Contextual classification of large volumes of satellite imagery for the production of land cover maps over wide areas

Dawa Derksen

THÈSE

En vue de l'obtention du
DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par l'Université Toulouse 3 - Paul Sabatier

Présentée et soutenue par

Dawa DERKSEN

Le 2 décembre 2019

Classification contextuelle de gros volumes de données d'imagerie satellitaire pour la production de cartes d'occupation des sols sur de grandes étendues

---

Ecole doctorale : **SDU2E - Sciences de l'Univers, de l'Environnement et de l'Espace**

Spécialité : **Surfaces et interfaces continentales, Hydrologie**

Unité de recherche :
CESBIO - Centre d'Etudes Spatiales de la Biosphère

Thèse dirigée par
Jordi INGLADA

Jury

M. Gabriele MOSER, Rapporteur
M. Sébastien LEFÈVRE, Rapporteur
Mme Florence TUPIN, Examinatrice
M. Jordi INGLADA, Directeur de thèse

Abstract

Land cover mapping is defined as the description of the nature and use of the surface of the Earth. A reliable, up-to-date knowledge of this is of a great interest for many applications. For example, urban development, climate, or catastrophic event detection (forest fires, floods, etc.). Today, most land cover maps are produced using satellite imagery, for instance, the multi-spectral time series of the Sentinel-2 satellites. These regular observations along the seasons are key for identifying agricultural classes, which are characterized by a particular temporal behavior. Moreover, these images are taken with a High Spatial Resolution (HSR) of 10m, which allows for many elements of the landscape such as roads, rivers, and isolated buildings to be seen. From a cartographic point of view, a high spatial resolution allows a finer delineation of the contours of the main objects in the image, and also allows the smallest among them to be seen. For example, mono-date images at a Very High Spatial Resolution (VHSR) of 1.50m, such as the ones produced by SPOT-7, can spot out streets and cars.

To produce a land cover map from satellite imagery, the "manual" approach, that consists in identifying each pixel from an image using expert knowledge, as is done for Corine Land Cover, is limited by the long update time it implies (6 years for CLC).

Thankfully, certain *supervised classification* algorithms have long since been designed to rapidly label the elements of a data set, using a set of already labeled samples. The production of the OSO map, which covers France, is done every year since 2016 using a supervised classifier.

One of the difficulties that comes forth is the discrimination of classes that depend more on the context of the pixel than on its content. This is the case for urban cover classes (discontinuous/continuous urban fabric), where the difference does not lie in the pixel features, but rather in certain properties of the neighborhood.

The objective of this Ph.D. is therefore to compare several methods of contextual inclusion, in order to improve the quality of land cover maps.

The standard definitions of the neighborhood (its shape) are the sliding window, and the object. However, an interesting intermediary representation is studied here : the superpixel. These segments are adaptive to local neighborhood, but have constraints on their size.

Then, in order to describe the contents of the neighborhood, there are two main ways in which to proceed, using a set of *contextual features*, or using a *model-based* approach. In the first case, certain descriptors of the neighborhood such as texture or shape are calculated. In the second case, the entire context is provided to the model, which can therefore be designed to apprehend the geometric elements in the image. This is the case of Deep Convolutional Neural Networks (D-CNN).

This Ph.D. presents a new method for contextual inclusion, which consists in calculating the histogram of predicted classes in the neighborhood. Starting with a pixel-based classification, it becomes possible to iterate several successive steps of classification, which recalculate the histograms using the previous classifications.

On two very different sets of experiments (Sentinel-2 and SPOT-7), it appears that the D-CNN provide maps with a lower geometric quality compared to contextual features. This translates as a rounding of the sharp corners, and a deformation or erasing of the fine elements. In terms of class recognition, the conclusions diverge. In the Sentinel-2 case, the use of class histograms provides a statistically equivalent class accuracy to D-CNN, whereas on the SPOT-7 case, the D-CNN have a higher class recognition rate.

Abstract

L'*occupation des sols* est définie comme la description de la nature et de l'usage anthropique de la surface de la Terre. Une connaissance fiable et à jour de celle-ci est d'un grand intérêt pour de nombreuses applications, par exemple, le développement urbain, le climat, ou pour la détection d'évènements catastrophiques (feux de forêt, inondations, etc.). Actuellement, la plupart des cartes d'occupation des sols sont produites à partir d'images satellitaires, par exemple, les séries temporelles multi-spectrales de Sentinel-2. Ces observations régulières au cours de l'année permettent d'identifier les classes agricoles, qui sont caractérisées par un comportement temporel particulier. De plus, ces images sont capturées à une résolution spatiale de 10m, ce qui permet de voir de nombreux éléments du paysage, comme les routes, les rivières, et certains bâtiments isolés. D'un point de vue cartographique, une haute résolution spatiale permet une détection plus fine des contours des objets principaux dans l'image, et de voir de plus petits objets. Par exemple, les images mono-date à Très Haute Résolution Spatiale, comme celles de SPOT-7 à 1,50m, capturent les ruelles et les voitures.

Pour produire une carte à partir d'images satellitaires, l'approche "manuelle" qui consiste à identifier chaque pixel d'une image à l'aide de connaissance experte, comme pour Corine Land Cover (CLC), est limitée par une durée de mise à jour trop longue (6 ans pour CLC).

Certains algorithmes de *classification supervisée* sont conçus pour étiqueter rapidement les éléments d'un jeu de données à partir d'un ensemble d'exemples connus au préalable. La production de la carte OSO se fait tous les ans depuis 2016 à l'aide d'un classifieur supervisé.

Une des difficultés rencontrées pour celle-ci est la différentiation de certaines classes qui dépendent plus du contexte du pixel que de son contenu. C'est le cas pour les classes urbaines, (urbain dense/urbain diffus), où la distinction ne se fait pas au niveau des primitives décrivant le pixel, mais de certaines propriétés dans son voisinage.

L'objectif de la thèse est donc de concevoir et de comparer plusieurs méthodes de prise en compte du voisinage des pixels, pour améliorer la qualité des cartes d'occupation des sols.

La définition de ce voisinage (sa forme) peut varier du voisinage carré à l'objet, en passant par le superpixel, qui est un intermédiaire entre les deux. Ces segments sont adaptatifs au voisinage, mais sont contraints au niveau de leur taille.

Ensuite, pour décrire le contenu du voisinage, deux familles de méthodes existent : les approches dites de *primitives contextuelles*, et les approches dites de *modélisation*. Dans le premier cas, il s'agit de calculer certains descripteurs du contexte, comme la texture ou la forme. Dans le deuxième cas, le contexte entier est fourni au modèle, qui peut alors être conçu pour appréhender les éléments géométriques de l'image. C'est le cas des Deep Convolutional Neural Networks (D-CNN).

Cette thèse présente une nouvelle méthode de prise en compte du contexte, qui consiste à calculer une primitive particulière : l'histogramme des classes dans dans un voisinage. En partant d'une classification des pixels, on peut alors itérer plusieurs étapes de classification successives, qui recalculent les histogrammes à l'aide des classifications précédents.

Sur deux jeux d'expériences très différents (Sentinel-2 et SPOT-7), on observe que les méthodes D-CNN fournissent des cartes avec une qualité géométrique dégradée par rapport aux primitives contextuelles. Cela se traduit par un arrondissage des coins et une déformation des éléments fins. En terme de précision thématique, on observe une divergence dans les conclusions. Dans le cas Sentinel-2, l'utilisation des histogrammes des classes fournit un résultat statistiquement équivalent aux D-CNN, tandis que sur le cas SPOT-7, les D-CNN ont une meilleure performance.

Acknowledgements

Working on one subject for three years is both demanding, and rewarding in many ways. This time frame seems long at first, but is in fact a short amount of time, given the scope of the problem at hand. Nonetheless, thanks to the help of the many people I have known and met during these years, it has been an incredible learning experience, both on a scientific and personal level. The rest of the manuscript mainly discusses the scientific aspects, but these paragraphs also show some of the personal lessons I have learned, thanks to my colleagues, friends, and family.

First and foremost, I would like to extend my deep gratitude to my Ph.D. director, Dr. Jordi Inglada. Through a full-hearted investement in the subject at every level, he has guided me towards a greater understanding of land cover mapping, but also of artificial intelligence, computer science, programming and software development. His unwavering presence and valuable insights were key in furthering this challenging topic without falling into the traps of unnecessary complexity. His dedication was greatly appreciated, both during the gruelling hours of debugging, and the many enriching discussions we shared.

I would also like to thank the Centre National d'Etudes Spatiales, for participating in the financing of this Ph.D., and for providing me with the necessary ingredients for research work. Beyond the new computer and the access to the High Performance Calculator, I was co-directed by Julien Michel, who introduced me to researchers and engineers at the space agency. He also provided me with many of the scientific insights regarding the subject, in particular about open-source software and image processing. His willingness to co-author scientific papers and spend time revising this manuscript was essential to the quality of this Ph.D work, and for that, I am very thankful. I also appreciated the help of Dr. Pierre Lassalle, whose continued investment in his Ph.D. work provided a solid stepping stone to my research. I would like to extend further thanks to Victor Poughon, Manuel Grizonnet, and the other CNES colleagues who followed my work from near or far.

This work would have been impossible without the help of Atos, who offered both financial and technical support. In particular, thanks to Tarik Habib, Chadi Jaber, Aurore Dorez and Alain Lefrançois.

Next, I would like to thank Vincent Poulain from Thales Services and Andrei Stoian from ThereSiS, for providing me both with insights on the Deep Learning side of the subject, but also for giving me access to their model and its results, and providing me the opportunity to co-author a paper.

I would also like to show my appreciation towards the Institut Géographique National (IGN), in particular Clément Mallet and Tristan Postadjian. They have given me many methodological insights regarding the application of Deep Learning models to VHSR imagery, and detailed data to evaluate and compare these methods.

It is also important for me to thank my colleagues at the Centre d'Etudes Spatiales de la BIOsphère, (CESBIO) laboratory, who helped in creating an enriching work environment. I greatly appreciated the help of Dr. Silvia Valero, who closely followed my work and encouraged me to teach programming and image processing at the university, and Vincent Thierion for his support regarding geodatabases.

I would also like to thank my fellow Ph.D. students, whom I felt were the only ones to truly understand how trying such a position can be, at times. First of all, Benjamin Tardy, whose work on different aspects of the same fundamental problem of land cover mapping was very enriching in many regards. His patience in demystifying incomprehensible compilation errors was extremely helpful in learning about software systems, and in saving time. Secondly, a huge thank you to Gaétan Pique, for being a worthy chess opponent, a patient teacher in the art of QGIS, and a true friend.

A further thanks to my intern, Lucas Schwaab, for his hard work and determination in providing a full and detailed analysis the of geometric precision of land cover maps in urban areas. His works were key to furthering

the research on what is in fact a complex, multifaceted issue. Some of the key results produced during his internship are in fact shown in this manuscript. I also feel grateful towards Arthur Vincent, for helping me understand the $iota^2$ processing chain, and for his willingness to integrate aspects of my work into it.

There are many other colleagues at the laboratory whose friendship and support I am appreciative for. I would like to thank Laurent Polidori, Mathieu Fauvel, Florian, Yann R., Yann P., Hoa, Colette, Juliette, Solène, Cédric, Bertrand, Marc, Marine and Emma, for the trips, coffee breaks, meals, or other moments enjoyed together.

Outside of the work sphere, I have also heavily relied on the unwavering support of my friends and family. Thank you to Nadège, Paul, Mel, Pierre, and Tristan for helping me broaden my knowledge of game theory, and to the Missing Ingredients, Andrea, Mark, Jean-Baptiste and Fabien for keeping my work in rythm with weekly improvisation sessions. I also offer my deepest gratitude to Nada, Tom, Ruben and Rosey, my dearest family, for unconditionally supporting me at every step along the way.

Contents

Introduction en français

L'*occupation des sols* est définie comme la description de la nature et de l'usage anthropique de la surface de la Terre. Une connaissance fiable et à jour de celle-ci est d'un grand intérêt pour de nombreuses applications, par exemple, le développement urbain, le climat, ou pour la détection d'évènements catastrophiques (feux de forêt, inondations, etc.). Actuellement, la plupart des cartes d'occupation des sols sont produites à partir d'images satellitaires, par exemple, les séries temporelles multi-spectrales de Sentinel-2. Ces observations régulières au cours de l'année permettent d'identifier les classes agricoles, qui sont caractérisées par un comportement temporel particulier. De plus, ces images sont capturées à une résolution spatiale de 10m, ce qui permet de voir de nombreux éléments du paysage, comme les routes, les rivières, et certains bâtiments isolés. D'un point de vue cartographique, une haute résolution spatiale permet une à la fois une détection plus fine des contours des objets principaux dans l'image, et permet de voir les plus petits éléments. Par exemple, les images mono-date à Très Haute Résolution Spatiale, comme celles de SPOT-7 à 1,50m, capturent les ruelles et les voitures.

Le projet dans lequel s'inscrit la thèse vise à exploiter les séries temporelles d'images optiques Sentinel-2 à hautes résolutions spatiale, spectrale et temporelle pour produire des cartes d'occupation des sols à échelles nationale et continentale. Ces chaînes de traitement seront ensuite implantées dans le Pôle THEIA, dont la mission est de fournir des produits thématiques sur les surfaces continentales comme l'occupation des sols, mais aussi la hauteur des lacs et rivières, les variables du cycle de l'eau etc.

Pour produire une carte à partir d'images satellitaires, l'approche "manuelle" qui consiste à idenfier chaque pixel d'une image à l'aide de connaissance experte, comme pour Corine Land Cover (CLC), est limitée par une durée de mise à jour trop longue (6 ans pour CLC).

Certains algorithmes de *classification supervisée* sont conçus pour étiqueter rapidement les éléments d'un jeu de données à partir d'un ensemble d'exemples connus au préalable. La production de la carte OSO se fait tous les ans depuis 2016 à l'aide d'un classifieur supervisé. Pour produire une carte, une étiquette de classe (urbain, maïs, forêt, etc.) est associée à chaque pixel de l'image satellitaire, qui est décrit par une série temporelle de réflectances dans plusieurs bandes ainsi que d'indices spectraux. Pour une nomenclature de classes donnée, la qualité d'une carte d'occupation des sols peut être décomposée en deux critères principaux.

1. La qualité thématique ou sémantique, qui traduit la précision de la reconnaissance de chaque classe.

2. La qualité cartographique ou géométrique, qui s'intéresse à la forme des contours des objets, et à leur localisation dans l'absolu. En quelque sorte, cette notion peut être vue comme la précision de la carte au sens spatial.

Au premier ordre, ces deux critères semblent entièrement dépendants du type d'imagerie utilisé. La qualité thématique dépend des dimensions temporelles et spectrales de l'image, qui devraient permettre d'identifier les classes de la nomenclature. Ensuite, la qualité cartographique dépend de la résolution spatiale, qui permet une meilleure localisation des contours et de la forme des objets. Ainsi, des images contenant plus de bandes spectrales, plus de dates d'acquisition, et à une fine résolution spatiale devraient produire de meilleures cartes.

Cependant, ce n'est pas toujours aussi simple que cela. L'analyse des cartes OSO met en avant l'existence de classes pour lesquelles la fine résolution spatiale semble dégrader la qualité thématique. En particulier, cet effet peut se manifester si la taille des pixels de l'image est plus petite que les objets décrits dans la nomenclature. Dès lors, la description des pixels peut ne pas suffire à caractériser entièrement la classe. C'est le cas pour les classes urbaines, (urbain dense ou urbain diffus), où la différence ne se fait pas au niveau des bandes spectrales décrivant

le pixel, mais de certaines propriétés dans le voisinage. C'est également le cas pour les pixels de sol nu au sein des exploitations agricoles, qui sont contenus dans plusieurs classes différentes et qui sont impossibles à distinguer avec l'information du pixel seul. L'utilisation de l'information présente dans le voisinage d'un pixel devrait alors améliorer la précision de classification pour ces classes, dites à *dépendances contextuelles*.

L'objectif de la thèse est alors de concevoir et comparer plusieurs méthodes de prise en compte du voisinage des pixels, pour améliorer la précision des cartes d'occupation des sols. Ces méthodes doivent être cohérentes avec le cadre de la production *opérationnelle*, qui est définie comme le développement et la livraison de produits robustes dans un délai de production prédéfini [Inglada et al., 2017]. De plus, on peut y ajouter les contraintes imposées par la grande étendue de la couverture des cartes désirées, qui se fait actuellement sur la France mais dont on devrait pouvoir envisager des applications à plus large échelle encore. Cela se traduit par des contraintes à la fois sur le plan théorique et pratique.

En effet, la classification d'une grande étendue ajoute de la *variabilité intra-classe*, en comparaison à la classification d'une petite zone. Les classes d'occupation des sols peuvent avoir un comportement très différent selon la zone géographique. Pour les classes de végétation, ceci peut être lié à l'écologie ou au climat de la région, qui varie sur un grand territoire. Il en est de même pour les classes urbaines. L'aspect des villes est rarement identique, et dépend souvent des matériaux localement disponibles pour la construction. Ce problème n'est pas directement abordé dans cette thèse, cependant, il se peut qu'il soit résolu dans une certaine mesure par l'utilisation d'une méthode de classification qui permet de modéliser des classes plus complexes en incluant de l'information contextuelle. On peut toutefois mentionner que la variabilité intra-classe induite par la classification sur de grandes étendues est réduite par une stratification éco-climatique, introduite par [Inglada et al., 2017], et définie dans la Section 3.5.1.

A cela viennent s'ajouter les contraintes pratiques d'un grand volume de données, qui pousse vers l'utilisation de méthodes peu coûteuses en temps de calcul.

Cela s'articule concrètement en trois facteurs à prendre en compte pour le choix de la méthode contextuelle à adopter :

1. Le type d'imagerie utilisé, à savoir les *primitives pixel* et les résolutions spatiale, spectrale et temporelle.
2. La donnée de référence, en particulier la nomenclature des classes et les dépendances contextuelles que celle-ci peut contenir. Un autre aspect intéressant est la densité relative des polygones ou des points d'apprentissage.
3. Les propriétés de l'infrastructure de calcul disponible, car certaines méthodes de classification contextuelles peuvent être très coûteuses.

Ces facteurs conditionnent les méthodes applicables au problème de l'occupation des sols opérationnelle à large échelle, et plus généralement forment le paradigme dans lequel s'inscrit cette recherche.

En premier lieu, il s'agit de trouver des approches entièrement automatiques, c'est-à-dire sans intervention manuelle ou réglage de paramètres, afin d'assurer que les procédés développés puissent être appliques à de très gros volumes de données pour un coût fixe.

Ensuite, la généricité vis-à-vis des classes cherche à être conservée, pour pouvoir appliquer la méthode à tout nombre et à tout type de classes.

Troisièmement, les méthodes développées ne devraient pas se limiter à l'imagerie optique. Des méthodes directement applicables à d'autres types d'imagerie (radar, hyperspectral), voire à des données multi-modales, sont recherchées.

Enfin, il y a un effort pour comprendre et analyser le processus d'apprentissage automatique à ses diverses étapes, et d'analyser autant que possible la façon dont le processus de décision est construit. Cela nous permet d'interpréter la cause de certaines erreurs, afin de continuer d'améliorer le processus sur le long terme.

Le manuscrit de thèse est divisé en cinq parties.

- **Partie I** Cette partie présente le contexte dans lequel s'inscrit cette recherche, ainsi que les objectifs principaux du doctorat. Le chapitre 1 présente les cartes d'occupation des sols et leurs applications principales, ainsi que les concepts de base associés à leur production, à savoir, la photo-interprétation, les approches par ontologie, et la classification supervisée. Grâce aux algorithmes d'apprentissage automatique (Machine Learning), le Centre d'Expertise Scientifique OSO du Centre de Données Theia [Leroy et al., 2013] produit une carte d'occupation des sols annuelle à l'échelle nationale, dans un délai raisonnable. Cependant, certaines classes, en particulier les classes à dépendance contextuelle ont des taux d'erreur élevés. La position de ce travail de thèse vis-à-vis de ce problème est présenté, ainsi que les contraintes et les objectifs qui en découlent. Ensuite, le chapitre 2 décrit les satellites d'observation de la Terre qui produisent les images utilisées dans les expériences, et comment ces satellites assurent une couverture globale. Enfin, le chapitre 3 présente la chaîne de production qui produit la carte OSO annuelle, des images jusqu'à la carte d'occupation des sols.

- **Partie II** Ici, les bases de l'évaluation expérimentale, ainsi que les définitions théoriques et analytiques qui sont fondamentales pour comprendre le travail réalisé ici sont présentées. Un des objectifs est d'établir une taxonomie des méthodes de classification contextuelle. Le chapitre 4 liste les diverses façons de définir le contexte, qui est caractérisé principalement par sa taille et sa forme. On peut aussi imaginer prendre plusieurs contextes de taille et de forme différentes, car les dépendances contextuelles dans les classes peuvent s'exprimer à plusieurs échelles. Ce problème n'est pas nouveau, et une revue de la littérature concernant les supports spatiaux communément utilisés pour des problèmes de télédétection est alors nécessaire. Un support spatial en particulier, le superpixel, est décrit avec détail car il fournit une représentation intermédiaire entre l'objet et le voisinage fixe. Ensuite, les descripteurs contextuels qui peuvent être utilisés pour caractériser le contexte sont étudiés en détail dans le chapitre 5, afin de déterminer lesquels d'entre-eux sont pertinents pour la classification de l'occupation des sols. Enfin, le chapitre 6 aborde le problème de l'évaluation des cartes de l'occupation des sols à base de données de référence éparses, en particulier vis-à-vis de la précision cartographique. Pour cela, une nouvelle métrique qui mesure la dégradation des coins en comparaison à une classification pixel est développée, et fait l'objet d'une contribution méthodologique.

- **Partie III** Les contributions méthodologiques principales sur le sujet de la classification contextuelle sont présentées dans cette partie. Le chapitre 7 présente la méthodologie adoptée pour appliquer l'algorithme de segmentation superpixel SLIC aux grandes images, car cela pose des problèmes théoriques. In fine, cela permet aux méthodes superpixel d'être appliquées et comparées aux autres supports spatiaux sur le jeu de données Sentinel-2. Ensuite, le chapitre 8 recherche diverses façons de réduire la grande dimension des données, tout en gardant la pertinence de celles-ci vis à vis du contexte spatial. Pour ce faire, une nouvelle primitive contextuelle est étudiée, l'histogramme des classes prédit par un classifieur pixellique. En effet, la proportion des classes dans un voisinage est un descripteur relativement léger en comparaison à beaucoup d'autres méthodes. Il représente un nombre de primitives égal au nombre de classes dans la nomenclature. La capacité de cette primitive à améliorer la qualité des cartes d'occupation des sols sera comparée aux méthodes de l'état de l'art, les Réseaux de Neurones Convolutionnels (CNN), qui sont introduits et définis dans le chapitre 9.

- **Partie IV** Cette partie montre les résultats des expériences qui ont été menées pour évaluer les méthodes présentées dans les parties II et III. Les chapitres 10 et 11 montrent respectivement l'analyse des résultats sur le jeu de données Sentinel-2, et sur le jeu de données SPOT-7. En premier, les images multi-temporelles de Sentinel-2 sont classifiées et évaluées à l'aide des mêmes données de référence que les cartes OSO. Cette évaluation représente un problème intéressant car l'inclusion du contexte n'a jamais été testée dans ce cadre. Afin d'être représentatif de l'occupation des sols sur le territoire, une variété de zones différentes sont choisies, avec chacune un paysage et un comportement éco-climatique unique. Ensuite, un problème d'imagerie THRS (1,50m) SPOT-7 est étudié. Ce problème a une nomenclature réduite à cause de l'aspect mono-date de ces images.

- **Partie V** Le chapitre final présente les conclusions et les perspectives de ce travail, à la fois d'un point de vue méthodologique et expérimental. Le chapitre 12 rappelle les éléments du problème de l'inclusion contextuelle, et synthétise les principales conclusions de la partie IV. Ensuite, la validité de ces conclusions vis-à-vis d'un critère multi-objectif est discutée, car les méthodes d'évaluation quantitative sont encore limitées. Cela mène à des suggestions pour améliorer les méthodes proposées, puis pour approfondir la recherche sur d'autres sujets similaires.

Part I

Introduction to land cover maps

1 The operational production of land cover maps

"Before computers, telephone lines and television connect us we all share the same air, the same oceans, the same mountains and rivers. We are all equally responsible for protecting them."
– Julia Louis-Dreyfus

This chapter introduces the context of this Ph.D. thesis work, namely, land cover maps, time series, the target classes and the data bases, and the objectives and scope.

1.1 Land cover maps

As far as we currently know, the Earth is the only place in the universe that harbors life, and it has been teeming with living organisms for at least three billion years. To understand our planet's history and hope to accurately forecast its future, we must take in account the role of the biosphere, the portion of our world's land, atmosphere and oceans within which life exists. Understanding our environment is a complex task because the vast, widespread mass of microbes, plants and animals on Earth are interlinked, forming a dynamic global system.

Today, satellites orbiting the Earth can capture high resolution images of the surface, giving a glimpse of the forests, fields, cities, roads, rivers and oceans that make up the world we live in. Pictures acquired remotely at different dates, or by different sensors, can be pieced together in order to form mosaics of images, which can cover countries, continents, or even the entire globe. By using Earth Observation techniques from space, we can monitor environmental change at an unprecedented scale. Modern satellite images are captured at a spatial resolution of the order of the meter (10cm - 10m), which is sharp enough to see cars, roads, houses, rivers, and the crown of most trees.

These remote observations can be used to describe our surroundings in terms of *land cover classes*, like the aforementioned forests, fields, and so on, to form a *land cover map*. These maps provide a first level of understanding of a satellite image, and therefore of the territory that is captured. This is done by categorizing the different objects in the image into easily interpretable classes, which form the *nomenclature* of the map.

There is a distinction to be made between the *land cover* (LC) and *land use* (LU) of a given area, the former describing its physical nature, and the latter describing the use which we make of it. For example, *slate tiles*, *asphalted surfaces*, and *grass* are land cover classes, while *residential area*, *highway* and *leisure park* could be corresponding land use classes. Ideally, a LU/LC map describes both the use and the cover of each area, by assigning one LC label and one LU label to each area [Comber et al., 2005]. However, as it turns out, many of the currently produced LU/LC maps only assign one class label to each area, with a nomenclature containing a mix of both LU and LC classes. For this reason, and for the sake of simplicity, the term *LC* will henceforth be used, rather than *LU/LC*.

Land cover maps are useful for several studies regarding the environment, a few examples of which are given here.

Yearly maps of urban areas show the way in which cities evolve over decades of time. Watching the construction of new industrial or residential areas and the establishment of a network of roads connecting them provides valuable information for urban planning. This can help to identify green areas such as parks and other natural areas in cities, or neighborhoods with a high risk of pollution. Land cover mapping techniques have also recently been used to survey slums, and other housing units built outside of a legal infrastructure [Kuffer et al., 2016]. While these are found mainly in developing countries, they are also present in many developed economies. These areas support

a large number of inhabitants, according to UN-Habitat, around 33% of the urban population in the developing world in 2012, or about 863 million people, lived in slums [United Nations Human Settlement Programme, 2012]. These areas can be difficult to map out in a timely and precise fashion using on-ground surveys and measurements. Figure 1.1 shows the particular structure of slums, which is very visible in satellite images.

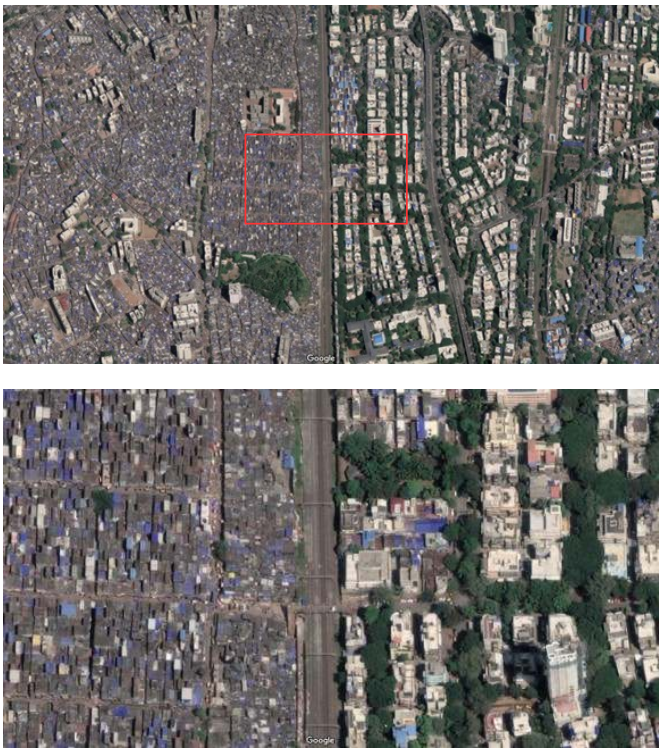


Figure 1.1: Satellite image taken over the Dharavi locality in Mumbai, India. An estimated 600,000 to 1,000,000 people inhabit this 277,000km^2 slum [Ragheb et al., 2016]. These images show the stark contrast between the structure of slum (left) and non-slum (right) neighborhoods.
Source: Google Maps https://www.google.com/maps

Land cover maps can also provide valuable information for food security and agricultural management; identifying the major crops can help governmental agencies administer the agricultural practices applied in their country, to adjust the subsidies and taxes for the years to come. For example, the European Common Agricultural Policy (CAP) [Fouilleux and Ansaloni, 2016] is based on detailed yearly crop maps. This information can also guide the management of water consumption for the irrigation of certain crops, through the detection of humidity levels, as is done in [Toureiro et al., 2017]. It can be used in conjunction with other measurements, like soil humidity, to detect drought events [Nicolai-Shaw et al., 2017], or to analyse their long-term impact [Schwalm et al., 2017].

Forests cover nearly a third of the surface of our land, and play a very important role in supporting the life of many of the plants and animals on Earth. Over the past two decades, more than half of the natural rainforests in Borneo have been replaced with orchards of oil palm trees [Austin et al., 2017]. These mass monocultures are known to support relatively few species, and for that reason much of the biodiversity of the island has unfortunately disappeared. Mapping out natural and agricultural areas like these provides quantitative arguments which can help maintain the rich variety of flora and fauna on our planet. Land cover maps can be used for the preservation of natural biodiversity, through the large-scale recognition of the habitats of various plant and animal species [Turner et al., 2003, Revermann et al., 2016].

Another interesting application of these maps is for studies concerning the climate, in particular, climate change. Land cover directly influences the climate by modifying water and energy exchanges with the atmosphere and by changing greenhouse gas and aerosol sources and sinks. Land cover information is considered an Essential Climate Variable (ECV) [Bojinski et al., 2014], as it serves as an input for estimating key variables like surface albedo, moisture levels, biomass, evapotranspiration, aerodynamic characteristics, and many others.

Remote Sensing data, and land cover maps in particular, can also be used for natural disaster risk analysis [Van Westen, 2013]. By pinpointing areas which are at the highest risk of falling victim to catastrophic events, we

can establish escape routes, construct emergency shelters, or create sophisticated alert systems. A few practical applications include mitigating the risk of floods, forest fires, volcanic eruptions, landslides and seismic events like earthquakes [Joyce et al., 2009].

Land Cover maps offer a wide variety of uses, and while the particular mapping problems studied in this Ph.D. do not address all of the issues mentioned above, furthering research on this topic can have a positive impact on many of these problems.

It is important to note that this work inscribes itself in the context of *operational* land cover mapping. We define this as describing the development and delivery of reliable data products within a pre-defined time schedule. This is different from purely experimental approaches for land cover mapping and monitoring, which focus on the development and performance testing of novel algorithms and models.

