}$$

The interpolation between two points from the latent space generates a spectral progression as pictured in the Fig. 6.4a. In comparison a linear interpolation applied on the raw spectral signatures produce uniformly distributed samples that might not be realistic. The latent vectors actually encode a path on the manifold of the spectra. Computing the barycenter between two spectral signatures does not necessarily has a physical meaning as the resulting vector might not be on the manifold of observable spectra. On the opposite the interpolation obtained using the GAN approach accurately represent the geodesic path on the manifold that connects the two ends.

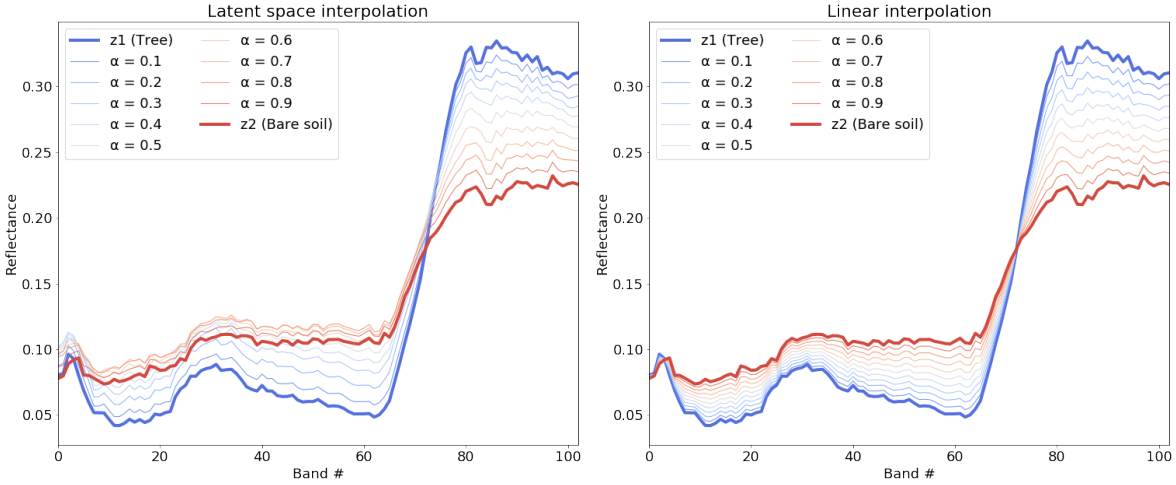
In the same idea, we can simulate spectral mixtures by interpolating between condition vectors instead of noise vectors, as shown in the Fig. 6.4b. The mixtures of materials observed in actual conditions often present non-linear properties due to terrain geometry, light reflections or shadows and occlusions. Once again the GAN produces samples that should belong on the manifold of the observed spectrum, while a linear interpolation walks an arbiratry path in a space with no physical meaning. Provided that the mixtured hallucinated by the generator are realistic, it would mean that this i way to achieve the inverse of the unmixing operation. Therefore it would be possible to map a suspected to its endmembers by mapping the complete latent space and the using a dictionary learning, nearest-neighbour classification or model inversion techniques [9].

6.1.4 Data augmentation

Since the generated samples are realistic and somewhat representative of the real spectra, we suggest to use them to enrich the existing labeled datasets as they might generate variations of real spectra that could introduce new invariances in the model. We test this idea on several datasets: Indian Pines (aerial, rural), Pavia University (aerial, urban), Pavia Center (aerial, urban) and Botswana (satellite, rural). The results using the supervised mode (GAN) and unsupervised mode (ss-GAN) are reported in the Table 6.2. Augmenting the dataset using fake spectra slightly improve the classifiers' accuracy.



(a) Interpolation between two latent vectors from the class “meadows”.



(b) Interpolation between the classes “tree” and “bare soil” with a fixed noise vector.

Figure 6.4: [I
nterpolations in the spectral latent space]Interpolating between two noise vectors or two conditioning vectors in the latent space is a way to continuously explore the spectral distribution. The GAN is trained on Pavia University for this figure. α is the interpolation factor.

However increasing too much the number of fake spectra does not increase the accuracy anymore and slightly degrades it. Indeed in this case the synthetic spectra have too much importance in the loss function and, as in the SVM case, they reduce the global accuracy.

Overall using GANs to generate synthetic spectra *ex nihilo* for data augmentation is not a complete game changer and only slightly increases the accuracy. Indeed GANs can only approximate the actual spectral distribution and possibly interpolate on the manifold. Nonetheless they cannot generate completely new observations outside this distribution. Therefore the information contained in the fake distribution cannot be significantly greater than the actual distribution. Since the classification consists in finding inter-class separations, the samples far from the class center are the more interesting for the model. Therefore the semi-supervised approach is a way to generate labeled spectra that present statistical features similar to unlabeled spectra, and therefore to augment the information quantity available to the classifier, in the same way a clustering used to propagate labels would. However the supervised approach quickly reaches its limits and saturates.

In conclusion, this work showed that GANs are powerful tools that can efficiently approximate complex statistical distribution in a pure data-driven approach without any expert knowledge. This is especially interesting since this might allow for hybrid techniques. The

Table 6.2: Accuracies of a 4-layers multi-layer perceptron on several hyperspectral datasets using various data augmentation policies. The datasets are split either in half (s) or by randomly sampling 3% of the pixels on the whole image (r).

Dataset	Pavia University		Pavia Center		Botswana		Indian Pines	
Augmentation	3% (r)	50% (s)	3% (r)	50% (s)	3% (r)	50% (s)	3% (r)	50% (s)
∅	92.72	86.22	98.93	96.26	86.90	84.87	79.44	74.00
GAN	92.95	86.47	99.00	96.26	87.72	84.60	80.01	74.81
ss-GAN	93.12	87.20	98.93	96.70	88.40	85.27	80.42	74.58

community has invested lots of time and effort in developing hyperspectral simulation models [5]. These simulators are based both on laboratory reflectance measures of known materials and physical models of sensor and atmosphere. However these models are approximative and simplified, therefore they necessarily introduce noise and errors. It is difficult for these models to deal with optical, atmospherical and electronical effets that are complex (noise provoked by component heating, parasites due to light, distorsion due to the atmosphere…). A hybrid approach combining data and physics could be based on conditioned GANs that maps the smooth spectra generated by the simulators to "realistic" observations, therefore letting the GAN approximating the complex phenomena that are expensive and difficult to compute. The GAN task would be only to make simulated data more realistic and in line with the observations.

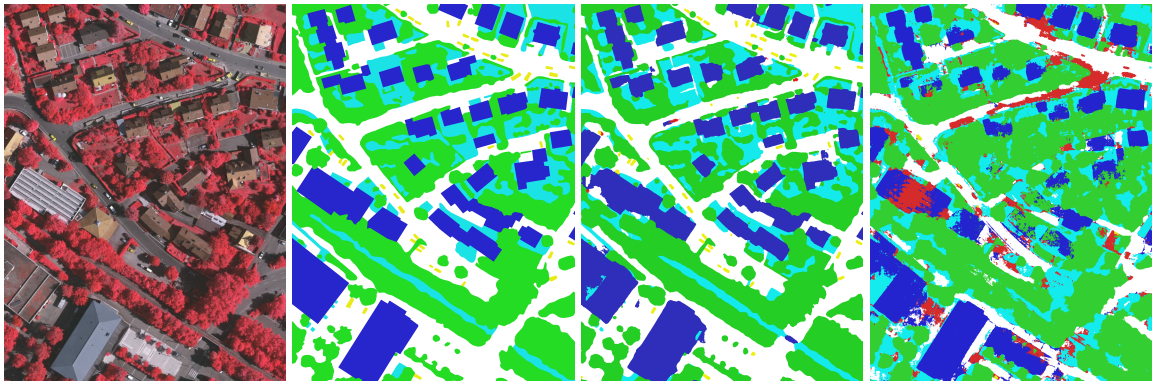
6.2 Scalability to large-scale datasets

6.2.1 Scene diversity

Until now we worked with datasets that cover only one scene, i.e. a unique geopgrahical area captured at a given time. The experiments from the Chapters 3 to 5 were done on the cities of Vaihingen and Potsdam, one at a time. However this is not exactly a actual use case. Earth Observation is done in the wild by multiple, partially overlapping, acquisitions all over the globe at several points in time. For these reasons it is necessary to evaluate how models can generalize on various environments that reflect the diversity of Earth's geography. Thecore question is to understand what we can expect of deep networks when applied on new unseen data. It is reasonable to expect some sort of decrease in the accuracy, since any dataset comes with its own intrinsic bias on which the model will overfit. However it is important to quantify this decrease.

As a first experiment we investigate transfer between two similar colour scenes: the ISPRS Potsdam and ISPRS Vaihingen. The two datasets are comprised of EHR aerial images with IRRG channels, acquired in urban areas and labeled for the exact same classes. Interestingly the two cities do not exhibit the same features: Potsdam has six times more inhabittants than Vaihingen and twice the density. The buildings are not built based on the same architectures and the overall town organizations are quite different. As a first we consider the SegNet model trained on the IRRG from Potsdam (cf. Chapter 3) that we apply at inference time on Vaihingen. The generated maps are shown as is in the Fig. 6.5. Overall the main components of the image are detected by the network, especially the roads and the buildings. However there are two lage confused areas: buildings with the clutter class (in red) and no vehicles. The latter can be explained by the difference in resolution between the two datasets. The convolutions for the model trained on Potsdam were optimized for images at 5 cm/px resolution. When applied on the Vaihingen image at 9 cm/px, vehicles are smaller than expected and therefore misclassified as the scale factor has changed. Overall the accuracy on Vaihingen for the model trained on Potsdam alone reaches 77%, which is significantly worse than the results presented in the Chapter 3. Training a model on one scene seems to generate

a significant bias that impedes its generalization to new unseen data.



IRRG tile 21 | Ground truth | SegNet prediction (Vaihingen) | SegNet prediction (Potsdam)

Figure 6.5: Semantic maps predicted on tile # 21 (Vaihingen) by a SegNet model trained respectively on Vaihingen and Potsdam. The transfer without fine-tuning from Potsdam to Vaihingen is able to detect the main objects but is significantly less accurate than a model trained on Vaihingen.

A realistic use case could be labeling a (very) small part of the target dataset (here, Vaihingen) and to perform a quick fine-tuning of the model pretrained on Potsdam. This would allow the model to adjust its weight to take into account the new images without retraining the full data-hungry network. We consider the same SegNet network, fine-tuned on the ISPRS Vaihingen dataset as follows:

- the weights of the last decoding block are optimized with a learning rate $\alpha = 0,01$,
- the learning rate for the other layers in the decoder is set constant to $\frac{\alpha}{10}$,
- the weights of the encoder are frozen.

The fine-tuning is tested in experiments where few data have been labeled, e.g. with $^1/_4$ of tile #3 or only tile #3 in full. To estimate the improvement on the Potsdam pretraining, we also compare the accuracies of SegNet models trained on the same images, but from scratch (i.e. the encoder is initialized with VGG-16 pretrained on ImageNet and the decoder is randomly initialized). Results are reported in the Table 6.3.

Table 6.3: Semantic segmentation results using transfer learning for the ISPRS Vaihingen dataset.

# tiles	pretraining	Imp. surfaces	Buildings	Low veg.	Trees	Vehicles	Accuracy
1/4	ImageNet	76.6	29.6	0.07	95.1	0.01	54.8
	Potsdam	80.3	55.0	16.0	95.8	43.3	65.5
1	ImageNet	91.8	78.8	50.4	93.5	47.6	81.0
	Potsdam	83.3	91.2	59.4	85.6	60.1	82.8

The model accuracy significantly decreases as we remove labeled tiles from the training set. Not only does this phenomenon appear when they are less dense labels available, it also occurs when existing annotations are made sparser and less complete. Maggiolo et al. [16] indeed showed that the accuracy of FCNs models trained on Vaihingen using a coarse ground truth significantly decreased. They considered an alternative version of the ground truth that only preserved 60% of the labeled pixels. Some objects are purely and simply omitted while other are only annotated by scribbles that coarsely approximate their respective shapes. The FCN is then trained on these pixels and learning is deactivated on unlabeled areas. The

overall accuracy drops by approximately 20%, which is similar to ours results obtained by training SegNet on a quarter of a tile with dense annotations.

This goes to show that if there are a least a few annotations on the target domain (in our case Vaihingen), pretraining on a source domain (Potsdam) significantly improves the generalization ability of the network at inference time. Fine-tuning makes it possible to alleviate the dataset bias from Potsdam and significantly reduce the influence of environmental conditions. As a practical use case, these result suggest that segmentation models could be adapted to new acquisitions with a moderate labeling effort. However we underline that the weights of the model trained on Potsdam are not as generic as those obtained by training on ImageNet for natural image classification. Indeed the latter draw their expressiveness by the large diversity of objects and classes exhibited by ImageNet. Reproducing those properties using remote sensing data requires a large-scale dataset that comes with a significant variability, both regarding semantics and observed geographic areas.

6.2.2 MiniFrance

City distribution | Orthoimage (Nice) | Annotations (UrbanAtlas)

Figure 6.6: Overview of the *MiniFrance* dataset.

Creating a remote sensing counterpart to ImageNet requires gathering and labeling a vast quantity of data. Although labeling natural images for classification and object recogntiion is relatively quick, dense segmentation annotations are significantly longer to obtain. Moreover in remote sensing distiinguishing between various objects types often require expert knowledge and some experience of photointerpreattion that is out of reach for the usual crowdsourcing strategies often deployed in computer vision and machine learning [4].

Therefore we choose to rely on semi-manual annotation of a large volume of data that have been labeled by an automatic classifier and then corrected by a specialist. The accuracy of such annotations are certainly lesser than those of human experts but this allows us to build a datatset bigger than those which currently exist. If learning on noisy or weak labels can require specific learning processes [14], in our case we start by training a fully supervised baseline to assess what is achievable with the dataset we built. We start with a country-scale dataset focused on France since it has a moderate climate, a strong environmental diversity (mountains, coasts, forests, crops, urban metropolis...) and – even more important – lots of data and annotations that we can leverage.

We collect a large-scale dataset on Metropolitan France. We use the BD ORTHO from the IGN as a data source to collect multiple aerial images with a spatial resolution of 50 cm/px. To ease the reproducibility of our findings, we consider only acquisitions performed between 2012 and 2015 that freely available and under an opence license. Overall these images cover 25 departements. The images are released by the IGN as RGB tiles. The mosaic has been split in square tiles of 10 000 px × 10 000 px, i.e. 25 km^2. Images are initially published in the JPEG2000 format although we convert them in GeoTIFF encoded in 8-bits integers for

faster decoding.

Concurrently, we also gathered land cover information from the Copernicus project *Urban Atlas* 2012 [3]. Urban Atlas is a land cover map of Europe for 17 urban classes and 10 rural classes that cover most large cities from the European Union that have more than 30 000 inhabitants. The annotations are semi-automated, generated by a classification and then inspected by deux experts and corrected based on high resolution satellite images acquired in 2012 (mostly SPOT-5 data). 82 extended urban areas are concerned in France. The intersection between the Urban Atlas data and the images from the BD ORTHO allow us to find 16 cities and their general suburbs with images ranging from 2012 to 2014 (to avoid a too large time gap between the *Urban Atlas* reference and the images). The cities we use are listed in the Table 6.4. We rasterize the corresponding shapefiles for each image of the BD ORTHO to generate semantic segmentation ground truth tiles. We leverage the hierarchical structure of the Urban Atlas taxonomy by grouping various labels with similar semantics into 14 land cover categories that are detailed ina Table 6.5.

As ancillary annotations that might be usable in the future, we also considered using the French building cadastre which is available under an open license. We integrated and rasterize for all tiles the cadastre released in the shapefile vector format by the governemental group Etalab[4]. We remove from the ground truth buildings that were added to the registry after January 1st, 2015, since they would not have been built when the images from the BD ORTHO were captured.

We name the complete dataset *MiniFrance*. An overview is given in the Fig. 6.6. The 16 cities we consider are mostly in the west of France although south-east, center and north are also represented. 8 towns are used for training and the remaining 8 are kept for evaluation.

Table 6.4: List of cities in the MiniFrance dataset.

	Conurbation	Tiles	% pixels	Colour
Training	Nice	170	8.01%	
	Nantes, Saint-Nazaire	226	10.65%	
	Le Mans	107	5.04%	
	Lorient	68	3.20%	
	Brest	88	4.14%	
	Caen	126	5.94%	
	Dunkerque, Calais, Boulogne-sur-Mer	150	7.07%	
	Saint-Brieuc	71	3.34%	
Ealuation	Marseille, Martigues	162	7.63%	
	Rennes	196	9.24%	
	Angers	123	5.79%	
	Quimper	79	3.72%	
	Vannes	73	3.44%	
	Clermont-Ferrand	150	7.07%	
	Lille, Arras, Lens, Douai, Hénin-Beaumont	275	12.96%	
	Cherbourg	57	2.68%	

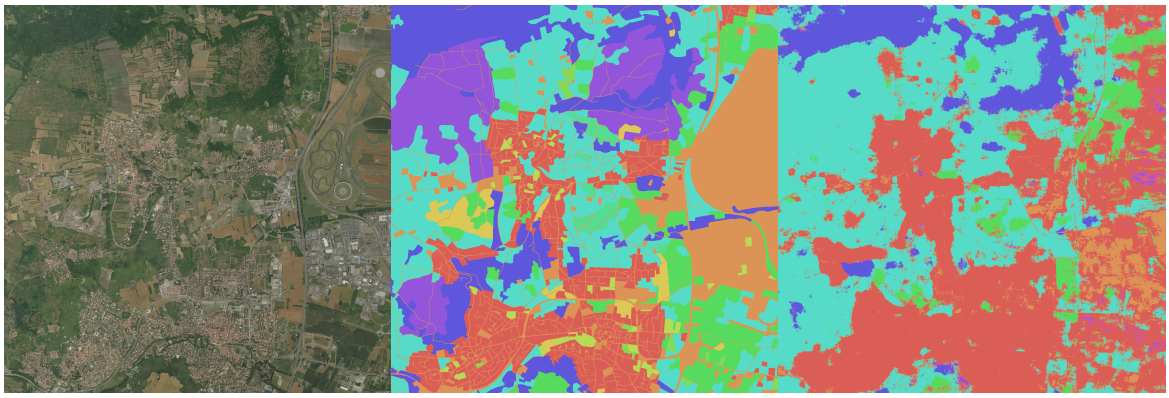
To obtain a baseline result on MiniFrance, we train a first SegNet model for semantic segmentation on the 14 land cover classes from *Urban Atlas*[5] The same hyperparameters are used as in the Chapter 3. Note that the considered classes are significantly more sophisticated than those from the datasets we worked with until now. Indeed the semantics of land covers

[3]Urban Atlas: https://land.copernicus.eu/local/urban-atlas

[4]Cadastre Etalab: https://cadastre.data.gouv.fr/datasets/cadastre-etalab

[5]In practice we only consider 12 classes since "Forests" and "Orchards" are absent from the MiniFrance dataset.

and land uses are abstract concepts generally linked to a whole area and not individual objects. Distinguishing commercial buildings from residential housing requires a more complete understanding of the classes than separating coarse object types (e.g. artificial structures against vegetation). Moreover the annotations from the *Urban Atlas* project are generally less accurate and noisier than the ground truth from the ISPRS dataset. The time gap also introduces new errors due to changes. Finally the diversity of cities and environments exhibited by MiniFrance require from the model a robustness to appearance variations and environmental conditions. From the statistical point of view, MiniFrance shows a significantly larger variance than Vaihingen with huge variations from town to town, as reported in the Tables 6.7a and 6.7b.



(a) RGB image (b) Ground truth (c) Prediction

Figure 6.7: Example of a semantic map generated on MiniFrance. An excerpt from the suburbs of Clermont-Ferrand.

The semantic segmentation metrics achieved by SegNet trained on MiniFrance are reported in Table 6.6. As expected the model obtains relatively low scores, with large F_1 variations between the various classes. Residential, commercial and agricultural land uses are well identified while rarer classes are barely learnt (mines, sport and leisure installations, water bodies...). Overall this first experiment shows that if the coarse classification between various land covers (built areas, vegetation, crops) is achievable, fine land use classification is much more complex. An example of semantic map produced by SegNet on a tile from MiniFrance is pictured in Fig. 6.7. These results remain encouraging considering the task difficulty. Especially the diversity of the observed conurbations and the scale of the dataset make it a very interesting challenge for future remote sensing image interpretation models.

Some of the works presented in this chapter have been presented in an international conference:

- Nicolas Audebert, Bertrand Le Saux, and Sébastien Lefèvre. "Generative Adversarial Networks for Realistic Synthesis of Hyperspectral Samples". In: *2018 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. July 2018, pp. 5091–5094

Table 6.5: Land cover taxonomy from UrbanAtlas 2012.