Space-borne observations are extremely valuable for mapping the cover of the Earth's surface, and are at the core of the research work done during this Ph.D. This chapter is organized in three parts. First, Section 1.2 introduces the basic notions and vocabulary of satellite imagery, which are necessary to understand how satellite images are converted in maps. This is done through a process known as *classification*, which is covered in Section 1.3. Then, the chapter ends with the definition of the principal research objectives and problems of this work, Section 1.4.

1.2 Remote sensing from space

Satellites have been capturing images of the Earth since NASA's TIROS (Television Infrared Observation Satellite) was used to monitor weather patterns in 1960 [Tatem et al., 2008]. Today, hundreds of Earth Observation satellites orbit our planet, gathering a variety of information regarding many aspects of the Earth's surface and atmosphere.

The most common Earth Observation satellites bear *passive* sensors, which measure the light reflected or emitted by the surface. Among these are optical sensors, which are very often designed for the visualization of images, by capturing either the overall luminance (gray-scale image), or the Red, Green and Blue wavelengths of the electromagnetic (EM) spectrum for color images. These sensors are in fact able to measure many different wavelengths of the spectrum, ranging from the near ultra-violet to the infrared portions of the spectrum. Optical sensors measure the sunlight reflected by the surface, and therefore can only be taken during the day, and are often obstructed by clouds. Famous optical Earth Observation satellites include the Landsats 1-8 [Markham and Helder, 2012], Sentinels 2 and 3 [Drusch et al., 2012], shown in Figure 1.2 and the Système Probatoire d'Observation de la Terre family, more commonly known as SPOT (1-7) [Chevrel et al., 1981]. Chapter 2 explains how these sensors are able to capture images at a global scale.

Other passive sensors measure the light emitted by the Earth in thermal infrared wavelengths (9-14 μm), rather than the light reflected from the Sun. This is useful to estimate surface temperatures which are linked to the amount of radiation in this portion of the spectrum. An example of such a satellite is the ASTER imager [Abrams, 2000].

Another technology involves actively scanning the surface of the Earth using a Radio Detecting And Ranging system, more commonly known as a Radar. These satellites emit signals in the radio wavelengths of the EM spectrum, which pass through clouds, but scatter off of hard surfaces and return back to the sensor a brief moment later. By measuring the time difference between emission and reception, and the polarization of the received wave, information regarding the nature of the surface can be derived, like roughness, topography, and the presence of tall, vertical structures. Images from sensors like ASAR aboard ENVISAT [Arnaud et al., 2003] have been used in tandem with optical images for characterizing the presence of irrigation in fields [Hadria et al., 2009].

Optical images have four main characteristics: spatial resolution, spectral resolution, temporal resolution, and covered area.

- *Spatial resolution* defines the physical distance between two points that are distinguishable in the image, not to be confused with the ground sampling distance, which defines the area covered by one pixel in the image. If the image is visualized at full resolution, these two have the same value, but images are often sampled with larger pixel sizes to avoid aliasing effects, in particular when applying transformations such as rotations. High Spatial Resolution (HSR) images like the ones Landsat-8 and Sentinel-2 have a spatial resolution of 10-100m, and are not to be confused with Very High Spatial Resolution (VHSR) images, which have spatial resolutions ranging from 0.1m-10m, like the SPOT-7 images. In general, the different wavelengths of an optical image are captured at different spatial resolutions, due to the technical constraints linked to the use of infrared sensors.

- *Spectral resolution* or *spectral range* specifies the wavelengths of light which are acquired by the imaging sensor. Each band is defined with one central wavelength and a spectral width, which represents how fine the band is. In order to distinguish the previously mentioned agricultural, natural, and artificial classes, many

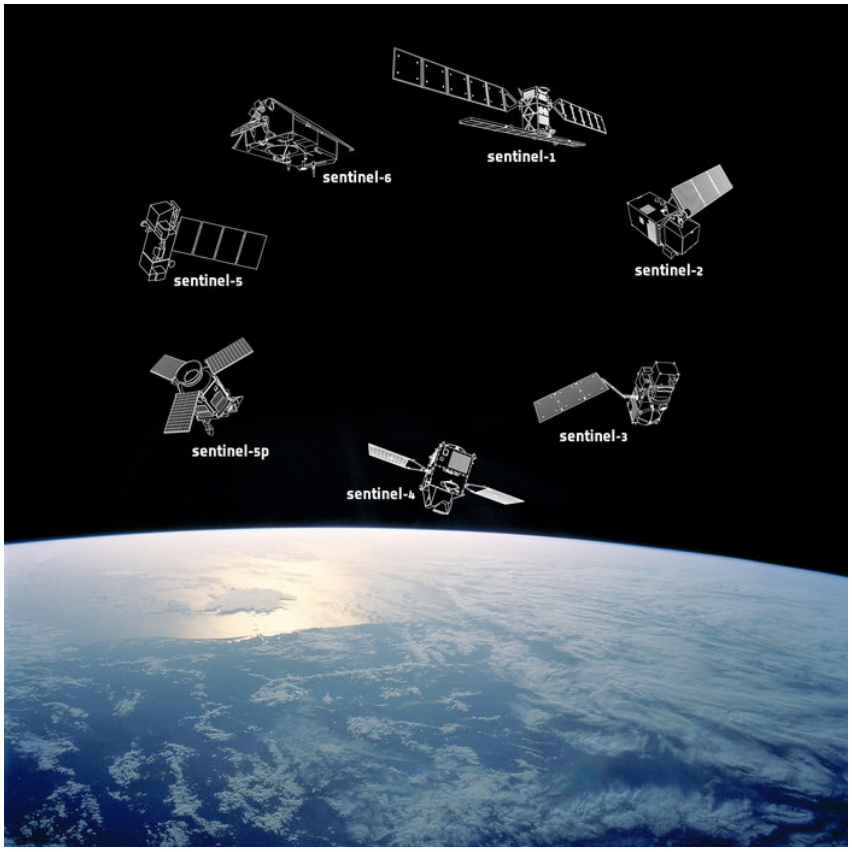


Figure 1.2: The six Sentinel missions provide a variety of different images and data through Europe's environmental monitoring Copernicus program.
Source: http://www.esa.int

land cover maps are based on images that have been captured in several spectral wavelengths. These contain rich information beyond the visible light that our eyes perceive. For example, vegetation reflects strongly in the infrared, unlike artificial surfaces, shadows, and water. Figure 1.3 shows SPOT7 imagery over an area in Brittany to illustrate the strong contrasts between the different land cover types that can be seen in multi-spectral images. By mapping the Near-Infrared, Red, and Green bands on the RGB channels of the image, vegetation appears in shades of red. Various materials also exhibit unique spectral patterns, which can provide extremely valuable information for characterizing them. Sensors like Landsat, SPOT, and Sentinel-2 capture multi-spectral images, both in the visible and infrared (450-2200nm). Hyper-spectral sensors like AVIRIS and Hymap capture images with a higher spectral resolution, and sample several hundreds of wavelengths [Teke et al., 2013].

- The *revisit time* or *temporal resolution* defines the rate at which the images are acquired through time. Imaging satellites orbit the Earth many times per day, and by precisely adjusting their trajectory, Earth Observation constellations like Sentinel-2A&B are able to capture an image of any given place every 5 days. This allows us to follow the temporal evolution of the land over a yearly cycle. We can see crops being planted and harvested, trees losing their leaves in the autumn, and snow falling on the mountainous areas. Such seasonal behavior is characteristic of certain land cover classes, and allows us to identify them by using observations throughout the year. An example of Sentinel-2 images over an area near Toulouse is provided in Figure 1.4, which illustrates the seasonal crop cycle of Winter and Summer crops.

- The *covered area* describes the location and footprint of the surface covered by the image or time series. Satellite optical images used for the production of land cover maps are often mosaics of images acquired at different dates, that have been blended together to create a coherent image covering a large area. For instance, the sensor aboard the Sentinel-2 satellites has as swath width of 290km (width of the area covered at each passage of the satellite), yet it produces time series covering the emerged surfaces of all latitudes

(a) Visible bands: Red Green and Blue

(b) False color: Near-Infrared, Red and Green mapped onto the RGB channels.

Figure 1.3: SPOT7 imagery at a 1.5m spatial resolution, over an area near the city of Brest. In the false color image, the Red band is replaced by the Near-Infrared band, the Green band by the Red band and the Blue band by the Green band. Infra-red is strongly reflected by vegetation, causing it to appear in shades of red.

between 56°S and 84°N, with a 10 day revisit time, or a 5 day time with both Sentinel-2A and 2B. Various resampling techniques are used to create homogeneous images or times series. This process, known as *temporal resampling* is described in detail in Part II, Section 3.2. The differently colored area near the top left corner of figure 1.4c is linked to the stitching of images acquired at different dates.

The requirements on these four characteristics depend on the target land cover classes. For example, a 10-100m spatial resolution is considered sufficient for most agricultural and natural classes. However, urban classes certainly benefit from higher spatial resolution, as the smallest objects that make them up range from 1-10m (narrow streets, canals, individual trees). On the other hand, agricultural classes require a frequent revisit time, as the different stages of the phenological cycle are all necessary to discriminate different crop types, as is shown in Figure 1.4.

Optical images are the main focus of the research work led in this Ph.D, as they are well known to provide valuable information for land cover mapping [Inglada et al., 2017, Gómez et al., 2016]. Moreover, the high quality, open availability, and mission lifetime of Sentinel-2 encourages the use of optical information. However, it is important to mention that none of the methods designed and proposed during this Ph.D. are in any way limited to optical imagery.

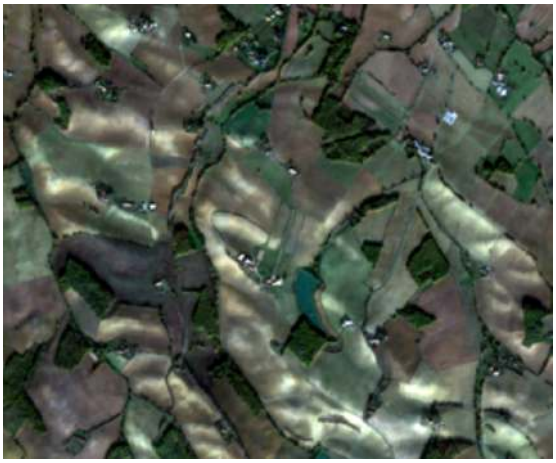
1.3 A classification task

The task of assigning categorical labels to each pixel in an image is called *classification*, *image classification* or *dense classification* in the Remote Sensing community. Interestingly, in the Computer Vision community, this same task is called *semantic segmentation*, as it consists in splitting (*segmenting*) the image into areas with *semantic* labels. The term *classification* or *image classification* is used when one label is assigned to an entire image to describe its content, for example saying "this is a picture of a house". Dense classification is quite different, as it involves determining precisely which pixels belong to the house and which belong to the sky, road, and background. To avoid confusion, the Remote Sensing terminology is used in the rest of the manuscript.

Creating a land cover map based on satellite imagery involves assigning a class label to each pixel in the image. This is done according to the image *features*, which depend on the type of imagery used. For example, they can be the multi-spectral channels of an optical image, the backscatter coefficients of a Synthetic Aperture Radar (SAR) image, a time series describing the temporal behavior of the pixel, altitude or height information coming from topographical measurements, or a combination of several of these. If multiple heterogeneous sources of information are used, the term *multi-modal* or *multi-source* data is employed.

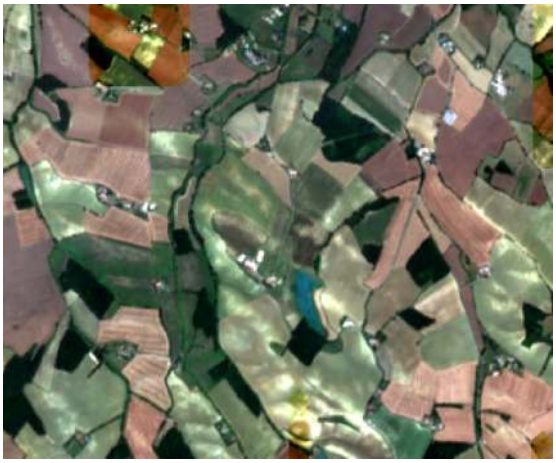
1.3.1 Photo-interpretation

Many land cover maps are made by human photo-interpretation, in other words, by an expert visualizing the images, and distributing the different classes across the area by hand. For instance, the Corine Land Cover (CLC) map [Bossard et al., 2000], illustrated in Figure 1.5 is produced in this way every 6 years. This map covers the entire European continent, and contains a very detailed nomenclature, particularly in the vegetation classes.



(a) Sentinel-2 image of January 2016. In the beginning of the year, crops and natural grasslands can easily be confused, as most agricultural land contains small amounts of vegetation during this period.

(b) Sentinel-2 image of May 2016. In the spring, winter crops reach the height of their growth phase, the land meant for summer crops is ploughed to bare soil.



(c) Sentinel-2 Image taken in August of 2016. Between May and August, the winter crops are harvested and prepared for the next cultural year, while summer crops are starting to grow.

(d) Sentinel-2 image of November 2016. By autumn, the summer crops have been harvested as well, and the only visible vegetation are forests, which usually contain a mix of broad-leaved and coniferous species.

Figure 1.4: Illustration of four dates of a time series of Sentinel-2 images over a small area, demonstrating the evolution of agricultural land throughout the year. Areas that remain green throughout the entire year are likely to be moorlands, grasslands, or coniferous forests.

Photo-interpreters make use of satellite images as a base for mapping land cover, but they also incorporate a degree of knowledge from external sources, in other words, information that is not present in the pixels themselves. Factors like the yearly climate, the geographical area, prior knowledge of the classes and their geometric layouts, and many more guide the expert decision process.

Photo-interpretation of a satellite image on a wide area is a slow and costly process, as it involves assigning a unique class label to a very large number of image elements. If the pixels of a high spatial resolution (10m) image were to be labeled one by one, mapping an area the size of France would mean labeling over 5 billion pixels. This illustrates the scale of the problem at hand, and is part of the reason why the CLC map is made with a Minimal Mapping Unit (MMU) of 25ha (500m×500m). While this MMU is sufficient for many land cover classes, in particular agricultural ones, other classes are impossible to describe at such a rough scale. Thin or narrow elements like streets, streams, canals and hedges, are absent from maps with this MMU, because they split larger objects in small areas (inferior to the MMU), and occupy a limited area themselves. The same can be said about isolated elements like lone houses in rural landscapes, which are absent from the CLC map, shown in Figure 1.5.

The fine spatial resolution of recent satellite images (1-20m) is sufficient to see many of the details of a landscape, which encourages their use for obtaining fine-grained maps. For this reason, the Remote Sensing community has long since been using automatic production methods, as they are adapted to deal with large amounts of data,

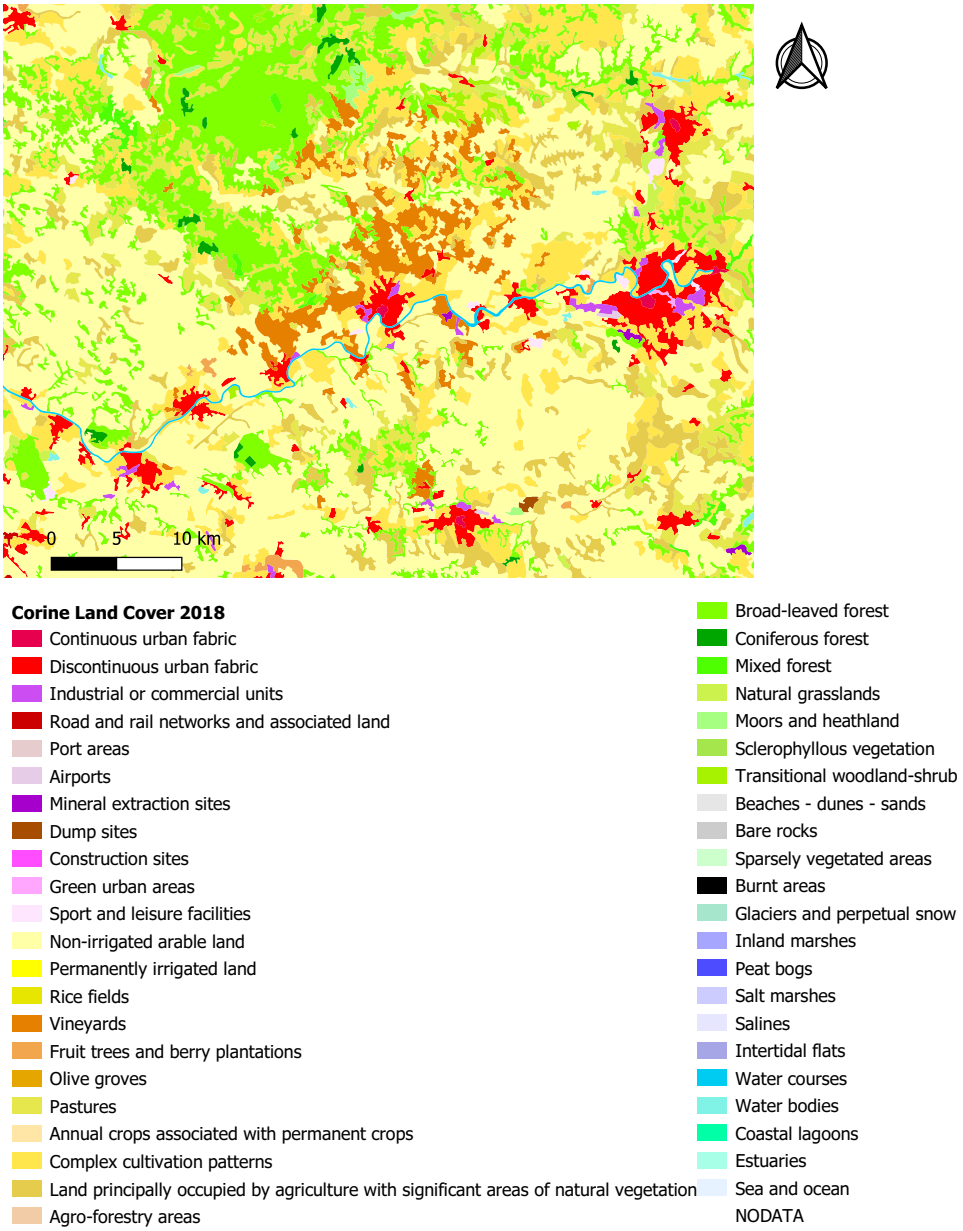


Figure 1.5: Illustration of the latest manually constructed Corine Land Cover map (2018) over the Gaillac area, and its nomenclature. The small city is surrounded by the famous Gaillac vineyards.
Source: https://land.copernicus.eu/pan-european/corine-land-cover/clc2018

given a sufficient amount of computational power. Many of these methods actually originate from studies on Computer Vision, which were developed for similar issues, but on smaller images. While current computer based methods are not yet as accurate as expert photo-interpreters, they have several advantages. Producing maps automatically brings the total production cost down significantly. It not only allows for a rapid distribution, which is key for many applications, it also generates spatially coherent and reproducible maps. This has encouraged Remote Sensing researchers to study artificial intelligence, machine learning, and supervised classification, to provide land cover maps at a lower cost, and within a shorter time. Given the right ingredients, these methods can surpass experts in terms of class recognition, thanks to their ability to "see" aspects of the problem that photo-interpreters cannot, such as the combination of several spectral or temporal features.

1.3.2 A set of decision rules

One way to classify the pixels of an image is to manually construct a set of decision rules, in an approach known as an *ontology-based classification* [Comber et al., 2005]. This translates preconceived knowledge of the different

classes into conditional statements regarding the features of a pixel and the class it should be attributed to. For instance, the following statement, "If a pixel is dark all year long, it must be water", can be converted into a rule separating the two *water* and *non-water* classes, based on a threshold on the feature values (dark or not) across the time series. Then, the non-water class can be separated again, by a rule like "If the pixel contains vegetation in the summer, and looks like bare soil in the winter, it must be a summer crop", and so on. This forms what is known as a *decision tree*.

By basing the decision tree on a hierarchical system of classes, called the *ontology*, our natural interpretation of the main land cover elements can be made into an automatic procedure, which can be applied to any pixel. Then, the simple programming of such a tree allows a machine to rapidly process very large amounts of data. A few examples of applications of such methods in land cover mapping include works from [Comber et al., 2004] and [Belgiu et al., 2013]. Figure 1.6 shows a possible tree-like structure describing the different hierarchical relations between land cover classes, which is based on the Corine Land Cover nomenclature. As this figure shows, providing an extensive description of the numerous land cover elements requires a significant amount of branches. To convert such a hierarchical description into a decision tree, each branch needs to be associated to one decision rule, with manually defined thresholds based on the image features.

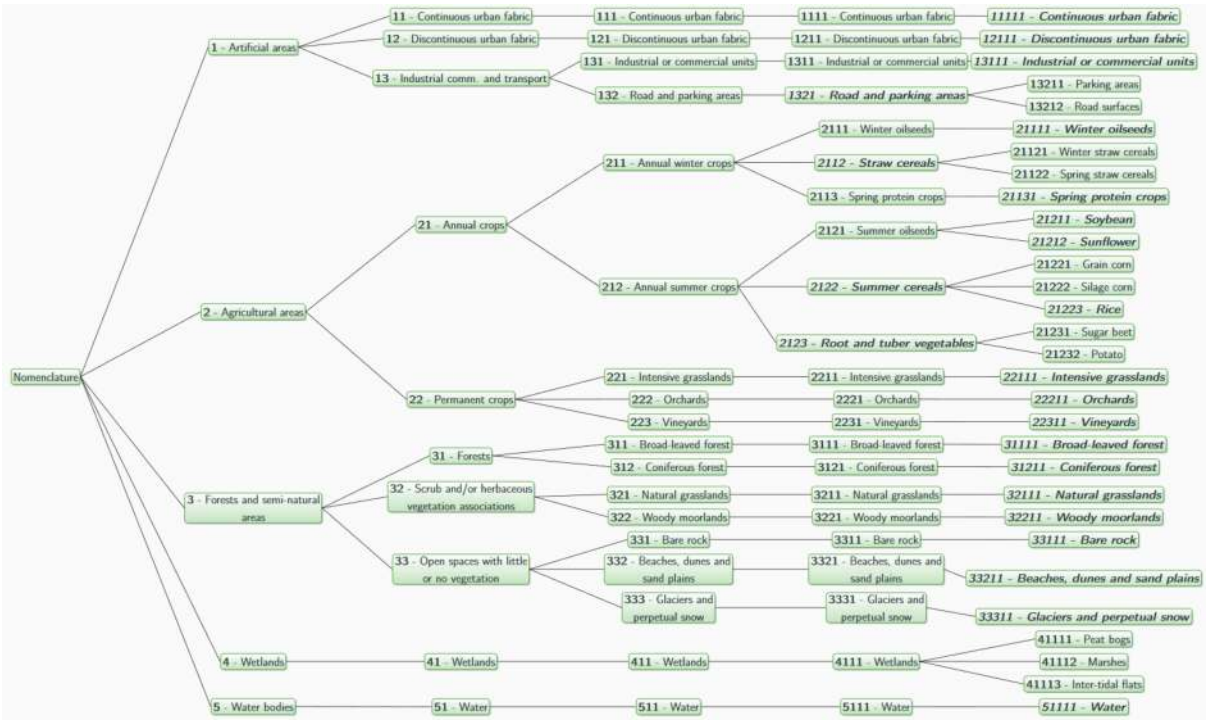


Figure 1.6: Hierarchical relations between different land cover elements, based on the Corine Land Cover nomenclature.

While this may be a good first approach for a simple two or three class classification problem, the complexity of land cover mapping with rich nomenclatures makes such decision trees very difficult to design at all. Indeed, time series contain a large amount of spectral, spatial, and temporal variability, which makes decision rules based on thresholds on the features difficult to determine. First of all, according to the geographical region, the behavior of land cover classes can be quite different. For vegetation classes, this is linked to the ecology and the climate of the area, which varies greatly across a large territory. In the case of urban classes, the aspect of different cities is rarely the same, as it usually depends on the availability of local construction materials. Second of all, the weather differs every year, which causes variability in the agricultural and vegetation classes, and would be difficult to take into account in such decision rules. For these reasons, a unique decision tree would need to be constructed for each region across the country, and once again every year.

Another reason that makes the ontology-based classification approach unsuited for such a problem is the requirement of large number of manually set thresholds. For simple problems, the precise value of the thresholds has a relatively low importance, but for such complex problems, tuning the value of the thresholds can undoubtedly allow for a more precise classification. Manually finding the ideal thresholds is nearly impossible, yet even untrained people are able to recognize many of the most challenging target classes. This raises an interesting question: *How is it that we can easily perform certain tasks while having no idea of how we are actually doing them ?*

In many cases, understanding how we make decisions is more difficult than actually making the decisions themselves. We can easily perform classification tasks that we are not truly able to describe in words, based on what might be called "experience" by some or "instinct" by others. For instance, distinguishing between different smells, or between the voices of people we know. Naturally, these all have physical origins which differentiate them, but our ability to classify them is instantaneous and relatively reliable, while our knowledge of the physics behind these phenomena may be limited or nonexistent.

We don't exactly understand (or need to understand) the decision process involved with each classification task we perform in our every day lives. Nonetheless, it is possible to use our knowledge of the way we learn how to perform these tasks efficiently, to inspire the design of an automatic classification system. Most of what we learn comes from our past experiences, which can be seen as a series of examples from which to optimize our decision process. By smelling hundreds of different roses and lavender flowers, or by listening to other peoples voices for enough time, one can learn to tell them apart, as each one produces a unique sensation registered somewhere in the brain. In the same way, machines can be taught to recognize different things, by learning from not hundreds but millions of different examples, in a process known as *machine learning*.

1.3.3 Supervised classification with machine learning

One very efficient way of generating land cover maps is to use a *supervised classification* method. This involves teaching an algorithm to automatically classify the various elements present in a data set. The basic idea is to automatically devise a decision process based on a set of already labeled examples, using a *learning algorithm*. These algorithms are also commonly known as *classifiers*. Their lifetime is divided in three main stages: training, testing, and prediction, which are represented in Figure 1.7.

During the training stage, the machine learns to recognize the classes on its own by observing data from a certain number of examples: the so-called *training data set*. These labeled samples are the basis from which the algorithm can categorize previously unseen data points. By analyzing the common points and differences between the features of tens of thousands of examples, the algorithm is able to establish a model, which aims to assign class labels to points described by these same features. In other words, training involves dividing the feature space in as many regions as there are classes, in order to later make a decision regarding the class label of an unknown pixel. The training data set is a fundamental aspect of the classification process as it contains observed instances of the natural phenomena that link the target classes, which are basically categorical concepts, to the features, which are specific measurements with real values.

The underlying objective of a supervised classification method is to *generalize*, in other words, to accurately predict the class of samples that are not present in the training data set. In a way, training a classifier involves transferring the information that is present in the labels of the training data set into a decision process, often called the *model*, which can assign labels to unknown samples. This is why the process is often named *learning*, as it involves first accumulating a large amount of individual observations, and transforming these observations into a decision process.

The term *supervised* in *supervised classification* comes from the fact that labeled training data points are used to define the target classes, and to train the model. Inversely, *unsupervised classification*, also known as *clustering* attempts to create groups of similar points in the data set, called *clusters*, without using prior knowledge of their class labels. Clustering can be useful for many image analysis purposes, for example, for identifying *outliers* (points which are very dissimilar from the others), or for compressing images. These are often used as a data analysis tool when no labeled data is available. Unsupervised classification alone is not sufficient for producing a land cover map as it only provides a *cluster label* for each pixel, which does not tell us its *class label*, in other words, if its a forest, a road, a field, etc. On the other hand, if an insufficient amount of training data is available, it is possible to perform clustering on the data and then label the clusters in a successive step.

Naturally, the quantity and quality of training data given to the algorithm is a very important factor for its success. The training data should provide a realistic representation of the various classes. It should be as comprehensive as possible, in order to account for variability within the data set. In land cover mapping, training data from several hundreds of different areas across the territory is used to account for geographical variability. Moreover, training labels should be as up-to-date as possible, in order to avoid falsely labeled points due to land cover changes.

The second stage involves evaluating how well the classifier has learned, and is called the *testing* stage. In practice, rather than training the classifier with all of the available labeled samples, a part of them are set aside in order to evaluate its performance. These form the so-called *test* data set, and provide valuable insights on how well the classifier is able to recognize different classes.

In multi-class classification problems, there are several ways of evaluating the performance of a classifier. The most commonly used tool is the confusion matrix. Each element c_{ij} of this matrix contains the number of

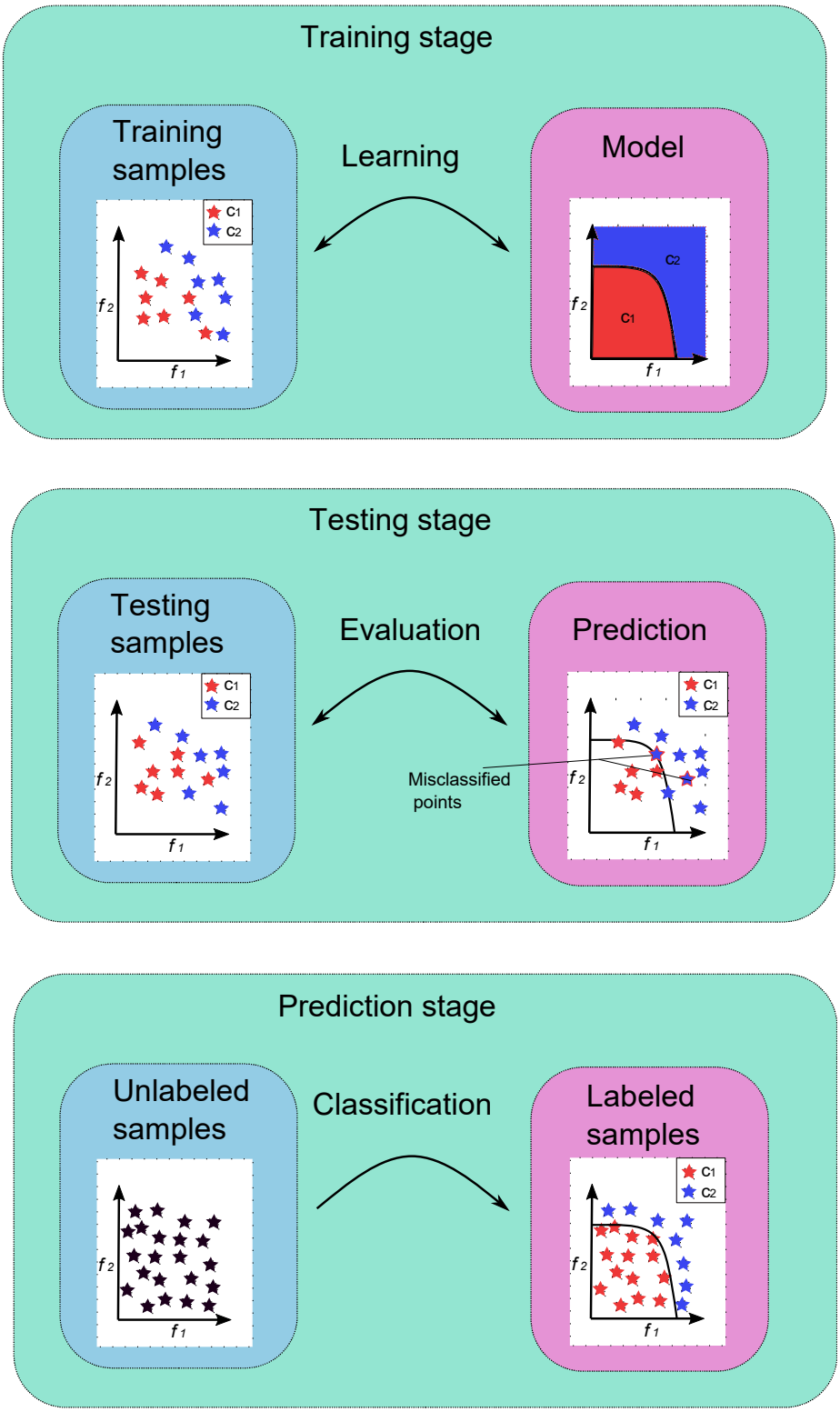


Figure 1.7: The three life stages of a classifier: training, testing, and prediction, demonstrated on a simple two-class problem. During the training phase, the labeled samples are used to divide the feature space (f_1, f_2) into two regions, one for each class (c_1 and c_2). The testing phase involves using a different set of labeled samples, to check if any misclassifications occur, and to calculate accuracy scores. Finally, the prediction phase consists in labeling a large number of samples, which can be done automatically, in a short amount of time, relative to manual labeling.

elements attributed to the class j by the classifier, and that are truly the class i. The diagonal elements show the

number of correct classifications, and the off-diagonals represent a confusion between two classes. In classification problems with a large number of classes, the confusion matrix becomes difficult to analyze efficiently, so a number of average performance scores are derived from the matrix. The most commonly used scores are the Overall Accuracy, which calculates the average accuracy over all classes, and the class F-scores which provide an indication of the recognition rate of each class. The precise definition of these scores is given in Part II, Section 6, page 87.