Code	Land cover	Class	Colour
10000	Artificial surfaces	1	
11000	Urban fabric	1	
11100	Continuous urban fabric	1	
11200	Discontinuous urban fabric	1	
11210	Dense discontinuous urban fabric	1	
11220	Moderately dense discountinuous urban fabric	1	
11230	Sparse discontinuous urban fabric	1	
11240	Very sparse discontinuous urban fabric	1	
11300	Isolated structures	1	
12100	Industrial, commercial, military or transport units	2	
12200	Road and rail networks and areas	2	
12210	Highway network and areas	2	
12220	Other road networks and areas	2	
12230	Railway network and areas	2	
12300	Port areas	2	
12400	Airports	2	
13000	Mines, building and dump sites	3	
13100	Mineral extraction sites	3	
13100	Dump sites	3	
13300	Construction sites	3	
13400	Unoccupied areas	3	
14000	Artificially vegetalized areas	4	
14100	Green urban areas	4	
14200	Sport and leisure facilities	4	
20000	Crops, semi-natural and humid areas	5	
21000	Non-irrigated arable lands	5	
22000	Permanent crops	6	
23000	Pastures	7	
24000	Complex and heterogeneous crops	8	
25000	Orchards	9	
31000	Forests	10	
32000	Mixed herbaceous vegetation	11	
33000	Sparsely or non-vegetated areas	12	
40000	Humid areas (marshes, bogs...)	13	
50000	Water bodies and courses	14	

Table 6.6: Semantic segmentation results of a SegNet model trained on MiniFrance (class-wise F_1 scores and overall accuracy).

Class	1	2	3	4	5	6	7	8	11	12	13	14	Overall
Ensemble	50.89	41.39	0.00	0.00	0.00	56.74	0.84	52.26	64.65	8.70	1.23	0.01	51.97

Table 6.7: Comparison of pixel-wise statistics in Vaihingen and MiniFrance.

(a) Pixel-level channel-wise statistics for Vaihingen.

Ensemble	Mean ± std		
	Infrared	Red	Green
Train	120.30± 54.47	81.64± 38.58	80.52± 36.62
Test	118.07± 56.23	80.69± 41.75	79.70± 40.37
Global	119.74± 54.93	81.40± 39.40	80.32± 37.59

(b) Pixel-level channel-wise statistics for MiniFrance.

Conurbation	Mean ± std		
	Red	Green	Blue
Nice	87.44± 67.04	95.70± 60.50	76.11± 60.19
Nantes, Saint-Nazaire	126.05± 46.64	132.81± 35.85	109.25± 38.09
Le Mans	108.06± 57.10	122.98± 44.05	85.93± 39.32
Lorient	89.43± 62.12	100.80± 53.21	87.12± 52.82
Brest	120.53± 76.08	134.98± 62.98	107.72± 62.76
Caen	127.55± 56.26	134.88± 40.76	114.36± 41.51
Dunkerque, Calais, Boulogne-sur-Mer	133.43± 66.10	138.65± 55.43	123.01± 56.93
Saint-Brieuc	116.91± 61.63	128.37± 50.72	105.12± 52.52
Marseille, Martigues	102.43± 62.58	109.71± 55.51	95.53± 57.45
Rennes	94.82± 46.42	110.57± 36.62	87.34± 28.17
Angers	123.04± 48.27	124.21± 33.14	97.28± 34.77
Quimper	115.04± 72.31	127.73± 58.71	104.76± 56.80
Vannes	75.70± 43.08	84.33± 32.72	68.00± 27.94
Clermont-Ferrand	93.74± 33.58	101.79± 25.79	77.41± 20.21
Lille, Arras, Lens, Douai, Hénin-Beaumont	120.14± 58.20	121.78± 47.45	100.94± 48.30
Cherbourg	123.90± 62.51	127.77± 57.14	114.54± 60.34
Train	115.30± 62.90	124.33± 52.42	101.94± 52.78
Test	106.81± 55.59	113.91± 45.41	93.00± 44.49
Global	110.83± 59.32	118.85± 49.14	97.24± 48.80

References

[1] Jacopo Acquarelli et al. "Convolutional Neural Networks and Data Augmentation for Spectral-Spatial Classification of Hyperspectral Images". In: (Nov. 15, 2017). arXiv: 1711.05512 [cs]. URL: http://arxiv.org/abs/1711.05512 (cit. on p. 136).

[2] Martin Arjovsky, Soumith Chintala, and Léon Bottou. "Wasserstein Generative Adversarial Networks". In: *Proceedings of the International Conference on Machine Learning (ICML)*. International Conference on Machine Learning. July 17, 2017, pp. 214–223. URL: http://proceedings.mlr.press/v70/arjovsky17a.html (cit. on p. 137).

[3] Nicolas Audebert, Bertrand Le Saux, and Sébastien Lefèvre. "Generative Adversarial Networks for Realistic Synthesis of Hyperspectral Samples". In: *2018 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. July 2018, pp. 5091–5094 (cit. on p. 147).

[4] Adela Barriuso and Antonio Torralba. "Notes on Image Annotation". In: (Oct. 12, 2012). arXiv: 1210.3448 [cs]. URL: http://arxiv.org/abs/1210.3448 (cit. on p. 145).

[5] Anko Börner et al. "SENSOR: A Tool for the Simulation of Hyperspectral Remote Sensing Systems". In: *ISPRS Journal of Photogrammetry and Remote Sensing* 55.5-6 (Mar. 1, 2001), pp. 299–312. ISSN: 0924-2716. DOI: 10.1016/S0924-2716(01)00022-3. URL: https://www.sciencedirect.com/science/article/pii/S0924271601000223 (cit. on p. 143).

[6] Yushi Chen et al. "Deep Feature Extraction and Classification of Hyperspectral Images Based on Convolutional Neural Networks". In: *IEEE Transactions on Geoscience and Remote Sensing* 54.10 (Oct. 2016), pp. 6232–6251. ISSN: 0196-2892. DOI: 10.1109/TGRS.2016.2584107 (cit. on p. 136).

[7] Amir Abbas Davari et al. "GMM-Based Synthetic Samples for Classification of Hyperspectral Images With Limited Training Data". In: *IEEE Geoscience and Remote Sensing Letters* 15.6 (June 2018), pp. 942–946. ISSN: 1545-598X. DOI: 10.1109/LGRS.2018.2817361 (cit. on p. 136).

[8] David A. van Dyk and Xiao-Li Meng. "The Art of Data Augmentation". In: *Journal of Computational and Graphical Statistics* (Jan. 1, 2012). DOI: 10.1198/10618600152418584. URL: http://amstat.tandfonline.com/doi/abs/10.1198/10618600152418584 (cit. on p. 136).

[9] Ian Gemp et al. "Inverting Variational Autoencoders for Improved Generative Accuracy". In: *NIPS Workshop on Advances in Approximate Bayesian Inference*. 2017. arXiv: 1608.05983. URL: http://arxiv.org/abs/1608.05983 (cit. on pp. 136, 141).

[10] Ian Goodfellow et al. "Generative Adversarial Nets". In: *Proceedings of the Neural Information Processing Systems (NIPS)*. NIPS. 2014, pp. 2672–2680. URL: http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf (cit. on pp. 136, 137).

[11] Ishaan Gulrajani et al. "Improved Training of Wasserstein GANs". In: *Proceedings of the Neural Information Processing Systems (NIPS)*. NIPS. 2017, pp. 5769–5779. URL: http://papers.nips.cc/paper/7159-improved-training-of-wasserstein-gans.pdf (cit. on p. 138).

[12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Proceedings of the Neural Information Processing Systems (NIPS)*. NIPS. 2012, pp. 1097–1105. URL: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf (cit. on p. 136).

[13] Hyungtae Lee and Heesung Kwoon. "Contextual Deep CNN Based Hyperspectral Classification". In: *2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. IGARSS. Beijing, July 2016, pp. 3322–3325. DOI: 10.1109/IGARSS.2016.7729859 (cit. on p. 136).

[14] Zhiwu Lu et al. "Learning from Weak and Noisy Labels for Semantic Segmentation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.3 (Mar. 2017), pp. 486–500. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2016.2552172 (cit. on p. 145).

[15] Andrew L. Maas, Awni Y. Hannun, and Andrew Y. Ng. "Rectifier Nonlinearities Improve Neural Network Acoustic Models". In: *ICML Workshop on Deep Learning for Audio, Speech and Language Processing*. 2013 (cit. on p. 138).

[16] Luca Maggiolo et al. "Improving Maps from CNNs Trained with Sparse, Scribbled Ground Truths Using Fully Connected CRFs". In: *2018 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. 2018 IEEE International Geoscience and Remote Sensing Symposium (IGARSS). July 2018, pp. 2103–2106 (cit. on p. 144).

[17] Konstantinos Makantasis et al. "Deep Supervised Learning for Hyperspectral Data Classification through Convolutional Neural Networks". In: *Geoscience and Remote Sensing Symposium (IGARSS), 2015 IEEE International*. Geoscience and Remote Sensing Symposium (IGARSS), 2015 IEEE International. July 2015, pp. 4959–4962. DOI: 10.1109/IGARSS.2015.7326945 (cit. on p. 136).

[18] Augustus Odena, Christopher Olah, and Jonathon Shlens. "Conditional Image Synthesis with Auxiliary Classifier GANs". In: *International Conference on Machine Learning*. International Conference on Machine Learning. July 17, 2017, pp. 2642–2651. URL: http://proceedings.mlr.press/v70/odena17a.html (cit. on p. 138).

[19] Fabian Pedregosa et al. "Scikit-Learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (Oct 2011), pp. 2825–2830. ISSN: ISSN 1533-7928. URL: http://www.jmlr.org/papers/v12/pedregosa11a.html (cit. on p. 139).

[20] Tim Salimans et al. "Improved Techniques for Training GANs". In: *Proceedings of the Neural Information Processing Systems (NIPS)*. NIPS. 2016, pp. 2234–2242. URL: http://papers.nips.cc/paper/6125-improved-techniques-for-training-gans.pdf (cit. on p. 139).

[21] Viktor Slavkovikj et al. "Hyperspectral Image Classification with Convolutional Neural Networks". In: *Proceedings of the 23rd ACM International Conference on Multimedia*. ACM Press, 2015, pp. 1159–1162. ISBN: 978-1-4503-3459-4. DOI: 10.1145/2733373.2806306. URL: http://dl.acm.org/citation.cfm?doid=2733373.2806306 (cit. on p. 136).

[22] Tijmen Tielman and Geoffrey Hinton. *Lecture 6.5—RmsProp: Divide the Gradient by a Running Average of Its Recent Magnitude*. 2012 (cit. on p. 139).

[23] Lloyd Windrim et al. "Hyperspectral CNN Classification with Limited Training Samples". In: (Nov. 28, 2016). arXiv: 1611.09007 [cs]. URL: http://arxiv.org/abs/1611.09007 (cit. on p. 136).

Chapter 7

Spatial structure of pixel-wise predictions

Homo sapiens is about pattern recognition, he says. Both a gift and a trap.

— William Gibson (Pattern Recogntion, 2002)

Contents

Summary:

Until now we focused our study on semantic segmentation through the lens of dense pixel-wise classification. The models we used for this task could leverage spatial context, the loss function was a simple average of the classification error on all pixels. However scene understanding can only be achieved through the extraction and the manipulation of concepts linked both through objects and their relationships, independently from the pixels they are made of.

This chapter lookes into various extensions of the fully convolutional network models to leverage spatial structure from the objects of interest in images.

First, we will try to add posterior structure on the pixel-wise predictions computed by the FCNs we train. More specifically we will show that it one can easily structure the pixel-wise prediction generated by the FCNs to accurately detect, segment and recognize vehicles in aerial images.

Second, we will study alternative representations of the ground truth that express the same information will making it possible to enforce spatial structure More precisely, we introduce an alternative loss function that operates on a continuous variant of the ground truth semantic annotations obtained by distance transform. This distance transform regression approach does not replace the standard classification model but instead complements it. In practice, we suggest to train a multitask network that performs simultaneously the euclidean distance trasnform maps regression and the pixel-wise classification. Tis method allows us to couple geometry and semantics in the loss function and regularize the segmentation.

7.1 Segment-before-detect

7.1.1 Regions and objects

Vehicle detection and recognition are two popular tasks addressed in remote sensing. Not only localization and identification of vehicles intervene in aerial surveilance and scene understanding, these informations are also extremely useful for 3D city reconstruction, optical flow estimation and co-registration to detect moving parts on an image and have a better understanding of the static parts and their geometry [31]. A lot of works have been focused on automating vehicle detection in VHR images using a large panel of techniques, ranging from HOG features and SVMs classifiers [37, 16, 27] to 3D pose estimation models [24] and deformable part models [42] or mixtures of rotation-invariant models [43]. Deep learning, and most notably the CNNs, has also been applied to this task [10]. Recently most methods use specifically tuned deep networks designed for object detection with region proposals such as Faster-RCNN [46] for Sommer, Schuchert, and Beyerer [51] or YOLO [45] for Van Etten [57]. However there have been surprinsgly few works that have looked into simultaneous localization and detection despite the introduction of the Vehicle Detection in Aerial Imagery (VEDAI) dataset by Razakarivony and Jurie [44] in 2016 and its baseline using experts features to characterize various vehicle classes in IRRGB images. Some older works mostly addressed this problem as a multi-scale segmentation task, for example combined with fuzzy logic rules [22] and linear discriminant analysis [14]. In particular, Eikvil, Aurdal, and Koren [14] show that performing a pre-segmentation of the image before the vehicle detection step was useful to reduce the number of false alarms.

Inspired by this statement, we suggest to study how, based on modern semantic segmentation techniques using deep networks – such as those introduced in Chapter 3 – we can obtain a complete vehicle detection and classification pipeline by successive refining. We present in this section a complete pipeline of segmentation for vehicle detection dubbed *segment-before-detect*. Going further than the usual bounding boxes generally used for detection, we show how this framework allows us to predict the mask and the type of vehicle instances in aerial image.

Our *segment-before-detect* pipeline is able to extract and classify vehicles based on VHR images in three steps, pictured in the Fig. 7.1:

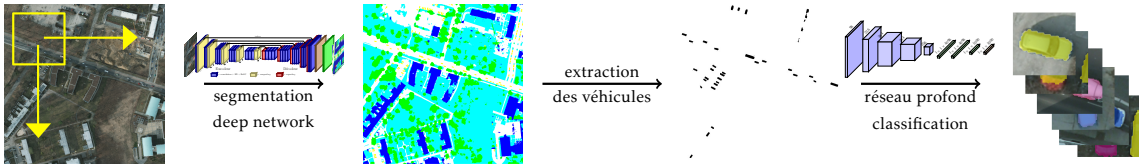
1. Semantic segmentation and inference of the pixel-evel vehicle mask using a FCN,
2. Instance detection by regressing the convex hull of connected components,
3. Vehicle classification of the identified objects using a CNN.

Small objects detection

As we have seen in the Chapter 3, a deep network such as SegNet is generally accurate enough to predict individual vehicles in very high resolution images ($< 50\,\text{cm/px}$). In that case we can simply extract the connected components of the vehicle masks to find the individual vehicle instances in the image. For each connected component, computing a bounding box is fast and simple.

However the predictions coming out from SegNet are often noisy. Especially CNNs applied to Earth Observation images tend to produce blurry inter-class boundaries [35]. Consequently we remove many false positives and separeate vehicles that might have been fused in the same connected component (i.e. the same "blob") by applying a morpholgocial opening on the semantic mask produced through SegNet (cf. Fig. 7.2)[1]. We then remove

[1] The loss of the exact object edges is not critical since we will classify the objects using a patch-based approach, centered on the connected component.



Sliding window on the RGB image | Semantic map | Vehicle mask | Classified vehicles

Figure 7.1: Illustration of the *segment-before-detect* pipeline for vehicle segmentation, detection and classification.

RGB image and segmentation | Vehicle mask | Mask after opening | Connected components

Figure 7.2: Localization of vehicle instances using a morphological opening and connected components extraction.

all objects with a surface lesser than a strict threshold to remove false alarms that would correspond to objects smaller than typical vehicles and that actually are misclassifications (e.g. air conditioners on roofs, garbage bins, etc.). Although this post-processing is very simple, it significantly improves the detection performance of the SegNet model.

Vehicle recognition using a CNN

Provided that we able to detect and find the location of the vehicles, the next step consists in finding their type. Therefore we hope to find whether the vehicle is car, a truck, a van, etc. This is a standard image classification problem that CNNs are well-known to solve efficiently. We use the classical approach that consists in fine-tuning a CNN [38, 61] model that has been pretrained on the ImageNet dataset [48] for vehicle recognition.

More specifically we compare the popular models from the literature used for small image classification ($\simeq$30 px $\times$ 30 px): LeNet [32], AlexNet [29] and VGG-16 [50].

Our goal is to train these classifiers on a large vehicle dataset (source domain) and then apply it on new data from another scene (target domain). Therefore we might be faced with the overfitting problem. Indeed we are looking to transfer knowledge from one dataset to another. To improve the model generalization ability, we can employ two techniques: domain adaptation and data augmentation. Domain adaptation will try to minimize the difference between the training dataset and the inference samples. Data augmentation will try to generate new synthetic training samples to robustify the classifier and improve its generalizability.

In our case, we suggest to normalize all vehicles so that they all exhibit the same azimuth, i.e. that all images containing a vehicle are aligned the same way, with the vehicle appearing horizontally. During training, we use bounding boxes (actually, convex hulls estimated from the segmentation mask) to estimate the vehicle direction and then we rotate the whole image to realign all the samples.

Finally we also increase the size of the training set using geometrical augmentation techniques to have more diverse samples: horizontal and vertical translations ($\pm$10 px), zooms

identity 90°rotation 180°rotation horizon. sym. vertical sym. zoom translation

Figure 7.3: Data augmentation on a vehicle from the VEDAI dataset.

(up to 1.25×), rotations (90°, 180°and 270°) and axial symetries, as pictured in the Fig. 7.3. When the realignement strategy is applied, we only consider the 180° rotation.

7.1.2 Vehicle segmentation

To train a CNN for vehicle classification, we have to possess a sufficiently large labeled dataset. To achieve this we rely on the VEDAI dataset [44] (cf. Appendix A.1.5) that contains many annotated vehicles in aerial images. VEDAI is used to train the initial CNN used for vehicle recognition, that will be applied at inference time on the ISPRS Potsdam dataset (cf. Appendix A.1.1). Classification results on VEDAI are obtained using a 3-fold cross-validation.

To validate our approach, we start by using the ISPRS Potsdam (cf. Fig. 7.4) dataset on which we manually annotated the vehicles into four sub-categories: cars, vans, trucks and pick-ups. The vehicles that were initially labeled in the "clutter" class (e.g. heavy duty construction vehicles) in the original ground truth are excluded. As reported in the Table 7.1, the dataset is mostly comprised of cars (94% of the vehicles).

We train a SegNet for semantic segmentation on this dataset and then apply the CNN pretrained on VEDAI for vehicle recognition. Results are obtained through cross-validation using 18 tiles for training and 6 tiles for validation. The spatial resolution from Potsdam is interpolated to 12.5 cm/px to match the GSD from VEDAI instead of the original 5 cm/px.

Then, we consider the NZAM/ONERA Christchurch dataset (cf. Appendix A.1.6). We recall taht the ground truth on this dataset is coarser than the one from Potsdam and is closer to the bounding polygon generally annotated for object detection tasks, as shown in the Fig. 7.4. We also extend this ground truth by manually annotating the vehicles into the same four sub-categories: cars, trucks, vans and pick-ups. Once again, the dataset is dominated by the car class (cf. Table 7.1), although this not surprising since most vehicles as tourism vehicles.

Since the NZAM/ONERA Christchurch ground truth is actually a set of shapefiles, i.e. polygons that might interesect, for trees, buildings and vehicles, we rasterize them into semantic maps. To do so we defined four classes of intrest: background, buildings, vegetation and vehicles. We build a dense ground truth by labeling first pixels that belong to a building bounding polygon, then pixels that belong to a vehicle bounding polygon and then pixels belonging to vegetation. The remaining pixels are assigned to the background class. This order accounts for the presence of vehicles on rooftop parking lots and that tree-like vegetation can mask cars – and in extreme cases buildings. To take into account the uncertainty on the bounding boxes around the objects, we eroded the ground truth by a disk of radius 5 px along the edges (about 60 cm). We deactivate learning on those pixels.

We train a standard SegNet for semantic segmentation on this dataset and then apply the CNN pretrained on VEDAI for vehicle recognition. The resutls are obtained by cross-validation using 3 tiles for training and 1 tile for validation. The spatial resolution is once again interpolated at 12.5 cm/px to match the GSD from VEDAI instead of the initial 10 cm/px. The hyperparameters described in the Chapter 3 are reused for this experiment.

Semantic segmentation

We report in the Table 7.2 the detailed F_1 scores and overall accuracy of the SegNet model trained on Potsdam. Let us recall that these results are obtained at resolution 12.5 cm/px, yet

(a) ISPRS Potsdam (b) NZAM/ONERA Christchurch

Figure 7.4: Annotations on the two datasets used for this experiment.

Table 7.1: Number of vehicles for each class in the three datasets.

Dataset	Cars	Trucks	Vans	Pick-ups	Boats	Camping-cars	Others	Planes	Tractors
VEDAI	1340	300	100	950	170	390	200	47	190
ISPRS Potsdam	1990	33	181	40	-	-	-	-	-
Christchurch	2267	73	120	90	-	-	-	-	-

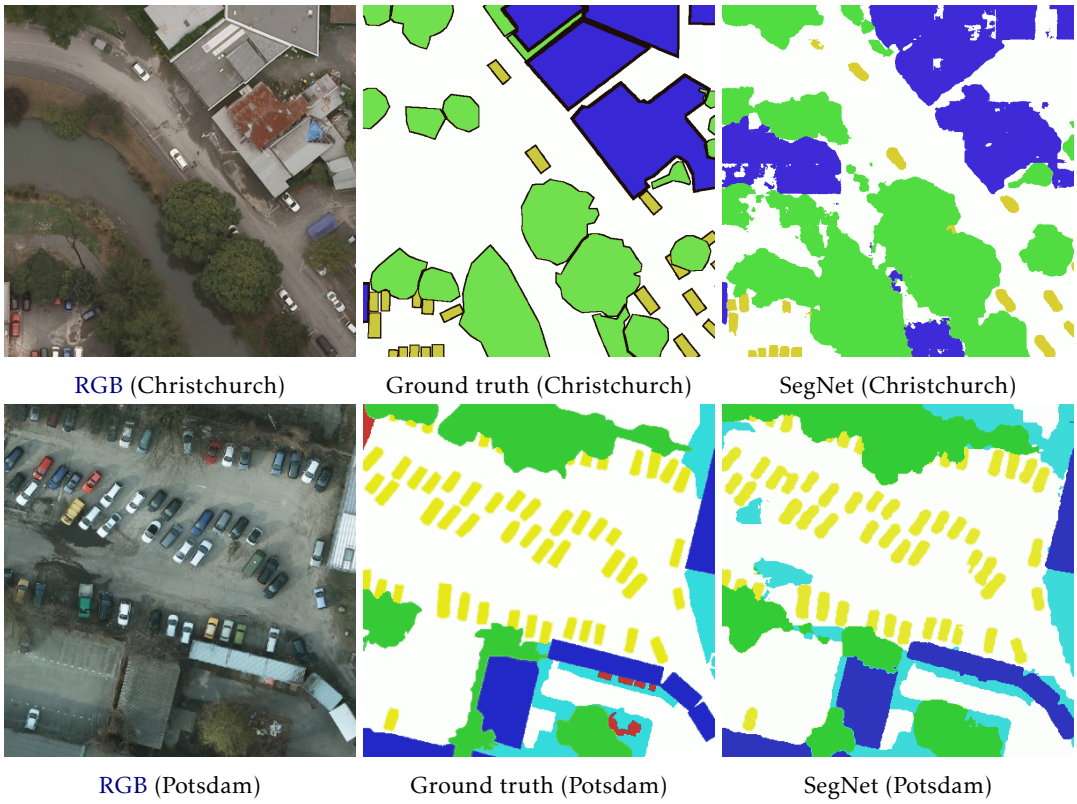
there are very close to those previously obtained using the initial 5 cm/px GSD. A qualitative segmentation sample is shown in the Fig. 7.5.

As illustraed by the Table 7.3, our SegNet model trained on the NZAM/ONERA Christchurch dataset achieves a F_1 score of 61.9% on vehicles, which is sufficient for our purpose and reasonable considering the coarseness of the ground truth compared to Potsdam. This finding has practical value since it shows that one can learn semantic segmentation models with coarse annotations, e.g. polygonal bounding boxes that were initially designed to train detection models. Inference on a tile from Christchuch takes approximately 120 seconds on a GPU NVIDIA Tesla K20c. A sample segmentation is exhibited in the Fig. 7.5.

7.1.3 Vehicle detection

We apply on the predictions inferred on both datasets a morphological opening of radius 3 px ($\simeq$ 35 cm uncertainty on the predicted vehicle shapes) to separate vehicles that might have been merged in the same connected component. We also filter out all connected components that cover less than 1.5 m^2 (100 px). Indeed an average tourism car covers around 4 m^2. Considering that occlusions might recover up to 60% of the vehicle, we set the treshold at 1.5 m^2. We then extract all connected components from the vehicle mask and we compute the convex hull and the bounding box for each component. On the ISPRS Potsdam dataset, as the initial ground truth is comprised of dense pixel-wise annotations, we also regress bounding boxes for each connected component and we manually corrected the occasional errors to obtain a detection ground truth.

We follow the usual guidelines for evaluating object detectors [15]: a true positive is defined as a predicted bounding box that whose intersection over union (IoU) with a bounding box from the ground truth is greater than 0.5. If several predictions exist for the same vehicle, we keep the one is the highest IsU and label the remaining predictions as false alarms. On the NZAM/ONERA Christchurch dataset, we use as a baseline the results obtained by Randrianarivo et al. [43] on the tile they selected for evaluation. Their model consists in a *Discriminatively trained Model Mixture* (DtMM) comprised of five models, one for each of the principal orientations.



RGB (Christchurch) | Ground truth (Christchurch) | SegNet (Christchurch)

RGB (Potsdam) | Ground truth (Potsdam) | SegNet (Potsdam)

Figure 7.5: Semantic segmentation samples obtained on Potsdam and Christchurch.
Colors: white: roads, blue: buildings, cyan: low vegetation, green: trees, yellow: vehicles, red: clutter.

Table 7.2: Semantic segmentation results on the ISPRS Potsdam dataset at 12.5 cm/px (F_1 scores and overall accuracy).

Dataset	Method	Imp. surfaces	Buildings	Low veg.	Trees	Vehicles	Accuracy
Validation 12.5 cm/px	SegNet RGB	92.4± 0.6	95.8± 1.9	85.8± 1.3	83.0± 2.1	95.7± 0.3	90.6± 0.6
Test 5 cm/px	SegNet IRRG	92.4	95.8	86.7	87.4	95.1	90.0
	FCN + CRF [49]	91.8	95.9	86.3	87.7	89.2	89.7

To understand the influence of the morphological opening preprocessing on the instance segmentation task, we compute and report in Table 7.4 the average IsU obtained on vehicle instances and precision/recall metrics for various preprocessing combinations. This shows that the naive morphological opening and small object filtering (inferior to 100 px) significantly improves the model accuracy by removing many false positives. This process is especially effective on the NZAM/ONERA dataset where the coarseness of the annotation results in less precise semantic maps at inference time. The complete process achieves a IsU score of 74% on Potsdam and more than 70% on Christchurch.

Finally we also report pure detection metrics in the Table 7.5. On Christchurch, our *segment-before-detect* pipeline achieve results significantly better than the two baselines: mixture of models and HOG+SVM. Although no other vehicle detection method had been applied on Potsdam yet, we also indicate our precision/recall scores on this dataset to set a future baseline. Qualitative detection examples are illustrated in the Fig. 7.6.

On Christchurch, where annotations are very coarse, our pipeline achieves a 0.81 F_1 score, and over 0.87 on Potsdam. In comparison the sophisticated deformable part model baseline [42] only reaches a 0.74 F_1 score on the same dataset. Moreover let us remind to the reader that we defined a true positive using the 0.5 IsU threshold. In the literature, it is quite common – considering similar spatial resolutions (< 30 cm) – to find works defining

Table 7.3: Semantic segmentation results on the NZAM/ONERA Christchurch dataset (F_1 scores and overall accuracy).

	Background	Buildings	Vegetation	Vehicles	Accuracy
RGB	75.6± 8.9	91.7± 1.3	55.2± 11.6	61.9± 2.4	84.4± 2.6

Christchurch Christchurch Potsdam

Figure 7.6: Sample detections on Potsdam and Christchurch (true positive are in green, false positive are in red and ground truth is in blue).

a true positive on the small vehicles using a 0.25 threshold instead. On a similar urban aerial dataset, Tang et al. [54] achieves a 0.83 F_1 score using the Faster-RCNN [46] detection network based on region proposals. Posterior to our work, Van Etten [57] introduced a vehicle detection dataset with GSD in the same range (between 10 cm and 1 m) and adapted the YOLO network [45] to aerial images. He achieves a F_1 score of 0.90 and predicts the number of vehicles in an image with a 5% margin (relative error). However this approach only predicts bounding boxes and defines a true positive using the 0.25 threshold, which is much more tolerant the the one we chose in this work. In conclusion it seems that the segmentation before detection pipeline is very competitive with the state of the art, including pure detection methods that have been tuned for bounding box prediction. Especially interesting is the fact that our pipeline can infer complete vehicle masks and not only coarse bounding boxes, even when trained with such annotations.

The Christchurch is more difficult for two reasons. First, the vehicle density is significantly greater than for Potsdam, as the city contains many vehicles that are packed in small areas (e.g. parking lots). Second, the coarse annotations from the ground truth make it hard for the FCN to accurately predict the object boundaries and result in blurry edges, i.e. imprecise vehicle masks (cf. Fig. 7.5, the mIsU on Christchurch is of 66.6% against more than 80% on Potsdam). This combination of factors makes the vehicle instance extraction problem more difficult, yet it still remains achievable using our approach. However there is a tendency of the resulting bounding boxes to contain more than one vehicle. Surprisingly, semantic segmentation metrics remain quite high despite training the network on coarse labels intented for object detection. For this reason the segmentation can be used as an intermediate step for detection tasks, which are currently addressed in the state of the art by sophisticated approaches using deep networks and region-proposal strategies. Moreover the connected component extraction could be significantly improved to generate a more effective bounding box generation strategy, for example by applying the *watershed* on the probability maps [6, 4], or by performing simultaneously the segmentation and the instance prediction inside the deep net [13, 21].

Table 7.4: Instance segmentation and vehicle detection results for various morphological preprocessings (mean Io, precision and recall).

Dataset	preprocessing	mIsU	Precision	Recall
NZAM/ONERA Christchurch	∅	60.0%	0.597	0.797
	Opening	69.8%	0.817	0.791
	Opening + small object removal	70.7%	0.833	0.791
ISPRS Potsdam	∅	70.1%	0.748	0.842
	Opening	73.3%	0.866	0.842
	Opening + small object removal	74.2%	0.907	0.841

Table 7.5: Vehicle detection results on Potsdam and Christchurch.

Dataset	Method	Precision	Recall
NZAM/ONERA Christchurch	HOG+SVM [37]	0.402	0.398
	DtMM (5 models) [43]	0.743	0.737
	Segment-before-detect	**0.833**	**0.791**
ISPRS Potsdam	*Segment-before-detect*	0.907	0.841

Vehicle density estimation

Once the vehicles have been detected in the images, a very simple task consists in counting how many cars there are in a given area. This is a common metric used for traffic estimation, urban planning and so on. We divide the two datasets in a grid of 1000×1000px cells ($125 \times 125\,\mathrm{m}^2$) and we compare the number of predicted vehicles compared to the number of actual vehicles present in the ground truth:

$$\mathcal{E}_{relative} = \frac{|\#\text{ predicted vehicles} - \#\text{ actual vehicles}|}{\#\text{ actual vehicles}} \,. \tag{7.1}$$

Results are averaged and rounded to the nearest integer on both datasets and detailed in the Table 7.6. On Potsdam as on Christchurch, estimations are correct with an error margin of less than 10% (i.e. ±5 vehicles in average). Estimations on Christchurch have a slightly greater error rate due to the less precise segmentation and detection steps.

Once the density has been calculated, it is possible to reduce the size of the cell to produce density maps based on vehicle occupation, as shown in the Figs. 7.7 and 7.8. This type of density maps can then be integrated to a GIS such as OpenStreetMap to automatically identify traffic jams, parking lots [27], road blocks, etc.

7.1.4 Vehicle classification

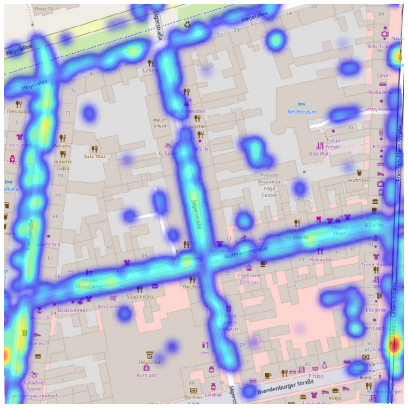
Now that we are able to locate and segment vehicles in the images, we can focus on recognizing the kind of vehicle we are faced with. To do so we compared three CNN architectures of increasing complexity: LeNet [32], AlexNet [29] and VGG-16 [50].

LeNet-5 is a small CNN that we train from scratch using a random initialization on the vehicles from the VEDAI dataset, using image patches of 32×32. AlexNet and VGG-16 are two networks that won the ILSVRC competition in 2012 and 2014. Preliminary experiments showed that initializing these networks using weights obtained after pretraining on ImageNet improved the overall accuracy of 10%, which is in line with previous findings [38, 39]. Consequently we simply fine-tune these CNNs on images of vehicles with dimensions respectively 224×224 and 227×227 for AlexNet and VGG-16. These image sizes are chosen so that we can keep the pretrained weights of the fully connected layers and only retrain the

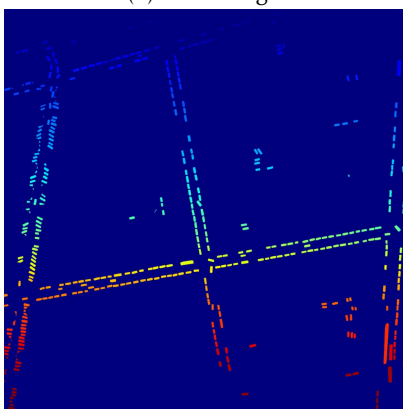
Table 7.6: Mean estimation error of the number of vehicles in a $125\,m^2 \times 125\,m^2$ cell.

Dataset	ISPRS Potsdam	NZAM/ONERA Christchurch
Absolute error (mean error/total from the ground truth)	3/52	6/66
Relative error	7.9%	9.1%

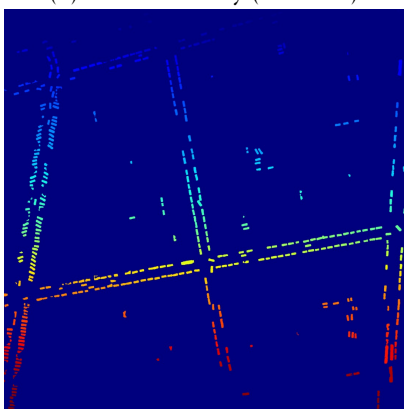
(a) RGB image



(b) Vehicle density (Potsdam)



(c) Ground truth (Potsdam)



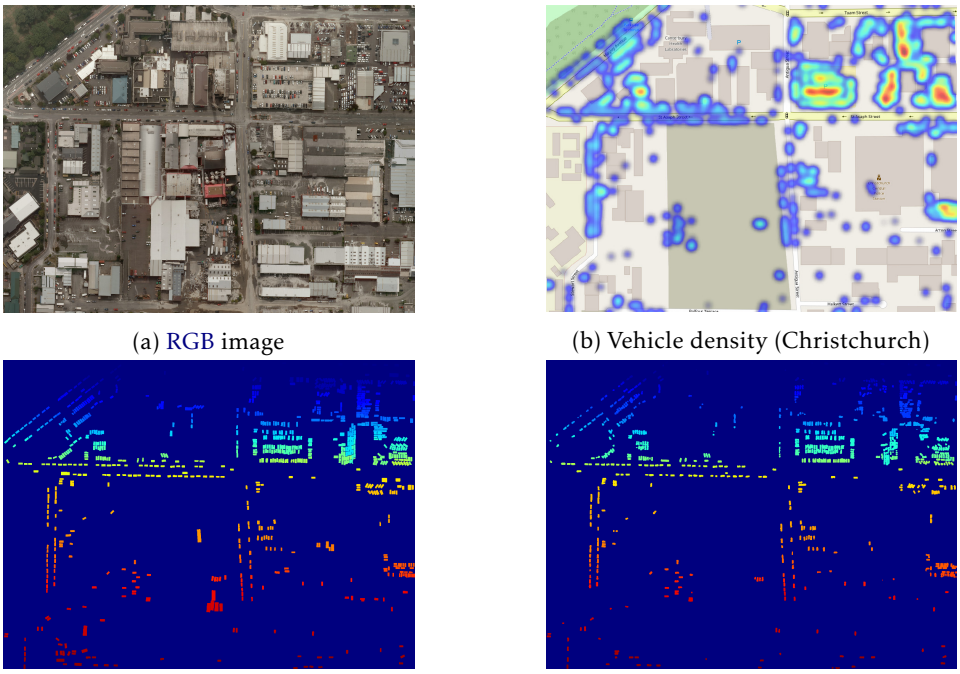
(d) Predicted vehicles (Potsdam)

Figure 7.7: Visualization of the vehicles from one the ISPRS Potsdam tiles.

last layer from scratch. In practice, considering the GSD of our aerial images, vehicles are actually around 25×25. We extract small image patches centered on the detected vehicles including a spatial context of 16 px in all four directions, this approach resulting the highest final accuracies. A larger spatial context tend to include other vehicles that might be close to the one we are zooming in, while a smaller spatial context removes some useful contextual hints. The image patches are then resized using bilinear interpolation so that their biggest dimension matches the one expected by the CNN, the smaller dimension being padded using white noise.

All models are trained (or fine-tuned) for 20 epochs on the dataset using the stochastic gradient descent with momentum algorithm. We use a batch size of 128 samples for AlexNet and LeNet and only 32 for VGG-16 as it has larger memory requirements. The learning rate is initially set to 0.01 and is divided by 10 after 75% of the training has been done. For the models we fine-tune, we retrain the whole network based on the pretrained weights, except the last layer which is randomly initialized and trained with a learning rate 10 times higher than the rest of the network. We apply some *Dropout* [53] on the fully connected layers to alleviate overfitting.

Unsurprisingly the performances of the CNN on VEDAI are proportional to their results



(a) RGB image (b) Vehicle density (Christchurch)

(c) Ground truth (Christchurch) (d) Predicted vehicles (Christchurch)

Figure 7.8: Visualization of the vehicles from one the NZAM/ONERA Christchurch tiles.

Table 7.7: Classification of results of various CNN on VEDAI (in %). OA = *Overall Accuracy*.

Model	Car	Truck	Boat	Tractor	Camping-car	Van	Pick-up	Plane	Others	OA	Time (ms)
LeNet	74.3	54.4	31.0	61.1	85.9	38.3	7.7	13.0	47.5	66.3± 1.7	**2.1**
AlexNet	**91.0**	84.8	81.4	83.3	98.0	**71.1**	85.2	91.4	**77.8**	87.5± 1.5	5.7
VGG-16	90.2	**86.9**	**86.9**	**86.5**	**99.6**	**71.1**	**91.4**	**100.0**	77.2	**89.7**± 1.5	31.7

on ImageNet as reported in the Table 7.7. However the most complex model (VGG-16) only slightly improves the classification metrics and comes with a huge computational overhead. In practice we could use any CNN for this task, including the expensive ResNet [20]. However we are satisfied with the accuracy achieved by AlexNet with respect to its execution time.

The Table 7.8 details the vehicle classification results achived on VEDAI using different preprocessing strategies. Data augmentation using geometrical transforms (noted *DA*) improves the mobel robustness and generalization ability. The realignment strategy *R* also improves the result and makes the network more robust by removing the need to learn a rotation-invariant classification. The combination of these two strategies result in the best metrics, therefore we will use both for our final model.

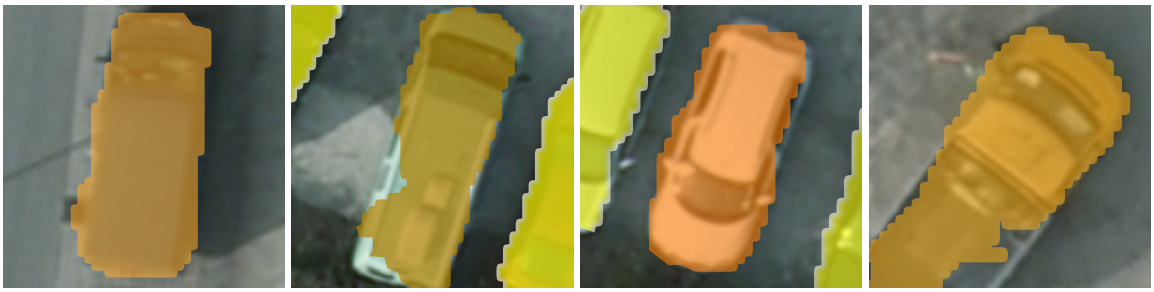
Transfer learning for vehicle classification

At this stage we have an effective vehicle detection model for Potsdam and Christchurch and a classifier trained on VEDAI. The Table 7.9 details the classification metrics achieved by the CNN trained on VEDAI and applied on the vehicles from Potsdam and Christchurch. Results are agregated through cross-validation on the same folds as the ones used for the semantic segmentation step. Since cars are majoritary in the datasets we are using, we also report the metrics that are obtained using a reference heuristic corresponding to a classifier that always predicts“car”. This constant classifier would be right 94% of the time but would fail on all vehicles that are not cars. The overall model accuracy would be stellar but its average accuracy over the various classes would be catastrophic. The CNNs are able to correctly predict several types of vehicles, significantly improving the average accuracy while maintaining a competitive overall accuracy. Some qualitative examples of

Table 7.8: Classification results of AlexNet on VEDAI using various preprocessings (in %). OA = *Overall Accuracy*, AA = *Average Accuracy*.

preprocessing	Car	Truck	Boat	Tractor	Camping-car	Van	Pick-up	Plane	Others	OA	AA
∅	90.4	66.7	80.4	**89.5**	96.6	63.3	78.7	92.6	75.0	83.9 ± 2.7	81.5 ± 1.9
DA	88.2	82.2	78.4	82.5	**97.4**	63.3	85.1	66.7	73.3	85.6 ± 1.4	77.3 ± 8.7
R	87.9	71.1	86.3	84.2	**97.4**	73.3	**87.2**	**100.0**	75.0	86.1 ± 0.9	84.7 ± 1.7
DA + R	**91.4**	**85.6**	**88.2**	87.6	**97.4**	**70.0**	**87.2**	**100.0**	**81.7**	**89.0** ± 0.5	**87.7** ± 1.5

DA = data augmentation, R = realignment.