In practice, classifiers are trained a great number of times with different parameters, and in some cases, several different classifiers are trained on the same data set, and participate in a voting system, which often increases the likelihood of a correct decision. This is known as an *ensemble* classification system [Rokach, 2009].

It is a commonly accepted notion that the training and test sets should be as independent as possible, in order to provide an unbiased evaluation of the performance of the classifier. Unfortunately, obtaining labeled data is one of the major difficulties in using a supervised classification approach, and the same data sources are often used for training and testing. This implies that biases in data collection can be reflected in the performance scores.

However, if the validation data set is representative of reality, in the sense that it encompasses the vast majority of cases that the classifier is likely to encounter during its prediction phase, the validation scores do provide accurate estimates of the final performance of the classifier. In other words, the *completeness* of the test data set is more important than how correlated it may be to the training data set. The implications of this point regarding the validation of land cover maps is discussed further in Part II, Section 3.1.2.

The third and final stage of a classifier's lifetime is the *prediction* phase, if the testing has shown sufficient performance scores. This involves classifying a very large number of unlabeled samples, and is the moment the classifier truly becomes useful, as it is able to perform such operations at great speeds. The following section describes a land cover map of France known as the OSO map, which has been produced every year since 2016, and the issues that it faces in classifying certain land cover types.

1.4 Definition of the problem and main research objectives

1.4.1 The OSO map

Many land cover applications use time series of multi-spectral satellite images covering a period of approximately one year, as this covers a phenological cycle for many of the land cover classes. The Occupation des SOls (OSO) [Inglada et al., 2017] land cover map describes the content of France with a yearly update. The most recent maps (2017 and 2018) have been made by using time series of Sentinel-2 images, at a 10m target spatial resolution (MMU of 0.1ha). Figure 1.8 provides a view of the evolution of the OSO products from 2014 to 2019. The colors corresponding to the different classes can be found in Figure 1.9, along with a close-up of the 2017 map. Note the fine grained detail brought by a 10m target spatial resolution in figure 1.9b. The production of this map is used as a baseline case for many of the experiments in Part IV.

The OSO Land Cover map uses a combination of the previously mentioned Corine Land Cover (CLC) [Bossard et al., 2000], for the vegetation and urban classes, as well as the Land Parcel Information System (Registre Parcellaire Graphique or RPG) [Cantelaube and Carles, 2014], which describes the main crop classes [Inglada et al., 2017], and the Randolph Glacier Inventory (RGI) [Pfeffer et al., 2014]. The detailed description of the OSO classes and their data sources is given in Chapter 3, Section 3.1.1.

Practically speaking, these data bases come under the form of labeled polygons, which each represent a geographic entity, like a field, a road, or a neighborhood. Figure 1.9a provides an illustration of the training data used by the OSO map, over an area of 10km $\times$ 10km, along with the OSO map of the corresponding area.

The map of 2018 was produced with an extended nomenclature containing 23 classes. The most recent extensions provide a more detailed description of the annual agricultural classes. Annual Summer Crops (ASC) are divided into 5 classes that describe the plant species: Soybean, Sunflower, Corn, Rice, and Root/tuber. The Annual Winter Crops (AWC) is split into 3 classes: Rapeseed, Straw cereals, and Protein crops. This extension is also planned for the map of 2019.

The OSO Land Cover Map uses a supervised classification method known as *Random Forest* [Breiman, 2001], which is described in detail in Section 3.4, in order to assign a class label to each pixel. For this purpose, time series of Sentinel-2 images are used to provide a multi-spectral description of each pixel, at various dates throughout the year. This allows the major natural, agricultural, and artificial classes to be distinguished. The detailed OSO class nomenclature is given in Table 1.1, along with the source used for the production of the 2016 OSO land cover map.

It can be noted that the CLC classes, which are only produced every 6 years, are used every year as labeled data for the production. These classes are considered to be relatively *perennial*, in the sense that major updates take place on a scale of several years, and are not subject to yearly changes or rotations like crops. That being said, this causes several mislabeled samples in the data base, the impacts of which are studied in detail in [Pelletier et al., 2017, Tardy et al., 2017]. In order to produce an updated land cover map, every single pixel of the image is therefore classified,

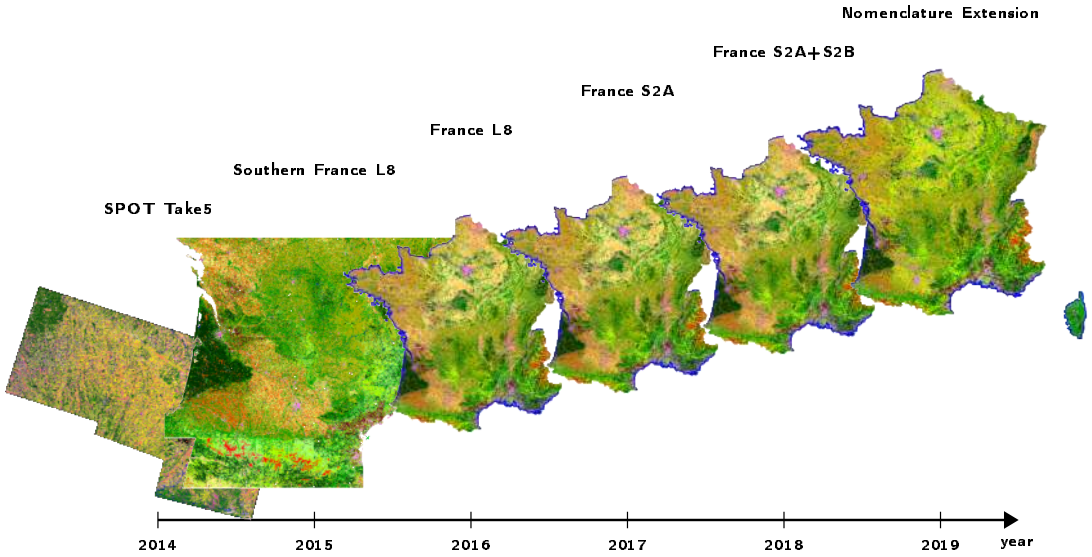


Figure 1.8: Occupation des SOIs (OSO) land cover maps have been produced and distributed since the year 2017. In 2014, the so-called Take-5 initiative, in which the characteristics of Sentinel-2 time series images were simulated by SPOT-4 over a small region near Toulouse [Hagolle et al., 2015] was used for product prototypes. In 2015 and 2016, more prototypes were made using Landsat-8 (L8) images, before the arrival of the Sentinel-2 (S2) time series in 2016, which have been used ever since.

including the ones which are already labeled in the reference data set. If the classifier has learned sufficiently well, it is even able to correct a certain number of the mislabeled samples. By analyzing disagreements between the map and the reference data, it is possible to detect and locate certain land cover changes.

Figure 1.10 shows the confusion matrix of the OSO map that was produced in 2016. The OSO method allows for high recognition rates of the major Annual Crops (AC), and Intensive Grasslands (IGL). Indeed, the multi-temporal information allows these classes to be relatively easily distinguished, due to the differences in the periods of seeding, growth, and harvest. Moreover, Broad-Leaved Forests, Coniferous Forests, Bare Rocks, Water and Glaciers are recognized quite well.

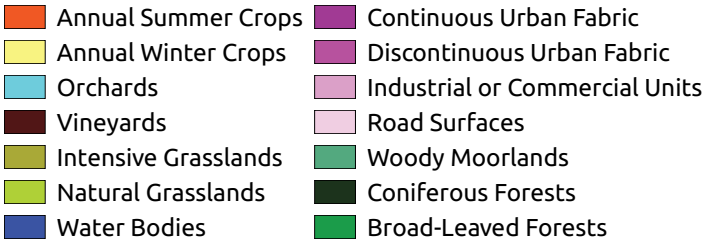
However, there are high rates of confusion between the different artificial classes (CUF, DUF, ICU and RSF). Other poorly recognized classes include the Woody Moorlands, Natural Grasslands, and Orchards. Many of these confusions can be linked to out-of-date or ill-defined training data, with large polygons containing a mix of different classes. However, this should not be the case for the four urban classes. As was mentioned earlier, cities evolve relatively slowly over time, and a sufficient number of labeled samples is available for training in the CLC, which was used as a data source for the 2016 OSO map.

In fact, the Urban Atlas (UA) database [Montero et al., 2014] provides a geometrically accurate description of the various artificial cover types, and in some cases, their density, on all cities with over 100,000 inhabitants. Unlike agricultural classes, urban classes are mostly perennial, and can be used on images from different dates with a limited amount of errors, as the construction or destruction of urban cover usually happens over several years. This implies that from one year to the next, the majority of built-up classes do not change. However, this mapping is only made on major cities, and is not updated every year. The impact of the integration of UA classes in the OSO production scheme is discussed in Part II, Chapter 1.3, but changing the data base does unfortunately not entirely solve the issue of confusions between urban classes in the land cover maps.

In a similar way, forests and shrublands are often confused with one another, as the difference between the two classes lies more in the density of tree cover than on the aspect of each tree, especially seen from above. Table 1.2 shows a few examples of classes that describe the density of tree cover.

The amount or quality of training data for these classes is not the reason behind the poor recognition of density-based classes.

A logical explanation for these confusions is the lack of contextual description. Practically speaking, a 10m × 10m area is not sufficient to characterize some of the target classes in the nomenclature. For example, the density of buildings cannot be described by pixel information alone, because each building is in fact made up of several pixels. The concept of density is linked to a wider area than just one pixel.



(a) Training data used by the supervised classifier of the OSO LC map. Labeled pixels are extracted from the polygons for training. The background shows the RGB bands of one date of the time series. A land cover map is created by classifying every pixel of the image.

(b) Result of the dense classification. Using high spatial resolution imagery brings a fine level of detail, for example, small roads between well outlined fields appear in the map. On the other hand, this comes with a certain level of noisy, isolated classifications, particularly in textured urban areas.

Figure 1.9: Principle of supervised classification in land cover mapping, illustrated with the OSO map. The objective is to determine the classes of the all of the pixels in the image, by using information contained in the labeled polygons.

Classes describing a notion of density (urban density, tree cover density), have a strong added value for analyzing the image, as they reflect statistical behavior of areas in the image. These are more challenging to recognize precisely, even by an expert photo-interpreter, especially if a very fine graduation of density is desired. In general, context dependent classes describe more than what is visible in each pixel of the image, which is what makes them such important land cover classes.

For this reason, features that describe the surroundings of the pixel can be useful to improve the recognition rates of density-based classes. These features are called *contextual features*, as opposed to *pixel features*, as they describe the context of a pixel. The quality of the OSO LC maps could be improved by the use of contextual information, especially as its nomenclature contains several classes describing density levels. The OSO LC mapping problem provides an interesting study case to further research in this topic. In the future, a possible application of this research could be extending the fine density graduations of the Urban Atlas classes to the smaller cities and generating yearly maps at a finer spatial resolution, by using supervised classification methods.

1.4.2 The importance of context in high-resolution image classification

When it comes to land cover mapping, creating spatially finer map products is a natural objective, and it is mainly conditioned by the spatial resolution of the images that are used.

The Sentinel-2 constellation was designed to acquire multi-spectral optical images of the continental surfaces of the Earth at a 10m spatial resolution. This level of detail allows for many elements of the landscape to be captured, such as cities and roads, as well as fields and forests.

However, small streets and individual cars require an even finer spatial resolution, which can be achieved using a pixel size of the order of 1.50m. For example, mono-date images such as the ones captured by the SPOT-7 satellite can be used.

Generally speaking, the first advantage of having a fine spatial resolution is that it allows for a more precise

Table 1.1: OSO Classes and their sources: Corine Land Cover (CLC) [Bossard et al., 2000], the Land Parcel Information Registry (Registre Parcellaire Graphique, or RPG) [Cantelaube and Carles, 2014], the National Topo Data Base (BD Topo) [Maugeais et al., 2011], and Randolph Glacier Inventory (RGI) [Pfeffer et al., 2014].

CODE	Class	Short description	Source
CUF	Continuous Urban Fabric	Buildings, roads and artificially surfaced areas cover more than 80% of the total surface.	CLC 111
DUF	Discontinuous Urban Fabric	Buildings, roads and artificially surfaced areas mixed with vegetated areas and bare soil.	CLC 112
ICU	Industrial or Commercial Units	Artificially surfaced areas with concrete, asphalt.	CLC 121
RSF	Road Surfaces	Motorway rest areas, parking areas, motorway networks, larger than 50 m.	BD Topo
ASC	Annual Summer Crops	Annual crops grown from March/June to August/September. Mainly corn and sunflower.	RPG
AWC	Annual Winter Crops	Annual crops grown from November/February to June/July. Mainly wheat, barley and rapeseed.	RPG
IGL	Intensive Grasslands	Dense grass cover, of floral composition, not under a rotation system.	RPG
ORC	Orchards	Parcels planted with fruit trees or shrubs.	RPG
VIN	Vineyards	Areas planted with vines.	RPG
BLF	Broad Leaved Forests	Forest of broad leaved trees.	BD Topo
COF	Coniferous Forests	Forest of coniferous trees.	BD Topo
NGL	Natural Grasslands	Low productivity grassland. Includes rocky areas, briars and heathland.	CLC 321
WOM	Woody moorlands	Spontaneous vegetation dominated by woody and semi-woody plants.	BD Topo
BDS	Beaches, dunes and sand plains	Beaches, dunes and expanses of sand or pebbles.	CLC 331
BRO	Bare rock	Scree, cliffs, and rock outcrops.	CLC 332
GPS	Glaciers and perpetual snow	Land covered by glaciers or permanent snowfields.	RGI
WAT	Water bodies	All water bodies longer than 20 m and all water courses larger than 7.5 m.	CLC 523 and BD Topo

Table 1.2: Different levels of tree density, following a classification similar to the Daubenmire cover classes [Daubenmire, 1959]. Possible land cover classes with an equivalent tree density are suggested here. These classes are challenging to classify with pixel information alone, as they depend on the spatial arrangement of the features in a wider vicinity.

Density of canopy cover	Possible land cover class
0-5%	Grassland, bare rock
5-25%	Shrublands, heathlands, moorlands
25-50%	Barren lands / fallow
50-75%	Open forest
75-100%	Closed forest

description of the outline of the objects in the image. This translates as an accurate localization of their borders, and of other salient geometric elements such as corners.

The second advantage is that it allows for smaller objects to be visible in the image, which increases the amount of potential target classes.

However, an increase in spatial resolution also comes with two challenges. First of all, the obvious increase in computational burden linked to the fact that a given area requires far more pixels in order to be covered at a higher spatial resolution. Second of all, and more importantly, if the size of the pixels is far smaller than the size of objects described by the target classes, the recognition rate of certain of these classes can be poor, if they are classified based on pixel information alone. This is particularly the case for classes presenting a form of texture, or classes based on a notion of density. Such classes are referred to as *context-dependent* classes.

The context of a pixel is defined as a connected region containing it, and formed by group of adjacent pixels, often called the *neighborhood*, *object* or *spatial support*, which is the preferred term here.

In order to accurately recognize context-dependent classes, information from beyond the pixel must be taken

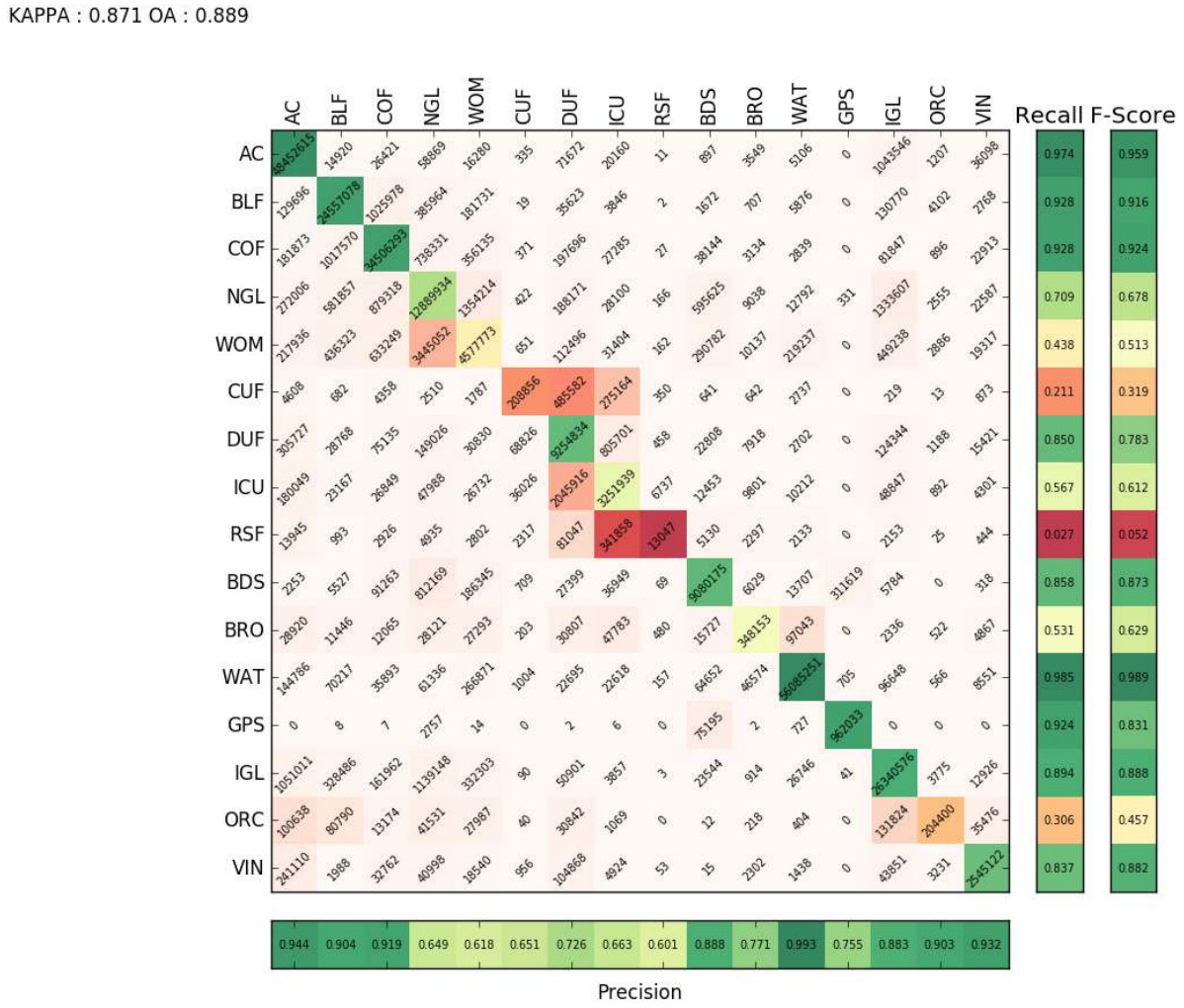


Figure 1.10: Overall Accuracy, Kappa, Confusion matrix, Precision, Recall, and F-scores of the 2016 OSO Land Cover Map [Inglada et al., 2017], which was produced at a national scale using Sentinel-2 imagery. The four urban classes, CUF, DUF, ICU and RSF are confused with one another, which reflects the inability of pixel features alone to fully describe them.

into account.

Including context in a supervised classification scheme raises several questions regarding the nature of the objects in the image, and the role that each pixel plays in forming an object.

To understand the underlying principle that causes confusion within context-dependent classes, the problem is illustrated on a first simple example, shown in the schematic in Figure 1.12. In this example, water, two types of crops, and orchards are used to illustrate how an undesired class, bare soil, can create confusion in a pixel-based classification.

On the bottom are the base components that are visible in a high spatial resolution image, with as much temporal and spectral information as is available to each pixel. This labeling is suggested here for the purpose of illustrating the phenomenon at hand. The upper layer contains the target, high-level classes. Physically speaking, these labels describe the training polygons, which are larger than the individual pixels, and therefore can contain several contain elements of the lower layer. This is indicated by a line joining the two class labels. The color and width of the lines represents the proportion of the lower level classes in each higher level class, with thick lines indicating a majority, and thin lines a minority.

The ideal case is shown first, where one higher-level class is directly recognizable at the target spatial resolution. Water is simple to identify using the multi-spectral and multi-temporal information, and is a very spatially homogeneous class.

However, other high-level classes are not so simple to recognize. This can be due to the presence of base components such as bare soil, which is present in both orchards and croplands.

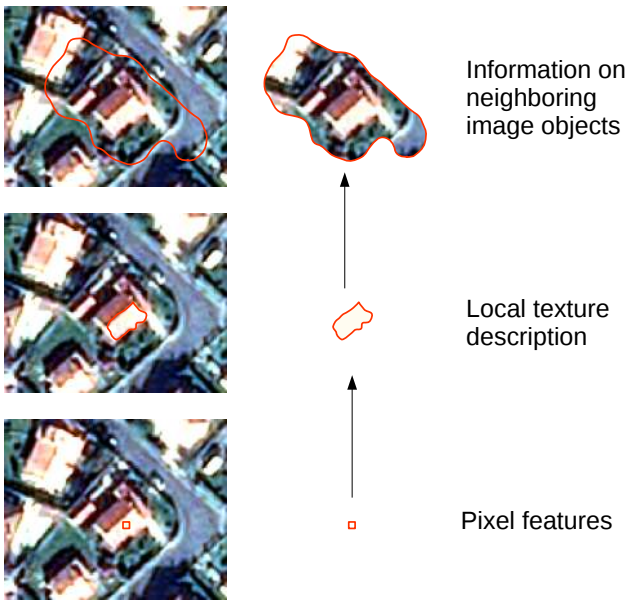


Figure 1.11: Illustration of several scales of context that describe the neighborhood of a pixel belonging to a residential building in a discontinuous urban area. The pixel information is not sufficient to describe the density of urban cover in a wider area. At a larger scale, the texture of the roof is captured. At the largest scale, the neighboring houses and surrounding vegetation can be seen.

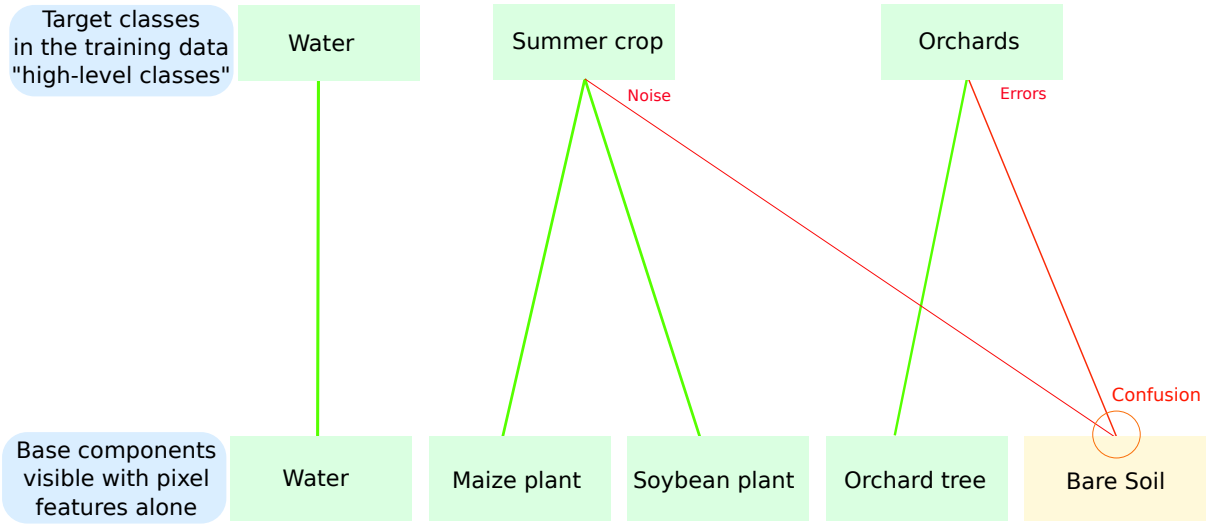


Figure 1.12: Hierarchical relations between base classes visible at a pixel level, and high-level classes in the nomenclature. The bottom layer represents a suggested labeling of what may be visible for each pixel at the given spatial resolution, with the totality of the pixel features. The labels in the upper layer represent the classes in the training polygons, which are often spatially larger than the pixels. The connections between the labels represent relations of inclusion, in other words, the fact that there exist pixels of the lower level classes in the training data of a higher level class. The thickness of these lines indicates the proportion of the base classes within the higher-level classes. When a lower level class is hierarchically included in more than one different higher-level class, this causes confusion within the class, which is represented by an orange label and a red-colored line. With the pixel information alone, any classifier would be subject to confusions for elements within these classes. This graph explains the source of the feature overlap which causes confusion within context-dependent classes.

Indeed, at a high spatial resolution, the bare soil class is clearly visible in the images. In fact, the areas of bare soil in between trees in an orchard should be classified as a part of the orchard, as they are present in the training

polygons of said class. Croplands sometimes also contain small quantities of bare soil, due to imperfections in irrigation, or other such phenonmena.

This lower-level class can cause confusions, as it is a part of both of the higher level classes. However, its impact on the summer crops is small, due to the minority proportion it represents, and it therefore manifests as noise.

This figure shows that confusions can occur when the lower level classes do not form a strictly inclusive hierarchical system with respect to the high-level classes. This is shown as an orange label on the lower level class. A class that has branches to several higher level members is bound to be a source of confusion in the classification, as pixels with the same features are likely to receive different class labels during training.

This graph is a way of representing feature overlap in high spatial resolution imagery. Shortly put, there are pixels that alone can be described as different of the labels in the nomenclature. In order to remove the confusions, each lower-level class should ideally be a included in only one of the target classes.

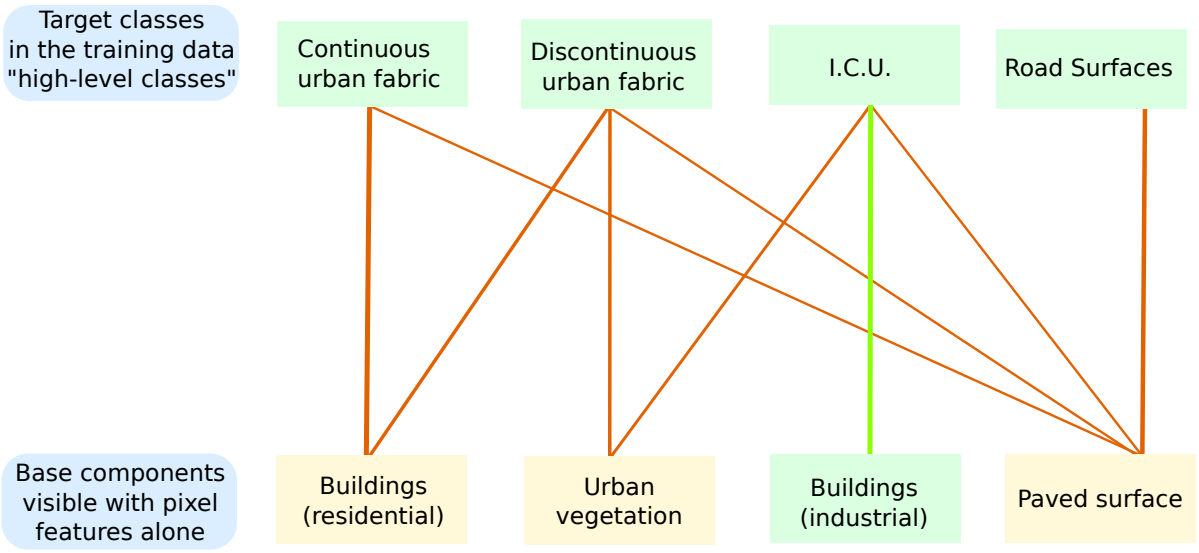


Figure 1.13: Hierarchical relations between base-level classes and the higher-level classes in the nomenclature in urban areas, following the same principle as Figure 1.12. In this illustration, three of the low-level classes are a part of more than one higher-level class: residential buildings, urban vegetation, and paved surfaces.

Looking at a similar representation as Figure 1.12 for the four urban classes, which is shown in Figure 1.13, it becomes clear that the situation is even more complex, as most of the lower-level classes are part of more than one higher-level class, which once again causes confusion.

Most cities are made up of four base elements: residential buildings, industrial buildings, roads, and to a varying degree, vegetation. Their spatial arrangement (texture), and local proportions (density) make up more complex classes such as continuous or discontinuous urban fabric. Because the pixels are smaller than the base elements of the classes, they cover an insufficient area to capture more than one of these base elements at a time. Individual pixels are unable to see the larger picture, which explains why supervised classifiers based on pixel information alone present can in some cases present high confusion rates when they attempt to distinguish higher-level classes.

These two figures explain the source of many of the confusions that occur in the land cover mapping process, namely the discrepancy between the high-level target classes, and the actual classes that would be theoretically visible using the pixel features alone.

A question for the classifier could be: for a pixel that ressembles urban vegetation, how to decide whether to label it DUF or ICU ? The pixel information alone is insufficient to perform this labeling accurately, which will result in a random attribution, causing classification errors.

In this case, the answer can come from looking at the pixels surrounding it. If it is part of a discontinuous urban area, it is likely to have pixels ressembling residential buildings and paved surfaces nearby. In the same way, if it is part of an ICU, there likely are industrial buildings and paved surfaces in the neighborhood.

The objective is to imagine a simple intermediate representation, which would be able to accurately classify labels with local contextual dependencies to other recognizable pixels in the vicinity. To illustrate this, consider the same case as the first example, Figure 1.12, with the bare soil, orchards, and croplands. In Figure 1.14, an an ideal intermediate layer of labels is added between the upper and lower levels.