(a) Van (truck predicted) (b) Van (truck predicted) (c) Voiture (van predicted) (d) Pick-up (van predicted)

Figure 7.9: Successful segmentation but wrong classifications on Potsdam.

successful segmentations but failed subsquent classifications and successful segmentations and classification are given in the Figs. 7.9 and 7.10.

The average accuracy is lower on Potsdam than on VEDAI because of the strong class-unbalance and partly due to the numerical sensitivity of the results. Indeed each train/test fold from the cross-validation contains approximately 15 truck and pick-ups samples. However the model is trained on VEDAI, where the class repartition is not as much unbalanced. Therefore the model learns a bias that does not transfer to Potsdam. Moreover the sensors used for VEDAI, Potsdam and Christchurch are not the same. The environments also differ (urban for Potsdam and Christchurch, rural for VEDAI) and this changes the semantics of the spatial context.

Variations due to sensors have been corrected by renormalizing the image colours in Potsdam and Christchurch. To do so we estimated the pixel-wise statistical moments on VEDAI to apply them on Potsdam and Christchurch:

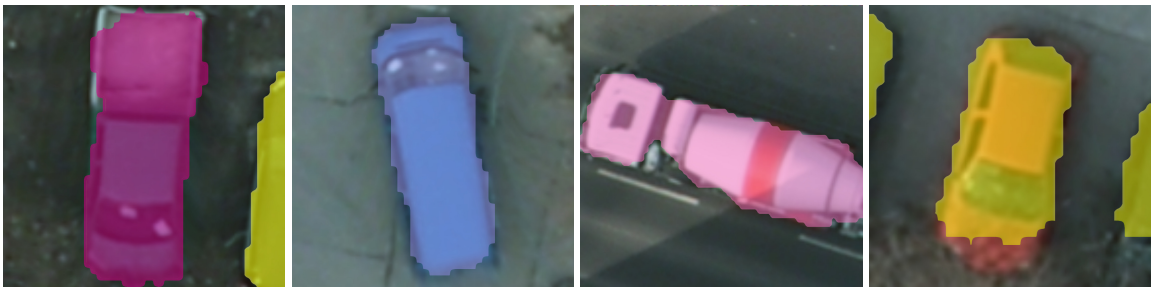
$$\mathrm{I}_{test} := \frac{\mathrm{I}_{test} - m_{test}}{\sigma^2_{test}} \cdot \sigma^2_{VEDAI} + m_{VEDAI} \tag{7.2}$$

where m is the value of the mean pixel in the dataset, σ the standard deviation and I the image to transform. This operation is applied channel-wise.

Despite this normalization the variations in appearance and environmental conditions, including the type of vehicles observed impact the performances. Notably vehicles, trucks and pick-ups from Christchurch are closer to the american bands present in the VEDAI datasets than the european vehicles from Potsdam. These variations from the environment and the objects can make the classifier out of its nominal inference range.

Some kind of regularization or even training on a more diverse vehicle dataset could alleviate all of these adverse effects. This type of problems in transfer learning are related to the more general concept of unsupervised domain adaptation [55, 12], which in itself is an active research field in remote sensing.

Finally, we showed that it is possible to *a posteriori* perform object-based analysis by adding structure on the output of an FCN directly from the pixel-wise classification. Notably we introduced the *segment-before-detect* pipeline with which we were able not only to detect



(a) Pick-up (b) Van (c) Camion (d) Voiture

Figure 7.10: Successful segmentations and classifications on Potsdam.

Table 7.9: Vehicle classification results on the augmented ground truth from Potsdam and Christchurch. OA = *Overall Accuracy*, AA = *Average Accuracy*.

Dataset	Classifier	Car	Van	Truck	Pick-up	OA	AA
Potsdam	Voitures seulement	100%	0%	0%	0%	94%	25%
	AlexNet	**98%**	**66%**	67%	0%	**95%**	58%
	VGG-16	92%	66%	**75%**	**33%**	89%	**67%**
Christchurch	Voitures seulement	100%	0%	0%	0%	94%	25%
	AlexNet	94%	40%	**67%**	**89%**	93%	73%
	VGG-16	**97%**	**80%**	**67%**	78%	**96%**	**80%**

vehicles in aerial images, but also recognizing their shape and identifying their type. In particular this method can be used even when the available annotations are coarse, such as bounding boxes from the NZAM/ONERA Christchurch dataset that were intented for detection. However as we have seen, finding the object instances require to use an *ad hoc* post-processing to regroup pixels that belong the the same object. Indeed, fully convolutional models are optimized to minimize a loss function that is calculated pixel-wise. Such as loss function cannot model correctly the spatial relationships that exist between pixels, especially the fact that several pixels belong to the same object (or the same object part). There seems to be some benefit to be gained by looking into alternative ways to express and enforce these dependencies when training the models.

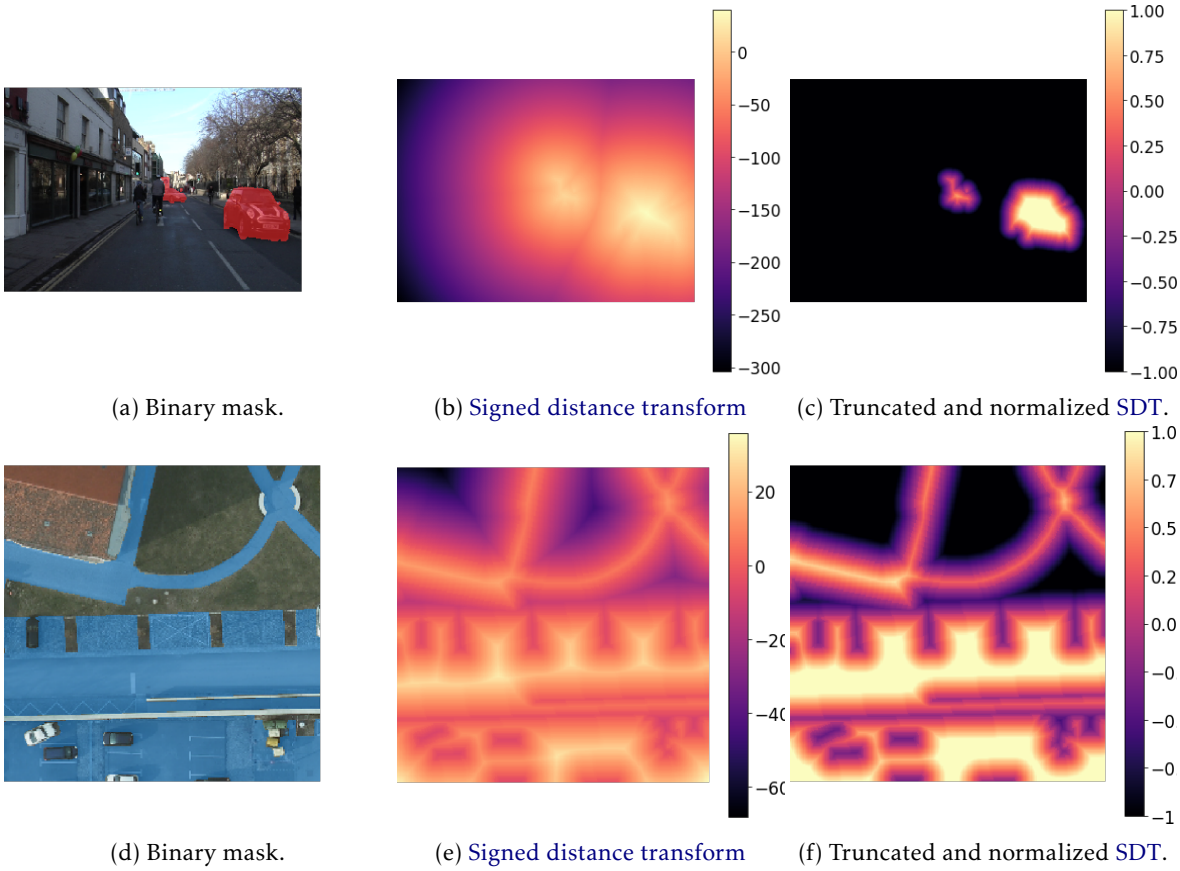
7.2 Distance transform regression for semantic segmentation

7.2.1 Semantic labeling and distance transforms

As we have seen in the previous section, it is possible to reconstruct *a posteriori* the structure at the object level of the semantic maps inferred by an FCN. However the literature often echoes problems related to blurry inter-class boundaries or even noisy segmentation that require post-processing regularizations to smooth the semantic maps [60, 9] or even *ad hoc* heuristics such as our *segment-before-detect* pipeline.

The computer vision community looked into various post-processing strategies that might improve segmentation edges and enforce geometrical constraints to match more closely the ground truth. Often, works on topic rely on graphical models added on top of the network [33] or expert knowledge [30, 5]. In particular, instance segmentation models have been designed to combine geometrical object localization with semantic recognition [21, 13].

We present here a direct approach that introduces an implicit regularization embedded in the ground truth representation. Indeed we suggest to use the regression of the distance transform computed on the ground truth semantic masks as an auxiliary task. Distance transforms express not only the fact that a pixel belong to a specific class, but also its



(a) Binary mask. (b) Signed distance transform (c) Truncated and normalized SDT.

(d) Binary mask. (e) Signed distance transform (f) Truncated and normalized SDT.

Figure 7.11: Equivalent representations of annotated segmentations.

proximity to other classes of interest and therefore can be used to model relationships between objects, a information more complete than the usual binary masks. This method is inspired by many works that have looked into using geometrical primitives for regularization in semantic segmentation, such as predicting object orientations [56] or the position of their center of mass [17]

By lightly modifying existing segmentation networks, we are able to obtain segmentations that are smoother without post-processing or prior knowledge.

We validate our approach on several fully convolutional architectures and for various applications in urban scene understanding, 2.5D image segmentation and Earth Observation.

Regularization through distance transform regression

The distance transform is an operation that transforms a binary mask into a strictly equivalent continuous representation. In our case, we work with signed truncated distance transform that we normalize in $[-1,+1]$. These reprsentations of the annotations are illustrated in the Fig. 7.11. We suggest that, although equivalent to the binary masks, this information expresses more explicitely the spatial structure of the images, since each pixel now contains its spatial distance to all classes of interest. This explicit representation makes more apparent the geometry of the scene as it is clear if a pixel is located along a class boundary or in the middle of an object. We argue that regressing the signed distance transforms (SDT) as an auxiliary task in semantic segmentation network has beneficial effects on the segmentation metrics.

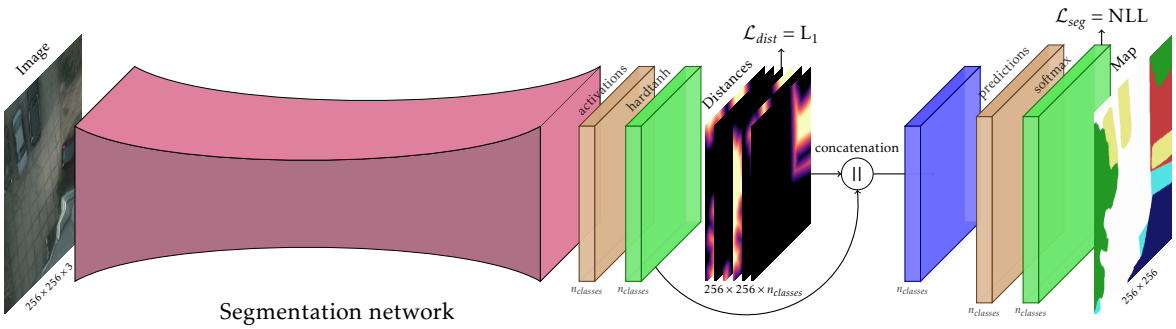


Figure 7.12: Multi-task learning (pixel-wise classification and distance transfrom regression). The convolutional layers are in blue, non-linear activations are in green and activation maps are in brown.

Distance transforms

Distance transforms (or distance maps) of a binary image assign to each pixel of the grid its distance to the nearest point that belongs to the mask of postivie values. This distance can be calculated based on various metrics, such as Manhattan distance or the euclidean distance. By convention, elements that belong to the foreground (positive mask) have a distance of 0. For example, the euclidean distance transform $\mathcal{D}$ maps an imagee I of shape $M \times N$ with a postivie mask I^+ into a distance map $\mathcal{D}(I)$ related by:

$$\forall i, j \in M \times N, \quad \mathcal{D}(I)[i,j] = \min_{I_{i',j'} \in I^+} (\| I[i,j] - I[i',j'] \|) \quad . \tag{7.3}$$

We will use in practice the signed distance transform [**q._z._ye_signed_1988**] that maps for each foreground pixel its positive distance to the nearest background pixel and for each background pixel the opposite of its distance to the nearest foreground pixel. Mathematically, it is defined as the transform $\mathcal{D}_s$ that maps an image I to the distance map:

$$\forall i, j \in M \times N, \quad \mathcal{D}_s(I)[i,j] = \begin{cases} +\min_{I_{i',j'} \in I^-} (\| I[i,j] - I[i',j'] \|), & \text{if } I[i,j] \in I^+, \\ -\min_{I_{i',j'} \in I^+} (\| I[i,j] - I[i',j'] \|), & \text{if } I[i,j] \notin I^+. \end{cases} \tag{7.4}$$

Usual semantic segmentation annotations exists in the "one binary mask per class format. Therefore one can convert these labels into continuous SDT counterparts. Let us stress that there is no information loss in this process, as the binary masks can be retrieved exactly by a simple thresholding of the SDT. We apply the signed distance transform to the semantic segmentation annotations using the exact linear-time algorithm from Maurer, Qi, and Raghavan [36].

To reduce side effects when pixels are too far from other objects, therefore moving out of the network receptive field, we add a saturation to the distances that we compute. More specifically the SDT is dividided by a scale factor – depending on the network receptive field – and then normalized in $[-1, +1]$ using the hardtanh which effectively clips the values going over the positive threshold (or under the negative one). These different representations are pictured in the Fig. 7.11.

7.2.2 Multi-task learning

The direct regression of the SDT does not improve the segmentation results compared to the usual dense pixel-wise classification in our preliminary experiments. However we suggest the employ of a multi-task learning strategy in which the network is trained both on pixel-wise classification and SDT regression.

More specifically, we alter the network architecture to first perform the regression of the SDT; then we add a new convolutional layer that fuses activations from the previous layers

with thse predicted distance maps in order to obtain the final classification (i.e. the semantic maps). The full network is then trained in a multi-task fashion, the SDT regression being used as a proxy task for classification.

The network is modified as follow. The last layer, usually followed by a *softmax*, is here used as a regression layer to predict the SDT. Since the distance havee been normalized in $[-1,+1]$, this layer uses the *hardtanh* non-linearity. Then we concatenate the activations from previous layers to the SDT predicted. The resulting tensor is fed to an additional convolutional layer followed by a *softmax* that performs the pixel-wise classification. The complete architecture is detailed in the Fig. 7.12. To achieve a fair comparison between the different models, all baselines used in this work include the same additional convolutional layer so that both original and altered networks contain the same number of trainable parameters.

The loss functions used in this work are the negative log-likelihood (NLL) expressed as the cross-entropy for classification and the L_1 for distance regression. Let $Z_{seg}, Z_{dist}, Y_{seg}, Y_{dist}$ respectively denote the classification after *softmax*, the predicted distance map, the ground truth annotations and the actual distance map. The global cost function we aim to minimize is:

$$\mathcal{L}_{totale} = \mathrm{NLL}(Z_{seg}, Y_{seg}) + \lambda \cdot \mathrm{L}_1(Z_{dist}, Y_{dist}) \tag{7.5}$$

where λ is a balancing hyperparametere that controls the regularization strength.

7.2.3 Experiments

To evaluate the effect of the distance transform regression, we train several networks based on either the SegNet [3] or PSPNet [59] reference architectures using various configurations ranging from pure regression to pure classification.

The SegNet encoder-decoder architecture [3] has already been detailed in Chapter 3. PSPNet [59] is a fully convolutional architecture that outperformed the current state of the art on several semantic segmentation datasets [11, 15]. Is is built on the ResNet model [20] and includes the concenation of a pyramidal activation tensor to learn from multiple spatial contexts. In our case we consider the reduced PSPNet based on the ResNet-101 model. As ResNet-101 generates feature maps at resolution 1:32, the resulting tensors are upsampled by deconvolution.

Datasets

Table 7.10: Cross-validated results on the ISPRS datasets (multi-task). Values reported indicate the overall accuracy and the class-wise F_1 score.

Method	City	Acc.	Roads	Buildings	Low veg.	Trees	Cars
SegNet* (regression)	Vaihingen	89.49	91.03	95.60	81.23	88.31	0.00
SegNet* (classification)		90.00	91.98	95.53	80.91	88.07	87.94
SegNet* (multi-task)		**90.43**	**92.46**	**95.99**	**81.30**	**88.34**	**88.16**
SegNet* (classification)	Potsdam	91.85	94.12	96.09	88.48	85.44	96.62
SegNet* (multi-task)		**92.22**	**94.33**	**96.52**	**88.55**	**86.55**	**96.79**

We validate our approach on several datasets to demonstrate its capacity to generalize on various binary and multi-label classification settings on multiple image types.

ISPRS 2D Semantic Labeling The ISPRS 2D Semantic Labeling dataset [47] is comprised of the two scenes from Potsdam and Vaihingen already introduced in the previous chapters and detailed in the Appendix A.1.1. The evaluation is achieved by 3-fold cross-validation.

INRIA Aerial Image Labeling Benchmark The INRIA Aerial Image Labeling dataset [34] consists in 360 RGB images of size 5000 px × 5000 px at a 30 cm/px GSD for 10 cities randomly selected on the planet. Half of the cities are selected for training and the relevant ground truth annotations for building footprints are publicly released. The remainder of the dataset is kept hidden for evaluation. More details are given in the Appendix A.1.4.

CamVid The CamVid dataset [7] is comprised of 701 images extracted from multiple videos acquired by a camera embedded in a car with a 360 px × 480 px resolution. We reuse the reference train/test split [3], i.e. 367 training images, 101 validation images and 233 test images. Annotations have been released for 11 classes of interest such as "building", "pedestrian", "car" or "sidewalk". More details are listed in the Appendix A.2.1.

SUN RGB-D The SUN RGB-D dataset [52] contains 10 335 RGB images alongside their depth map. These images have been annotated for 37 classes of interest covering the furniture, walls, groundl... Its goal is to provide a benchmark for semantic segmentation of indoor autonomous navigation images, with objects located at less than 10 m. More information regarding this dataset is available in the Appendix A.2.2.

Data Fusion Contest 2015 The Data Fusion Contest 2015 dataset [8] consists in 7 aerial images of shape 10 000 px × 10 000 px at resolution of 5 cm/px, acquired on the town of Zeebruges (Belgium). 8 classes of interest (the 6 classes from the ISPRS dataset, water and boats) are labeled. We keep two images for the test set, one image for the validation set and the rest is used for training. More details about this dataset are given in the Appendix A.1.2.

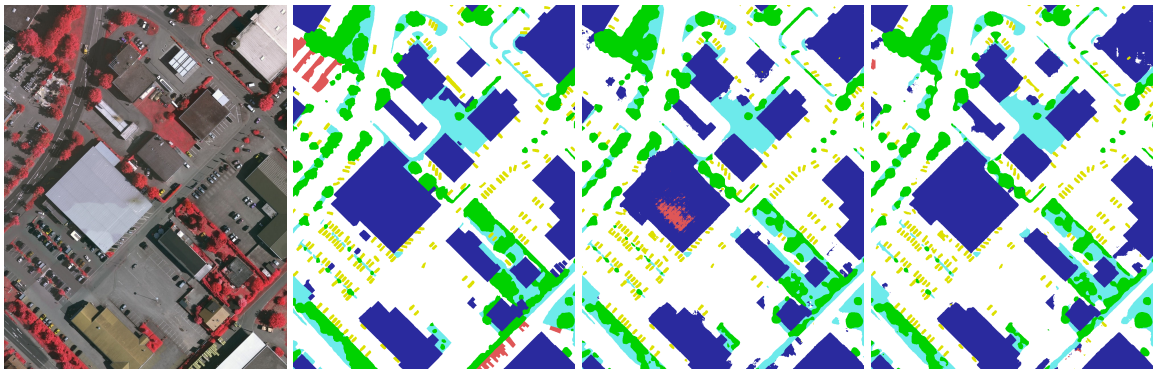
Experimental setup

The SegNet and PSPNet-101 architectures are trained and used as follows. SegNet is trained for 50 000 iterations on batches of size 10. We use the stochastic gradient descent algorithm with an initial learning rate of 0.01, divided by 10 after 25 000 and 45 000 iterations. The weights of the encoder are initialized using the pretrained weights of VGG-16 [50] on ImageNet. The decoder weights are randomly initialized using the policy from He et al. [19]. On the SUN RGB-D dataset, we use the FuseNet model [18] to validate our approach in a multi-modal setting. This model uses a dual-stream SegNet that learns a joint representation of both the color image and the depth map (cf. Chapter 5). On aerial images, we perform data augmentation during training by randomly cropping 256 × 256 patches (384 × 384 for the INRIA Aerial Image Labeling dataset) from the high resolution tiles in addition to random mirroring and flipping. Inference is performed using a sliding window using the same shape and a 75% overlap.

We train a PSPNet on CamVid for 750 000 iterations on batches of 10 images using stochastic gradient descent with an initial learning rate of 0.01, divied by 10 after 500 000 iterations. We randomly crop 224 × 224 patches from the images and apply random mirroring (horizontal symetry). Following the practice from [25], we then fine-tune the network on the full-scale images for 200 000 iterations. Our implementation of PSPNet is based on ResNet-50 pre-trained on ImageNet and do not use the auxiliary classification loss for deep supervision [59].

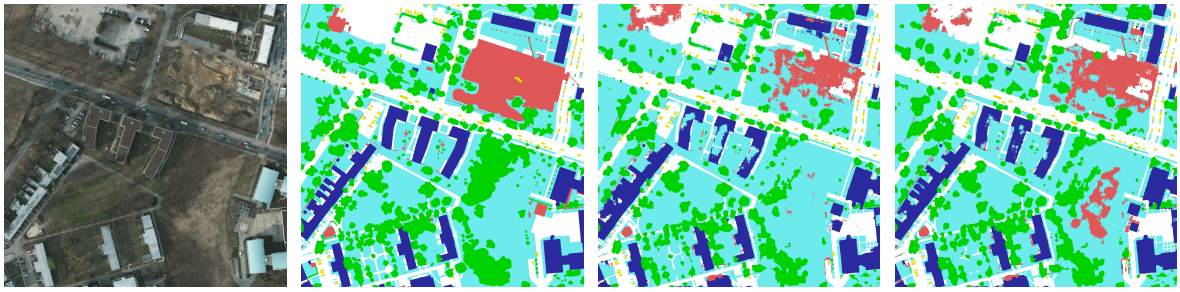
Finally, we use median-frequency balancing to alleviate the class unbalance from SUN RGB-D and CamVid.

All experiments are performed using the PyTorch library [40]. The SDT are computed on CPU using the Scipy library [26] are cached in memory on-the-fly to avoid recomputation. Calculating the SDT slightly slows down the training during the first epoch before they are stored in memory. For actual high-performance applications, an on-line GPU implementation [58] could make this high memory overhead disappear while removing the extra computation time.



IRRG image | Ground truth | SegNet (classification) | SegNet (multi-task)

Figure 7.13: Excerpt of the segmentation results on the ISPRS Vaihingen dataset.
Colors: white: roads, blue: buildings, cyan: low vegetation, green: trees, yellow: vehicles, red: clutter.



RGB image | Ground truth | SegNet (classification) | SegNet (multi-task)

Figure 7.14: Excerpt of the segmentation results on the ISPRS Potsdam dataset.
Colors: white: roads, blue: buildings, cyan: low vegetation, green: trees, yellow: vehicles, red: clutter.

Results

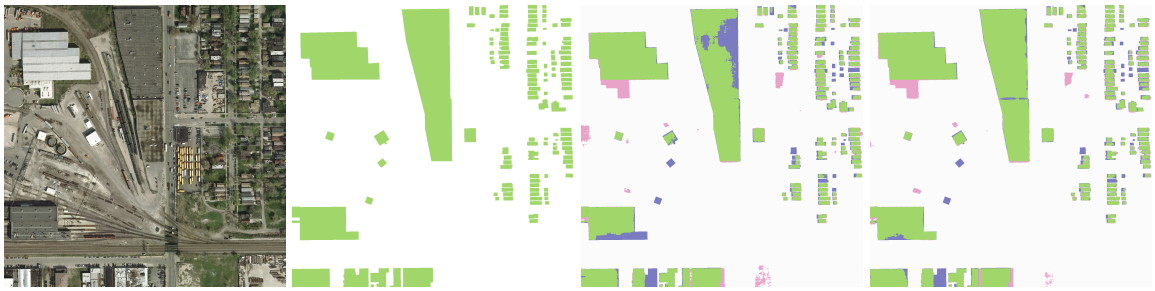
Models suffixed by an asterisk ("*") in the Tables 7.10 to 7.12 and 7.14 are those we introduced and implemented ourselves in this study. The other models are reference baselines from the state of the art.

ISPRS The cross-validated results on the ISPRS Vaihingen and Potsdam datasets are reported in Table 7.10. All classes seem to benefit from the distance transform regression. On Potsdam, the class "trees" is significantly improved as the distance transform regression forces the network to better learn its closed shape, despite the absence of leaves that make the underlying ground visible from the air. Two example tiles are shown in Fig. 7.13 and Fig. 7.14, where most buildings strongly benefit from the distance transform regression, with smoother shapes and less classification noise. Moreover, we also tested to perform regression only on the Vaihingen dataset, which slightly improved the results on several classes, although it missed all the cars and had a negative impact overall. It is also worth noting that our strategy succeeds while CRF did not improve classification results on this dataset as reported in [35].

INRIA Aerial Image Labeling Benchmark Results on the INRIA *Aerial Image Labeling* dataset are detailed in the Table 7.11. Using the distance transform regression on this task significantly improves the IsU score. As pictured in the Fig. 7.15, building shapes are more regular in the multi-task prediction setting and fit better to the original objects. Buildings that were already detected are overall cleaner. Our results are competitive with those from other methods of the first year of the benchmark [23] that use various tricks such as soft-Jaccard index as a loss function.

Table 7.11: Building extraction results on the INRIA *Aerial Image Labeling* dataset.

Method	Bellingham		Bloomington		Innsbruck		San Francisco		East Tyrol		Global	
	Io	Exac.	Io	Exac.	Io	Exac.	Io	Exac.	Io	Exac.	Io	Exac.
AMLL [23]	67.14	96.64	65.43	96.73	72.27	96.66	**75.72**	**91.80**	74.67	97.70	**72.55**	**95.91**
NUS [23]	**70.74**	**97.00**	66.06	96.74	**73.17**	**96.75**	73.57	91.19	**76.06**	**97.81**	72.45	95.90
Raisa [23]	68.73	96.79	60.83	96.23	70.07	96.31	70.64	89.52	74.76	97.64	69.57	95.30
Inria [34]	56.11	95.37	50.40	95.27	61.03	95.37	61.38	87.00	62.51	96.61	59.31	93.93
SegNet* (classif.)	63.42	96.11	62.74	96.20	63.77	95.44	66.53	89.18	65.90	96.76	65.04	94.74
SegNet* (multi)	68.92	96.94	**68.12**	**97.00**	71.87	96.72	71.17	89.74	74.75	97.78	71.02	95.63



RGB image | Ground truth | SegNet (classification) | SegNet (multi-task)

Figure 7.15: Excerpt of the segmentation results on the INRIA *Aerial Image Labeling* dataset. True positives are in green, false postivies are in pink and false negative are in blue. The multi-task setting produces maps that respect more closely the object structures.

SUN RGB-D We report in the Table 7.12 the detailed segmentation results on the SUN RGB-D dataset. Moving from the classification FuseNet to the multi-task network imroves the average precision and the overall accuracy with a very slight decrease in IsU. These results show that using the distance transform regression can also be useful in multi-modal settings with dual stream networks. Moreover our results are competitive with those obtained by Qi et al. [41] using a sophisticated 3D graph convolutional network that learns from a richer information.

Data Fusion Contest 2015 The Table 7.13 reports semantic segmentation results obtained by training SegNet with and without SDT regression on the DFC 2015 dataset. As a baseline, the best approach from the initial contest is also reported [8]. The quantitative improvements in all metrics are close to those obtained on the ISPRS Vaihingen and Potsdam datasets. Indeed most classes benefit from the regularization, especially the vegetation; the annotations on the vegetation are clearly more regular in the ground truth compared to the chaotic nature of the actual trees. Overall the model accuracy is improved by 0.64% in the multi-task

Table 7.12: Results on the SUN RGB-D dataset (224 px × 224 px images).

Method	Accuracy	Io	Precision
3D Graph CNN [41]	-	42.0	55.2
3D Graph CNN [41] (multiéchelle)	-	**43.1**	55.7
FuseNet* [18]	76.8	39.0	55.3
FuseNet* (multi-task)	**77.0**	38.9	**56.5**

Table 7.13: Semantic segmentation results on the DFC 2015 dataset (class-wise F_1 scores and overall accuracy).

Method	Acc.	Roads	Buildings	Low veg.	Trees	Cars	Clutter	Boats	Water
AlexNet (*patch*-based) [8]	83.32	79.10	75.60	78.00	79.50	50.80	63.40	44.80	98.20
SegNet (classification)	86.67	**84.05**	**82.21**	82.24	69.10	79.27	65.78	**56.80**	98.93
SegNet (multi-task)	**87.31**	84.04	81.71	**83.88**	**80.04**	**80.27**	**69.25**	50.83	**98.94**

Table 7.14: Semantic segmentation results on the CamVid.

Method	Io	Acc.	Building	Tree	Sky	Car	Sign	Road	Pedestrian	Fence	Pole	Sidewalk	Biker
SegNet [3]	46.4	62.5	68.7	52.0	87.0	58.5	13.4	86.2	25.3	17.9	16.0	60.5	24.8
DeepLab-LFOV [9]	61.6	–	81.5	74.6	89.0	**82.2**	42.3	92.2	48.4	27.2	14.3	75.4	50.1
DenseNet56 [25]	58.9	88.9	77.6	72.0	92.4	73.2	31.8	92.8	37.9	26.2	32.6	79.9	31.1
DenseNet103 [25]	**66.9**	**91.5**	**83.0**	**77.3**	**93.0**	77.3	**43.9**	**94.5**	**59.6**	37.1	**37.8**	**82.2**	50.5
PSPNet* (classification)	60.3	89.3	74.7	64.1	89.0	71.8	36.6	90.8	44.5	38.5	25.4	77.4	50.3
PSPNet* (multi-task)	62.2	90.0	76.2	66.4	88.8	78.0	37.6	90.7	47.2	**40.1**	28.6	78.9	**51.2**

setting.

CamVid The test results on the CamVid dataset are reported in Table 7.14 that also includes a comparison with other methods from the state-of-the-art, notably [25]. Some examples are shown in Fig. 7.16 where the distance transform regression once again produces smoother segmentations. The PSPNet baseline is competitive with those other methods and its mean IoU is improved by 0.5 by switching to the multi-task setting including the distance transform regression. Most classes benefit from the distance transform regression, with the exception of the "road" and "sky" classes. This is due to the void pixels, that are concentrated on those classes and that result in noisy distance labels.

Discussion

To understand better how the weighting between classification and regression intervenes in the loss function, we train several models with different values for the λ hyperparameter on the ISPRS Vaihingen dataset. This allows us to adjust the relative influence given to the SDT regression compared to the classification cross-entropy. As shown in the Table 7.10, we compare the simultaneous regression and classification to each task alone. In practice, it appears that SDT regression only obtains lower classification accuracies than the usual classification setup. This corresponds to the mode $\lambda \to +\infty$, while classification alone corresponds to $\lambda = 0$. We therefore compare the performance achieved by models trained with intermediate values of λ.

As illustrated in the Fig. 7.17, setting $\lambda > 0$ (i.e. taking into account the distance maps) significantly improves the classification results. Two values seem particularly interesting at 0.5 and 2. The first suffer from a high variance depending on the experiment while the second is achieves a slightly lower accuracy although more consistently. In practice all values of λ in the considered range improved the FCN performance, therefore making this hyperparameter fairly easy to tune.

The multitask learning incorporating the distance transform regression in the semantic segmentation model helps the network to learn spatial structures. More precisely, it constrains the network not only to learn if a pixel is in or out a class mask, but also the Euclidean distance of this pixel w.r.t the mask. This information can be critical when the filter responses

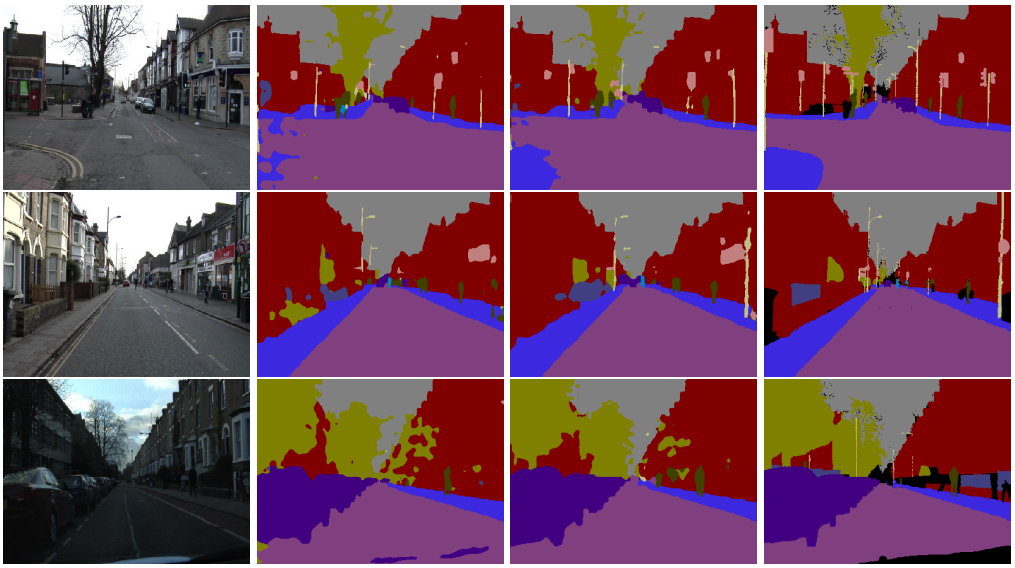
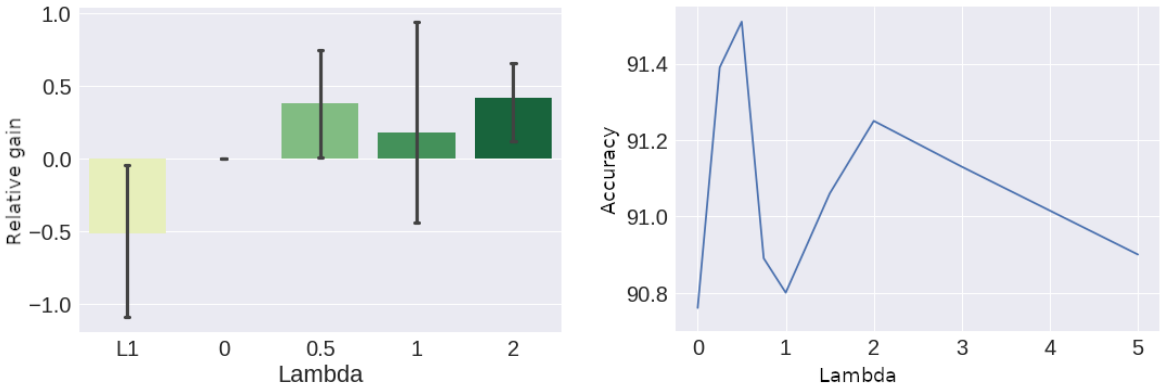


Figure 7.16: Examples of semantic segmentation results on the CamVid dataset. From left to right: RGB image, PSPNet (classification), PSPNet (multi-task), ground truth.



(a) Accuracy improvement with respect to λ.

(b) Classification accuracy with respect to λ.

Figure 7.17: Exploration of various values for λ on the ISPRS Vaihingen dataset.

are ambiguous. For example, trees from birdviewX might reveal the ground underneath during the winter, as there are no leaves, although annotations still consider the tree to have a shape similar to a disk. Spatial proximity helps in taking these cases into account and removing some of the salt-and-pepper classification noise that it induces, as shown on the ISPRS Vaihingen and Potsdam and DFC2015 datasets. Moreover, as the network has to assign spatial distances to each pixel w.r.t the different classes, it also learns helpful cues regarding the spatial structure underlying the semantic maps. As illustrated in Fig. 7.15, the predictions become more coherent with the original structure, with sharper boundaries and less holes when shapes are supposed to be closed.

Finally, an interesting research direction for the SDT reside in the panopatic segmentation paradigm [28]. This task consists in performing at the sime time semantic segmentation and instance segmentation. Indeed objects can be defined instance-wise but many areas and surfaces (the sky, the ocean, the roads...) cannot be clearly defined as a set of instances. The distance tranform maps could express these two concepts using the level sets, where class/instance-boundaries are encoded by the level 0.

The study from the Section 7.1 has been published in an international journal:

- Nicolas Audebert, Bertrand Le Saux, and Sébastien Lefèvre. "Segment-before-Detect: Vehicle Detection and Classification through Semantic Segmentation of Aerial Images". In: *Remote Sensing* 9.4 (Apr. 13, 2017), p. 368. DOI: 10.3390/rs9040368

The experiments from the Section 7.2 have been presentend at national conference:

- Nicolas Audebert et al. "Segmentation Sémantique Profonde Par Régression Sur Cartes de Distances Signées". In: *Reconnaissance Des Formes, Image, Apprentissage et Perception (RFIAP)*. Marne-la-Vallée, France, June 2018. URL: https://hal.archives-ouvertes.fr/hal-01809991 (visited on 08/27/2018)

References

[1] Nicolas Audebert, Bertrand Le Saux, and Sébastien Lefèvre. "Segment-before-Detect: Vehicle Detection and Classification through Semantic Segmentation of Aerial Images". In: *Remote Sensing* 9.4 (Apr. 13, 2017), p. 368. DOI: 10.3390/rs9040368 (cit. on p. 173).

[2] Nicolas Audebert et al. "Segmentation Sémantique Profonde Par Régression Sur Cartes de Distances Signées". In: *Reconnaissance Des Formes, Image, Apprentissage et Perception (RFIAP)*. Marne-la-Vallée, France, June 2018. URL: https://hal.archives-ouvertes.fr/hal-01809991 (cit. on p. 173).

[3] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Scene Segmentation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.12 (Dec. 2017), pp. 2481–2495. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2016.2644615 (cit. on pp. 167, 168, 171).

[4] Min Bai and Raquel Urtasun. "Deep Watershed Transform for Instance Segmentation". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). July 2017, pp. 2858–2866. DOI: 10.1109/CVPR.2017.305 (cit. on p. 159).

[5] Gedas Bertasius, Jianbo Shi, and Lorenzo Torresani. "Semantic Segmentation With Boundary Neural Fields". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016, pp. 3602–3610. URL: https://www.cv-foundation.org/openaccess/content_cvpr_2016/html/Bertasius_Semantic_Segmentation_With_CVPR_2016_paper.html (cit. on p. 164).

[6] Serge Beucher and Fernand Meyer. "The Morphological Approach to Segmentation: The Watershed Transformation. Mathematical Morphology in Image Processing." In: *Optical Engineering* 34 (1993), pp. 433–481 (cit. on p. 159).

[7] Gabriel J. Brostow, Julien Fauqueur, and Roberto Cipolla. "Semantic Object Classes in Video: A High-Definition Ground Truth Database". In: *Pattern Recognition Letters*. Video-based Object and Event Analysis 30.2 (Jan. 15, 2009), pp. 88–97. ISSN: 0167-8655. DOI: 10.1016/j.patrec.2008.04.005. URL: http://www.sciencedirect.com/science/article/pii/S0167865508001220 (cit. on p. 168).

[8] Manuel Campos-Taberner et al. "Processing of Extremely High-Resolution LiDAR and RGB Data: Outcome of the 2015 IEEE GRSS Data Fusion Contest Part A: 2-D Contest". In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 9.12 (Dec. 2016), pp. 5547–5559. ISSN: 1939-1404. DOI: 10.1109/JSTARS.2016.2569162 (cit. on pp. 168, 170, 171).

[9] Liang-Chieh Chen et al. "DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.4 (Apr. 2018), pp. 834–848. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2017.2699184 (cit. on pp. 164, 171).

[10] Xueyun Chen et al. "Vehicle Detection in Satellite Images by Hybrid Deep Convolutional Neural Networks". In: *IEEE Geoscience and Remote Sensing Letters* 11.10 (Oct. 2014), pp. 1797–1801. ISSN: 1545-598X. DOI: 10.1109/LGRS.2014.2309695 (cit. on p. 154).

[11] Marius Cordts et al. "The Cityscapes Dataset for Semantic Urban Scene Understanding". In: *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Las Vegas, United States, June 2016, pp. 3213–3223. DOI: 10.1109/CVPR.2016.350 (cit. on p. 167).