This intermediate representation can afford to be more specific than the basic classes, as benefits from a sup-

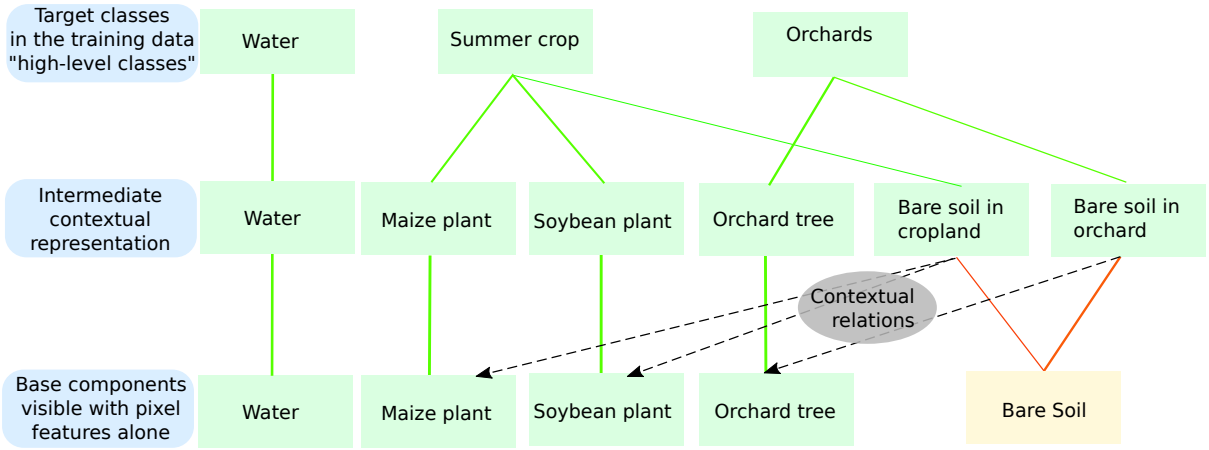


Figure 1.14: Hierarchical relations with an intermediate representation, that is located in between the pixel-based components and the high-level classes. This representation contains a more precise labeling that describes the surroundings of the pixels, and is therefore able to theoretically resolve the confusions in the classification. At a first order, the surroundings of a pixel are generally composed of elements of the same high-level class as itself. This is represented by a dotted black arrow, and is called a *contextual relation* here. This illustration only shows the relations of the Bare soil class with regards to the other base-level classes.

plementary source of information. This representation considers that the pixels in its context might bear certain recognizable aspects. This labeling is capable of dividing the lower level classes into classes that form a strictly inclusive hierarchy with the target high-level classes. This is done by using contextual relations, illustrated in dotted lines, which represent a link between a pixel in the intermediate representation, and the "true" elements in the surroundings, as defined by the higher level class.

To understand this figure, consider the case of a bare soil pixel p, that in reality belongs to an orchard. Based on the pixel features alone, p cannot be accurately labeled as "Cropland" or as its true class, "Orchards". Because p is in fact a member of the high-level class "Orchards", the training polygon that that contains p should also contain orchard trees. This allows it to be intermediately characterized as a "Bare soil in orchard", and later, simply as "Orchard". The link between the pixel and its immediate surroundings is represented in the graph by a black dotted arrow, going in this case towards the lower-level label "Orchard tree". This arrow represents how local contextual relations could exist between a bare soil pixel and the orchard trees around it.

It should be noticed that the black arrows are only drawn towards low level classes that are a part of the same high level class. A pixel does not necessarily have contextual relations with all of the lower level classes. The intermediate representation is able to place itself at a higher scale, based on information from neighboring pixels that are a part of the same higher-level class.

Drawing and defining precisely what these links are meant to signify is a task that should be left for the classifier to perform. Indeed, mapping out precisely which contextual relations exist between different classes of a particular nomenclature is not a simple task. Ideally, a supervised classifier should be able to learn these relations based on the data it is presented with, provided that contextual information is included in some way.

It is therefore not advisable to use the lower level class labels that are shown here, as they are meant for an illustration of a more general problem in image classification. In fact, it would be quite complex to draw accurate graphs like these for even more complex nomenclatures. This figure also suggests that the proportions of local neighboring lower-level classes could be a discriminating factor for classifying a pixel. However, this intermediate representation does not explain the totality of contextual relations between objects in an image, as relations between different neighboring objects can also exist. This idea is pursued further in Part III, Chapter 8.

Understanding the complexity of how pixels and classes are inter-linked, in general, can guide the design of the appropriate ways in which to help the classifier bridge the gap between the available features and the target classes. Indeed, thinking in terms of intermediate representations prepares the necessary ingredients to allow the classifier to incorporate contextual information in a practical and efficient manner.

This justifies the need for an evaluation of various ways to include context in a supervised classification scheme, keeping in mind the constraints of land cover mapping at high/very high spatial resolutions. Three main factors enter into play for the design of such a method:

1. The spatial, spectral, and temporal resolutions of the image, which make up the pixel features.

2. The reference data, which describes the target classes, and specifically the contextual dependencies that may

be present. Another interesting aspect is the relative density of available training points or polygons.

3. The computational cost of the method, as the data sets that are to be classified represent very large volumes.

1.4.3 Challenges of a large scale production

Recent High Resolution Remote Sensing images span over several thousands of square kilometers. ESA's Copernicus initiative, previously known as the Global Monitoring for Environment and Security (GMES) [Harris and Browning, 2003] is responsible for over 30 imaging satellites, the most recent of which are the Sentinel family (Figure 1.2). Their mission is to observe the Earth's surface in a variety of ways, and to openly distribute these images. In fact, these sensors are one of the biggest data producers in the world, yielding several terabytes of images every day. In fact, the Sentinels are not only the largest producers of satellite data, they also generate more data than internet giants such as Facebook and Youtube.

Exploiting such large images is a delicate task, and requires awareness of the scale of the problem at every step. This very important factor must be constantly taken into account in the design and analysis of the various methods, if they are to be implemented in an operational processing scheme.

Classifying wide stretches of the Earth poses both theoretical and practical problems for land cover mapping.

First of all, in theory, covering more land means dealing with an increased intra-class variability compared to small areas, as was mentioned earlier, different land cover classes can exhibit a very different behavior according to the geographical location. While this issue is not directly addressed in this Ph.D. work, it may be indirectly improved by working on a classification strategy that includes context, and therefore allows for more complex classes to be modeled. It is nonetheless worth mentioning that the issue of intra-class variability over wide areas is currently handled thanks to an eco-climatic stratification strategy, which was introduced by [Inglada et al., 2017], and is described in more detail in Section 3.5.1.

Second of all, in practice, temporal, spectral, and contextual information are all necessary in order to reach a precise classification with a complex nomenclature, like the one of the OSO Land Cover map. This implies that the images not only cover a wide extent (very large number of pixels), but that each pixel also contains several hundreds of features.

Dealing with data sets of such size imposes practical constraints on two fundamental aspects of processing: the required memory, and the computation time.

Memory can simply be defined as any physical element able to reliably conserve information through time. In standard computers, memory units can be categorized into two main groups: volatile memory, and non-volatile memory.

Volatile memory requires an external energy source, like an electrical current, to store information. Generally situated near the computation unit (CPU), it stores relatively small pieces of information, but provides rapid access times. The main volatile memory unit is the Random Access Memory, or RAM, which is used to stock all of the data and instructions during a processing cycle.

Non-volatile memory stores the information in a physical manner, which means it does not require an energy source to keep the information. Hard drives, flash drives, optical disks, and Solid State Drives (SSDs), are able to store hundreds of terabytes of data, and can therefore store satellite images without difficulty.

The other important aspect of processing is the amount of time necessary for running the various algorithmic steps that produce a result, in this case, a land cover map. In order to provide up to date land cover information, the images should be processed as rapidly as possible after their acquisition. For example, the order of magnitude of the production times of the OSO land cover maps can be counted in days, which is reasonable, but should not be much longer.

Usually, the amount of total storage space available in a memory unit, (volatile or non-volatile) is proportional to the access time necessary to read the data. For example, reading and writing data in volatile memory units like the RAM is several orders of magnitude faster than in an optical disk.

For this reason, processing can not be done directly in the storage unit, as it would be extremely slow. The best practice consists in first copying the data from the disk to the RAM, and then to run the computations in the volatile memory.

A commonly used technique to reduce computation time involves distributing the computation tasks to different processing units. In a *shared memory* scheme, the different processors all have access to the same physical volatile memory unit, and are able to rapidly communicate with each other. This is also commonly known as *multi-threading*.

However, the size of the largest currently available RAM units is in the order of hundreds of GB, which is far too small to contain an entire multi-spectral multi-temporal satellite image requiring several dozens of TB. This implies that the data must be dealt with in a *piece-wise* fashion. In other words, the data set is split into portions, such that each piece is small enough to fit in the RAM. Then, each portion is processed individually, and the results

are copied in the storage unit. This scheme is also known as *streaming*, *piece-wise* or *tile-wise processing* in the case of images, as they are generally split into rectangular tiles.

This also allows the computations to be done using multiple physical volatile memory units, each using their own storage unit. This effectively increases the total amount of available volatile memory, which in fine decreases the total computation time. This mechanism is known as a *distributed memory* scheme, and is only truly useful when dealing with very large data sets. Figure 1.15 shows the difference between shared and distributed memory management.

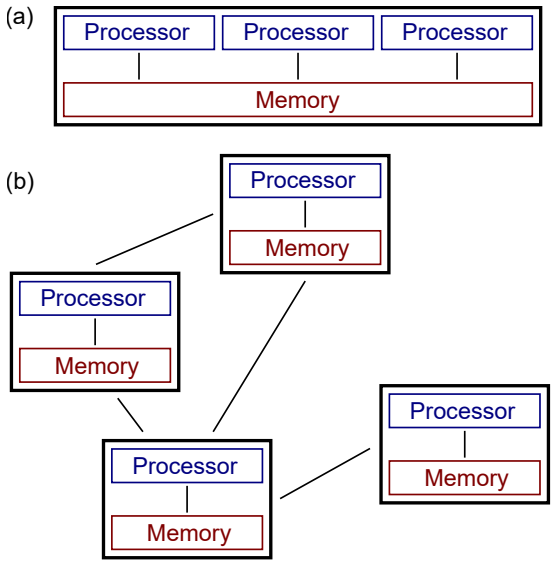


Figure 1.15: (a) Shared memory scheme. Here, the processors operate on the same physical memory unit. The amount of available memory is limited to the size of one physical unit.
(b) Distributed memory. Each processor has access to a distinct physical memory unit, which effectively multiplies the total amount of available memory.

Algorithms are almost always based on sequential steps, where each step depends on the result of a previous computation. However, they can sometimes be broken down into sub-parts that can be run independently. This implies that different processors can operate simultaneously to perform the tasks required for the result to be produced. However, rapid and efficient communication between the different processors is required for significant speed-ups to be attained. This must be taken into account in the design of the algorithms, and is not always a simple task; designing a complex processing scheme that can run in a parallel environment is a challenge in itself, and can sometimes have implications on the design and properties of the algorithm itself.

1.4.4 Objectives and scope

The underlying objective of this Ph.D work is to provide guidelines for selecting the most appropriate way of including context, in an operational land cover mapping problem over wide areas. To this end, several smaller objectives can be defined. First of all, this requires designing a fully automatic process. The solutions investigated here do not require manual intervention, or any tuning of parameters once the training is finished. This allows the processes that are developed to be applied on large amounts of data, at a relatively low cost. Next, a *class-generic* mindset is respected, in the sense that the solutions proposed are not specific to a particular nomenclature. So long as training data is available, the idea is to design methods that can be applied to any number or kind of classes. Moreover, the methods are not necessarily limited to optical imagery, as many of them could be relevant on different types of images, or even on multi-source data sets. Finally, there is an effort made to understand the machine learning process at every step, in other words, to analyze how the decision process is constructed. This allows us to interpret the causes of certain errors, in order to keep improving the process in the long term.

According to the objectives stated above, the manuscript is divided into five parts.

- **Part I** This part presents the background of the research, and the main objectives of the Ph.D. Chapter 1 includes a presentation of land cover maps and their main applications, as well as the basic notions linked to their production. This is commonly done by using various classification methods: photo-interpretation,

ontologies, and supervised classification. Thanks to machine learning technology, the OSO Scientific Expertise Center of the Theia Land Data Center [Leroy et al., 2013], produces a yearly land cover product at a national scale, in a relatively short time. However, certain classes, in particular classes with contextual dependencies, exhibit high error rates. The position of the Ph.D. work with regard to this problem is presented, along with the challenges and the objectives of the associated research. Then, Chapter 2 describes the Earth Observation satellites that provide the images used in the experiments, and how they achieve a global coverage. Finally, Chapter 3 presents the processing chain that generates the yearly OSO map, from the remotely sensed acquisitions to the final land cover products.

- **Part II** The basic experimental setup, theoretical definitions and analytical tools that are paramount for understanding this Ph.D. work are presented in this part. One objective of this work is to establish a taxonomy of the many different possible ways of including contextual information in a dense image classification scheme. Chapter 4 lists various ways to define the context in the first place. Indeed, different methods use spatial supports of varying shape and size. In many cases, it is also relevant to include several scales of context at once. As this is not a new problem, a brief literature review of several of the spatial supports commonly used in Remote Sensing is made. One spatial support shape, the superpixel, is studied with particular attention, as it provides an intermediate representation between the sliding window and the object. Next, the various contextual descriptors that can be used to characterize the content of these spatial supports are described in depth, in Chapter 5, to determine which of these might be relevant for land cover mapping. Finally, Chapter 6 addresses the issue of evaluating land cover maps (and dense classifications in general) using sparse reference data, and proposes a metric based on corner detection and matching.

- **Part III** Here, the main methodological contributions with regard to contextual classification are presented. Chapter 7 presents the methodology adopted to apply the SLIC superpixel algorithm to large images, as this poses theoretical problems to segmentation algorithms. This allows for superpixel methods to be evaluated alongside the other spatial supports on the entire Sentinel-2 data set. Then, in Chapter 8, a new contextual feature is proposed, based on the histogram of classes predicted by a pixel-based classifier. This feature is developed mainly for the purpose of image classification with a large number of features per pixel. Estimating the distribution of classes in a neighborhood is indeed a light-weight way of characterizing a context, as it does not involve high-dimensional multi-spectral and multi-temporal features. The performance of this feature is then compared to the state-of-the-art of context-aware classification methods: Convolutional Neural Networks, which are introduced and defined in Chapter 9.

- **Part IV** Chapters 10 and 11 present the results of the experiments led to evaluate the various methods presented in Part III. A performance evaluation is made on two data sets, which represent commonly encountered problems in land cover mapping. First of all, the Sentinel-2 time series classification problem, with the same nomenclature as OSO, represents a state-of-the art problem, in which the inclusion of context has never been done before. The experiments are run on a variety of different areas, with unique landscapes and eco-climatic behavior, spread across France. Then, a VHSR SPOT-7 imagery classification problem with fewer classes is studied. While this problem also includes a nomenclature with contextual dependencies, it is quite different as it is based on an image at only one date, but at a 1.5m spatial resolution. These two experiments are run in order to provide general conclusions on two very different image classification problems.

- **Part V** In the final chapter, conclusions and perspectives are drawn, both from a methodological and experimental point of view. Chapter 12 presents an overview of issue at hand and the possible solutions, before summarizing the most important results that are shown in Part IV, in regards to the multi-criteria quality objectives. Then, the validity of the conclusions is discussed, as the limits of many of the proposed methods are shown. This leads to suggestions on further research topics and different potential studies and applications.

2

Operational optical imaging systems for land cover mapping at a global scale

"Ever since Newton, we've done science by taking things apart to see how they work. What the computer enables us to do is to put things together to see how they work: we're now synthesized rather than analysed."

– Douglas Adams

Optical sensing, from the Greek *optikos*, meaning *seen*, involves measuring the quantity of sunlight reflected by a surface, in the same way as our eyes perceive the brightness and hues of objects around us. By measuring the energy of the incoming light in different portions of the spectrum (red, green, and blue in our case), we can distinguish different objects that exhibit unique patterns, characteristic of the materials that compose them.

Images captured from an overhead view are an interesting potential source of information for many of the common land cover elements; trees, buildings, and streets can all be characterized by what they look like from above. In fact, different target nomenclatures can require very different types of images.

Many of the target land cover mapping classes, such as annual summer and winter crops, exhibit a unique temporal pattern over a period of one year. Automatically detecting such classes requires the ability to capture multiple images of the globe, at dates regularly spread throughout the four seasons. The recognition of agricultural classes, namely annual crops, orchards, vineyards and intensive grasslands is only possible if images at several dates are available. For example, there is no way to distinguish between different crop types in a certain agricultural area if the only available image shows the bare soil stage.

The desire to obtain images with a frequent revisit is mainly due to two aspects. First of all, a high temporal resolution is useful to recognize agricultural classes, which are characterized by abrupt events at specific dates, such as seeding, flowering, or harvest. Crops such as rapeseed or mustard only flower for a brief period of time in the year. One image captured at the appropriate moment can therefore be a distinguishing factor for agricultural classes. Second of all, acquiring a greater number of images increases the chance of obtaining cloud-free ones, which once again allows more of the different phenological states of the vegetation to be seen.

In terms of spatial resolution, the type of classes can also have an influence. Indeed, to identify a certain class in an image, the size of the pixels in the image should be smaller (or of equivalent size) than the smallest objects in that class. For example, many of the artificial classes, in particular streets and isolated buildings, require images with spatial resolutions in the order of 1-10m. At a first order, artificial classes seem not to benefit from multi-temporal aspects, as man-made surfaces such as concrete and tiles do not evolve throughout the year. However, in order to observe that an area not changed through time requires more than one date of imagery. The use of time series allows to contrast classes that are stable through time with classes that show a temporal dynamic. Moreover, if the target nomenclature contains classes that describe a notion of density of urban cover, through a class such as Discontinuous Urban Fabric for instance, the mix of vegetation and urban cover does follow a yearly temporal evolution.

Finally, a fine spectral resolution is beneficial for most of the common land cover classes. For vegetation classes, which reflect strongly in the infrared bands, this comes down to the capacity to precisely measure the bands of the vegetation *red-edge*. This is a small region of the spectrum (700-780nm) in which vegetation sees a rapid change in reflectance, due mainly to the presence of chlorophyll.

In short, the potential accuracy of global land cover mapping with rich target nomenclatures, depends on the spectral, spatial, and temporal resolutions of the satellite images that are used.

The first group of satellites to be addressed in this chapter are known as *continuous acquisition sensors*. These capture a constant stream of images of the surface that passes underneath the satellite. Thanks to this, they are able to acquire a large number of images throughout the year, creating what is known as a *time series* of images. By aligning their orbits with the rotation of the Earth around the Sun, these satellites are guaranteed to pass over a given place at precisely the same time every day. Such orbits are known as *Sun-Synchronous Orbits* (SSO), as the orbital plane remains at a constant angle with respect to the direction of the Sun. Figure 2.1 shows this solar angle, as well as a few of the other orbital characteristics.

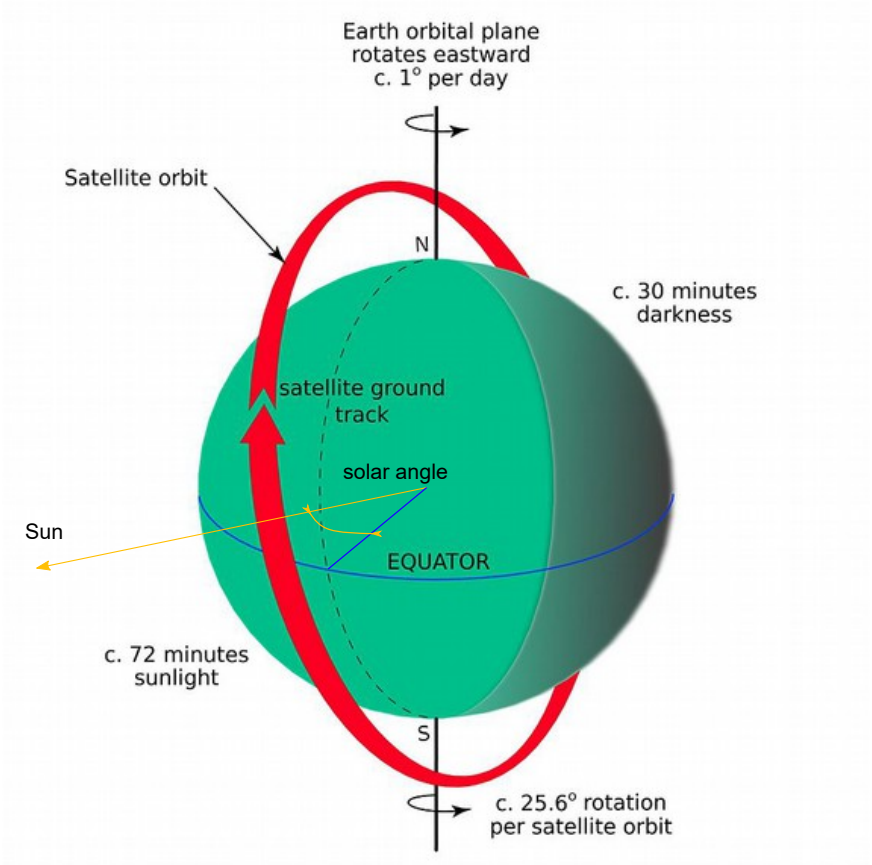


Figure 2.1: Characteristics of a Sun synchronous (SSO) orbit, The solar angle is kept constant as the year goes by, because the orbital plane rotates at the same rate as the Earth around the Sun.

This is done so to ensure the consistency of time series, by making the angles constant so that the values of the pixels are not affected by the fact that the surfaces have bidirectional reflectance patterns which vary both with the illumination and observation angles. Often, the images are taken in the late morning (10:30 AM), when the atmospheric turbulence and the probability of cloud cover is lower than at other moments in the day.

Using SSO orbits also implies that the revisit time must be a round number of days. Furthermore, clouds are a natural obstacle to the optical wavelengths, and will necessarily be captured by such satellites. These need to be dealt with in some way with in order to accurately map the surface underneath. The way in which the OSO map overcomes the issue of cloud cover is explained in Chapter 3, in Section 3.2.

The other group of satellites is known as *programmable acquisition sensors*, which unlike continuous sensors are designed to capture specific areas. These sensors often have high spatial resolutions, which limits the size of the images they can capture at a specific moment in time. This implies that such sensors can achieve global coverage by acquiring an image of each area once a year. For this, the images at different dates are stitched together to form a global image. This creates a *mono-date* mosaic, in the sense that each area is captured once during the year, but is in fact made up of images from several different dates. An example of this is illustrated in Figure 2.2, which shows the intersection area of a summer and winter date, in a region to the south-east of Toulouse. As only one

date for each area is required, such sensors can afford to capture the clearest possible image of each area, meaning no clouds are present in the final global image.

It is possible to visualize mono-date mosaics on free platforms such as Google Earth, or Google Maps. These are today an almost irreplaceable source of information in many areas of study, and even in our every day lives. The images available on Google Earth are actually a combination of aircraft and satellite images at different spatial resolutions. Areas with high demand for imagery, in particular urban areas, are often covered by drone or aircraft images with a higher spatial resolution.

Figure 2.2: Intersection of different dates of a mono-date mosaic visible on the popular Google Maps platform. This image shows the difference between the phenological stages of crops on either side of the intersection. **Source:** Google Maps https://www.google.com/maps

Continuous and programmable acquisition systems are schematically illustrated in Figure 2.3. These technologies are currently used to obtain images covering the entire globe, that are in a way complimentary in the characteristics of the images that they offer.

1. Continuous acquisition sensors, produce time series of images. These also generally have lower spatial and spectral resolutions, as will be seen in Section 2.1.

2. Programmable sensors, which acquire images of a requested area on demand are shown in Section 2.2.

Many of the past land cover mapping studies [Markham and Helder, 2012, Inglada et al., 2015, Griffiths et al., 2019, Demarez et al., 2019] are based on the use of images from the Landsat family. This program had its first successful launch in 1972, and the latest addition is Landsat-8, which began operating in 2013. However, the recently launched Sentinel-2 A&B satellites have different properties, which make them more interesting for our land cover mapping problem. In terms of high/very high spatial resolution mono-date mosaics, the SPOT family [Chevrel et al., 1981], which also began in the 1970s has also been used for land cover mapping, but with different target classes.

This chapter serves as an introduction to the two types of imagery used in the experiments of this Ph.D. work. The following sections describe the properties of two currently operational optical imagery constellations: Sentinel-2 A&B, which are both continuous acquisition systems, in Section 2.1, and SPOT-7, which is a programmable imager, in Section 2.2.

2.1 Properties of the Sentinel-2 constellation

Sentinel-2 is made up of two satellites, A and B, which have been placed on opposite ends of the orbit, in other words, at a constant angle of 180°, with respect to the center of the Earth. This was done in order to divide the revisit time by two, compared to the case where only one satellite is used. For each satellite, the maximal revisit time at the equator of each sensor is 10 days. This implies that the revisit time for the entire constellation, at each location, is only 5 days, which is far more frequent than the 16 day revisit time of Landsat-8.

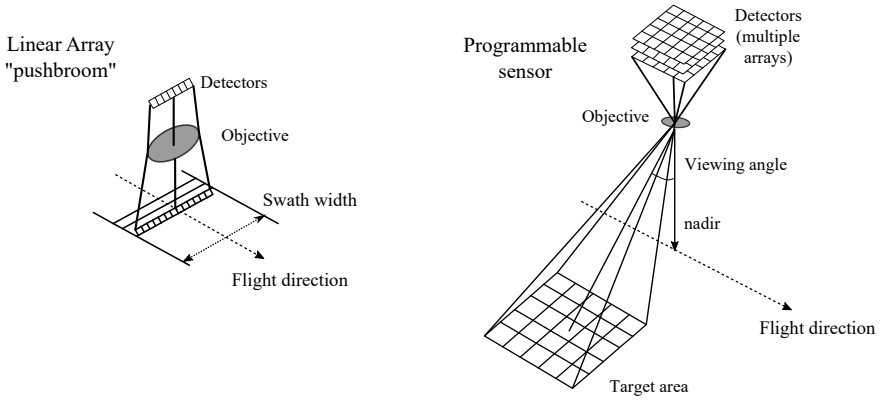


Figure 2.3: Continuous scanning compared to programmable acquisitions. In the first case, a linear array of pixels known as the *push-broom* always captures the area under the satellites passage, by using a sensor placed in a direction perpendicular to the flight direction. Programmable satellites can be oriented off the nadir axis to take pictures of areas at an angle, and can either use a push-broom array or a square array of pixels, as is shown in the illustration on the right side of the figure.

The Sentinel-2 satellites measure the Red, Green, and Blue optical bands, as well as the Near Infrared and Short Wave Infrared wavelengths, as is shown in Table 2.1. These images are adapted for mapping out different types of vegetation, as the vegetation red-edge is captured in three relatively close and narrow bands (5-6-7).

Table 2.1: Optical properties of the Sentinel-2A satellite [Drusch et al., 2012]. The properties of the Sentinel-2B are almost identical.

	Sentinel-2A		
Band Number	Central wavelength (nm)	Bandwidth (nm)	Spatial resolution (m)
1 - Coastal aerosol	442.7	21	60
2 - Blue	492.4	66	10
3 - Green	559.8	36	10
4 - Red	664.6	31	10
5 - Vegetation red edge	704.1	16	20
6 - Vegetation red edge	740.5	15	20
7 - Vegetation red edge	782.8	20	20
8 - NIR	832.8	106	10
8A - Narrow NIR	864.7	22	20
9 - Water vapor	945.1	21	60
10 - SWIR - Cirrus	1373.5	30	60
11 - SWIR	1613.7	94	20
12 - SWIR	2202.4	185	20

The finest bands (visible and near infrared) are captured at a 10m spatial resolution, and are the bands used for most common applications, such as the visualization of images in natural or false color. This 10m spatial resolution is kept as the target land cover resolution, although the other bands used are captured at a lower spatial resolution. The 20m bands include specific infrared bands, particularly the vegetation red edge and short wave infrared, which are key to distinguish the target land cover classes. In the OSO processing chain that is presented in Chapter 3, these bands are re-sampled at 10m using bi-cubic interpolation. This may cause a lower spatial resolution classification result in areas containing vegetation classes, for which the 20m bands are relevant. The 60m bands are not used in the OSO classification scheme under its current state, because they are designed to characterize the atmospheric properties, which are useful for atmospheric corrections which are carried out prior to land cover mapping.

Sentinel-2 A&B, like all push-broom scanners, use a line of sensors arranged perpendicular to the flight direction [Gupta and Hartley, 1997]. During an acquisition, the sensor is maintained pointed towards the nadir, and captures linear images of the ground that passes underneath. These linear acquisitions are stitched together to

form an image of the area underneath the track. The width of the captured area is known as the *swath width*. The Sentinel-2 A&B sensors have a swath width of 290km.

Figure 2.4 shows how Sentinel-2 data is organized in *tiles* for referencing and storage purposes, over France. Indeed, the sensor is only activated when the satellite passes over land, and the majority of the continental surfaces of the Earth are covered in this way. This figure also shows which tiles are selected for experimentation in Part IV.

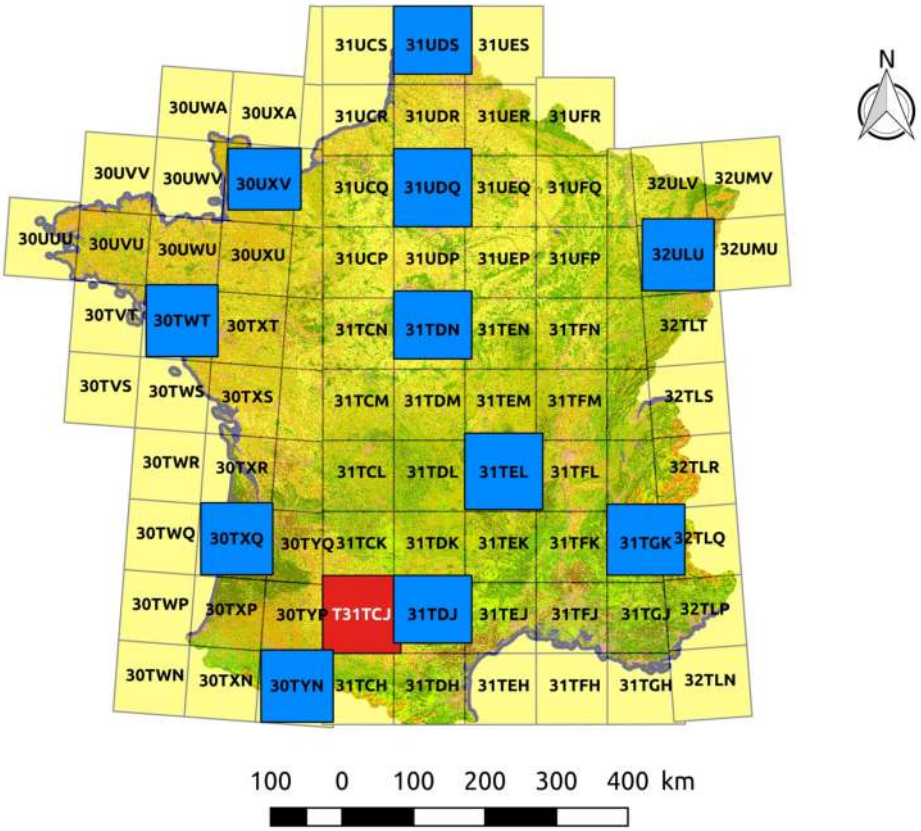


Figure 2.4: Tiles of Sentinel-2 data. The images are organized in this way to allow for a simple retrieval and storage. Each tile is 110×110km. The tiles used for the experiments in Part IV are marked in blue and red. The tile containing Toulouse, T31TCJ, is the primary focus of much of the analysis.