[12] Nicolas Courty et al. "Optimal Transport for Domain Adaptation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2016). URL: https://hal.archives-ouvertes.fr/hal-01377220 (cit. on p. 163).

[13] Jifeng Dai, Kaiming He, and Jian Sun. "Instance-Aware Semantic Segmentation via Multi-Task Network Cascades". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Las Vegas, United States, 2016, pp. 3150–3158. DOI: 10.1109/CVPR.2016.343 (cit. on pp. 159, 164).

[14] Line Eikvil, Lars Aurdal, and Hans Koren. "Classification-Based Vehicle Detection in High-Resolution Satellite Images". In: *ISPRS Journal of Photogrammetry and Remote Sensing* 64.1 (Jan. 2009), pp. 65–72. ISSN: 09242716. DOI: 10.1016/j.isprsjprs.2008.09.005. URL: http://linkinghub.elsevier.com/retrieve/pii/S092427160800097X (cit. on p. 154).

[15] Mark Everingham et al. "The Pascal Visual Object Classes Challenge: A Retrospective". In: *International Journal of Computer Vision* 111.1 (June 25, 2014), pp. 98–136. ISSN: 0920-5691, 1573-1405. DOI: 10.1007/s11263-014-0733-5. URL: http://link.springer.com/article/10.1007/s11263-014-0733-5 (cit. on pp. 157, 167).

[16] Joshua Gleason et al. "Vehicle Detection from Aerial Imagery". In: *Robotics and Automation (ICRA), 2011 IEEE International Conference On*. IEEE, 2011, pp. 2065–2070. URL: http://ieeexplore.ieee.org/abstract/document/5979853/ (cit. on p. 154).

[17] Zeeshan Hayder, Xuming He, and Mathieu Salzmann. "Boundary-Aware Instance Segmentation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2017 (cit. on p. 165).

[18] Caner Hazirbas et al. "FuseNet: Incorporating Depth into Semantic Segmentation via Fusion-Based CNN Architecture". In: *Computer Vision – ACCV 2016*. Asian Conference on Computer Vision. Springer, Cham, Nov. 20, 2016, pp. 213–228. DOI: 10.1007/978-3-319-54181-5_14 (cit. on pp. 168, 170).

[19] Kaiming He et al. "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification". In: *Proceedings of the IEEE International Conference on Computer Vision*. IEEE International Conference on Computer Vision (ICCV). Dec. 2015, pp. 1026–1034. DOI: 10.1109/ICCV.2015.123 (cit. on p. 168).

[20] Kaiming He et al. "Deep Residual Learning for Image Recognition". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Las Vegas, United States, June 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90 (cit. on pp. 162, 167).

[21] Kaiming He et al. "Mask R-CNN". In: *Proceedings of the International Conference on Computer Vision*. International Conference on Computer Vision (ICCV). Mar. 20, 2017 (cit. on pp. 159, 164).

[22] Ashley C. Holt et al. "Object-Based Detection and Classification of Vehicles from High-Resolution Aerial Photography". In: *Photogrammetric Engineering & Remote Sensing* 75.7 (2009), pp. 871–880. URL: http://www.ingentaconnect.com/content/asprs/pers/2009/00000075/00000007/art00007 (cit. on p. 154).

[23] Bohao Huang et al. "Large-Scale Semantic Classification: Outcome of the First Year of Inria Aerial Image Labeling Benchmark". In: *2018 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. July 22, 2018. URL: https://hal.inria.fr/hal-01767807/document (cit. on pp. 169, 170).

[24] Pranam Janney and David Booth. "Pose-Invariant Vehicle Identification in Aerial Electro-Optical Imagery". In: *Machine Vision and Applications* 26.5 (July 1, 2015), pp. 575–591. ISSN: 0932-8092, 1432-1769. DOI: 10.1007/s00138-015-0687-9. URL: https://link.springer.com/article/10.1007/s00138-015-0687-9 (cit. on p. 154).

[25] Simon Jégou et al. "The One Hundred Layers Tiramisu: Fully Convolutional DenseNets for Semantic Segmentation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. Honolulu, United States, July 2017, pp. 1175–1183. DOI: 10.1109/CVPRW.2017.156 (cit. on pp. 168, 171).

[26] Eric Jones, Travis Oliphant, Pearu Peterson, et al. *SciPy: Open Source Scientific Tools for Python*. 2001–. URL: http://www.scipy.org/ (cit. on p. 168).

[27] Dmitri Kamenetsky and Jamie Sherrah. "Aerial Car Detection and Urban Understanding". In: *2015 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*. 2015 International Conference on Digital Image Computing: Techniques and Applications (DICTA). Nov. 2015, pp. 1–8. DOI: 10.1109/DICTA.2015.7371225 (cit. on pp. 154, 160).

[28] Alexander Kirillov et al. "Panoptic Segmentation". In: (Jan. 2, 2018). arXiv: 1801.00868 [cs]. URL: http://arxiv.org/abs/1801.00868 (cit. on p. 172).

[29] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Proceedings of the Neural Information Processing Systems (NIPS)*. NIPS. 2012, pp. 1097–1105. URL: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf (cit. on pp. 155, 160).

[30] TT. Hoang Ngan Le et al. "Reformulating Level Sets as Deep Recurrent Neural Network Approach to Semantic Segmentation". In: *IEEE Transactions on Image Processing* 27.5 (May 2018), pp. 2393–2407. ISSN: 1057-7149. DOI: 10.1109/TIP.2018.2794205 (cit. on p. 164).

[31] Franz Leberl et al. "Recognizing Cars in Aerial Imagery to Improve Orthophotos". In: *GIS: Proceedings of the ACM International Symposium on Advances in Geographic Information Systems*. Jan. 1, 2007, p. 2. DOI: 10.1145/1341012.1341015 (cit. on p. 154).

[32] Yann LeCun et al. "Gradient-Based Learning Applied to Document Recognition". In: *Proceedings of the IEEE* 86.11 (Nov. 1998), pp. 2278–2324. ISSN: 0018-9219. DOI: 10.1109/5.726791 (cit. on pp. 155, 160).

[33] Ziwei Liu et al. "Deep Learning Markov Random Field for Semantic Segmentation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.8 (Aug. 2018), pp. 1814–1828. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2017.2737535 (cit. on p. 164).

[34] Emmanuel Maggiori et al. "Can Semantic Labeling Methods Generalize to Any City? The Inria Aerial Image Labeling Benchmark". In: *Proceedings of the IEEE International Symposium on Geoscience and Remote Sensing (IGARSS)*. IEEE International Symposium on Geoscience and Remote Sensing (IGARSS). July 23, 2017. DOI: 10.1109/IGARSS.2017.8127684. URL: https://hal.inria.fr/hal-01468452/document (cit. on pp. 168, 170).

[35] Dimitrios Marmanis et al. "Classification With an Edge: Improving Semantic Image Segmentation with Boundary Detection". In: *ISPRS Journal of Photogrammetry and Remote Sensing* (2017). DOI: 10.1016/j.isprsjprs.2017.11.009. arXiv: 1612.01337 (cit. on pp. 154, 169).

[36] Calvin R. Maurer, Rensheng Qi, and Vijay Raghavan. "A Linear Time Algorithm for Computing Exact Euclidean Distance Transforms of Binary Images in Arbitrary Dimensions". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 25.2 (Feb. 2003), pp. 265–270. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2003.1177156 (cit. on p. 166).

[37] Julien Michel et al. "Local Feature Based Supervised Object Detection: Sampling, Learning and Detection Strategies". In: *2011 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, July 2011, pp. 2381–2384. ISBN: 978-1-4577-1003-2. DOI: 10.1109/IGARSS.2011.6049689. URL: http://ieeexplore.ieee.org/document/6049689/ (cit. on pp. 154, 160).

[38] Keiller Nogueira, Otávio Penatti, and Jefersson A. dos Santos. "Towards Better Exploiting Convolutional Neural Networks for Remote Sensing Scene Classification". In: (Feb. 3, 2016). arXiv: 1602.01517 [cs]. URL: http://arxiv.org/abs/1602.01517 (cit. on pp. 155, 160).

[39] Otávio Penatti, Keiller Nogueira, and Jefersson A. dos Santos. "Do Deep Features Generalize from Everyday Objects to Remote Sensing and Aerial Scenes Domains?" In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). June 2015, pp. 44–51. DOI: 10.1109/CVPRW.2015.7301382 (cit. on p. 160).

[40] *PyTorch: Tensors and Dynamic Neural Networks in Python with Strong GPU Acceleration*. http://pytorch.org/. 2016–. URL: http://pytorch.org/ (cit. on p. 168).

[41] Xiaojuan Qi et al. "3D Graph Neural Networks for RGBD Semantic Segmentation". In: *Proceedings of the International Conference on Computer Vision*. International Conference on Computer Vision (ICCV). 2017. URL: http://openaccess.thecvf.com/content_iccv_2017/html/Qi_3D_Graph_Neural_ICCV_2017_paper.html (cit. on p. 170).

[42] Hicham Randrianarivo, Bertrand Le Saux, and Marin Ferecatu. "Urban Structure Detection with Deformable Part-Based Models". In: *2013 IEEE International Geoscience and Remote Sensing Symposium - IGARSS*. 2013 IEEE International Geoscience and Remote Sensing Symposium - IGARSS. July 2013, pp. 200–203. DOI: 10.1109/IGARSS.2013.6721126 (cit. on pp. 154, 158).

[43] Hicham Randrianarivo et al. "Contextual Discriminatively Trained Model Mixture for Object Detection in Aerial Images". In: *International Conference on Big Data from Space (BiDS'16)*. Spain, Mar. 2016 (cit. on pp. 154, 157, 160).

[44] Sébastien Razakarivony and Frédéric Jurie. "Vehicle Detection in Aerial Imagery: A Small Target Detection Benchmark". In: *Journal of Visual Communication and Image Representation* 34 (2016), pp. 187–203. DOI: 10.1016/j.jvcir.2015.11.002. URL: http://www.sciencedirect.com/science/article/pii/S1047320315002187 (cit. on pp. 154, 156).

[45] Joseph Redmon et al. "You Only Look Once: Unified, Real-Time Object Detection". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). June 2016, pp. 779–788. DOI: 10.1109/CVPR.2016.91 (cit. on pp. 154, 159).

[46] Shaoqing Ren et al. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.6 (June 2017), pp. 1137–1149. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2016.2577031 (cit. on pp. 154, 159).

[47] Franz Rottensteiner et al. "The ISPRS Benchmark on Urban Object Classification and 3D Building Reconstruction". In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 1 (2012), p. 3. URL: https://t3sec3.rrzn.uni-hannover.de/cmsv021a.rrzn.uni-hannover.de/uploads/tx_tkpublikationen/isprsannals-I-3-293-2012.pdf (cit. on p. 167).

[48] Olga Russakovsky et al. "ImageNet Large Scale Visual Recognition Challenge". In: *International Journal of Computer Vision* 115.3 (Apr. 11, 2015), pp. 211–252. ISSN: 0920-5691, 1573-1405. DOI: 10.1007/s11263-015-0816-y. URL: http://link.springer.com/article/10.1007/s11263-015-0816-y (cit. on p. 155).

[49] Jamie Sherrah. "Fully Convolutional Networks for Dense Semantic Labelling of High-Resolution Aerial Imagery". In: (June 8, 2016). arXiv: 1606.02585 [cs]. URL: http://arxiv.org/abs/1606.02585 (cit. on p. 158).

[50] Karen Simonyan and Andrew Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition". In: *Proceedings of the International Conference on Learning Representations (ICLR)*. May 2015. URL: http://arxiv.org/abs/1409.1556 (cit. on pp. 155, 160, 168).

[51] Lars Wilko Sommer, Tobias Schuchert, and Jürgen Beyerer. "Fast Deep Vehicle Detection in Aerial Images". In: *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*. 2017 IEEE Winter Conference on Applications of Computer Vision (WACV). Mar. 2017, pp. 311–319. DOI: 10.1109/WACV.2017.41 (cit. on p. 154).

[52] Shuran Song, Samuel P. Lichtenberg, and Jianxiong Xiao. "SUN RGB-D: A RGB-D Scene Understanding Benchmark Suite". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). June 2015, pp. 567–576. DOI: 10.1109/CVPR.2015.7298655 (cit. on p. 168).

[53] Nitish Srivastava et al. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting". In: *Journal of Machine Learning Research* 15 (2014), pp. 1929–1958. URL: http://jmlr.org/papers/v15/srivastava14a.html (cit. on p. 161).

[54] Tianyu Tang et al. "Vehicle Detection in Aerial Images Based on Region Convolutional Neural Networks and Hard Negative Example Mining". In: *Sensors (Basel, Switzerland)* 17.2 (Feb. 10, 2017). ISSN: 1424-8220. DOI: 10.3390/s17020336. pmid: 28208587. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5335960/ (cit. on p. 159).

[55] Devis Tuia, Claudio Persello, and Lorenzo Bruzzone. "Domain Adaptation for the Classification of Remote Sensing Data: An Overview of Recent Advances". In: *IEEE Geoscience and Remote Sensing Magazine* 4.2 (June 2016), pp. 41–57. ISSN: 2168-6831. DOI: 10.1109/MGRS.2016.2548504. URL: http://ieeexplore.ieee.org/document/7486184/ (cit. on p. 163).

[56] Jonas Uhrig et al. “Pixel-Level Encoding and Depth Layering for Instance-Level Semantic Labeling”. In: *Pattern Recognition*. German Conference on Pattern Recognition. Lecture Notes in Computer Science. Springer, Cham, Sept. 12, 2016, pp. 14–25. ISBN: 978-3-319-45885-4 978-3-319-45886-1. DOI: 10.1007/978-3-319-45886-1_2. URL: https://link.springer.com/chapter/10.1007/978-3-319-45886-1_2 (cit. on p. 165).

[57] Adam Van Etten. “You Only Look Twice: Rapid Multi-Scale Object Detection In Satellite Imagery”. In: (May 24, 2018). arXiv: 1805.09512 [cs]. URL: http://arxiv.org/abs/1805.09512 (cit. on pp. 154, 159).

[58] Francisco de Assis Zampirolli and Leonardo Filipe. “A Fast CUDA-Based Implementation for the Euclidean Distance Transform”. In: *International Conference on High Performance Computing Simulation*. International Conference on High Performance Computing Simulation (HPCS). July 2017, pp. 815–818. DOI: 10.1109/HPCS.2017.123 (cit. on p. 168).

[59] Hengshuang Zhao et al. “Pyramid Scene Parsing Network”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Honolulu, United States, July 2017, pp. 2881–2890. DOI: 10.1109/CVPR.2017.660. URL: http://openaccess.thecvf.com/content_cvpr_2017/html/Zhao_Pyramid_Scene_Parsing_CVPR_2017_paper.html (cit. on pp. 167, 168).

[60] Shuai Zheng et al. “Conditional Random Fields as Recurrent Neural Networks”. In: *Proceedings of the IEEE International Conference on Computer Vision*. IEEE International Conference on Computer Vision (ICCV). Dec. 2015, pp. 1529–1537. DOI: 10.1109/ICCV.2015.179 (cit. on p. 164).

[61] Weixun Zhou, Zhenfeng Shao, and Qimin Cheng. “Deep Feature Representations for High-Resolution Remote Sensing Scene Classification”. In: *2016 4th International Workshop on Earth Observation and Remote Sensing Applications (EORSA)*. 2016 4th International Workshop on Earth Observation and Remote Sensing Applications (EORSA). July 2016, pp. 338–342. DOI: 10.1109/EORSA.2016.7552825 (cit. on p. 155).

Chapter 8 Conclusion and future works

It is good to have an end to journey toward; but it is the journey that matters, in the end.

— Ursula Le Guin (The Left Hand of Darkness, 1969)

The plethora of remote sensing data being acquired everyday is a goldmine for the scientific community. Thanks to the joint effort of multiple actors that invest into Earth Observation programs, we can access high resolution images of the whole globe at high frequencies that were unimaginable 20 years ago. In France, aerial images at 20 cm/px GSD on the whole country are released every 3 years by the IGN, while the CNES performs a complete observation at 1.5 m/px GSD every year using the SPOT constellation. Meanwhile the European Copernicus program has deployed the Sentinel-2 satellites which provide us with weekly multispectral images every 5 days at 10 m/px on the whole globe. Added to these instutional actors are the many private and military operators (Worldview, Pléiades) but also the many radar, Lidar and hyperspectrales sensors.

The human resources available to process this data are however largely unsufficient to transform this mass of raw data into knowledge. Manual photointerpreation is slow and expensive. Its automation is a major challenge for scientific research today and tomorrow in ecology, urbanism, meteorology or agriculture. Our goal in this thesis was to propose machine learning tools suited to the problem of cartography automation. Building on deep artificial neural networks, we designed statistical models for semantic segmentation of aerial and satellite images from low to extremely high resolution for a variety of optical sensors. To leverage ancillary information such as digital surface models and open geographic datasets, we introduced multi-modal deep network architectures for data fusion that are able to learn from heterogeneous information. Finally, we showed the current limits of these approaches on limited and very large-scale datasets and suggested alternatives, notably data augmentation using generative models and the introduction of anew country-scale dataset called *MiniFrance*.

Therefore the works presented in this manuscript allowed us to confirm the position of deep networks as excellent tools for automatic remote sensing image understanding. These models reach the accuracies that are reasonably expected by specialists from the applicative fields. Indeed automating the production of semantic maps from optical images, either in colour, multispectral or hyperspectral, seems reachable at an industrial scale in a few years, including in a multi-modal setting. While deep learning for remote sensing was only in its premises a few years ago, it is now at forefront of the research in Earth Observation. This thesis contributed to this by reiterating once again that neural networks are the state of the art for image interpreation, and can be extended to the sensors that are commonly used in remote sensing and were until then ignored by the computer vision community. We introduced good practices and guidelines for the deployment of convolutional neural networks for Earth Observation image understanding by studying extensively the impact of pretraining, multi-scale learning, data fusion and segmentation regularization. These techniques, still rarely studied in the beginning of the 2010s, are now established on solid experimental grounds.

First, we demonstrated that state of the art region-based classification techniques could be replaced with much more efficient fully convolutional neural networks. Especially we showed that the unsupervised segmentation pre-processing is on the critical path of semantic

segmentation and puts an upper bound on the achievable accuracy. We adapted semantic segmentation neural networks to aerial remote sensing IRRG and RGB images. Then we extended this approach to Sentinel-2 multispectral satellite images on which we showed that including spectral bands outside the visible domain significantly improved the classification results. We also studied deep learning techniques for hyperspectral image classification, showing that 3D convolutional networks were particularly efficient and effective to process the hypercubes. These contributions allowed us to produce very accurate semantic maps that meet the operational requirements of users and consumers of geographic data.

Second, we studied how to introduce ancillary information into image-based deep models to learn from all available input sources on the scenes of interest. Especially we designed two multi-modal convolutional network architectures for semantic segmentation that can fuse data coming from heterogeneous inputs. By combining optical images on the one hand and digital surface models on the other hand, but also optical images and OpenStreetMap data, we were able to significantly reduce the error rate of our convolutional networks deployed for semantic mapping, notably using the residual correction module. We introduced a new way to take into account OSM data which encourages us to study how to automatically update this kind of GIS using a bootstrapping approach.

Then, we investigated the behaviour of statistical models on small and large-scale datasets. In a hyperspectral setting, we noted that very few labeled datasets were available to train deep networks and that these images were particularly small compared to usual databases. We therefore designed generative models based on GAN to synthesize new artificial spectra successfully to generate fake datasets. In addition we validated our semantic mapping models on several large scale datasets with various properties. Notably we created our own large-scale high-resolution dataset that covers many urban areas in France. We showed that the networks we introduced previously were able to scale to the country level and could generalize to diversified scenes. The introduction of the *MiniFrance* dataset will allow in the long run to pretrain supervised networks at the largest scale ever reached for remote sensing image understanding models.

Finally we studied regularization techniques to structure the semantic maps produced by the networks, especially for an object-based image analysis framework. We designed a segment-before-detect approach to locate and recognize vehicles with a fine-grained classification in aerial images. Our approach is based on the semantic segmentation task and results in outputs the precise shapes of the vehicles in addition to their location. Overall it generated less false postivies than the previous state of the art in remote sensing. Moreover we introduced an alternative formulation of the semantic segmentation task by expressing it as the regression of signed distance maps. This allowed us to implicitly regularize and enforce some geometrical structure on the semantics maps produced by deep networks. This improved quantitatively and qualitatively the resulting segmentation which modeled more accurately the relationships between pixels.This scheme also gives a research direction to solve in an unified way to panoptic segmentation of images, either natural or remotely sensed, in which we aim to simultaneously identify object instances and non-structured areas. These findings open the door to new research direction rarely – even never – investigated up until now. Although this thesis focused on image semantic segmentation, let us recall that geographical analysis often looks at temporal aspects. From the application viewpoint many research topics involve understanding the evolution of the maps, from typhoon monitoring to understanding the dynamics of deforestation. However if we want to detect changes that occurred between two acquisitions or produce a complete time series, generating complete semantic maps for each acquisition could become expensive quickly. On the opposite image comparison techniques or time series classification methods, for example using recurrent neural networks, could be used to produce incremental maps at each timestep that leverage the history of a scene in a more efficient and more expressive way.

Equally interesting is the topic of data fusion. Here we focused on multi-modal architec-

tures taking rasters as input, but it would be interesting to extend these concepts to sparse 3D data or even non-structured information such as street-level paranoma or textual annotations. Some works have already strated using modular neural networks that can learn from multiple data sources, even with missing data. This is especially important for optical sensors that are sensitive to weather conditions and cloud cover, since SAR data could alleviate most of these problems for automated cartography at high frequency. Nonetheless radar signals are very different from the usual colour images and as complex signals, they would require approaches specifically tailored to the physics of these sensors. For scene interpreation, learning from geometry either by reconstruction from an image or by leveraging additional sensors (e.g. Lidar) would help produce high resolution 3D semantic models of cities. In remote sensing this encourages the community to study approaches combining geometry and semantics, at the edge between photogrammetry and semantic segmentation, for tasks such as 3D modeling, orthorectification and co-registering.

Finally, if deep learning has made it possible to reach excellent empirical performance, the question of how to help final users interpret these results remain a major obstacle before a large-scale adoption. The representations learnt by the statistical models are difficult to reuse and understand by human users and do not carry a complete information. Interpreting the models, especially to help experts understand why the models predicted such or such class, is a prerequisite for an efficient collaboration between the users and the machine. This requires two elements: a) matching the features learnt by the neural network to high level semantic concepts graspable by the human user, b) make it clear how the model took a specific decision and what features are factored the most in this choice. Particularly, this would greatly help active learning process that include human knowledge when training the network, both regarding its expertise regarding the task but also how the user usually decides, so that the machine can mimick it and learn from both.

Appendix

A Datasets

One accurate measurement is worth a thousand expert opinions.

— Grace Hopper

A.1 Remote sensing datasets

There are many classification datasets focused on optical remote sensing data. Let us cite to this end the *UC Merced* dataset [13], comprised of 2100 aerial images for 21 land cover classes, *Brazilian Coffe* [8] containing SPOT images for crop classification and SAT-4/SAT-6 [1] containing respectively 500 000 and 405 000 aerial images for various land use classes. These datasets share two drawbacks. First the images are quite small (256 px × 256 px for *UC Merced* and *Brazilian Coffe*, 28 px × 28 px for SAT) and the annotations are scarce. Indeed these datasets have been conceived for image classification and therefore are not suited to cartography, which is closer to semantic segmentation. However several datasets with dense ground truth annotations have been released.

A.1.1 ISPRS 2D Semantic Labeling

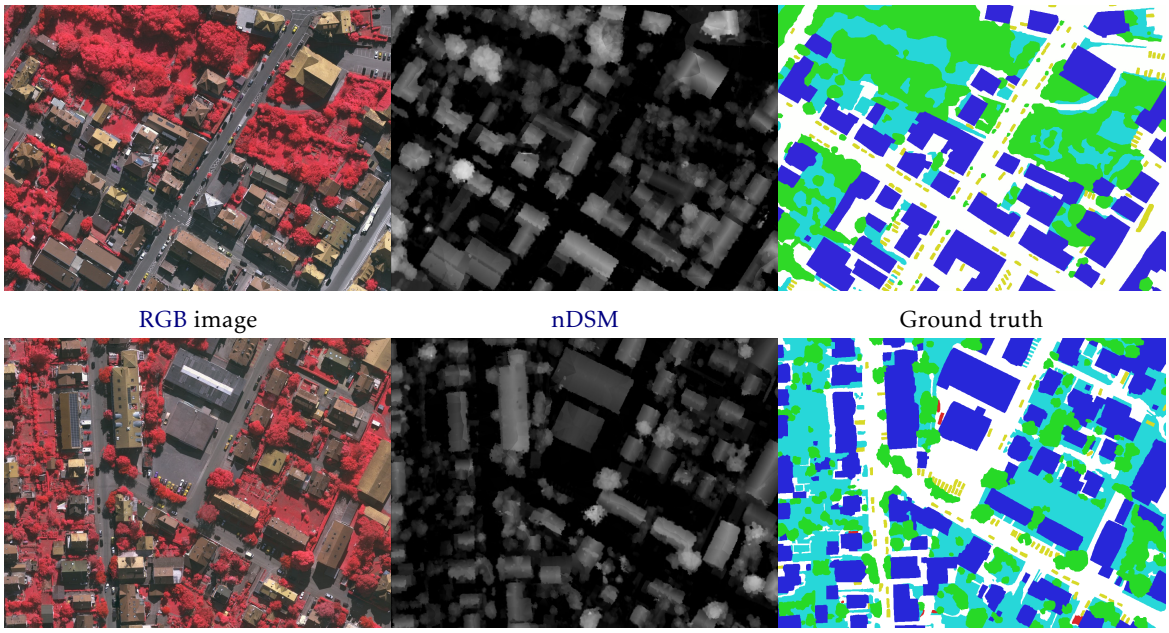


Figure A.1: Ortho-rectified images and nDSM on the ISPRS Vaihingen dataset.

The ISPRS 2D Semantic Labeling dataset [11] consists in two sets of extremely high resolution (EHR) aerial images released by the working group (WG) II/4 of the International Society for Photogrammetry and Remote Sensing. Both acquistions are urban scenes that have been labeled for semantic segmentation in five classes: impervious surfaces (roads,

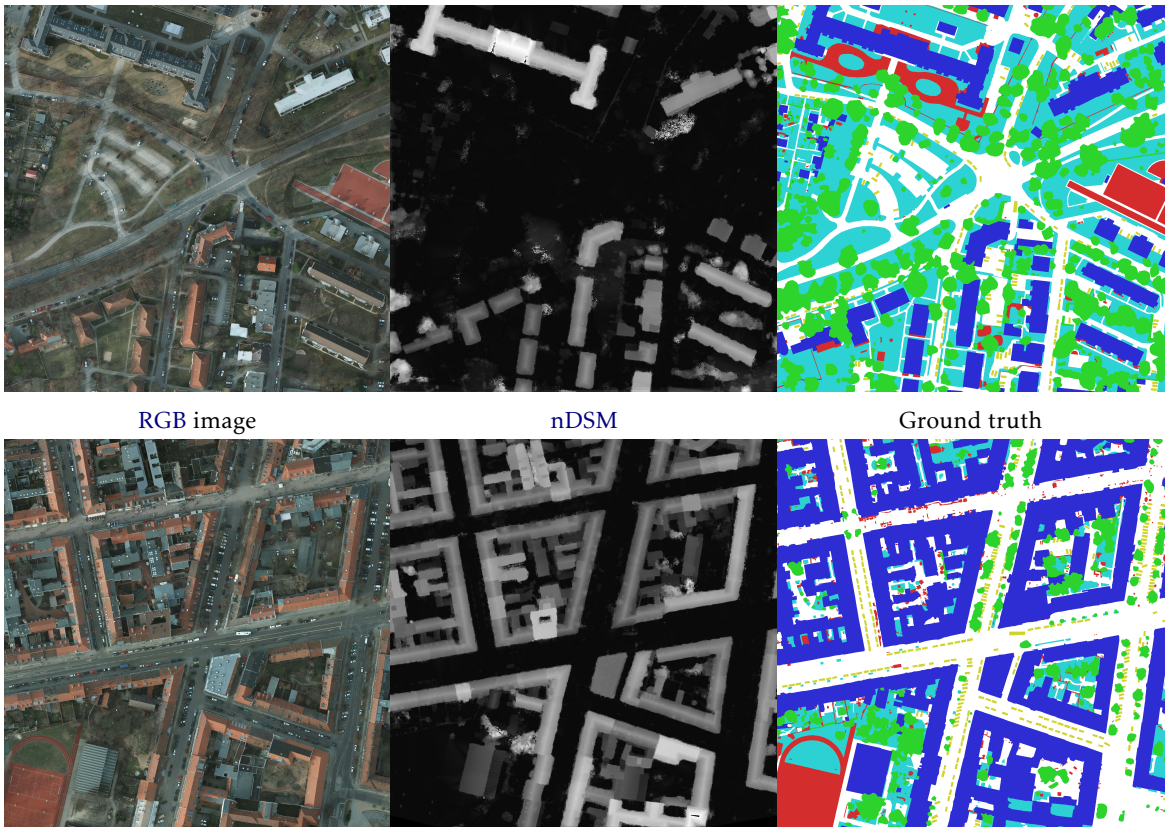


Figure A.2: Ortho-rectified images and nDSM on the ISPRS Potsdam dataset.

sidewalks, parking lots...), buildings, low vegetation, trees and vehicles. A reject class is also labeled[1] containing the urban clutter (benches, garbage bins, containers...) and other materials (basketball courts, construction sites, fountains...).

There are two scenes in the dataset. scènes. The first one is based on an aerial acquisitions over the city of Vaihingen (Germany) and consists in a mosaic of 33 IRRG ortho-rectified tiles at a spatial resolution of 9 cm/px. The optical acquisition is completed by a Lidar point cloud at the same GSD from which a DSM has been rasterized. A pre-computed nDSM [4] calculated from the DSM is also publicly available. The ortho-images are published in the TIFF format, encoded on 8 bits integers, while the DSM is encoded on 32 bits floats. All the data have been co-registered on the same pixel grid. The images are approximately 2600 px × 1900 px in average, i.e. an approximate surface of around 40 000 m^2. Vaihingen is a medium-sized town (28 853 inhabitants in 2009), characterized by a an average urbanization consisting in mostly individual houses and green urban areas.

The second scene is also an aerial acquisition, but on the city of Potsdam (Germany) and consists in a mosaic of 38 IRRGB tiles ortho-rectified at a spatial resolution of 5 cm/px performed by BSF Swissphoto. The dataset is released by the *Deutsche Gesellschaft für Photogrammetrie, Fernerkundung und Geoinformation* (DGPF)[2]. All tiles have the same dimensions (6000 px × 6000 px, i.e. a surface of 90 000 m^2). An DSM and its corresponding nDSM are also available, rasterized from a Lidar point cloud. Dense annotations are published for the same classes as the previous dataset on 24 images. Once again, all modalities have been co-registered on the same pixel grid and images are released as 8 bits integer TIFF while the digital surface models are released as 32 bits floats. Potsdam is a quite large urban town (161 468 inhabitants in 2013), characterized by many large buildings and a dense road

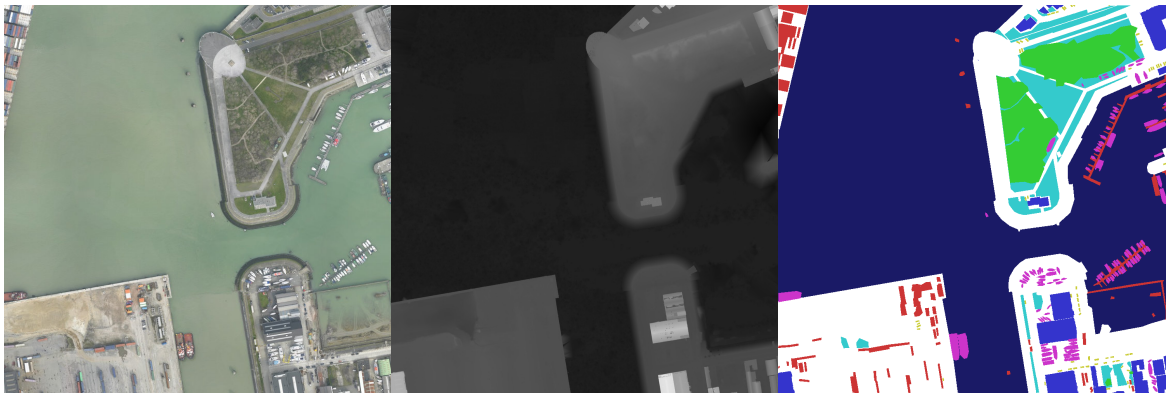
[1] But not evaluated.
[2] http://www.ifp.uni-stuttgart.de/dgpf/DKEP-Allg.html

network. There is also a water channel and many active construction sites at the time the images were acquired.

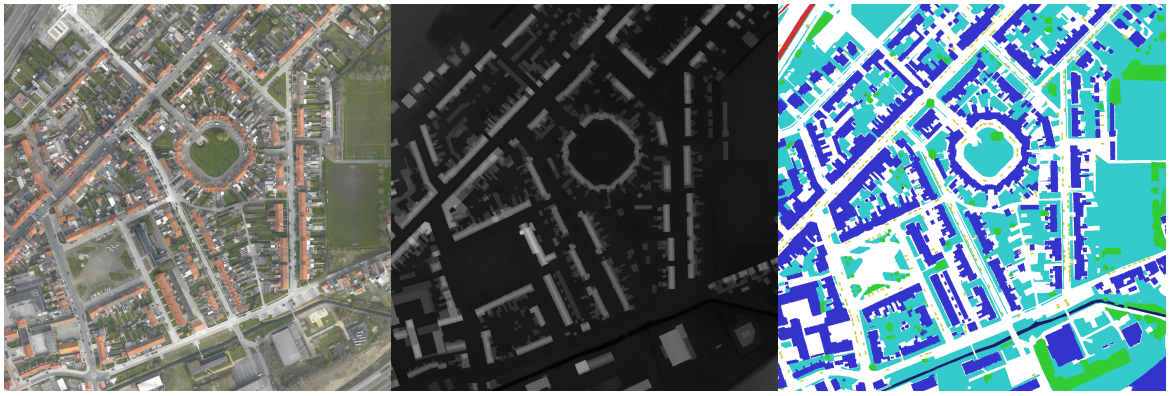
Some representative samples from the two acquisitions are pictured in the Figs. A.1 and A.2. The number of pixels in each class is reported in the Fig. A.10.

The images for which the annotations are not publicly released are still labeled, although the ground truth is used to evaluate blindly the submissions to the benchmark. The WG II/4 commission from the ISPRS manages a public leaderboard[3][4] that reports the official results obtained by various methods from the state of the art.

A.1.2 Data Fusion Contest 2015



RGB image nDSM Ground truth



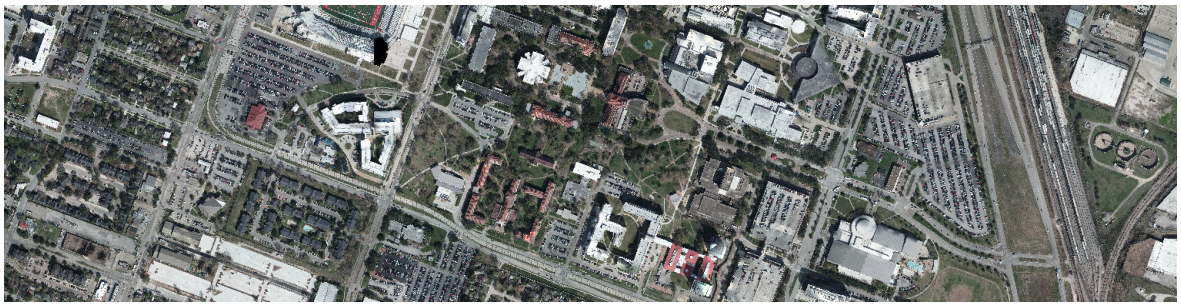
RGB image nDSM Ground truth

Figure A.3: Ortho-rectified images and nDSM from the DFC 2015 dataset.

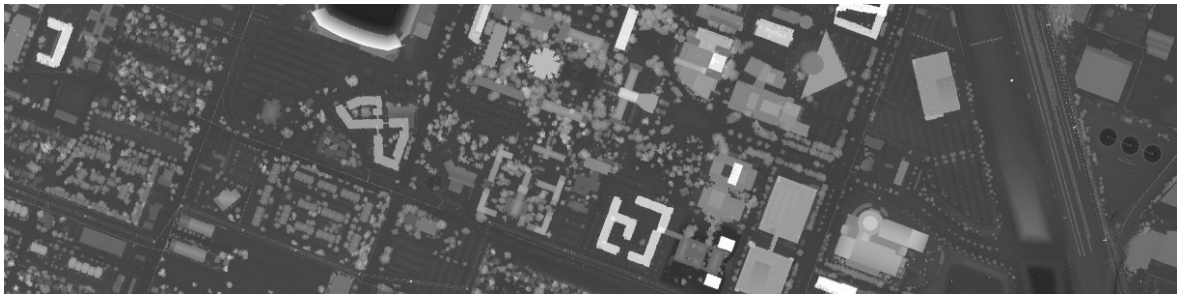
The DFC 2015 [3] dataset is the product of a data fusion competition organized by the GRSS workgroup from the IEEE. This dataset consists in a mosaic of 7 ortho-rectified colour images of size 10 000 px × 10 000 px with a GSD of 5 cm/px, i.e. a surface per tile of 250 000 m^2. The acquisition was realized on the port area of Zeebruges (Belgium) in March, 2011 by the *Communication, Information, Systèmes & Senseurs* (CISS) department of the *École royale militaire de Belgique*. There is also a Lidar acquistion of about 65 points/m^2 separated by 10 cm. The colour data are released in the TIFF (8 bits integers) format and the Lidar is released as a rasterized DSM (32 bits floats) and a point cloud. It is a urban scene focused mostly on port installations and structures.

A dense labeling of the area has been performed by the *Office national d'études et de recherches aérospatiales* (ONERA) [5] for the classes boats, cars, low vegetation, trees, buildings,

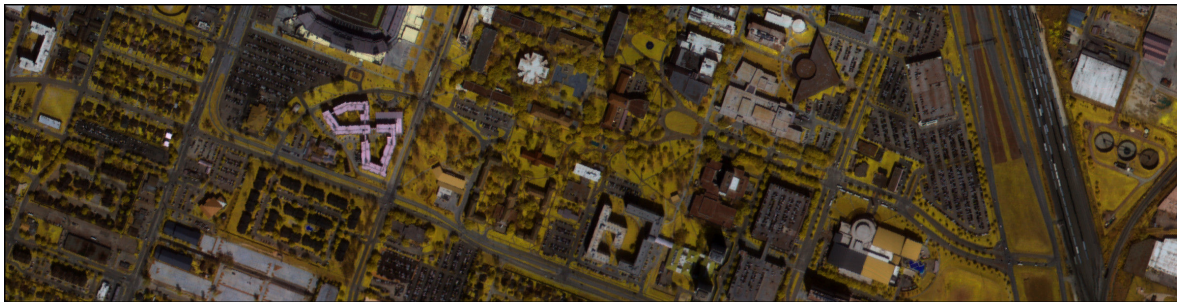
[3] http://www2.isprs.org/commissions/comm2/wg4/vaihingen-2d-semantic-labeling-contest.html
[4] http://www2.isprs.org/commissions/comm2/wg4/potsdam-2d-semantic-labeling.html



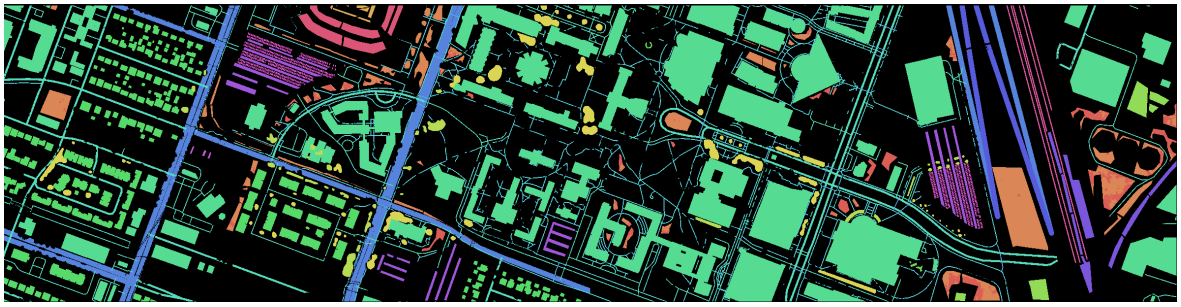
(a) RGB image



(b) nDSM



(c) Hyperspectral image (false colours)



(d) Ground truth

Figure A.4: Training data from the DFC 2018.

water and impervious surfaces. The Fig. A.3 illustrates some images extracted from the dataset and the Fig. A.11a reports the distribution of the pixels in the different classes.

A public leaderboard for this dataset is managed by the IEEE GRSS[5] to allow the comparison of several classification techniques.

A.1.3 Data Fusion Contest 2018

The DFC 2018 dataset [6] is a also a product of the Data Fusion Contest (DFC) organized by the IEEE GRSS. It consists in 14 ortho-rectified aerial RGB images at VHR (5 cm/px) of size 12 000 px × 12 000 px and one large hyperspectral image with 48 bande at 1 m/px between

[5] http://dase.ticinumaerospace.com/

380 and 1050 nm. Also available is a multispectral Lidar acquisition with a resolution of 0.5 m/px, from which a nDSM has been rasterized. All the data are georeferenced and have been co-registered. The data have been acquired by the *National Center for Airborne Laser Mapping* over the city of Houston (United States of America) in February, 2017. The image covers mostly Houston University and its surroundings. This is a very urbanized scene including massive installations (train station and railroads, baseball stadium). Partially dense annotations have been released on half the dataset for several urban classes of interest, the other half is kept hidden for the evaluation phase. The training set is pictured in the Fig. A.4 and the Fig. A.11b reports the pixel distrbution in the different classes.