After being captured by the satellite, images must go through several pre-processing steps in order to be used in a land cover mapping application. This involves two main aspects. First of all, transforming the measured voltages into physical units (calibration), and secondly, projecting the image onto the Earth's nearly spherical surface, so that the precise latitude and longitude of each pixel in the image is well defined. These processing steps are common for many optical satellite sensors, and are sometimes known as *image processing levels* [Biesemans et al., 2007].

Radiometric correction: Each light-receiving detector is slightly different, and therefore has unique measurement biases. It is imperative that the brightness is measured in the same way by all of the different detectors, because remote sensing applications often depend on analyzing small changes in pixel values. The detectors can be calibrated according to different sources with well known brightness levels, such as deep space, the Sun, or an on-board calibration system. This step, also known as *inter-detector equalization and calibration*, involves calculating absolute calibration coefficients for each pixel that can be used to convert the measured pixel values into real irradiance measurements.

Projection: In this processing step, the images are associated to a standard cartographic map projection system, using information regarding the position of the sensor at the moment of the acquisition. This is an important step when creating global images, as each scene needs to be well localized to be correctly assembled with the others. Moreover, this step is essential for land cover mapping to superimpose the reference data from different sources over the appropriate areas in the image.

Atmospheric correction: Earth Observation satellites orbit 700-800km above the surface (LEO) capture what

is known as the Top-Of-Atmosphere (TOA) reflectance. Seeing as the final aim here is to map out the classes covering the surface, the Bottom-Of-Atmosphere (BOA) reflectance is what should be measured. Indeed, the atmosphere can perturb the observations, making certain images lighter or darker for no reason other than the weather in the high atmosphere on a particular day. Atmospheric correction is applied in order to obtain spatially consistent images over wide areas, where the pixel values truly represent the classes at the surface. Figure 2.5 shows the subtle difference between a TOA image on the left, and a BOA image on the right. This is the level at which the Theia Land Data center distributes Sentinel-2 images [Leroy et al., 2013].

Figure 2.5: Left: The image of TOA reflectance is slightly lighter due to the presence of aerosols and water vapor in the atmosphere. Right: The image after atmospheric correction represents the BOA reflectance, in other words, the true reflectance at the surface of the Earth.
Source: https://sentinel.esa.int/

The other popular technology for capturing optical data involves taking off-axis images, using an agile programmable satellite, such as SPOT-7.

2.2 Properties of SPOT-7

Programmable acquisition systems are in fact designed in this way due to their desire to capture very high resolution global images. In general, a higher spatial resolution implies a lower swath width. In practice, this means the area covered by the satellites passage is narrower, which means attaining global coverage is difficult, or even impossible using continuous acquisitions at such a spatial resolution. For this reason, programmable acquisition sensors are designed with high quality attitude control systems, which enables them to control their absolute angular position with great precision. These *agile* satellites can aim for a specific area during an overhead pass, which allows for the acquisitions to be made on-demand. This allows such satellites to achieve a yearly global coverage, in other words, in this way they are able to capture each area of the continental surface at least one time. This comes with several advantages and drawbacks.

For example, a satellite can capture a specific zone numerous times during one overhead pass, or during a few successive passes. This is useful to target areas with an urgent need of recent images, following a natural disaster for example, to avoid cloudy images, or to capture areas with different viewing angles for 3D stereoscopy.

For land cover mapping applications, the main disadvantage of such global mosaics is the mono-date aspect, which restricts the recognition potential for agricultural classes that are only present at certain times of the year.

One such sensor is the SPOT-7 satellite, which is used to provide yearly global mosaics at a much finer spatial resolution than Sentinel-2. Indeed, by combining a panchromatic band at 1.50m and spectral bands (R, G, B, NIR) at 6m in a process known as pan-sharpening, a multi-spectral image at 1.50m spatial resolution can be artificially generated. At such a fine spatial resolution, objects like narrow streets, cars and the details of the roofs of buildings are captured. Moreover, the infra-red band is a key indicator for vegetation classes.

However, providing fine-grained land cover maps of very high spatial resolution images remains a challenging issue, as certain context-dependent classes remain difficult to recognize, even with state-of-the-art methods [Bruzzone and Carlin, 2006, Kim et al., 2011, Postadjian et al., 2017]. Indeed, at such fine spatial resolutions, the ratio between the size of the objects in the nomenclature to the size of the pixels in the image is quite large.

Like Sentinel-2 images, SPOT-7 images go through a radiometric correction step, as well as a projection step before being made available. On the other hand, they are not subject to atmospheric correction, for two reasons. First of all, only cloud-free images are acquired by the sensor, meaning the atmospheric effects are likely to be small. Moreover, as only one date is provided for each pixel, there is no need for radiometric coherence between successive dates, as is the case for time series.

Figure 2.6 shows the SPOT-7 image that is the subject of the experiments presented in Chapter 11. This image is used for the discrimination of 5 classes: urban, crops, water, roads, and vegetation, in an identical fashion to the classification problem addressed by [Postadjian et al., 2017]. The low variety of classes is linked to the properties of this kind of image, in particular the mono-date aspect, which causes crops and vegetation to be grouped into only two classes. Moreover, the reference data that is available for a fine distinction of buildings and streets comes from the National Topo Data Base (BD Topo), described in Section 3.1.1, and which does not contain urban density classes.

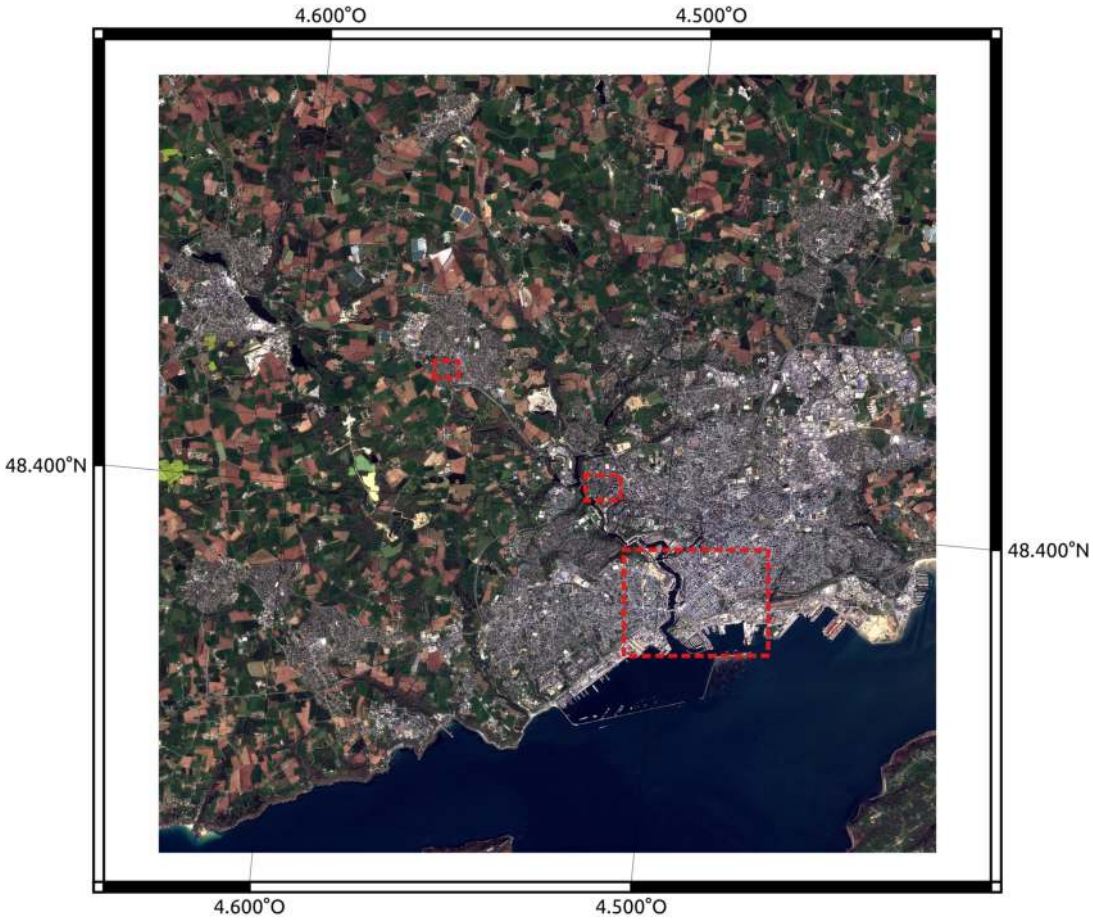


Figure 2.6: SPOT-7 image used in the experiments. It covers an area of $16.5km \times 16.5km$ at a 1m50 spatial resolution containing the city of Brest. The red rectangles represent the areas on which a zoom of the classification results is shown in Figures 11.2 and 11.3, pp165-166.

This chapter has shown two operational technologies for generating optical satellite images at a global scale,

each with unique characteristics that can be useful for different land cover mapping problems. For the issue of mapping out agricultural classes, it appears that the use of time series of images from continuous acquisition sensors such as Sentinel-2 is preferred. This is due to the temporal evolution of the natural classes (crops and vegetation), that can only be observed with several dates. The applicability of supervised classification is ensured by the fact that these sensors provide coherent pixel values from one date to another. This is achieved thanks to the constant illumination and observation angles guaranteed by the SSO orbit, and the atmospheric correction that removes a source of temporal variability due to the atmospheric weather on a given date. The time series of Sentinel-2 images covering the year 2016 is exploited in the first set of experiments which is presented in Part IV, Chapter 10.

Second of all, satellites with programmable acquisitions, SPOT-7 for example, guarantee cloud-free images with a higher spatial resolution than Sentinel-2. However, the mono-date aspect of the global images delivered by such sensors is restrictive for the recognition of certain land cover classes. Nonetheless, these can serve as a basis for generating fine-grained maps with fewer classes, for instance basic urban classes such as roads and buildings for which the need of temporal information is not binding. While the classification of time series of images is the true core of this work, it is interesting to analyze how the image characteristics can influence the results. For this reason, a data set based on a SPOT-7 image is used to provide experimental observations on this subject, in Part IV, Chapter 11.

3

Production of the OSO land cover map

"Don't try to make the problem fit your tools; get yourself the tools that fit the problem."
– Tony Berger

Rapidly presented in Section 1.4.1, the OSO land cover map has been produced on a yearly basis since 2017, following prototype products in 2014-2016, using an open-source processing chain called $iota^2$ which is freely available http://tully.ups-tlse.fr/jordi/iota2. This chapter presents the general methodology used for the production of the OSO land cover maps, from the satellite images to the final usable products. Some characteristics of the maps have evolved over time, for instance, when the first Sentinel-2 images became available in 2016, the spatial resolution was refined from 30m to 10m, and the most recent changes for the 2019 map include a nomenclature extension. However, the general $iota^2$ processing method is independent of these parameters, and such modifications can be made without altering the overall processing chain. Figure 3.1 illustrates this processing chain, and the various steps that are followed in order to obtain a land cover map. The following sections provide details on the sequence of steps that make up the processing chain.

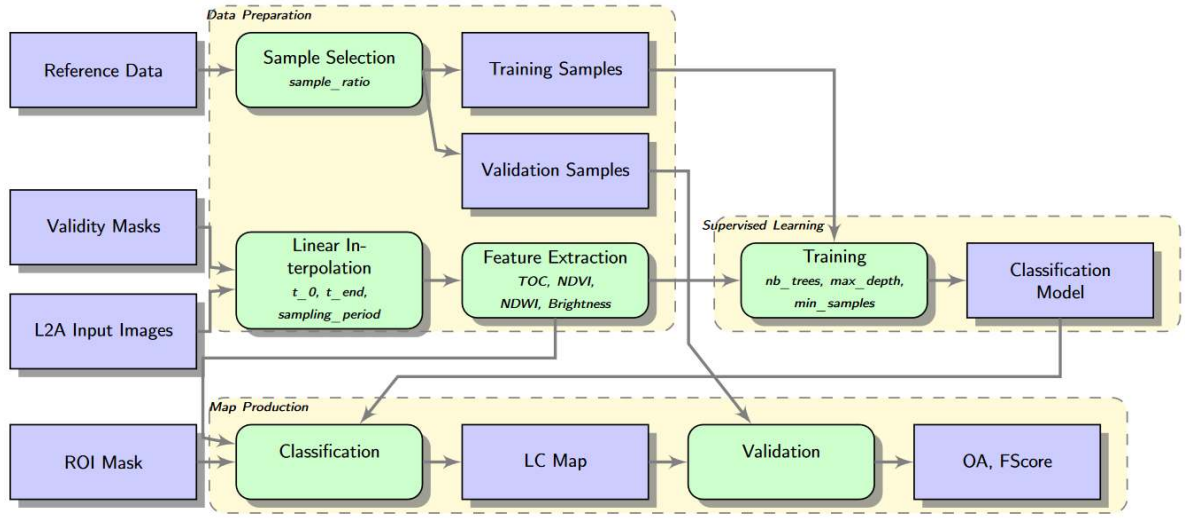


Figure 3.1: Schematic of the $iota^2$ processing chain showing the three main steps: data preparation, supervised learning, and map production. Sections 3.1-3.4 present these steps in detail.

3.1 Reference data and sample selection

3.1.1 Data sources

The reference data used for the production of the OSO land cover map comes from several sources. This is due to the variety of classes in the target nomenclature, which do not exist in one regularly updated database. The

construction of these training data sets is based on the principle that certain land cover classes, called *perennial classes*, have a lower chance than others of being subject to change over time. In practice, cover types like water, artificial surfaces, and forests, only evolve over several years. This means that reference data from the past can be used to classify more recent images, so long as the time difference is relatively low. The data sources used for producing the OSO land cover map [Inglada et al., 2017] are shown in Table 1.1, on page 30.

1. **Corine Land Cover (CLC)** [Bossard et al., 2000] is based on manual photo-interpretation, and is updated every 6 years. It contains a rich description of the many land cover elements, with a nomenclature of 44 classes. Only the perennial classes (urban cover, water, beaches, bare rocks, and natural grasslands) are kept for the production of the OSO maps. The maps of 2014-2017 were all produced using classes from CLC 2012.

2. **Urban Atlas (UA)** [Montero et al., 2014] covers all cities with over 100,000 inhabitants, and is updated every 6 years, like CLC. It contains 17 classes describing different levels of urban density, as well as other urban features like construction sites, sports and leisure sites, with a MMU of 0.25ha ($\equiv$ 50m×50m area). The full nomenclature, as well as possible equivalent OSO classes are shown in Table 3.1.

3. **National Topo Data Base (BD Topo)** [Maugeais et al., 2011] is a continuously updated data base made by the French National Geographical Institute (IGN), with an average update time of 3 years for the entire territory. The forest database describes the main woody cover classes (woody moorlands, broad-leaved and coniferous forests). The urban data base gives the outline of buildings, which will be useful for the SPOT-7 experiments, but does not provide an indication of the urban density. This is why it is only used for the road surface class on the Sentinel-2 problem.

4. **Land Parcel Information Registry, Registre Parcellaire Graphique (RPG)** [Cantelaube and Carles, 2014] is another product of the IGN which describes arable lands based on a graphical declaration system from the farmers cultivating the land. It contains an up-to-date description of the main agricultural classes.

5. **Randolph Glacier Inventory (RGI)** [Pfeffer et al., 2014] contains a worldwide description of the glaciers, and is updated every 1 or 2 years.

When combining data sets from different years and from different sources, disagreements can occur. Any area that is described by two different class labels is therefore discarded from the data set, in order to reduce confusions.

The polygons are also eroded by 1 pixel, which is done to mitigate the effect of errors in the geometrical alignment of the different images in a time series. This also removes many of the so-called *mixels* (pixels which are on the edge of two different land cover types) from the data set.

Table 3.1: Urban classes as defined by the OSO Nomenclature and the Urban Atlas nomenclature. The table shows how the Urban Atlas classes could be grouped according to the more basic OSO classes.

Urban Atlas class	Equivalent OSO Class
Continuous Urban Fabric (S.L. > 80%): 11100	Continuous urban fabric: 41
Discontinuous Dense Urban Fabric (S.L. 50% - 80%): 11210 Discontinuous Medium Density (S.L. 30% - 50%): 11220 Discontinuous Low Density Urban (S.L. 10% - 30%): 11230 Discontinuous Very Low Density (S.L. < 10%): 11240 Isolated structures: 11300	Discontinuous urban fabric: 42
Industrial, commercial, public, Military and private units: 12100 Port areas: 12300	Industrial and commercial areas: 43
Fast transit roads and associated Land: 12210 Other roads and associated land: 12220 Railways and associated land: 12230 Mineral extraction and dump sites: 13100 Construction sites: 13300 Sports and leisure facilities: 14200 Land without current use: 13400	Road surfaces: 44

3.1.2 Split of training and evaluation sets

In order to provide a realistic evaluation of a land cover map, the test data set should be sampled from different polygons than the training data set. This means that the test set will not contain samples from the same geographical

units (field, neighborhood, road, etc.) as the training data set. Extracts of the training and validation polygons used in the OSO approach are given in Figure 3.2.

Such a sampling ensures that the test data set is representative of the data that the classifier is likely to encounter during its operational phase, in the sense that it contains only samples from areas that the classifier has never seen before. If samples from the same polygons are used for training and testing, the performance scores are generally higher than what they would be in reality, in other words, during the operational phase. This deformation of the scores can also lead to an incorrect parametrization of certain models, as they might learn to memorize the training samples (which is known as over-fitting), rather than seek to generalize. Moreover, misevaluated scores can cause a biased hyper-parametrization or non-optimal model selection, in the case that several different models are tested by the user. A recent study by [Liang et al., 2017] shows that this can lead to an over-consideration of nearby pixels when using contextual methods, and artificially inflates the performance scores.

(a) Training polygons used to sample points for the learning stage of the classifier.

(b) Testing or validation polygons, used to evaluate the performance of the classifier.

Figure 3.2: Extract of training and test data sets in a small area near Toulouse show how the training and testing polygons are split at a *polygon* level. This ensures scores that are representative of the *generalization* power of the classifier, as the validation samples have never been seen by the classifier before.

3.2 Cloud-filling and temporal interpolation

In order to conserve an automatic paradigm that is independent on the area or nomenclature choices, an approach with no selection of specific dates is chosen. This avoids introducing biases into the data towards specific land cover classes or landscape types, and takes advantage of the power of the classifier to select and use the most relevant dates.

In the procedure proposed in [Inglada et al., 2017], each pixel is characterized by a time series of image features: surface reflectances and spectral indices, specified in Section 3.3. In order to provide comparable features for the classifier, a homogeneous set of dates should be used on the entire area, which must be the same for training, testing, and operation phases.

One downside to using optical imagery is that clouds block the view of the Earth's surface in optical wavelengths which makes acquiring clear winter images relatively difficult in certain areas. Even recent optical sensors like the Sentinel-2 A&B, which work in tandem to provide images with a frequent revisit of 5 days are sensitive to the effect of cloud presence. Figure 3.3 shows the number of valid (cloud-free) acquisitions of each pixel in the time series, for year of Sentinel-2 data. The pixels on the overlap of neighboring satellite tracks are bright as they are captured many more times. Overall, the northern area of the country appears darker, due to the heavier cloud cover.

The Theia Land Data Center [Leroy et al., 2013] distributes Level 2A Sentinel-2 data as tiles of 110km×110km with a 10km overlap, shown in Figure 2.4. In the experiments done on the OSO land cover problem in Part IV, different tiles are used to test the methods on a variety of landscapes.

In order to provide a homogeneous land cover map with a supervised classification approach, the acquisition dates must be identical over the entire area, and be free of clouds.

In [Inglada et al., 2015], the authors propose a linear interpolation method, which fills in the values of the obscured pixels based on previous and following cloud-free dates. This interpolation should be applied on all of the surface reflectance values before the computation of spectral indices.

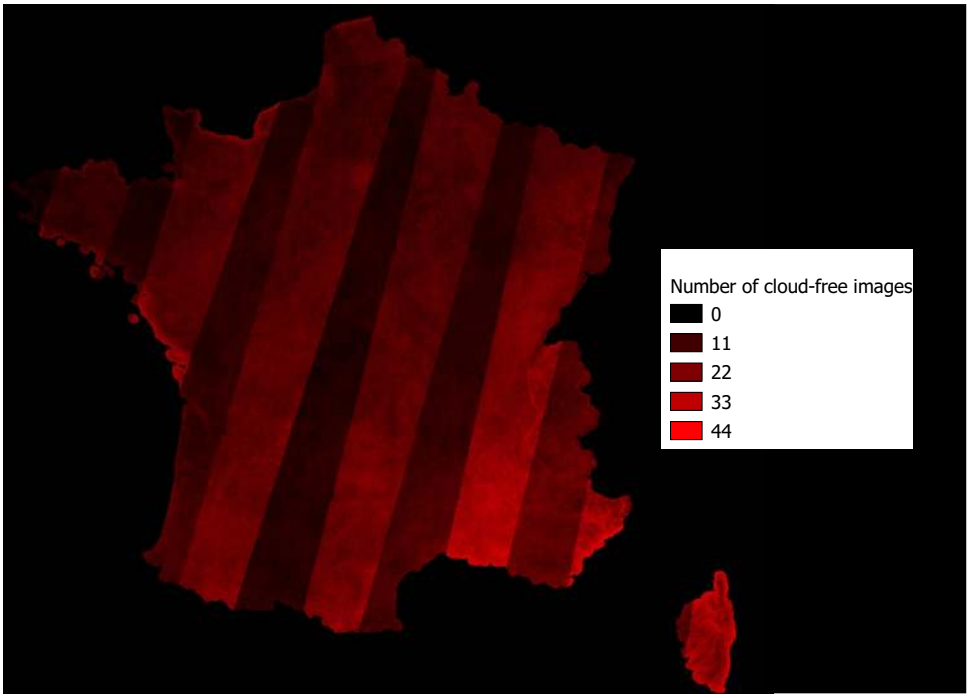


Figure 3.3: Number of valid images per pixel, over France in 2016. Dark pixels indicate a low number of images per pixel. Areas in bright red, which have been captured many times, show the overlap of satellite orbits, and differences in cloud cover between the north and south of the country.

Interpolation can be used for more than just accounting for cloudy images, as it allows for the values of any date to be estimated, based on images captured nearby in time. This solves the issue of temporal shifts between adjacent satellite tracks, as one set of common dates for the entire area can be selected. The time series begins at the first acquisition date, and the temporal step can be defined according to the revisit date of the imagery used. It is often taken equal or lower than the maximal revisit time of the satellites, to avoid over-sampling effects.

3.3 Feature extraction

The production of the latest OSO land cover maps is based on time series of Sentinel-2 images, which measure *reflectances* in several bands of the EM spectrum. Reflectance is defined as the proportion of solar energy reflected by a surface, on a given spectral interval. The spectral characteristics of Sentinel-2 images are given in Chapter 2, in Table 2.1.

Several spectral indices can be calculated from these reflectances, which are commonly used to characterize land cover elements such as vegetation, water, or urban areas. An example of these indices on Sentinel-2 imagery is shown in Figure 3.4.

- **Normalized Difference Vegetation Index** or NDVI is the normalized difference between the Red (R) and Near Infrared (NIR) bands. It ranges between -1 and 1, with strong values often indicating the presence of vegetation .

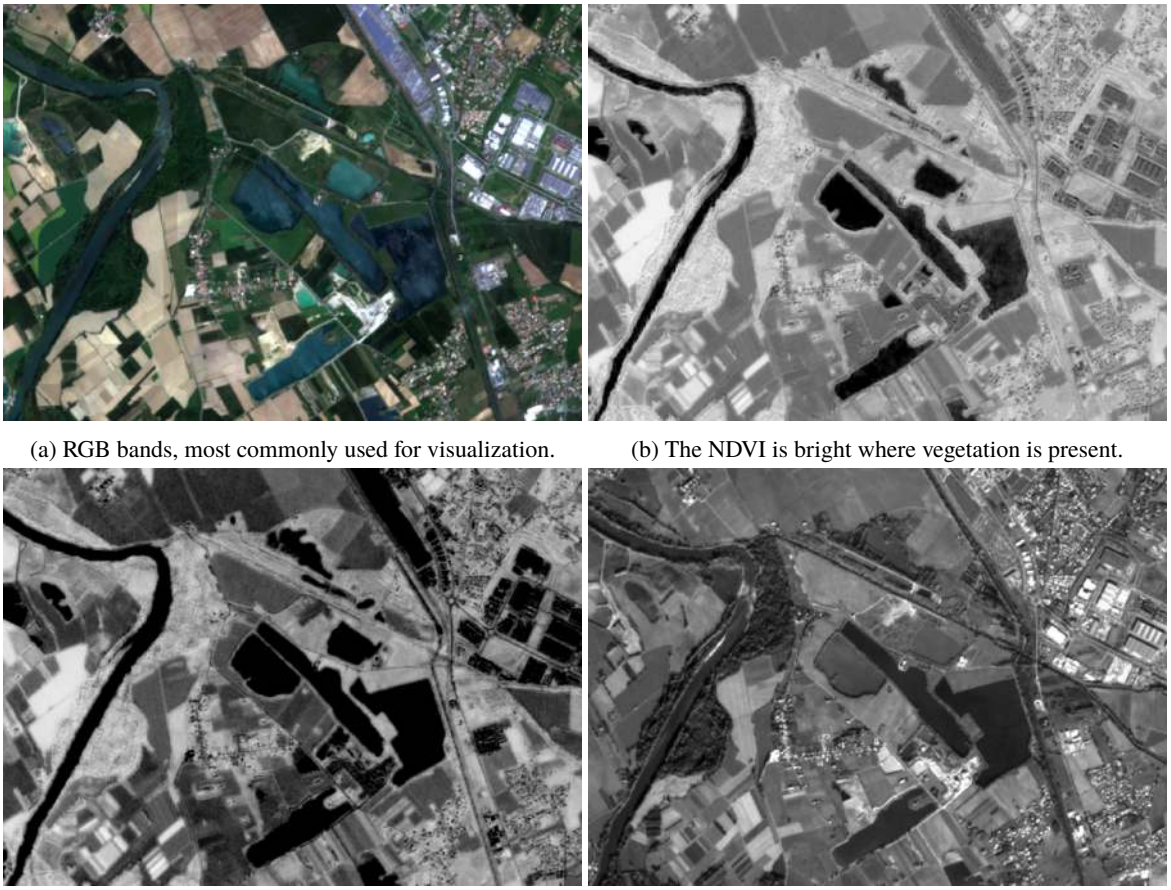
$$NDVI = \frac{NIR - R}{NIR + R}$$

- **Normalized Difference Water Index** or NDWI, uses the NIR and SWIR bands, and is used to characterize the presence of liquid water in vegetation [Gao, 1996].

$$NDWI = \frac{NIR - SWIR}{NIR + SWIR}$$

- **Brightness** (BR) is the rooted sum of the square values of the N spectral bands B_i, $i \in [1...N]$, in other words, the L_2 norm of the spectral vector. It is often used to characterize urban areas, which exhibit large contrasts in brightness.

$$BR = \sqrt{\sum_{i=1}^{N} (B_i^2}$$



(a) RGB bands, most commonly used for visualization.

(b) The NDVI is bright where vegetation is present.

(c) The NDWI is bright in humid vegetative areas, but very dark for water.

(d) The Brightness underlines contrasts in urban areas.

Figure 3.4: Different spectral indices used to produce the OSO map, shown on a Sentinel-2 image of January 2016. Each index characterizes different basic elements of land cover, like vegetation (NDVI) and water (NDWI), or overall spectral behavior in the case of Brightness.

In the Sentinel-2 configuration, each mono-date image contains 10 reflectance features (4 at a 10m spatial resolution, 6 at a 20m spatial resolution) and 3 spectral indices (NDVI, NDWI, BR), for a total of 13 features per date.

3.4 Details of the supervised learning algorithm: Random Forest

The supervised classification algorithm used to produce the OSO maps is the *Random Forest*, which is described here.

In order to learn how to recognize the various classes in the training data set, the Random Forest [Breiman, 2001] constructs a large amount of *decision trees*, also known as Classification And Regression Trees (CART) [Breiman, 1984], based on binary rules like the ones mentioned in Section 1.3.2.

A decision tree is grown by randomly selecting one feature at a time to split the training data set into two parts, the objective being to split away certain classes from the others, based on the selected feature. For this, the training set is first sorted according to the chosen feature, and then split according to a purity criterion. A possible classification tree using two features (f_1, f_2) to separate two classes c_1 and c_2 is shown in Figure 3.5.

When learning a decision tree, a purity criterion is used to optimize the quality of the splits made during training, the objective being to split the feature space in areas with only one type of label. In practice, the training is *boosted*, which means that at each split, the purity of the left and right branches is calculated for more than one

random feature (usually, the square root of the total number of features), and the feature providing the purest nodes is selected.

This process is applied iteratively, split after split, until a stopping criterion is met. This can be either a maximal number of splits, a certain desired level of purity reached, or a minimum number of elements in the nodes. The final splits provide the *leaves* of the decision tree. Each leaf therefore contains a subset of the training data, and the associated labels. In order to determine the class of an unlabeled pixel, the decision branches are applied to its features, and it is assigned the label of the majority class present in the leaf in which it arrives.

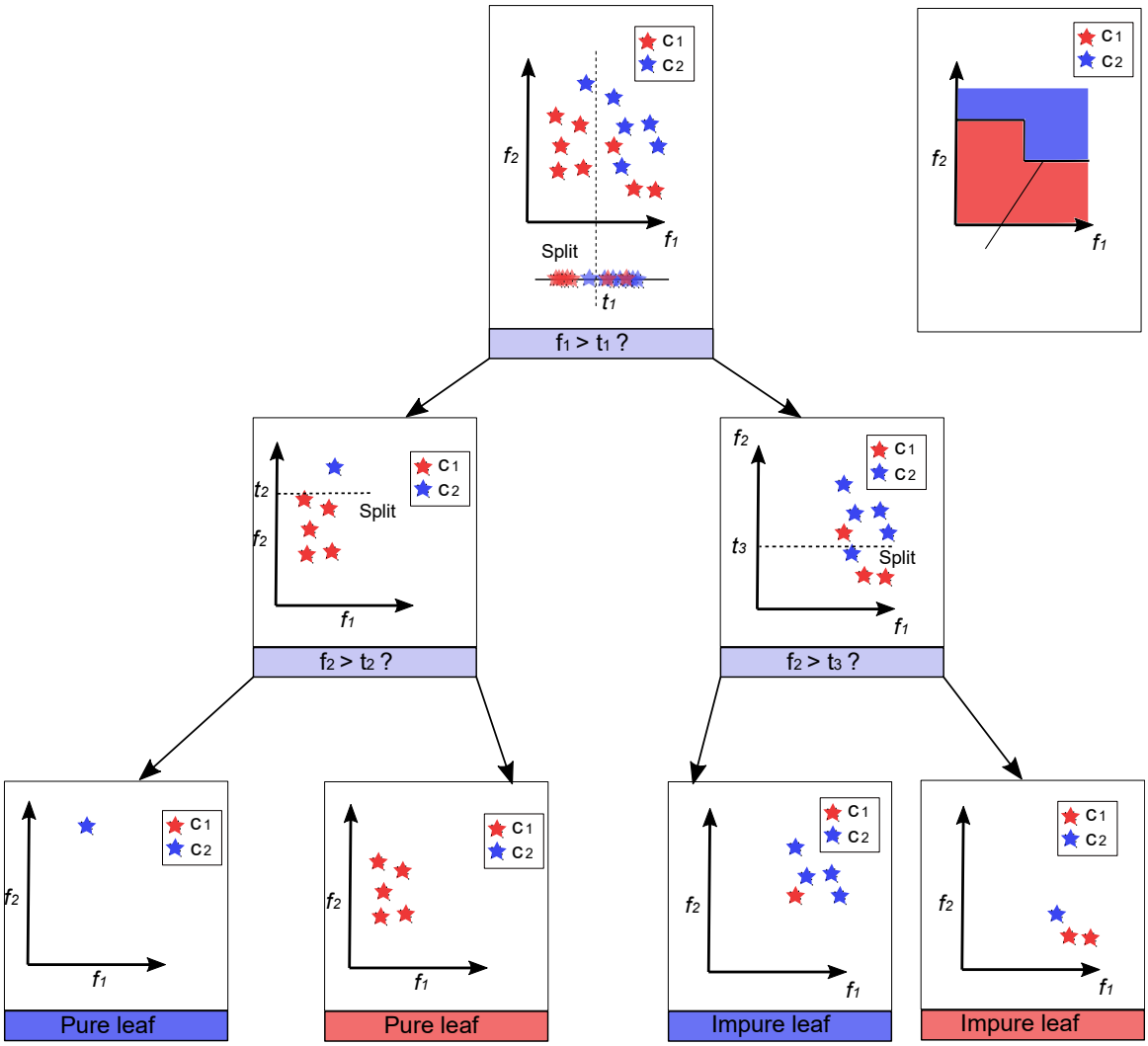


Figure 3.5: Classification tree like the one used in Random Forest, illustrated on a simple two class problem. The training data is split iteratively from the root node, using thresholds (noted t_i) on the different features. At each step, the split threshold is selected to minimize the Gini Impurity of the right and left nodes. At the end of this process, the data set is partitioned into groups, which should be as pure as possible in terms of class labels. Presented with an unlabeled sample, the tree follows the decisions until a leaf is reached, and then assigns the sample with the majority label in the leaf.

3.4.1 Purity criteria

A commonly used criterion is the Gini criterion which evaluates if a set of categorical labels contains a majority of one type of label, or if the set of labels contains various different classes. Practically speaking, Gini Impurity is calculated as the probability of incorrectly classifying a randomly chosen element in the data set if it were randomly labeled according to the class distribution alone.

$$G = \sum_{i=1}^{C} p(i) * (1 - p(i)) \tag{3.1}$$

C is the number of classes and $p(i)$ is the probability of randomly picking an element of class i, also known as the *class prior probability*. The Gini criterion is evaluated for each possible branch of the decision tree, in order to optimize the purity of the splits.

To understand the Gini Impurity criterion, consider a data set with 9 samples sharing the same label l_1, and 1 sample carrying a different label l_2. The probability of of randomly selecting a sample labeled l_1 is therefore $p(l_1) = \frac{9}{10}$, and $p(l_2) = \frac{1}{10}$. If an element of l_1 is sampled, the probability of incorrectly classifying it by using a random decision based only on the class distribution is $1 - p(l_1) = \frac{1}{10}$. The probability of both selecting and incorrectly classifying an element of the majority label, l_1, is thus $p(l_1) \times (1 - p(l_1)) = \frac{9}{10} * \frac{1}{10} = 0.09$, or 9%. In the same way, there is a 9% chance of incorrectly classifying an element of the minority label l_2, as there is only one. The Gini Impurity of this data set is the sum of these two probabilities, which in this case is 0.18. This relatively low number indicates a high degree of purity in the data set. A balanced data set, with the same number of elements for each of the C class has a Gini Impurity of $1 - 1/C$, for instance, a data set containing 10 differently labeled elements would have relatively high Gini Impurity of 0.9.

3.4.2 Ensemble methods

While CARTs are efficient way of conducting supervised classification, it seems natural that in many cases, binary decisions do not accurately reflect the ideal decision boundaries, which are more likely to be complex, especially in high-dimensional spaces. This means that individual decision trees are not a very strong deciders, especially in areas of the feature space presenting a large inter-class overlap. In order to achieve complex decision boundaries in the feature space, each tree must randomly select the same key features several times, which implies that decision trees perform best on feature spaces with a low number of dimensions.

Unfortunately, when dealing with very complex problems, individual decision trees encounter generalization issues, as is shown in [Criminisi et al., 2012]. Each decision tree is biased by the early thresholds, which are in most cases random, and do not necessarily reflect real class behavior in the data set. Moreover, these basic models, with a relatively low number of variables, are unable to fully represent complex problems.

For this reason popular ensemble methods like Random Forest and XG-BOOST [Chen and Guestrin, 2016] train a large number of trees, which participate in a voting system. One important characteristic of these so-called ensemble systems [Rokach, 2009] is that they perform better with a certain degree of individuality. It seems logical that the power of the ensemble can be lost if the totality of the population always produces the same prediction. The first form of randomness that is introduced in the trees is in the unique selection of features that are tested at each split (boosting).

In the Random Forest proposed by [Breiman, 2001], each individual decision tree learns from a random subset of the training data, through a process known as *bootstrap aggregating*, or *bagging*. Considering a training data set T with a total of n samples, bagging involves extracting m different data sets called *bags*, noted T_i. This is done by randomly selecting n' samples from T with replacement, which implies that some of the samples may be repeated in each bag. In fact, an average ratio of $\frac{1}{e} \approx 0.632$, i.e. 63.2% of unique samples are likely to be selected.

It may seem illogical to train the voters with fewer training samples than are available, however, this allows for the ensemble to exhibit several interesting properties, which are listed here.

1. It reduces correlation in the ensemble, as each voter will have been trained on a unique part of the data set.
2. Bagging allows for the density distribution of the data set to be incorporated to a certain degree. Areas of the feature space that contain many points will be present in most of the m bags, whereas areas with very few samples, which might be outliers, will only be seen by some of the members of the ensemble.
3. The samples which are not selected for training can be used as intermediate validation points on which to calculate an estimation of the generalization error of each tree, known as the Out-Of-Bag (OOB) error.

The main advantage of the Random Forest method over others is the speed at which the training can be accomplished. Even with mono-thread implementations, work on land cover mapping with time series led by [Inglada et al., 2015] shows that Random Forests are orders of magnitude faster than Support Vector Machines. The experiments led during this Ph.D., while not discussed here, have shown that this is also the case compared to Neural Networks, which are described in Part III.

Once the first iterations of training and testing are finished, the classifier can be applied to unlabeled data, to produce the actual land cover maps. This is known as the prediction phase. Section 3.5 describes the design of the various steps involved in this phase, which is in fact not as simple as applying one classifier to the entire data set.

3.5 The final prediction phase

In order to produce land cover maps, a supervised classifier must first be trained, and this requires several pre-processing steps on the data set, which have been explained in the previous sections. However, due to the large dimension of the data set, both in terms of pixel features and extent, a few theoretical and practical issues arise. The following sections explain how the production of OSO land cover is adapted to overcome these issues.

It is important to mention here that feature reduction schemes, such as feature selection or extraction are avoided for the production of the OSO land cover maps. Feature selection methods could involve choosing certain key dates or using fewer bands and indices. This would however be a choice specific for the class nomenclature, which would need revision if the target classes change over time. Second of all, feature extraction methods such as Principal Component Analysis (PCA), create new features as combinations of the existing ones, in an effort to preserve the quantity of information in the data set. Both of these methods imply theoretical losses of description power of the training data set, in one way or another. In fact, the task of feature extraction is better performed by the supervised classifier, as the training labels are available to it. The classification quality should guide the feature extraction, rather than the preservation of information in lower dimensions.

The desire to preserve the totality of the features is a key aspect of the methods researched and evaluated in the rest of the manuscript. In fact, the operational production of the OSO map shows that there are ways to overcome the high dimension of pixel features, and to provide all of the available pixel features to the classifier.

3.5.1 Eco-climatic stratification

One problem that can be expressed in classification terms as a *high intra-class variability* can cause confusion in the land cover mapping process. This occurs when a class contains pixels with very different aspects, which increases the risk that they might overlap with other classes. One way to address this issue is to simplify the problem, by dividing the data set into groups, following knowledge from the comprehension of the problem itself.

In land cover mapping terms, this source of external knowledge can come from meteorological observations over long periods of time. By dividing the image into areas called *eco-climatic regions* that fit with certain weather patterns, the time series become more expressive of the regional classes. In other words, they are not influenced by pixels at the other side of the country, where the climate is totally different.

To incorporate this eco-climatic information into a land cover mapping scheme, an elegant solution appears: training one model on each eco-climatic region. This allows for two interesting advantages.

1. A part of the intra-class variability is eliminated at the start of the problem.

2. Each model is trained on a lower amount of data, and the different models can be trained in parallel to save time.

Using eco-climatic stratification was proven to be beneficial over learning with samples from the entire data set by [Inglada et al., 2017]. Figure 3.6 shows a map of the different areas used in the OSO eco-climatic stratification model. These 8 types describe the overall climate of an area, and were calculated based mainly on meteorological observations over long periods of time [Joly et al., 2010].

The question of the edges of the regions is an interesting point. In fact, observations near these edges show that the effect of the edge is hardly visible in the land cover map.

3.5.2 Tile-based classification and mosaicking

As was mentioned earlier, predicting a large amount of data imposes computational constraints. In practice, the currently available hardware is only able to stock one Sentinel-2 tile in memory. For this reason, only one tile can be classified at a time. Due to the eco-climatic stratification, there are in fact 8 models that need to classify not tiles, but entire regions, that can overlap on different tiles. The following strategy is adopted.

1. For each eco-climatic area, all of the tiles that intersect with the area are classified using the corresponding model.

2. The results of a tile are then stiched together using the eco-climatic regions, and their land cover map of the tile. This second step is known as the mosaicking step.

Figure 2.4 on page 43 shows the Sentinel-2 tiles that cover France, over the OSO map of 2016. They are indeed far smaller than an eco-climatic region.

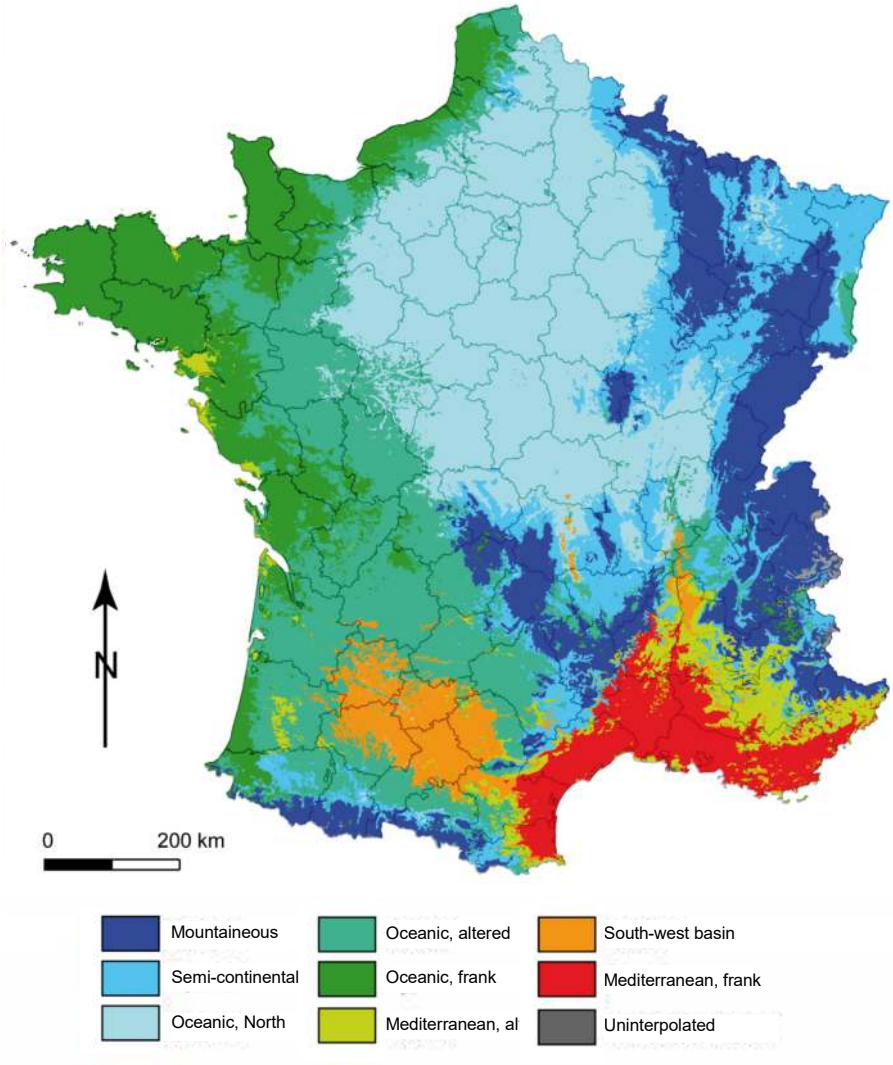


Figure 3.6: Eco-climatic areas used for the production of the OSO land cover maps. Each region groups together areas that have a similar climate throughout the year. A more precise definition of each typology is given in [Joly et al., 2010]. One small adjustment is made for the stratification process: areas smaller than $100km^2$ are removed [Inglada et al., 2017].

3.5.3 Validation

Once the map has been produced, it can be validated, that is to say, its overall quality can be assessed. The very first level of validation is the one performed by the classifier itself, as most classifiers are able to provide an indication of their confidence. Figure 3.7 shows the confidence map of the classification of the year 2016, with green pixels indicating a high degree of confidence, and dark pixels indicating uncertainty. This is meant to be used alongside the map in order to interpret its results.

Then, the map can be tested on various data sets to assess its accuracy. This is done in two stages, an internal validation followed by an external validation. The internal validation data set is made up of parts of the reference data that are set aside for this precise purpose. These are the scores that are meant to represent the generalization power of the classifier regarding the problem at hand, and are a standard procedure for any supervised classification problem.

However, as the test data originates from the same sources as the training data, it can contain a bias with regards to the problem. For instance, if the reference data used for training and testing contains no elements of a certain type, it can be systematically misclassified by the classifier, and this will only appear in the maps. Another form of bias can appear if the reference data contains labeling errors that are systematic across the data set. The classifier will learn these errors, and it will not be reflected in the internal validation scores.

For this reason, an external validation can be set up. In the case of the OSO map, this involves several aspects.

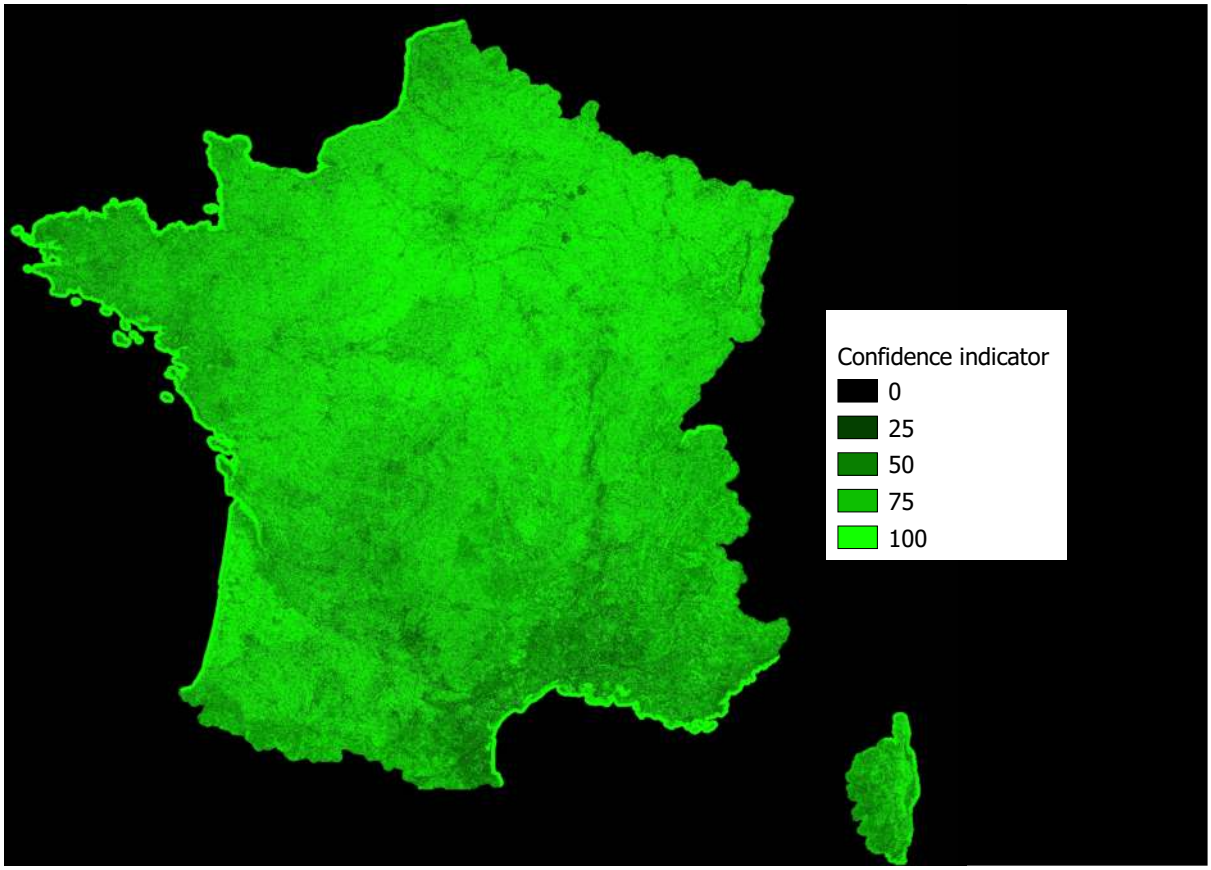


Figure 3.7: Confidence map of the OSO land cover map. This is an internal score evaluated by the classifier, based on the label uncertainty in the ensemble. It appears that water, as could be expected, is classified with a great degree of certainty. Moreover, large cities, such as Paris and its suburban area appear darker in the image, due to the presence of urban classes.

First of all, locally collected reference data that is not used for training OSO can be used for an external validation. The other way of proceeding involves photo-interpreters performing the classification on a small area, using very high resolution imagery, and comparing the results to the produced map. This was performed for the 2016 map. The executive report reads as follows:

> *This report provides the evaluation results of the CESBIO OSO 2016 10m layer and the CESBIO OSO 2016 20m layer.*
>
> *The thematic accuracy assessment was conducted in a two-stage process:*
>
> 1. *An initial blind interpretation in which the validation team did not have knowledge of the product's thematic classes.*
> 2. *A plausibility analysis was performed on all sample units in disagreement with the production data to consider the following cases:*
>
> - *Uncertain code, both producer and operator codes are plausible. Final validation code used is producer code.*
> - *Error from first validation interpretation. Final validation used is producer code*
> - *Error from producer. Final validation code used is from first validation interpretation*
> - *Producer and operator are both wrong. Final Validation code used is a new code from this second interpretation.*
>
> *Resulting to this two-stage approach, it should be noticed that the plausibility analysis exhibit better results than the blind analysis.*
>
> *The thematic accuracy assessment was carried out over 1,428 sample units covering France and Corsica.*

The final results show that the CESBIO OSO product meet the usually accepted thematic validation requirement, i.e. 85% in both blind interpretation and plausibility analysis. Indeed, the overall accuracies obtained are 81.4±3.68% for the blind analysis and 91.7±1.25% for the plausibility analysis on the CESBIO OSO 10m layer. The analysis on the 20m layer shows us that the overall accuracy for the blind approach is 81.1±3.65% and 88.2±3.15% for the plausibility approach.

Quality checks of the validation points have been made by French experts. It should be noticed that for the blind analysis, the methodology of control was based mostly on Google Earth imagery, no additional thematic source of information that could provide further context was used such as forest stand maps, peatland maps, etc.

Source: http://www.cesbio.ups-tlse.fr/multitemp/?p=12869

These elements define the environment in which this work progresses, that any new methodology must respect in order to be applied in an operational manner. The key notions involved are listed here.

1. The pixel-based classification map has a strong recognition of many of the classes, thanks to the high dimension of the data.
2. This high dimension, both in terms of features and of extent, imposes significant computational constraints.
3. The reference data is only available in a sparse form, which as the following chapters will show, causes difficulty both in training the supervised classifier and evaluating the quality of the land cover maps it generates.

Part II

Basics of contextual classification

4

Defining the spatial support

"There are two cultures in the use of statistical modeling to reach conclusions from data. One assumes that the data are generated by a given stochastic data model. The other uses algorithmic models and treats the data mechanism as unknown. The statistical community has been committed to the almost exclusive use of data models. This commitment has led to irrelevant theory, questionable conclusions, and has kept statisticians from working on a large range of interesting current problems. Algorithmic modeling, both in theory and practice, has developed rapidly in fields outside statistics. It can be used both on large complex data sets and as a more accurate and informative alternative to data modeling on smaller data sets. If our goal as a field is to use data to solve problems, then we need to move away from exclusive dependence on data models and adopt a more diverse set of tools. "

– Leo Breiman, Statistical Modeling: The Two Cultures [Breiman et al., 2001]

In order to improve the recognition rates of classes that depend on the context of the pixel, such as density based classes, it is desirable to add contextual information into the classification process. This chapter defines a taxonomy of the various manners in which context can be included in a dense classification scheme, paying particular attention to the shape of the context which is used.

The notion of context is based on the physical distance between different pixels in the image. This information is contained in the image itself, and is extremely valuable as it is an inherent part in the way we percieve images. In fact, when we see a picture, it is virtually impossible to not see it as a whole. If the same pixels were jumbled up, the image would make absolutely no sense. Our eyes constantly percieve spatially structured information, and even have specialized receptors for the central and peripheral areas. This allows us to understand the many layers surrounding each point we see, which is undoubtedly important for our survival.

The relevant context in many classification cases goes beyond a small neighborhood of one or two pixels. For example, in computer vision problems, the presence of blue pixels in the top of an image can indicate the scene is outdoors, increasing the chance of there being trees and grass in an entirely unrelated area of the image. Long range dependencies like these are the most difficult to take into account in a classification process, as just defining the necessary size of neighborhood depends largely on the problem at hand, and is also often conditioned by the available computational power.

In the context of land cover mapping, the issue comes from the incapacity of the pixel features to describe certain aspects of the target classes, as these express notions that exist on a much larger scale than each individual pixel, such as density.

The first way of including context is to use our knowledge of image elements to define a set of *contextual features*, which are then combined with a supervised classification system like the one introduced earlier, in Section 3.4. The idea behind this is to condense the large amount of information present in the context of each pixel into a few relevant descriptors, which are linked to notions that we understand, like texture, spatial frequencies, sharp corners, and more.

The second way involves directly providing the entire context to the classifier as features, in other words, a list of every single feature of every single pixel in the neighborhood. This equates to an extremely large number of features if a large scale of context is desired, and for that reason, a classification model tailored to interpret such information can be needed. These is referred to here as a *model-based* approach.

The following sections define three commonly used spatial support shapes: sliding windows, objects and superpixels. Figure 4.1 shows an example of these three spatial support shapes over a homogeneous area (agricultural lands) and a textured area (urban cover).

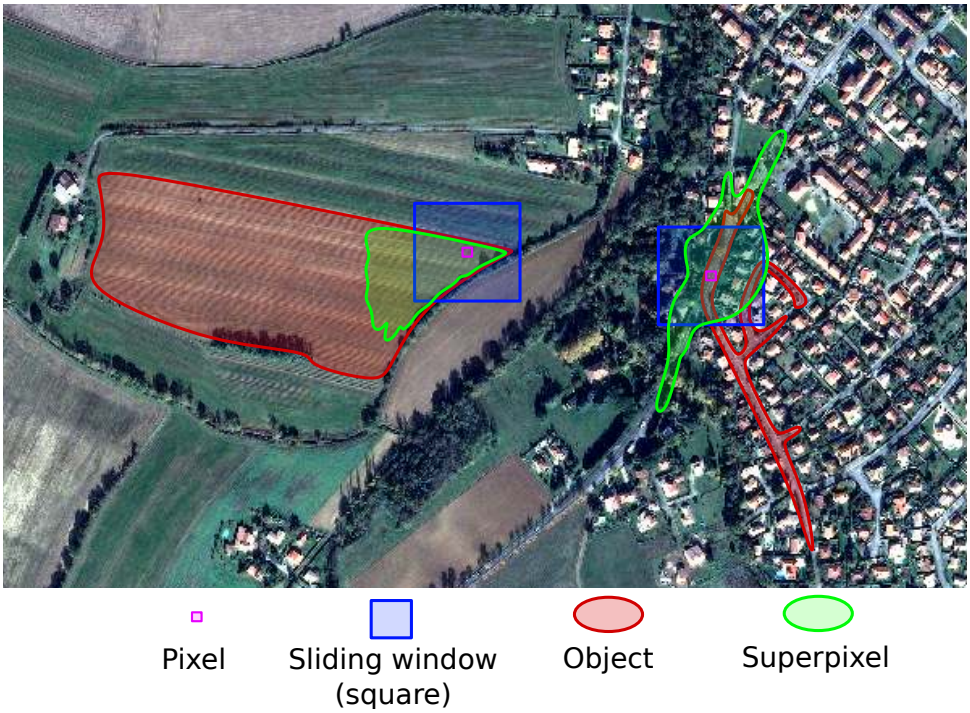


Figure 4.1: Illustration of the three neighborhood shapes: Sliding windows, Objects, and Superpixels over two areas with different contextual characterizations.

4.1 Sliding windows

As an intuitive definition of the context, one can consider that all of the pixels within a certain physical distance are a part of the context. Using the Euclidean (L2) distance, this brings a circular context shape, and using a Manhattan (L1) distance (on a grid of pixels), this brings a square context shape. This kind of spatial support is often called a *fixed-shape neighborhood* or a *sliding window*, as the context has the same shape for every pixel in the image.

The square spatial support is by far the most commonly used shape, for efficiency reasons. Indeed, extracting a square from an image is simple and fast to do, as the pixels already form a grid. The size of this square determines the scale of the context that is used. This is naturally adapted for gradients and local maxima, as they describe a local behavior on the smallest possible scale (3x3 pixels). However, at larger scales, the diversity of information present in a square neighborhood rapidly increases, and is not always relevant for describing the central pixel.

Model based approaches almost always use square spatial supports, as they require consistent information on each pixel to be given to each input. This means that the input must have the same number of features, wich requires a fixed-shape neighborhood. Moreover, the features should be comparable and describe the same aspect of the samples. This is coherent with the use of sliding window neighborhoods. Two model based approaches will be discussed here, as they use a square spatial support as a base for their definition of a context.

One of these methods is known as Markov Random Fields (MRF) [Melgani and Serpico, 2003, Schindler, 2012, Moser and Serpico, 2013, Kang et al., 2014] or Conditional Random Fields (CRF) [Zhao et al., 2016]. The objective of these methods is to extract a model of the probabilities of the various pixel combinations, which allows for local dependencies, particularly textures, to be taken into account. However, these methods make strong assumptions on the probability density functions, which are difficult to extend to very high dimensional spaces. Random Fields are also difficult to scale, as a global energy function must be recalculated on the entire image at each iteration, at least with the most common solvers. More details on the subject of Random Fields are given in Part III, in Section 8.1.

Another approach involves structuring certain well known classification methods called Neural Networks, invented in 1943 [McCulloch and Pitts, 1943]. The Neocognitron, ancestor of the modern Deep Convolutional Neural Networks was created in 1980 [Fukushima et al., 1983], and was based on our understanding of the visual perception system. However, this research was all but abandoned, as the computational power at the time was insufficient for this to reach practical applications. In recent years, these models have been largely reconsidered, and achieve state of the art performance in many computer vision problems.

Originally used for classifying small patches of images, Convolutional Neural Networks (CNNs) have been

extended to dense classification problems on larger images on topics such as land cover. More details, as well as the advantages and drawbacks of these methods are discussed in Part III, Chapter 9.

Finally, square spatial supports are most often used to calculate contextual features, also known as contextual descriptors. These features can range from basic local image statistics such as mean and variance to more complex descriptors such as the Haralick texture features [Haralick, 1979], wavelet based textures [Huang et al., 2008], or Gaussian Pyramid features [Binaghi et al., 2003]. Details on these features are given in Chapter 5.

The geometric degradations that are expectable when using a sliding window as spatial support can come under two forms.

First of all, it has a tendency to smooth out sharp corners and to erase fine elements in the image, as its shape is not adaptive to the content of the image. Pixels belonging to sharp corners or fine elements are mainly surrounded by pixels from very different classes. For example, a pixel belonging to sharp corner like the pixel in the corner of the field in Figure 4.1, is partially surrounded by elements belonging to neighboring fields. As these pixels are all included in the spatial support, their contribution to the contextual information may overshadow the contribution from the target class, which might lead to a misclassification. Sharp corners and fine elements do not contain many pixels, but they are in fact very important, as they define the fine details of the geometry of the classification map, which provides a visually and geographically accurate result for the end user. Another way of understanding this smoothing phenomenon is to consider the spatial frequencies present in the image. Sharp corners and fine elements are linked to the high spatial frequencies of the image, and may therefore be sensitive to low-pass filtering, for instance, by using a sliding window with an isotropic feature.

Secondly, there can be an effect where the square shape of the neighborhood appears in the aspect of the certain features, under the form of marked edges where none appear in the image. This is visible in figure 5.1a.

4.2 Objects from an image segmentation

Due to the smoothing of high spatial frequency areas, other methods attempt to define a spatial support that is adaptive to the strong gradients in the image. In other words, the shape of the neighborhood is adapted to fit with the edges of the object that contains the pixel. The underlying idea is to preserve the geometry of the objects in the image, by considering that a spatial support should be formed by pixels that are not only nearby in the image, but also similar in terms of features.

Including context in an adaptive neighborhood allows groups of similar pixels in the image to exhibit identical features, which increases their chance of receiving the same class label. This translates the idea that pixels that are nearby in the image, and are similar to each other probably belong to the same class. This idea is the base of Object Based Image Analysis [Blaschke, 2010], a group of methods in which *objects*, also referred to as *object segments*, are used as a base unit for classification rather than pixels.

4.2.1 Object Based Image Analysis (OBIA)

A paradigm for including contextual information known as Object Based Image Analysis (OBIA) [Blaschke, 2010] has seen several practical applications in remote sensing classification problems. This method makes use of image segmentation, which is the process of splitting an image in non-overlapping regions, called segments, that attempt to optimize feature homogeneity and sometimes a shape criterion. Segments from a segmentation method like the ones commonly used in OBIA, such as Mean Shift [Comaniciu and Meer, 2002] and Region Merging [Baatz, 2000], are called object segments, object neighborhoods, or objects, in reference to Object Based Image Analysis, and in contrast with superpixel segments that are defined in Section 4.3.

In most OBIA approaches, object segments serve as spatial supports for calculating contextual features. If a hierarchy of segmentations is used, such methods can also include information from several scales, as is done in [Bruzzone and Carlin, 2006]. Furthermore, most OBIA methods make use of the spatial characteristics of the segments, i.e. their shape, size, perimeter, and other such descriptors. These features, described in more detail in Section 5.5, add a level of spatial information that can help in describing the context. This is shown to have a positive impact on the classification accuracy on various remote sensing problems [Walker and Blaschke, 2008, d'Oleire Oltmanns et al., 2014].

However, such methods may have difficulty in characterizing highly textured areas, due to over-segmentation. Indeed, most common image segmentation algorithms generate segments that adhere to the strong gradients in the image, but do not necessarily include diverse pixels, as the primary objective of these methods is to maximize feature homogeneity in the segments. An illustration of a Mean Shift segmentation on a Sentinel-2 image, given in figure 4.4b. The various spatial support shapes, figure 4.1 show that in urban areas, object segmentation methods often isolate individual houses, streets and gardens, rather than groups of buildings, due to the strong local gradients

in these areas. However, the relevant contextual information is not contained in these segments because it is the spatial arrangement of the streets, houses and gardens that truly characterizes the urban density. Generally speaking, it is the diversity of pixels and their spatial arrangement that provides a meaningful characterization of the context.