A public leaderboard is maintained by the IEEE GRSS to facilitate comparison between various classification techniques.

A.1.4 Inria Aerial Image Labeling

Ortho-image (Chicago) Ground truth (Chicago) Ortho-image (Vienna) Ground truth (Vienna)

Figure A.5: Image samples from the *Inria Aerial Image Labeling*.

The *Inria Aerial Image Labeling* dataset [7] contains 360 RGB ortho-rectified images of size 5000 px × 5000 px with a 30 cm/px GSD, i.e. a surface of 2.25 km^2. Images have been agregated from the USGS database for Austin, Chicago, Kitsap County, Bellingham, Bloomington and San Francisco and from various regional Austrian geographic agencies for Tyrol, Vienna and Innsbruck. All imags are ortho-rectified aerial images that have been resampled at 30 cm/px and released in 8 bit colour format. The building footprints annotations have been obtained from local cadastres. Half of the images can be used freely to train building extraction models, the remainder being for evaluation by the dataset's authors. A few images and their ground truth are shown in the Fig. A.5.

Among the cities present in the dataset, some are large conurbations characterized by a high building density that mixes personal housing, large-scale constructions (trains tations, hospitals, factories…) and high-rises. On the opposite, other areas are sparsely populated with an important relief and lots of vegetation, especially in Tyrol. This diversity of the observed areas is intentional and aims to evaluate the models capacity to generalize to multiple environments.

The organizers manage a public leaderboard[6] to compare the results obtained by various methods.

A.1.5 VEDAI

The Vehicle Detection in Aerial Imagery (VEDAI) database [10] is a collection of ortho-rectified aerial images, initially published by the *Automated Geographic Reference Center* from Utah. The images were acquired in spring 2012 at a 12.5 cm/px GSD on 4 channels: RGB and infrared. Data is encoded on 8 bit integers. The original images have been split in 1210

[6] https://project.inria.fr/aerialimagelabeling/leaderboard/

Figure A.6: Sample images from the VEDAI dataset.

tiles of shape 1024 px × 1024 px. A downsampled version of the dataset at 25 cm/px using tiles of sizes 512 px × 512 px also exist.

The vehicles present in the imagse have been labeled for detection using rectangular bounding boxes, accompanied by a label corresponding to the vehicle type. Nine classes of interest have been identified: plane, boat, camping-car, car, pick-up, tractor, truck, van and a "other" class. There are also annotations regarding the coordinates of the center of the vehicle and the angle corresponding to its main orientation.

This dataset covers mostly rural areas with a low vehicle density. The images exhibit a large contextual diversity, from parking lots to small airports, highways and field roads, but also crops and small houses. Some examples are illustrated in the Fig. A.6.

A.1.6 NZAM/ONERA Christchurch

The NZAM/ONERA Christchurch dataset is comprised of 4 RGB images, ortho-rectified at a spatial resolution of 10 cm/px acquired after the earthquake that hit the city of Christchurch (New-Zeland) on February 22, 2011. The images have been released under the Creative Commons Attribution 3.0 license by the *New Zealand's Land Information Office* [7]. All images (≈ 5000 × 4000px) have been labeled by the ONERA/DTIS [9] for the following classes: "buildings" (797 objects), "vehicles" (2357 objects) and "vegetation" (938 objects). These objects are annotated by a polygonal bounding box, i.e. the annotations are coarser than the pixel-accurate ground truthes from the ISPRS datasets, for example. Sample images are illustrated in the Fig. A.7.

[7] http://www.linz.govt.nz/land/maps/linz-topographic-maps/imagery-orthophotos/christchurch-earthquake-imagery

Figure A.7: Images and annotations extracted from the NZAM/ONERA Christchurch dataset.

A.2 Jeux de données en interprétation de scènes

A.2.1 CamVid

The CamVid dataset (*Cambridge-driving Labeled Video Database*) [2] is an image database extracted at 1 Hz from an RGB video of a 10-minutes drive in a car in the city of Cambridge (United-Kingdom). 367 training images and 233 test images are collected at a resolution of 360 px × 480 px from the videos and manually labeled for 11 clases of interest such as roads, buildings, other vehicles, pedestrians, traffic signs, sidewalks and so on. Overall these images are representative of autonomous driving situations at a moderate pace in a urban environment with many moving objects. The Fig. A.8 illustres some labeled images from the training set.

A.2.2 SUN RGB-D

The SUN RGB-D dataset [12] consists in 10 335 indoor images Red-Green-Blue + Depth acquired using various sensors (Kinect, Xtion, RealSense). Every image is actually a pair of RGB and grayscale depth map. All images have been annotated pixel-wise for 37 classes of interest as objects or surfaces such as “chair”, “ground”, “wall” or “table”. Images are generally resized at 224 px × 224 px. Overall 146 617 objects in 2D have been labeled as non-overlapping polygons, which results for every image in a 2D semantic segmentation ground truth (with some unlabeled pixels). Other annotations are also availabel such as the scene category in 2.5D from 47 possible categories or 800 types of 3D objects identified by a bounding box. Sample images and ground truthes are illustrated in the Fig. A.9.

Figure A.8: RGB images (first row) and pixel-wise annotations (second row) extracted from the CamVid dataset.

Figure A.9: RGB images, depth maps and annotations from the SUN RGB-D dataset.

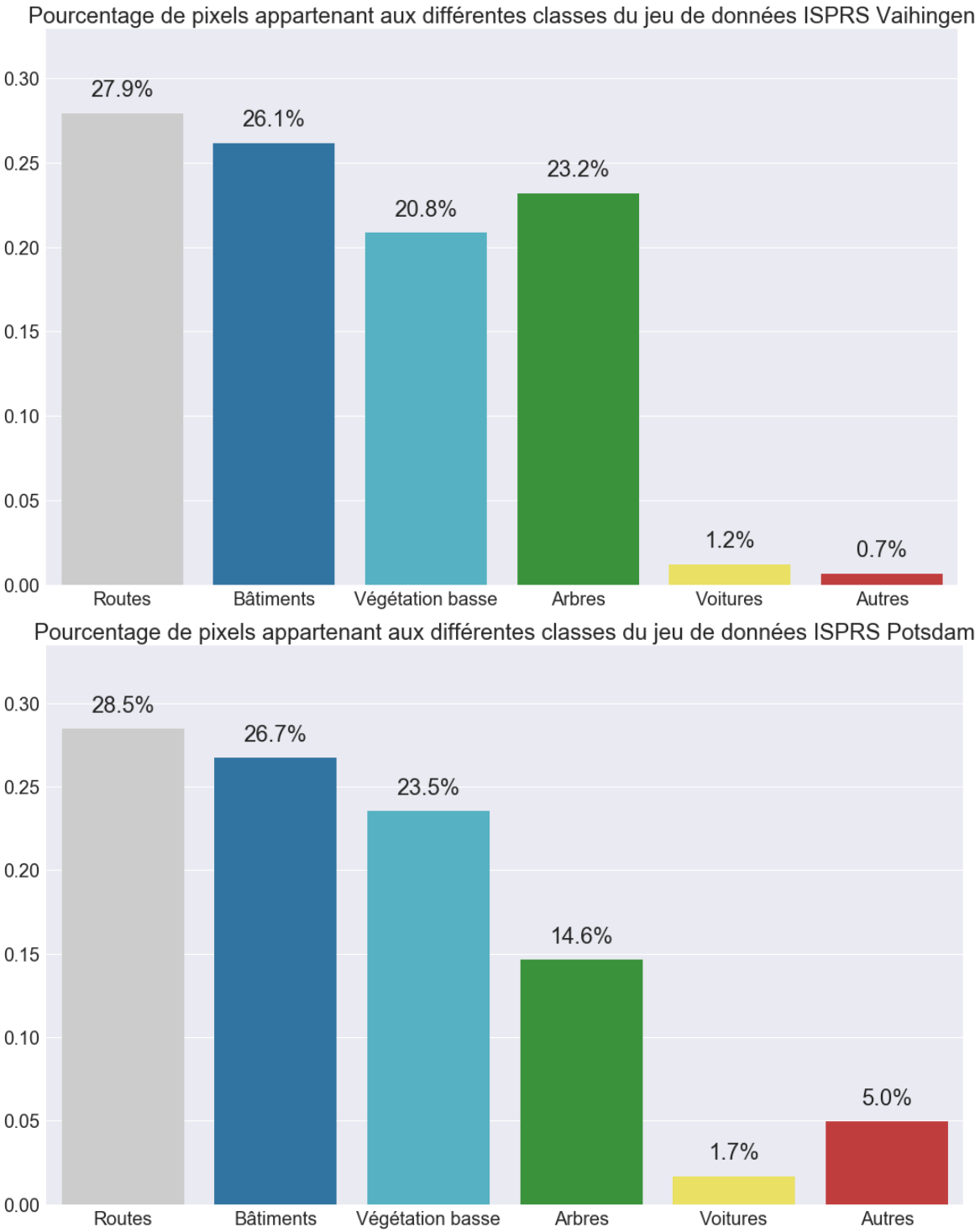
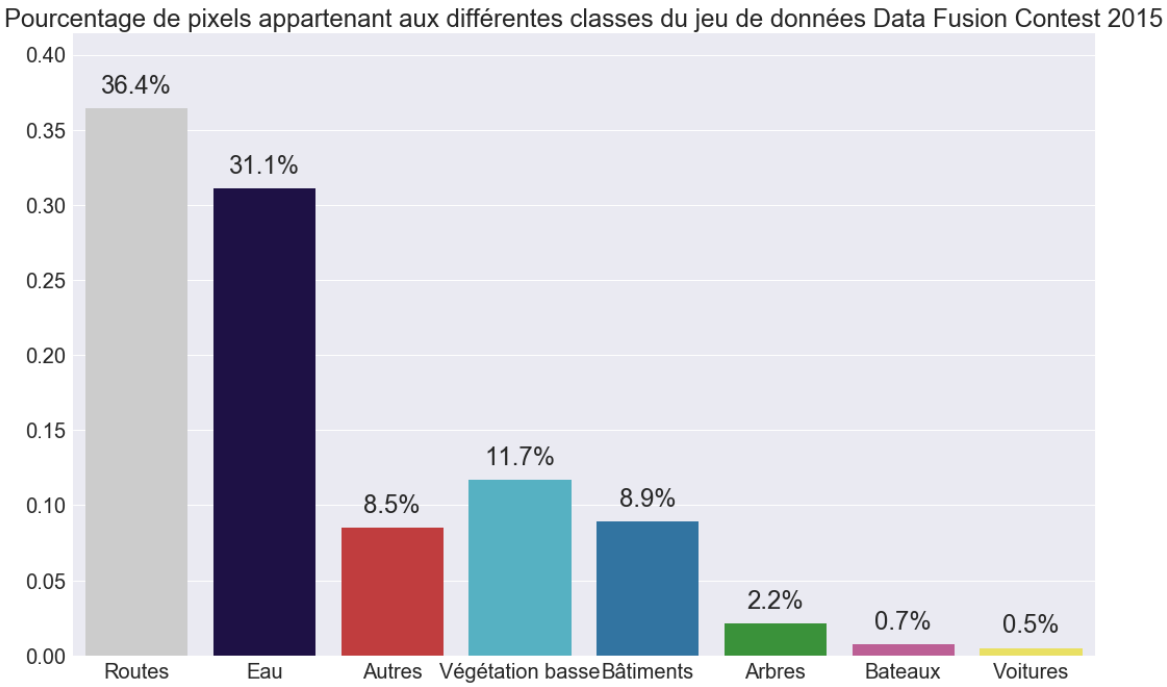
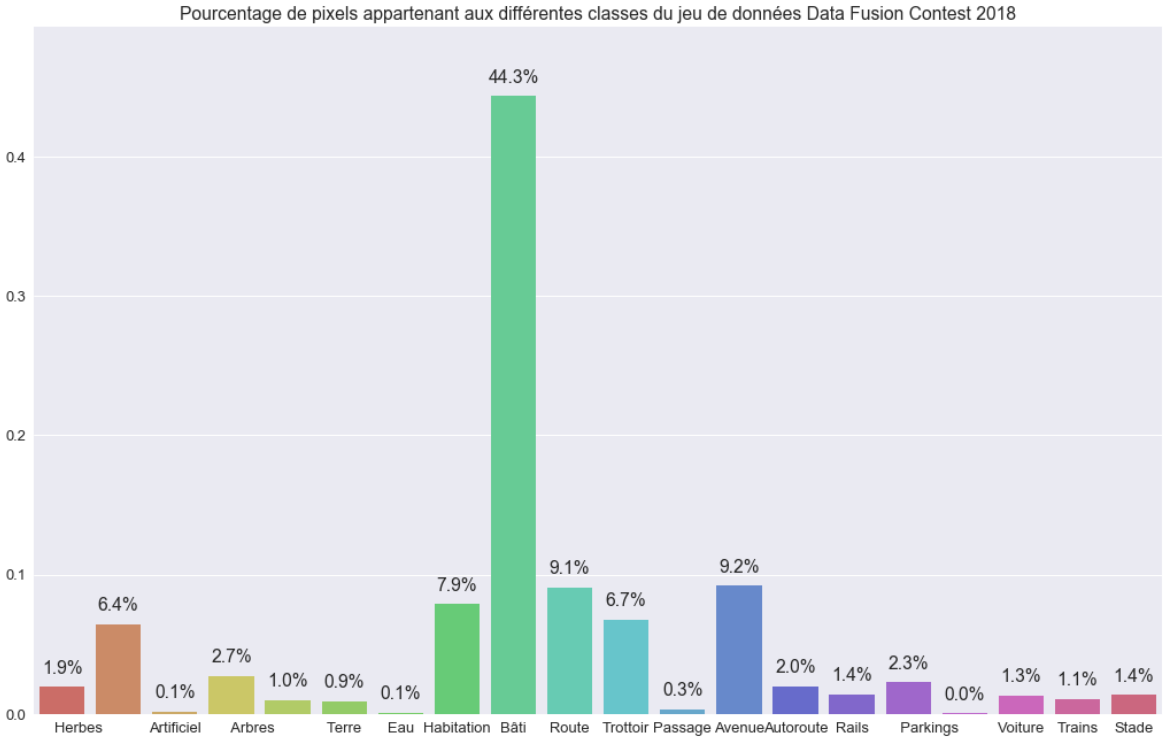


Figure A.10: Pixel distribution amongst the classes of the ISPRS dataset.



(a) Pixel distribution amongst the classes of the DFC 2015 dataset.



(b) Pixel distribution amongst the classes of the DFC 2018 dataset.

Figure A.11: Pixel distribution amongst the classes of the DFC datasets.

Bibliography

[1] Saikat Basu et al. "DeepSat: A Learning Framework for Satellite Imagery". In: *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems*. SIGSPATIAL '15. New York, NY, USA: ACM, 2015, 37:1–37:10. ISBN: 978-1-4503-3967-4. DOI: 10.1145/2820783.2820816. URL: http://doi.acm.org/10.1145/2820783.2820816 (cit. on p. I).

[2] Gabriel J. Brostow, Julien Fauqueur, and Roberto Cipolla. "Semantic Object Classes in Video: A High-Definition Ground Truth Database". In: *Pattern Recognition Letters*. Video-based Object and Event Analysis 30.2 (Jan. 15, 2009), pp. 88–97. ISSN: 0167-8655. DOI: 10.1016/j.patrec.2008.04.005. URL: http://www.sciencedirect.com/science/article/pii/S0167865508001220 (cit. on p. VII).

[3] Manuel Campos-Taberner et al. "Processing of Extremely High-Resolution LiDAR and RGB Data: Outcome of the 2015 IEEE GRSS Data Fusion Contest Part A: 2-D Contest". In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 9.12 (Dec. 2016), pp. 5547–5559. ISSN: 1939-1404. DOI: 10.1109/JSTARS.2016.2569162 (cit. on p. III).

[4] Markus Gerke. *Use of the Stair Vision Library within the ISPRS 2D Semantic Labeling Benchmark (Vaihingen)*. International Institute for Geo-Information Science and Earth Observation, 2015. URL: https://www.researchgate.net/profile/Markus_Gerke/publication/270104226_Use_of_the_Stair_Vision_Library_within_the_ISPRS_2D_Semantic_Labeling_Benchmark_(Vaihingen)/links/54ae59c50cf2828b29fcdf4b.pdf (cit. on p. II).

[5] Adrien Lagrange et al. "Benchmarking Classification of Earth-Observation Data: From Learning Explicit Features to Convolutional Networks". In: *2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. 2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS). July 2015, pp. 4173–4176. DOI: 10.1109/IGARSS.2015.7326745 (cit. on p. III).

[6] Bertrand Le Saux et al. "2018 IEEE GRSS Data Fusion Contest: Multimodal Land Use Classification [Technical Committees]". In: *IEEE Geoscience and Remote Sensing Magazine* 6.1 (Mar. 2018), pp. 52–54. ISSN: 2473-2397. DOI: 10.1109/MGRS.2018.2798161 (cit. on p. IV).

[7] Emmanuel Maggiori et al. "Can Semantic Labeling Methods Generalize to Any City? The Inria Aerial Image Labeling Benchmark". In: *Proceedings of the IEEE International Symposium on Geoscience and Remote Sensing (IGARSS)*. IEEE International Symposium on Geoscience and Remote Sensing (IGARSS). July 23, 2017. DOI: 10.1109/IGARSS.2017.8127684. URL: https://hal.inria.fr/hal-01468452/document (cit. on p. V).

[8] Otávio Penatti, Keiller Nogueira, and Jefersson A. dos Santos. "Do Deep Features Generalize from Everyday Objects to Remote Sensing and Aerial Scenes Domains?" In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). June 2015, pp. 44–51. DOI: 10.1109/CVPRW.2015.7301382 (cit. on p. I).

[9] Hicham Randrianarivo, Bertrand Le Saux, and Marin Ferecatu. "Urban Structure Detection with Deformable Part-Based Models". In: *2013 IEEE International Geoscience and Remote Sensing Symposium - IGARSS*. 2013 IEEE International Geoscience and Remote Sensing Symposium - IGARSS. July 2013, pp. 200–203. DOI: 10.1109/IGARSS.2013.6721126 (cit. on p. VI).

[10] Sébastien Razakarivony and Frédéric Jurie. "Vehicle Detection in Aerial Imagery: A Small Target Detection Benchmark". In: *Journal of Visual Communication and Image Representation* 34 (2016), pp. 187–203. DOI: 10.1016/j.jvcir.2015.11.002. URL: http://www.sciencedirect.com/science/article/pii/S1047320315002187 (cit. on p. V).

[11] Franz Rottensteiner et al. "The ISPRS Benchmark on Urban Object Classification and 3D Building Reconstruction". In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 1 (2012), p. 3. URL: https://t3sec3.rrzn.uni-hannover.de/cmsv021a.rrzn.uni-hannover.de/uploads/tx_tkpublikationen/isprsannals-I-3-293-2012.pdf (cit. on p. I).

[12] Shuran Song, Samuel P. Lichtenberg, and Jianxiong Xiao. "SUN RGB-D: A RGB-D Scene Understanding Benchmark Suite". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). June 2015, pp. 567–576. DOI: 10.1109/CVPR.2015.7298655 (cit. on p. VII).

[13] Yi Yang and Shawn Newsam. "Bag-of-Visual-Words and Spatial Extensions for Land-Use Classification". In: *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*. GIS '10. New York, NY, USA: ACM, 2010, pp. 270–279. ISBN: 978-1-4503-0428-3. DOI: 10.1145/1869790.1869829. URL: http://doi.acm.org/10.1145/1869790.1869829 (cit. on p. I).

Appendix B Code

It's still magic even if you know how it's done.

— Terry Pratchett (A Hat Full of Sky, 2004)

B.1 FCN for semantic mapping

Website: https://github.com/nshaud/DeepNetsForEO

This open source code implements the reference SegNet model from Chapter 3 for semantic segmentation of RGB and multispectral aerial and satellite images. Written in Python, this software uses the Pytorch library to run both the model training and inference either on GPU or CPU. Some parameters can be configured to reproduce the experiments described in the Chapters 3 to 5 on reference or custom datasets.

B.2 *DeepHyperX*

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

This open source code is the modular toolbox for hyperspectral image classification described in the Chapter 4. Written in Python, this software relies on the Pytorch and scikit-learn libraries. It is designed for two types of audience:

- Machine learning experts that aim to design, implement and validate new deep neural network architectures for hyperspectral data in a standard framework,
- Hyperspectral specialists that want to apply state of the art neural networks on their data.

B.3 *MiniFrance*

Website: https://gitlab.inria.fr/naudeber/FranceDataset

This open source code gathers the scripts used to build the *MiniFrance* dataset described in the Chapter 6. Written in Python and bash, these scripts can be used to convert the images from the BD ORTHO and rasterize the data from the French cadastre and *UrbanAtlas* on the same mosaic using the rasterio, fiona and geopandas libraries.

B.4 HyperGANs

Website: https://github.com/nshaud/HyperGANs

This open source code is a reference implementation of the conditioned Wasserstein-GAN for synthetic spectrum generation described in the Chapter 6. Written in Python, this software uses the Pytorch library. It can be used to reproduce the experiments in spectrum generation described in this manuscript on various datasets.