The true challenge in defining a spatial support is finding the "right amount" of context. If the context is too large, very fine details will be blurred out by the mixed characteristics of the many neighboring objects. Inversely, if the context is too small, the classification nears a pixel-based approach.

In fact, there is usually a trade-off to be made between the adaptability of a spatial support and its ability to include diverse pixels. Sliding windows can be placed at one end of the spectrum, as they allow the inclusion of diverse pixels but are not at all adaptive to strong gradients in the image. On the other end are segments from an object segmentation method, which are very adaptive, but do not allow the inclusion of diverse pixels. Figure 4.4 shows an example of the Mean Shift algorithm applied to Sentinel-2 imagery, and illustrates the oversegmentation phenomenon that occurs in urban areas.

4.2.2 Mean Shift segmentation algorithm

The Mean Shift segmentation algorithm [Comaniciu and Meer, 2002] is based on the *mean shift* process, which is a non-parametric iterative mode-seeking technique. The objective of this process is to locate the local maxima of the density function of a data set, which are known as the *modes* of the distribution. These modes represent common behaviour patterns that are found within a data set.

Figure 4.2 shows an illustration of a synthetic *bi-modal* data set, superimposed with the density function. The objective of a mode-seeking technique like mean shift is to automatically determine the location of these modes in the feature space. The term *non-parametric* refers to the fact that the number of modes is automatically determined by the algorithm.

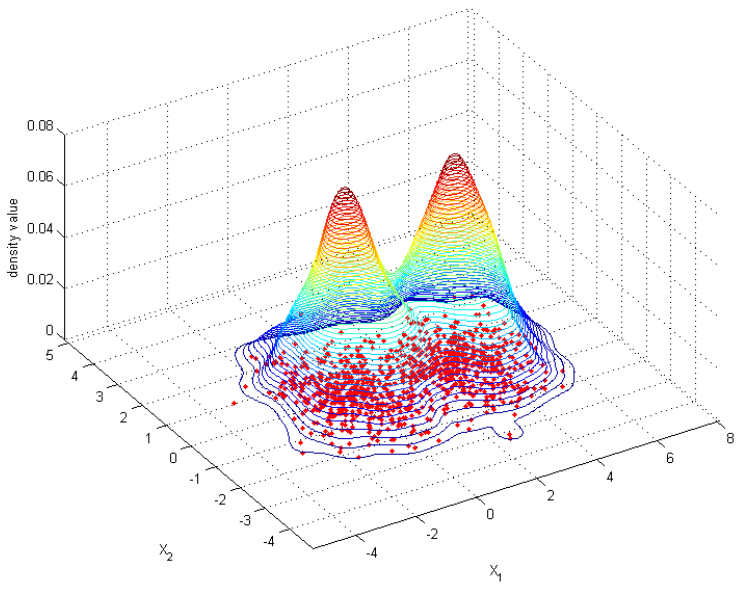


Figure 4.2: Example of an artificial bi-modal data set. The colored surface represents the density function, which is a mixture of two normal distributions. The modes are the mean values of these two distributions.

The mean shift process operates in an iterative manner, following these steps, for each point in the data set:

1. Find other similar points in the data set, within a certain predefined distance.
2. Calculate the mean of these points.
3. Repeat steps 1 and 2 using the mean, and update the value of this mean until convergence.

Step 1 requires defining a distance, which in many cases is taken as the Euclidean (L2) distance on the feature set. Step 2 requires calculating a mean, which is often taken to be the sample arithmetic mean of the features. Finally, convergence is reached when the distance between successive values of the mean falls beneath a certain threshold.

When applying such a process to images, it is most interesting to operate in a feature space that contains both the image features (pixel values) and the positions of the pixels (X, Y) in the image. This feature space is known as the *joint domain*. Each mode therefore represents both a set of features (a color for example), as well as a position in the image, where the density function reaches a local maximum. These modes can be linked to the different objects present in an image. In other words, the features of the mode represent the typical behavior (color, time series, etc.) of an object, while the position of the mode can be seen as the centroid of this object.

The mean shift algorithm is mainly dependent on two parameters: the *spatial* and the *spectral* radii, which control the size of the windows in the joint domain.

Figure 4.3 shows a schematic of the steps of the mean shift process applied to one pixel of an image.

The mean shift process associates a pixel with its nearest mode, in other words, the local density maximum in the joint space (features and locations). In order to achieve a segmentation, this process is applied to each pixel in the image. Then, all of the pixels that converge towards an identical (or almost identical mode) are grouped together to form a segment.

The image in figure 4.4b shows a Mean Shift segmentation applied to a time series of Sentinel-2 images. The advantages are that the number of segments are automatically determined, sharp gradients are well respected, and the objects are overall very homogeneous. The issue, as was stated before, lies in the over-segmentation of textured areas, which contain a large amount of local density maxima that the mean shift process is not able to group together.

4.3 Superpixels

There is another type of segmentation, known as superpixel segmentation, which aims to extract spectrally homogeneous regions, but that includes another constraint: the segments should all have similar sizes, and should be equally distributed throughout the image [Achanta et al., 2012].

Superpixels can be seen as an intermediate representation between sliding windows and object segments, because they are adaptive to strong gradients, but include a variety of different pixels in highly textured areas, due to the size constraints. Another interesting property is that when using superpixels, all of the segments have a similar size, which means that the contextual features at each scale are comparable to each other, in terms of the extent of the scale that they have considered. Feature comparability is a desirable property for classification.

There are different ways of obtaining a superpixel segmentation of an image. With the right parameters, some well known segmentation methods such as Watershed [Vincent and Soille, 1991], Mean Shift or Region Merging can generate superpixels. However, these provide fewer guarantees and less control on the superpixel parameters than custom-made superpixel methods.

There are two main families of superpixel methods: Graph based, and Gradient-ascent based. Graph based methods consider each pixel of the image as a node in a graph. Neighboring nodes are connected by a vertex, the weight of which depends on the similarity between the pixels. By minimizing a global objective function over the image, superpixels can be created. Gradient-ascent based methods, on the other hand, start with a poor initial segmentation, usually a regular square grid, and then relabel the pixels in successive steps to optimize an objective function.

An example of gradient-ascent based methods are the lattice methods, in which one considers the segmentation as a lattice (horizontal and vertical lines) which is then iteratively adapted to the image gradients, [Moore et al., 2008, Chai et al., 2017].

The most popular superpixel segmentation method, introduced by [Achanta et al., 2012], is called Simple Linear Iterative Clustering (SLIC), and belongs to the group of gradient-ascent based methods. It provides better segment adherence to boundaries when compared to other state of the art superpixel methods. SLIC is also chosen among other superpixel algorithms, because of the execution speed of the algorithm, even in very large dimensional spaces, which is necessary in order to deal with the volume of multi-spectral time series data.

Figure 4.4a shows an extract of a SLIC segmentation applied to a Sentinel-2 image time series, containing 33 images with 10 spectral bands each, covering the entire year of 2016. Cloud detection and gap-filling are applied to the image stack, as is done in [Inglada et al., 2017]. The segmentation is applied on all 33 dates, but the background image shown in Figure 4.4 shows the RGB bands of the first date. The natural boundaries between objects are relatively well respected, and the segments are indeed all compact and similar in size. Figure 4.4b also shows a mean shift segmentation of the same area, with a spatial radius of 6, and a spectral radius of 500. While the fields are not over-segmented, the segments in the urban area are very small and do not include spectrally diverse pixels. This shows that object-based methods have difficulty segmenting out semantically relevant objects in textured areas.

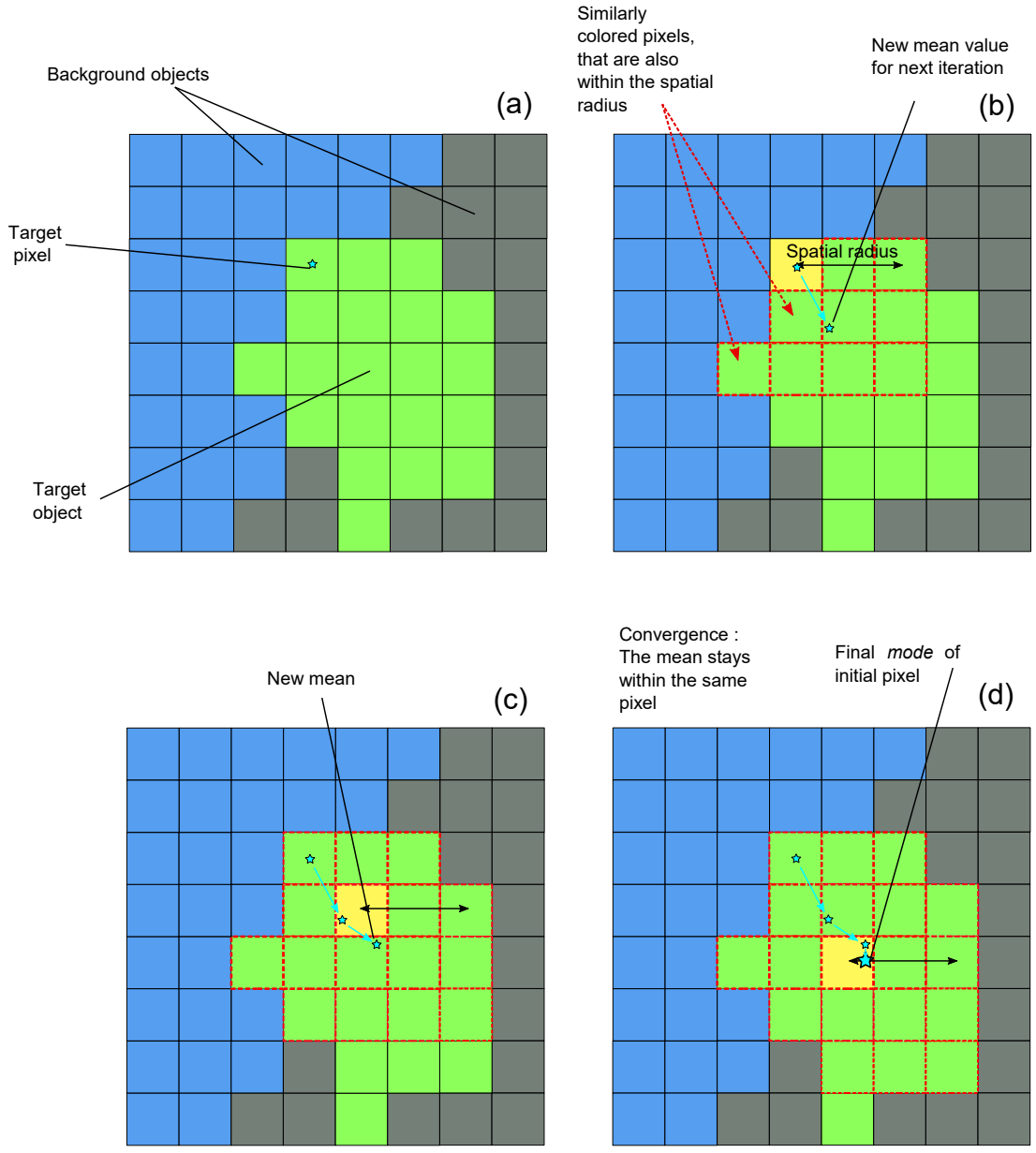


Figure 4.3: Steps of the mean shift process, applied to a grid of pixels. (a) The target pixel is marked with a blue star. The objective is to calculate the *mode* of this pixel, contained within the green object. (b) All of the pixels within a spatial radius of 3 pixels, and that are also similar in terms of features (green) are averaged, to calculate a new mean position. (c) This process is repeated, as long as the new mean value is in a different pixel in two successive iterations. (d) The final mode of the initial pixel is found.

SLIC is an algorithm similar to the K-Means clustering algorithm, and its application to images is relatively straightforward.

Let I be an image of dimensions $X \times Y$ with N pixels , in which each pixel p_n at position (x_n, y_n) contains D features called $f_n^d, n = [1..N], d = [1..D]$. For example, in a remote sensing image, the spectral bands as well as any number of derived indices (NDVI, etc.) can be used as features.

Each segment is represented by a centroid $C_k = [X_k, Y_k, F_k^d], k = [1..K], d = [1..D]$. The couple (X_k, Y_k)

(a) SLIC superpixels. In the textured area, superpixels group together diverse pixels, and are therefore able to describe the density of the urban cover.

(b) Mean-Shift segmentation. The urban fabric in the center is prone to over-segmentation, due to the high spectral variability in the area. These small segments are unable to correctly describe the context.

Figure 4.4: Different segmentations of a Sentinel-2 image time series, on a discontinuous urban fabric. Background: RGB bands of the first date. All of the spectral bands and dates of the yearly time series are used to obtain these segmentations.

represents the spatial mean (geometric center of gravity), while the F_k^d are the mean features on the segment k.

At the initialization step, the image is split into K square segments following a regular grid. The initial grid size (length of the side of the square grid elements) is a parameter of the algorithm, called the spatial width parameter. This very important parameter defines the initial size of the superpixels, and therefore the total number of superpixels. The algorithm then proceeds as follows. During the iterative step, each pixel is associated to its nearest centroid, following a similar principle as in K-means. The distance in this step is calculated as a weighted sum of the spatial and feature Euclidean distances, shown in equation (4.1), which is similar to the definition in [Hsu and Ding, 2013]. The distance weight parameter, DW allows the user to give more importance to either segment compactness or segment homogeneity.

$$\begin{aligned} dist(p_n, C_k) &= dist_{feature} + DW \times dist_{spatial} \\ &= \sqrt{\sum_{d=1}^{D} \left(F_k^d - f_n^d\right)^2} + \\ & DW \sqrt{\left((X_k - x_n)^2 + (Y_k - y_n)^2\right)} \end{aligned} \tag{4.1}$$

In the original algorithm, the candidate centroids are chosen in a limited search radius, to avoid calculating unnecessary distances [Achanta et al., 2012]. Once all the pixels are updated, the centroids are recalculated, and the next iterative step is started. Iterations stop when a convergence criterion is met: either a maximal number of iterations, or a total residual (average square distance between the centroids of two consecutive steps) smaller than a given value.

The most significant parameters of the algorithm are defined as follows.

- The spatial width, abbreviated as SpW, is the length of the side of each segment in the initial square grid. This parameter controls the total number of superpixels and their average size. Figure 4.5 shows an illustration of the impact of this parameter on the SLIC segmentation, on a VHSR SPOT6/7 image over the city of Brest.
- The distance weight, DW, is the weight of the spatial distance with respect to the feature distance. The combined distance is given in equation (4.1).
- The search radius, SR gives the search radius for candidate centroids.

SLIC, like K-Means applied to an image, does not necessarily generate connected segments. This is due to the composite distance criteria: a pixel can be similar in terms of features to a centroid but spatially distant, meaning that other pixels can isolate a pixel from its centroid. The authors of [Achanta et al., 2012] propose a post processing

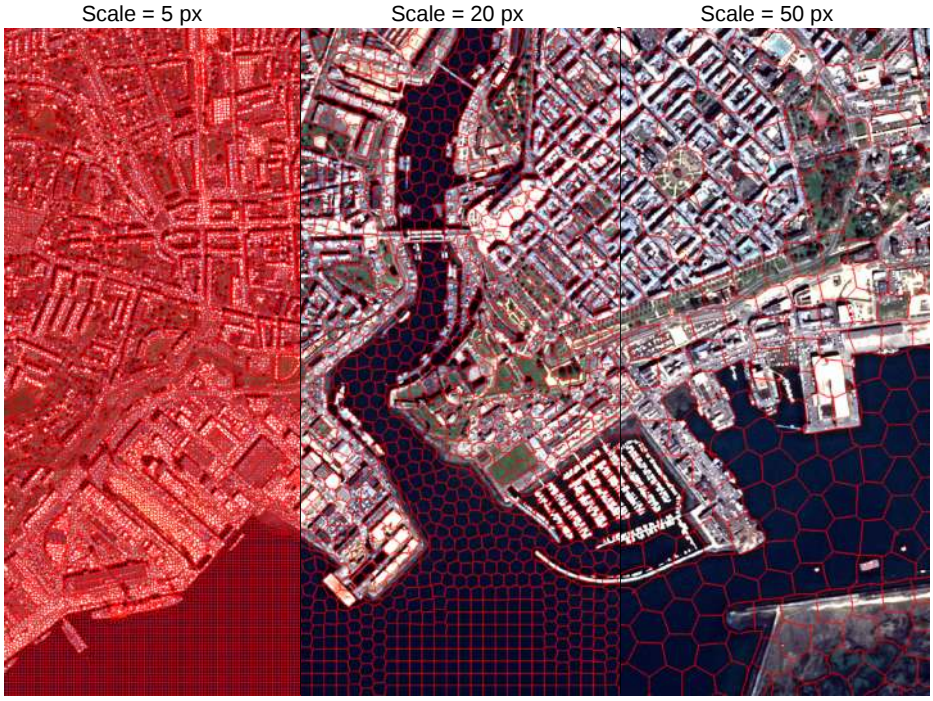


Figure 4.5: SLIC segmentation over Brest. The spatial width parameter offers control on the average size of the superpixels, which allows for a contextual description at a fixed scale. The illustration shows 3 scales of superpixel: 5, 20, and 50 pixels. These indicate the size of the initial grid used by the SLIC algorithm. In the segmentations, each superpixel contains in average 25, 400, and 2500 pixels.

step which involves removing the isolated regions by fusing them with a neighboring segment, based on the same composite distance. A region is considered isolated if the centroid of its segment does not belong to the region itself. Simply put, this post-processing step will keep only the regions that contain the centroids. Such a post-processing step has two downsides:

1. If the segment centroid happens to be outside the segment, for instance in the case of a crescent shaped segment, it will be incorrectly fused with a neighbor.

2. It is costly, as it requires the isolated regions to be found, and distances with their neighbors to be calculated.

In exploring the application of SLIC to videos, [Chang et al., 2013] expose a way to overcome the issue of the post-processing step. The solution is based on the concept of simple points, introduced in [Bertrand, 1994] and more recently in [Han et al., 2003]. A point is characterized as simple if changing its label cannot alter the topology of its 4-neighboring segments. For instance, a pixel in the middle of a segment is not simple, because if its label is changed, a hole will appear in the segment. On the other hand, a pixel on a straight edge between two segments will be simple, because it can be safely changed without altering the segments' topology. Calculating if a point is simple can be done in constant time, as it depends only on the labels in the 8-neighborhood, as shown in [Han et al., 2003]. Figure 4.6 shows the difference between simple and non-simple points.

In SLIC, the initial segments are fully connected, so if only the simple points are changed, this connectivity will be insured until the end of the algorithm. The other concept introduced by [Chang et al., 2013] is the log-likelihood of the segmentation. This evaluates the quality of a Gaussian fit on the data, both in the feature and the spatial domains. The log likelihood is used to find the best neighbor for a label change during the algorithm.

In the algorithm proposed by [Chang et al., 2013], the iterative step is modified in the following way. For each pixel:

1. Check if simple;

2. If simple, find which 4-neighbor fusion brings the highest change in log likelihood;

3. Assign label of that neighbor;

4. In case of change, update centroids of the two segments.

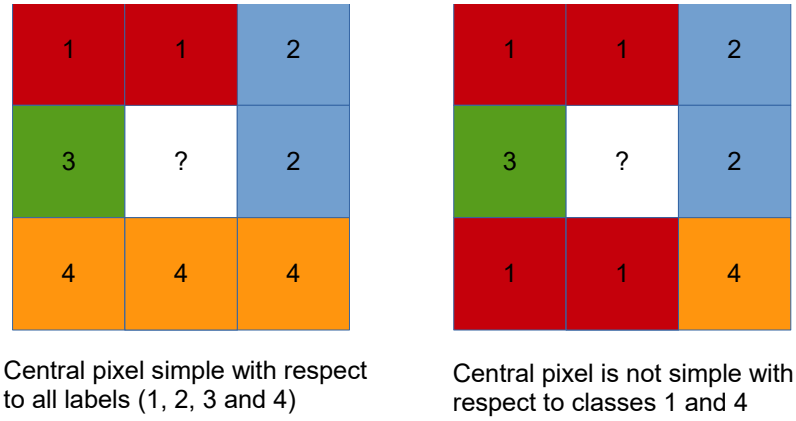


Figure 4.6: Simple points can be changed without altering the topology of the neighboring segments.

Following the logic of the original SLIC algorithm from [Achanta et al., 2012], the proposed modifications involve associating the pixel to the nearest centroid in terms of composite distance in the 4-neighborhood, instead of calculating the rather costly log-likelihood which contains both variance and mean terms.

Therefore, in the modified algorithm proposed here, the iterative step becomes, for each pixel:

1. Check if simple;
2. If simple, find which of the 4-neighbor's centroid is the nearest in terms of composite distance;
3. Assign label of that neighbor;
4. In case of change, update centroids of the two segments.

Compared to the original SLIC algorithm, the modified algorithm has the following advantages.

1. Fewer distances need to be calculated.
   - Distances are only calculated if the pixel is a simple point.
   - The original SLIC algorithm already restricts the nearest centroid search space by only calculating distances to centroids in a limited search radius. In the modified algorithm, the search space is reduced even further, to the 4 neighboring segment centroids.
2. The search radius for candidate centroids parameter SR, is no longer needed.
3. No post-processing step is required.

The issue of how to apply this method to large images is discussed in detail in Part III, as it is an original contribution of this Ph.D.

Superpixels provide interesting candidate spatial supports for the evaluation of contextual features in land cover mapping, however, they can only capture one scale of information at a time. The following section introduces multi-scale representations, and what shapes and spatial supports they can use.

4.4 Multi-scale representations

In natural images, and in Remote Sensing images alike, the context of a pixel can be relevant at several different *scales*.

In this situation, the term *scale* refers to the overall size of the context that is considered. This is not to be confused with the cartographic scale, which is the ratio between two points in a map and the actual distance between these points on the Earth's surface. Although this definition of cartographic scale does not make sense in digital maps, some insist on using it; the MMU is a more appropriate concept for digital vector maps, while spatial resolution is preferred for raster maps.

At a local scale, gradients and local maxima describe texture, and the precise contours of objects. Such features provide information on the local variability, which tells us if the pixel is in a smooth or textured area, if it is near

the edge of a strong gradient, and if it is very different from its close-range neighbors. Then, at a slightly larger scale, the average color and shape become apparent, as well as shadows, slow gradients, and small objects. At even larger scales, these shapes and shadows can assemble to form objects with complex geometries, sometimes containing several sub-objects with different characteristics.

For example, Figure 1.11, given on page 32 shows how different scales of context can be useful to describe a pixel belonging to a house, in an discontinuous urban area. The smallest scale, i.e. the pixel, describes the color of the roof, which already indicates that the pixel belongs to an impervious surface, like a building. However, with this scale of information alone, it is impossible to tell if this building belongs to a dense city center, a discontinuous suburban area, or an industrial and commercial unit. By including a larger scale of context, the texture of the roof can be characterized, which already helps in telling apart residential buildings from many industrial units. With an even larger context, the vegetation, and the average distance between neighboring houses surrounding the building can be seen. This provides relevant information for distinguishing continuous from discontinuous urban areas.

The idea of a multi-scale spatial support is to integrate information from several scales of context at once. The objective behind this is to be able to differentiate between long-range information and short-range information, from the point of view of the classifier. Indeed, the context of a pixel can contain valuable information both near and far, and it is interesting to study how both of these sources can be combined in a fitting way for land cover mapping.

One simple way of creating a multi-scale description is through the use of sliding windows. Seeing as their size is directly parametrizable, it is not difficult to accumulate features calculated in windows of increasing sizes. This is sometimes called a pyramid, for instance, the cognitive pyramid of [Binaghi et al., 2003]. However, it seems quite obvious that these spatial supports might fall victim to the same issues as mono-scale sliding windows, which were mentioned in Section 4.1. As these supports are the most simple ones to compute, they will be evaluated in the experimental section, in Part IV.

The second option is based on objects from a so-called *hierarchical* segmentation algorithm. Such an algorithm divides the image into connected regions several times using different parameters. The set of parameters and method is chosen carefully so that the resulting segmentations follow a *hierarchy*. The lowest level of segmentation contains a local description of objects, in other words, a probable over-segmentation of the area, and the highest level contains much larger segments, which can therefore group together objects of different classes. Consider N segmentations S_i, $[i \in 1...N]$. Saying that they follow a hierarchy is equivalent to saying that each segmentation S_i is hierarchically *included* in S_{i+1}. This expresses the notion that lower-level segments are entirely included within a higher-level segment, so that they form a hierarchical tree of segments. Such a tree can be generated during the iterations of an algorithm, for instance, Region Merging [Baatz, 2000].

Such a hierarchical object segmentation was applied by [Bruzzone and Carlin, 2006] on a VHSR problem, which suggested that the use of several scales was almost always beneficial over using only one.

Seeing the theoretical value of the superpixel as a spatial support, it is interesting to try to extend it to a multi-scale description as well. First of all, in the same way as for sliding windows, it is simply possible to accumulate superpixel segmentations with different spatial width parameters. This will be called a *multi-scale superpixel* spatial support. The multi-scale superpixel can be seen as a pyramid that adapts to the shapes of the nearby objects in the image at all of its levels. Each pixel can therefore be described by features calculated not in one but all of the superpixels that contain it. This comes with two advantages:

1. The benefits of superpixels, namely their adaptivity to local gradients is conserved at all scales.

2. The notion of scale is conserved through the multi-scale description.

This second point is important in view of the classification step. Indeed, it is advisable that a given feature consistently describes a certain aspect of the pixel or its surroundings. This is one of the fundamental aspects of supervised classification, and is called *feature comparability*. This notion encompasses the idea that a given feature should have values that express the same notion regarding each of the samples in the training and testing data set. This translates the general idea that one can only compare what is comparable. In the case of hierarchical segmentation, the segments at a given level do not entirely express the same notions, as some of them can be very small and therefore express local relations, while others are very large and express long-range relations. On the other hand, with a multi-scale superpixel description, all of the segments at a given level have approximately the same scale, and therefore describe either a close-range or a long-range relation, which was the initial goal of using a multi-scale description.

However, pyramids and hierarchical descriptions also come with a downside: the information that is present in the local scales is also contained in the larger scales. This means that the largest scales express the context over a wide area, but not necessarily the context at a certain specific distance. A very strong pixel value, such as a missed cloud, can therefore corrupt all of the scales of context. Giving the classifier the ability to incorporate

both long-range and short-range information is the goal of a multi-scale description, but it could be interesting to differentiate between the two in the design of the spatial support. The idea is therefore to create a spatial support that only expresses the context at a certain distance of the pixel. However, the basic idea of taking a circular or square ring around the pixel is avoided, as there is a more interesting way of using the superpixel segmentation that was defined earlier. This idea is the based on the notion of *adjacency layer*, which emerges from graph theory.

Indeed, it is possible to consider an image segmentation as a graph, noted G, where each node n represents a segment, and the vertices express the fact that two nodes are adjacent in the image. The graph in Figure 4.7 shows a basic graph, and the successive adjacency layers of the node numbered 6.

The adjacency layer i of a node is noted $A^i(n)$. It represents the set of nodes that are at a minimal graph distance of exactly i nodes. The minimal graph distance, noted $d_G(n, m)$ is the length of the shortest path that connects two nodes n and m in a graph G.

$$A^i(n) = \{m \in G | d_G(n, m) = i\}$$

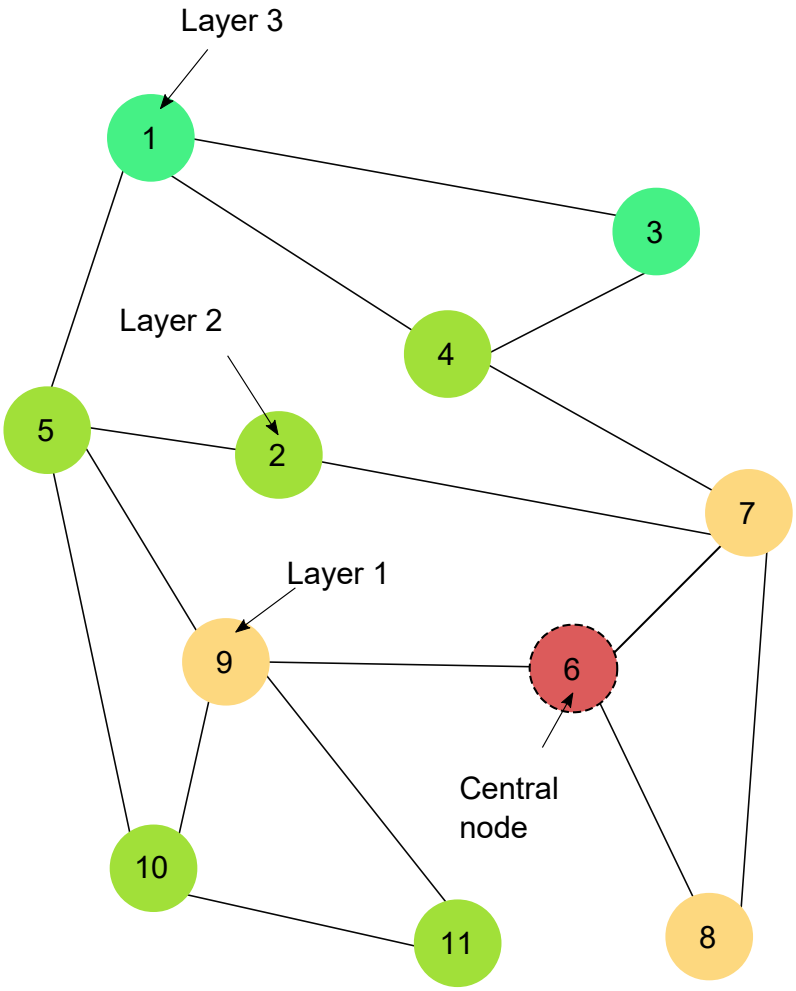


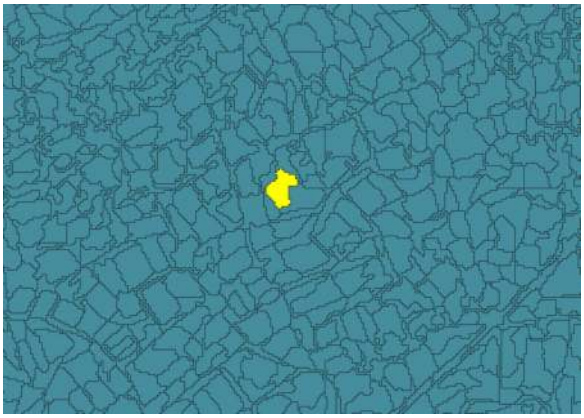
Figure 4.7: Example graph to illustrate the notion of adjacency layers. The central node is the node number 6. The color code indicates the layers. Its first layer contains the nodes 7, 8 and 9. The second layer contains the nodes 2, 4, 5, 10 and 11, and the third layer contains the nodes 1 and 3.

In theory this principle can be applied to any segmentation, but in this case it will be applied only to superpixel segmentations. Figure 4.8 shows the successive adjacency layers of a superpixel. Seeing as all of the superpixels have similar sizes, it makes sense that the adjacency layers draw out relatively circular shapes. This translates as two interesting properties for this spatial support, regarding its impact on the supervised classification:

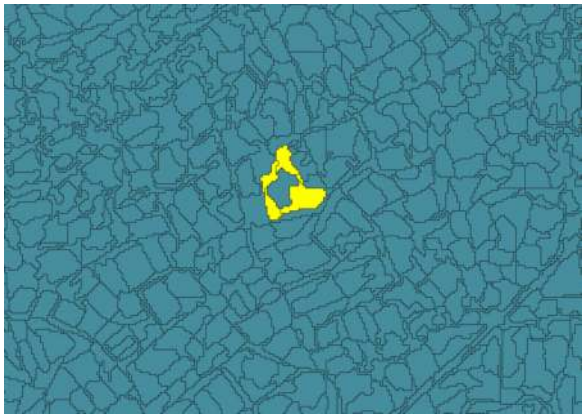
1. It conserves a kind of rotation invariance, which can be an interesting property for land cover classification, as the target classes generally do not depend on a particular orientation in the image.

2. The feature comparability is ensured, as each adjacency layer expresses an aspect of the context at a certain distance from the pixel.

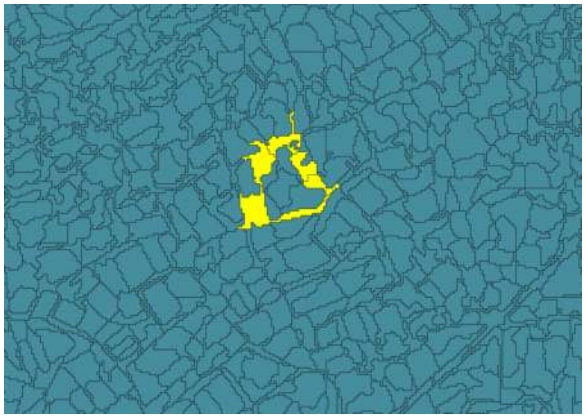
This second property would not be verified on a different type of segmentation, such as an object segmentation.



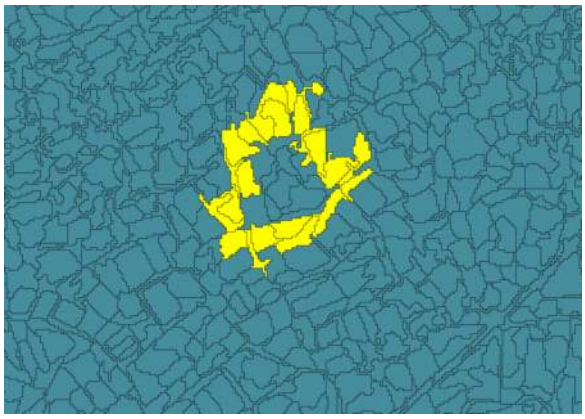
(a) Level 0.



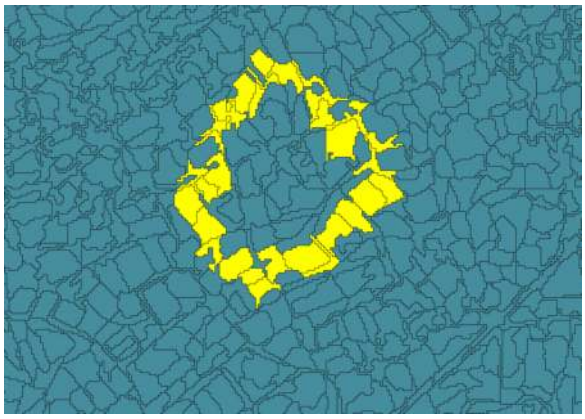
(b) Level 1.



(c) Level 2.



(d) Level 3.



(e) Level 4.

Figure 4.8: Successive adjacency layers ranging from 1-4 around a superpixel. Each adjacency layer is formed by the set of neighbors of the previous layer, going outwards. These spatial supports conserve a form of rotation invariance, and more importantly, provide a spatial support that expresses a notion at a certain scale, in this case, at a certain distance from the pixel.

These multi-scale spatial supports are evaluated as candidates for calculating contextual features in Part IV, and their impact on the geometry of the map is studied with particular attention.

4.5 Overview of the spatial supports

Defining the context is the first step in achieving a contextual classification. Indeed, if information from the neighboring pixels is to be used to classify a pixel, it must first be defined what exactly is considered a neighboring pixel. This section has shown that there are several definitions for this purpose, and has identified four of them that are suitable for evaluation on the two experimental data sets:

1. Sliding windows,
2. Objects,
3. Superpixels,
4. Adjacency layers of superpixels.

The next step involves describing the pixels and their spatial arrangement within in these spatial supports, which is the subject of much of the discussion in this Ph.D. Chapter 5 presents a review of possible methods for achieving this purpose, and discusses their applicability to high-dimensional imagery.

5 Contextual features

"The beauty of a living thing is not the atoms that go into it, but the way that the atoms are put together. The cosmos is also within us, we are made of starstuff. We are a way for the cosmos to know itself."

– Carl Sagan, Cosmos: A Personal Voyage, 1980

The task of describing the context of a pixel is challenging due to the sheer amount of information present in the many pixels surrounding every pixel in an image. In fact, due to the two dimensional nature of images, this number increases quadratically with the desired scale of context. With the high spatial resolution necessary to provide an appropriate description of the edges and sharp contours of the image content comes a significant challenge: designing and training a model capable of interpreting such a vast amount of data.

There are several ways to include context into a classification scheme, once a spatial support has been defined (Chapter 4). This is often done through the bias of *contextual features* or *contextual descriptors*, which seek to describe the content of the spatial support. They generally express notions like texture, the presence of salient elements, or contour shapes (elongation, perimeter, area). An extensive review of the various contextual features commonly used in Remote Sensing is provided in the following sections.

Most standard classification methods are well suited for interpreting features from different sources, in other words, dealing with multi-modal data. Combining contextual and pixel-based information is indeed multi-modal, as the contextual descriptors can be expressed in entirely different units than spectral reflectances or spectral indices. This can be done by using a Multi-Kernel SVM as is done in [Thanh Noi and Kappas, 2018, Cui et al., 2017], or with the Random Forest [Breiman, 2001] which is used for the OSO map, and which requires no form of feature normalization. Alternatively, any other simpler classifier combined with some form of normalization can be used.

The choice of which contextual features to use is not an easy one, because it depends on many factors, such as the target class nomenclature, the spatial resolution, the features in the original image (pixel features), the amount and quality of training samples, and the shape of the spatial support.

The subject of model-based approaches such as Convolutional Neural Networks (CNN) that attempt to learn these contextual features directly from the data is addressed in Part III. Here, the choice of handcrafted contextual features rather than a model-based approach comes from the desire of finding methods with low training and evaluation times, such as the Random Forest. Therefore, this section focuses on identifying a set of light-weight features, that have the potential to improve the context-dependent classes of the Sentinel-2 problem.

5.1 Isotropic features

Many contextual features seek to describe general aspects of the neighborhood. To do this, they average a certain feature over all of the pixels in the spatial support.

Depending on the shape of the spatial support, this causes rotation invariance, and to a lesser degree, translation invariance in the contextual features. Indeed, the particular orientation of the spatial support has no impact if the feature is averaged over a rotation invariant spatial support. This is the case for sliding window shapes with a central or nearly central symetry (circular, square). Moreover, segmentation based spatial supports (objects and superpixels) are rotation invariant, as their shape is already based on the layout of the pixels, the shape of the segments will be similar no matter the orientation of the image. For this reason, this group of features is named *isotropic features* here.

Among these isotropic features, low order statistics like sample mean and variance are often used as a first attempt, but other features also take into account the layout and structure of the pixels, and average certain properties, like gradients or local correlation, over the spatial support. The following sections detail these two groups of isotropic contextual features.

5.1.1 Local statistics: the sample mean and variance

The sample mean describes the average behavior of the pixels in a spatial support, while the variance gives a first order estimation of the variability of the pixels.

Consider that each spatial support, which can be a sliding window, superpixel, or object, contains N_k pixels, where k serves as an index for the spatial support. N_k is constant for sliding windows, but can vary for superpixels or objects. Each pixel also contains D features, which may represent for instance the spectral reflectances at different dates. Defining each pixel as p_n^d, with $n \in [1...N_k]$ and $d \in [1...D]$, the mean M_k^d and variance V_k^d of a spatial support k is shown respectively in equations 5.1 and 5.2. These features are often used because of their simplicity, but also because they can represent certain basic aspects of a context, such as typical behavior in the spatial support, as well as the overall pixel heterogeneity. Illustrations of these two features on the RGB bands of the first date of the Sentinel-2 time series are given in figure 5.1.

$$M_k^d = \frac{\sum_{n=1}^{N_k} p_n^d}{N_k} \tag{5.1}$$

$$V_k^d = \frac{\sum_{n=1}^{N_k} (p_n^d - M_k^d)^2}{N_k - 1} \tag{5.2}$$

Using the mean and variance is a good first approach for any problem, and is evaluated on the OSO land cover problem in Part IV, Chapter 10. These features have also been recenly used by [Yu et al., 2018] in sliding windows on Landsat-like high resolution images for land cover mapping.

5.1.2 Structured texture filters

One downside of the mean and the variance is that they are first order statistics, in the sense that they do not consider the structure of the pixels within the spatial support. The neighboring pixels could be totally jumbled up, and would still have the same mean and variance. By definition, variance only describes the variability of the area, which is not always linked to the texture. For this reason, structured texture filters are often based on local gradients, averaged over the spatial support. The following sections describe two structured textured filters that have been applied to remote sensing image analysis: edge density and spatial autocorrelation. These features, like mean and variance, are rotation invariant, given the right spatial support.

Edge density

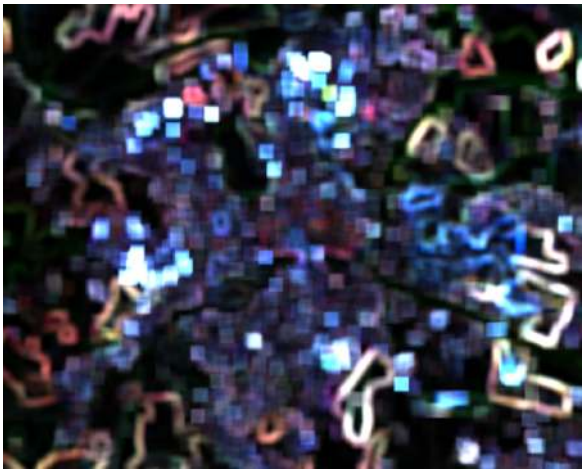
The edge density, as developed in [Trias Sanz, 2006], aims to quantify the amount and strength of local gradients. It is calculated as the average magnitude of the 3x3 gradient in a given neighborhood. For multi-variate data, the edge density is calculated separately for each band, providing a number of edge density features equal to the original number of features in the image. It provides a structured measurement of the local variations, giving an indication on the roughness of the surface texture, averaged over all directions. The 3x3 gradient is shown in figure 5.2a, and then figures 5.2b, 5.2d and 5.2c respectively give an illustration of the edge density feature calculated in a sliding window, a superpixel, and a Mean Shift object, on the RGB bands of the first date of the time series. The superpixel and object supports seem to provide a feature with relevant values, even near object edges.

Spatial autocorrelation

Another group of features is based on the measure of the *auto-correlation* of pixels, in other words, the probability that neighboring pixels bear similar values. Introduced by [Goodchild, 1986], Geary's index and Moran's index are high if neighboring elements are similar, negative if they are not, and near zero if the distribution is random. Figure 5.3 shows an illustration of how Moran's Index behaves in these three cases. Spatial autocorrelation features are often calculated as the average of the local correlation, between neighboring pixels.

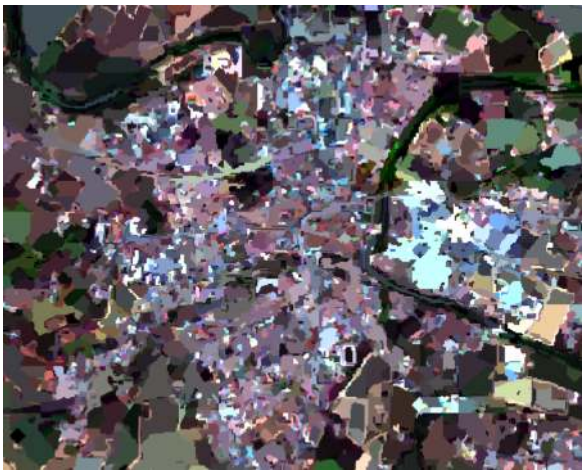
In Moran's Index, this local correlation is equal to $(p_i - M_k) \times (p_j - M_k)$, where M_k is the mean value over the entire spatial support as given in equation (5.1). Then the auto-correlation is defined as the average correlation

(a) Mean in a sliding window of 11x11.



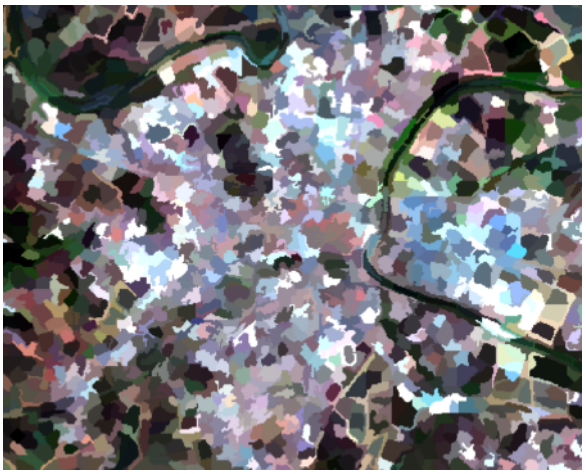
(b) Variance in a sliding window of 11x11.

(c) Mean feature in a Mean Shift object.



(d) Variance feature in a Mean Shift object.

(e) Mean feature in superpixel, average size of 100 pixels.

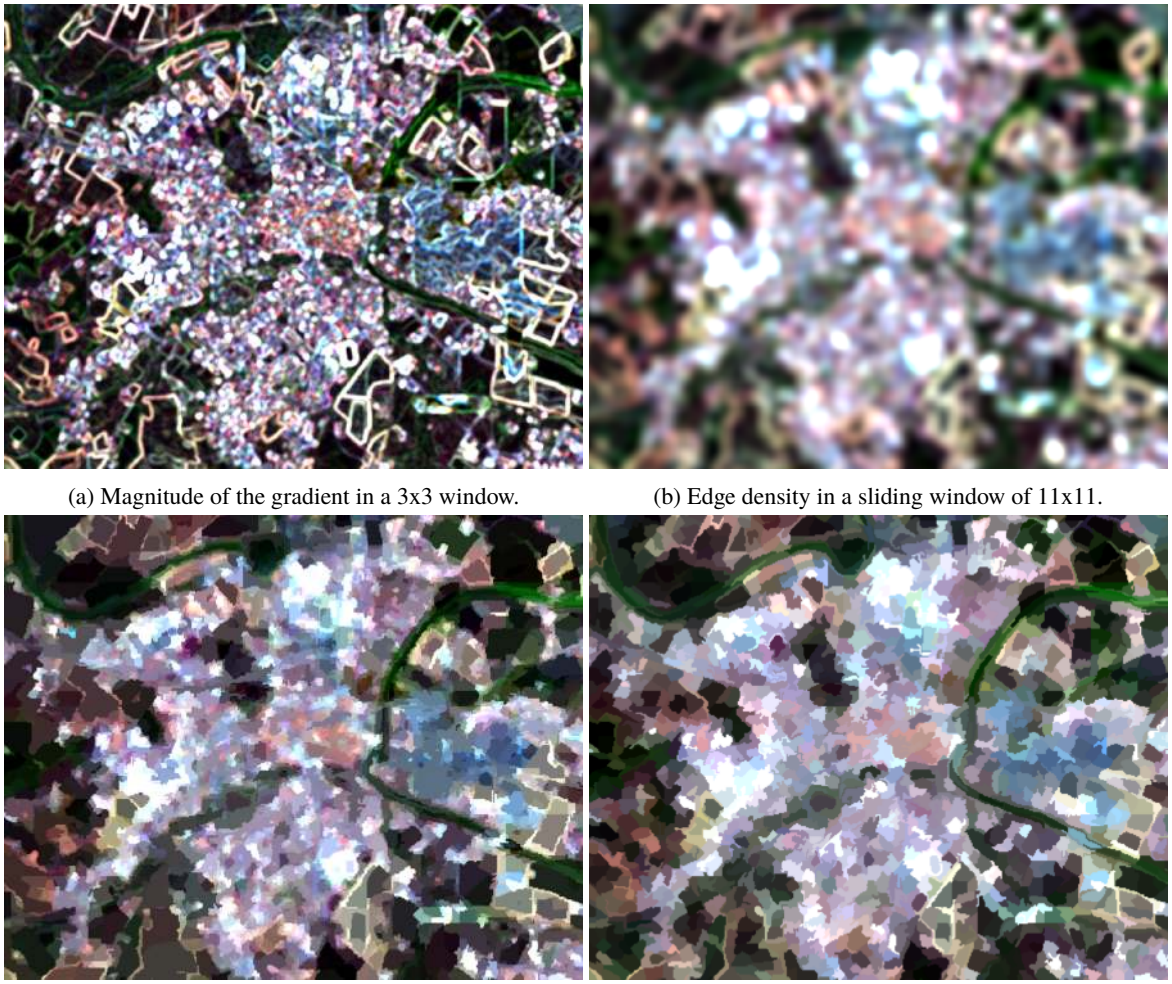


(f) Variance feature in a superpixel, average size of 100 pixels.

Figure 5.1: Example images of the mean and variance features on the RGB bands of the first date, calculated in the three spatial support types.

over all of the local neighborhoods centered on pixels within the spatial support. Taking the same notations as before, this can be written as in equation (5.3). Geary Index calculates the correlation slightly differently, as it takes the product of the pixel values $p_i^d p_j^d$ as a measure of the local correlation, equation (5.4).

In these equations, W_{ij} is an element of the adjacency matrix, which has a value of 1 if the pixels indexed i and j are adjacent, and 0 otherwise. The metric is also normalized by W, which is the sum of all of the values of



(a) Magnitude of the gradient in a 3x3 window.

(b) Edge density in a sliding window of 11x11.

(c) Edge density in a Mean Shift object.

(d) Edge density in a superpixel, average size of 100 pixels.

Figure 5.2: Illustration of the edge density feature in the three spatial support types.

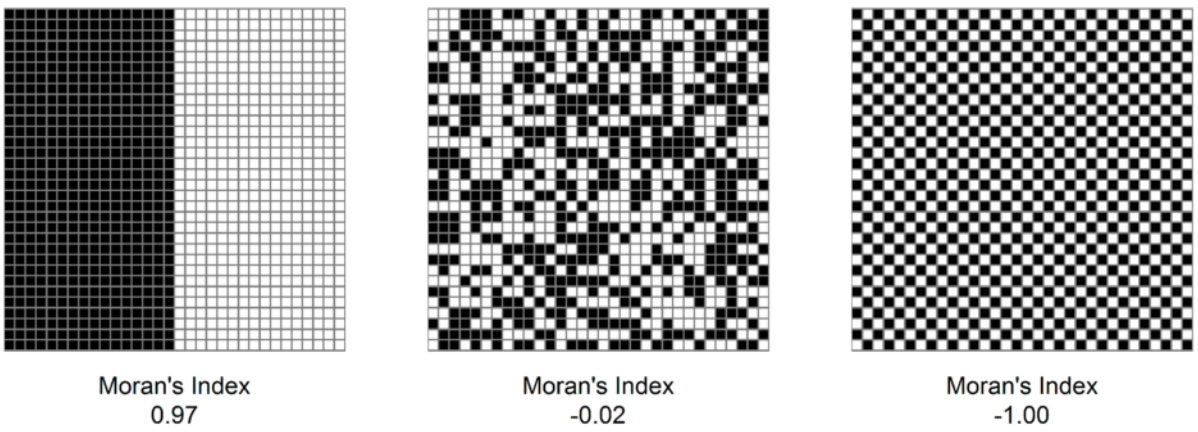


Figure 5.3: Illustration of Moran's index in three cases. In the first case, there is a strong amount of auto-correlation, in other words, neighboring pixels have a strong chance of being similar. In the second case, the pixels are randomly distributed, and the Moran's index is near zero. Finally, if there is a negative auto-correlation, neighboring pixels have strong chances of being different, which indicates a particular kind of texture.

the matrix, and V_k which is the variance within the spatial support.

$$Moran_k^d = \frac{\sum_{j=1}^{N_k} \sum_{i=1,i\neq j} (p_i^d - M_k^d)(p_j^d - M_k^d) W_{ij}}{S^2 W} \tag{5.3}$$

$$Geary_k^d = \frac{\sum_{j=1}^{N_k} \sum_{i=1,i\neq j} p_i^d p_j^d W_{ij}}{V_k^d W} \tag{5.4}$$

A review of these indices and their applications to characterize remote sensing images was made by [Wulder and Boots, 1998]. More recently, these have been used on radar images to detect landslide events by [Mondini, 2017], and to characterize crop types using hyperspectral images by [Zhang et al., 2016].

Figure 5.4 shows the application of the Moran's index on one date of the time series, over a discontinuous urban area, both in sliding windows and superpixels. The presence of buildings creates a close-knit texture, which translates as low autocorrelation values. The surrounding fields, which have little texture, therefore have high values of auto-correlation.

However, the added value of this feature compared to the simpler structured texture filters like edge density is questionable, as both of these features measure the average strength of the local gradients in a spatial support. In the preliminary experiments on very small areas, these features showed identical performance to the edge density feature, and were therefore not retained for further experiments.

5.2 Oriented texture filters

The next group of contextual features are oriented texture filters, which as their name suggests, are not rotation invariant. These filters characterize not only the strength or roughness of the texture, but also describe a directional aspect.

5.2.1 Describing oriented repeatability

The most commonly used oriented texture filters are Haralick's textures, which are based on the Grey Level Co-occurence Matrix (GLCM) [Haralick, 1979]. This matrix defines the distribution of the co-occuring pixel values at a given offset. An $n \times m$ area A with p different pixel values provides a $p \times p$ matrix for each offset $(\Delta x, \Delta y)$, in which the value at coordinates (i, j) gives the total number of pixels with value p_i in the unshifted area, and value p_j in the shifted area. This is expressed in equation (5.5).

$$C_{\Delta x,\Delta y}(i,j) = \sum_{x=1}^{n} \sum_{y=1}^{m} \begin{cases} 1, & if \quad A(x,y) = i \quad and \quad A(x+\Delta x, y+\Delta y) = j \\ 0, & otherwise \end{cases} \tag{5.5}$$

Haralick's textures involve calculating the GLCM for each pixel, using a square spatial support as the area A, for 8 offsets that surround the central pixel. This matrix has strong values if a repeated pattern is present at a given offset. To characterize the presence of texture, Haralick's original model involves calculating seven features from the co-occurence matrix: Uniformity, Entropy, Maximum probability, Contrast, Inverse difference moment, Correlation, and Probability of a run of maximal length [Haralick, 1979]. Recent studies have considered these features for characterizing the texture of radar images, to improve early crop type identification in conjunction with time series of optical images [Inglada et al., 2016, Demarez et al., 2019].

In essence, the underlying idea of many textural features is to detect the presence of repeated patterns, which express themselves as a strong spatial autocorrelation in a given direction, within the image. Mathematically speaking, high auto-correlations in a function are linked to the presence of certain specific spatial frequencies, defined by the Fourier transform of the function [Lim, 1990]. This is based on the principle that any function can be decomposed into an infinite series of base functions, so long as they verify certain properties. The most commonly used functions are the cos and sin functions, which when applied to images, decompose the two dimensional signal into magnitude and phase components. When a repeated pattern of a given frequency is present in the signal, the coefficients of the sinusiodal base functions at the corresponding frequency present a high value in magnitude.

In imaging terms, the infinite series translates as a convolution of the image, with each base function being an adapted convolution filter. Using appropriate filters, an image can be decomposed into its spatial frequency components. Rather than using the sin and cos functions, one can choose other base functions, known as *wavelets* due to their shapes, which ressemble small waves. With the two dimensional extension of the simple 1D Haar filters ϕ and ψ shown in figure 5.5, the image can be decomposed into its horizontal, vertical, and diagonal gradients, at a

given scale. The two dimensional filters are the four cross-products: $\phi(x)\phi(y)$, $\psi(x)\phi(y)$, $\phi(x)\psi(y)$ and $\psi(x)\psi(y)$. These have been previously applied to classify land cover using very high spatial resolution optical imagery [Huang et al., 2008, Yu et al., 2016].

Smoother wavelets seem more adapted to natural images, as the point spread functions of diffracted light are in general differentiable at higher orders. Using Gaussian distributions as base functions for the decomposition, Gaussian Pyramid features can be extracted, which are often used for image compression, and contextual classification of land cover mapping [Binaghi et al., 2003]. The Gabor filters [Turner, 1986], combine the Gaussian component to a sinusoidal component to create the convolution filters, and have been used for characterizing texture in natural images [Yang and Newsam, 2008]. A bank of such filters is illustrated in figure 5.6. The forumulas for obtaining the filter functions are given in equations (5.6) and (5.7). In these equations, A and B represent normalization factors that are to be determined according to the image. The coefficient σ can be varied to change the size of the neighboring area to be analyzed.

$$G_c[i,j] = Ae^{-\frac{(i^2+j^2)}{2\sigma^2}} \cos(2\pi f(i\cos\theta + j\sin\theta)) \tag{5.6}$$

$$G_s[i,j] = Be^{-\frac{(i^2+j^2)}{2\sigma^2}} \sin(2\pi f(i\cos\theta + j\sin\theta)) \tag{5.7}$$

5.2.2 Local binary patterns

Another oriented texture filter is the Local Binary Pattern (LBP), which was recently applied to hyperspectral image classification [Jia et al., 2017]. The LBP can be defined for every pixel in the image as the list of neighboring pixels thresholded by the value of the central pixel. In other words, it forms a list of 1s and 0s that indicate if a neighboring pixel is brighter or darker than the central pixel. This list is then converted into a binary number, as is shown in figure 5.7, which allows the feature to be quite compact. If this is combined directly with a supervised classifier, the feature is indeed not rotation invariant, as each feature describes the behaviour in a given direction with respect to the central pixel. This idea will be explored further in Part III, Section 8.1.

Overall, oriented texture filters, including the LBP, provide a very large number of descriptors for each band of the image, and in many cases are not useful for describing classes without a particular orientation in the image. In land cover mapping, the absolute orientation of a class is rarely relevant, apart for in mountaineous areas where the direction of the slope can have an influence on the vegetation that can grow. In practice, these cannot be used directly on each date of an optical time series of images.

5.3 Key-point based methods

Key-point based features aim to describe context by detecting and describing the high spatial resolution features, like sharp gradients and local extrema in the vicinity of a pixel. Features like the Scale Invariant Feature Transform (SIFT) [Lowe, 1999], the Speeded-Up Robust Feature (SURF) [Bay et al., 2006], were used by [Russell et al., 2006] in computer vision problems, for issues like object recognition and image matching, and found superior to Gabor Filters on VHSR image classification [Yang and Newsam, 2008]. This is achieved by extracting so-called keypoints, which are meant to characterize the points of interest in the image, and should help describe its content. This way, a pixel can be characterized by statistical information regarding the keypoints in its surroundings. For each keypoint, a normalized histogram of oriented gradients is calculated, which represents a rotation invariant shape and strength descriptor of the gradients that form the keypoint. On a side note, this can also be done in a dense manner (on every pixel in the image), and the resulting feature is known as the Histogram of Oriented Gradients (HOG), which also has found practical applications in computer vision [Dalal and Triggs, 2005, Larios et al., 2011]. The applications of these methods to remote sensing classification was reviewed by [Cheng et al., 2017].

More recently, this was improved upon by calculating the covariance of these gradients, around local extrema (maxima and minima), in an approach known as the Point-Wise Covariance of Oriented Gradients (PW-COG) [Pham et al., 2016].

Key-point based methods remain difficult to apply to time series of images, as they require a very large number of features to directly describe the magnitude and orientation of the gradients present in the surroundings.

5.4 Level set methods

Another popular contextual feature for the classification of VHSR imagery is the Extended Morphological Profile (EMP), which is based on a series of mathematical operations like closing and opening by reconstruction. This

feature describes the scale at which an area in the image is distinguishable from its neighborhood, and whether the area is lighter or darker than its surroundings [Benediktsson et al., 2005]. Some attributes also describe geometrical properties like elongation and squareness. The downside is the large number of features generated by the EMP, which makes dimensionality reduction necessary when the images present a large number of pixel features [Dalla Mura et al., 2010]. Moreover, these contextual features describe the size and shape of the object containing the target pixel, with respect to one layer of surrounding objects, but these near-range relationships are not always sufficient to describe the surrounding elements in a wider sense. Figure 5.8 shows an example of an EMP on SPOT6/7 imagery over the city of Brest.

There are also low-dimensional features, like the Shape Feature Set (SFS) [Huang et al., 2007], which describes the context of a pixel by constructing a set of segments at regular angular intervals, with one extremity on the central pixel. The segments are expanded out until a dissimilar pixel is encountered. This characterizes the shape of the object containing the pixel, as well as the position of the pixel in the object. Secondly, a histogram of the segment lengths is calculated, and 6 characteristics describing this histogram are used as contextual features. Figure 5.9 shows SFS features over the same area as Figure 5.8.

5.5 Shape features

One of the advantages of OBIA is being able to exploit the shape of the segments given by the segmentation [Blaschke, 2010]. One very common way of doing so is to include features such as the compactness, area, or the squareness. An more exhaustive list of possible shape features is given in [Van der Werff and Van der Meer, 2008]. Some of these features are used in the experimental section, to see if they bring any discrimination power to the OBIA approach, on the Sentinel-2 land cover mapping problem.

$$C_p = 4\pi \frac{a^2}{p} \tag{5.8}$$

The perimeter-based compactness, defined in equation (5.8) describes how close the perimeter to area ratio is to that of a circle. In the formula, a and p respectively designate the area and the perimeter. The area and the perimeter themselves can also be used as features. These three features provide information on whether or not the segment has a compact shape, and its overall size. This feature seems more pertinent for object supports than for superpixel supports, because the latter are similarly sized, and have a relatively compact shape. A more recent study by [Liu et al., 2014] suggests the use of a tree of shapes, to extract shape co-occurence patterns.

5.6 Overview

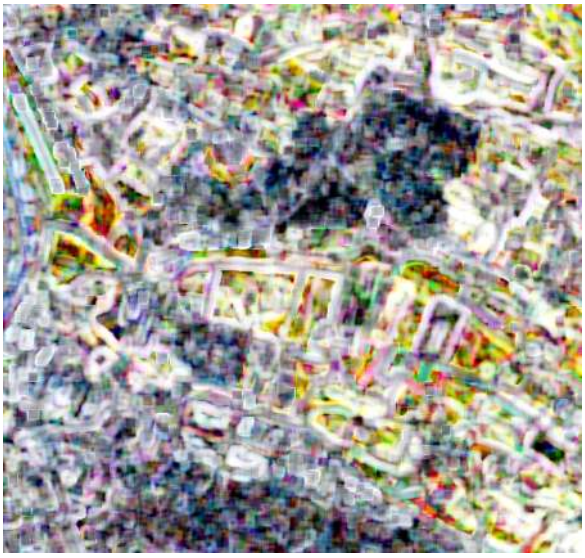
The previous sections have defined numerous contextual features that could potentially be used in land cover mapping. Before attempting more complex features, it is necessary to start with the basic local statisics, the mean and variance. Second of all, the edge density feature which describes unstructured local information is also retained for the experiments, as it describes aspects of the context that the mean and variance cannot. Moreover, it requires the same number of features as the initial image, making it applicable in at least one scale of information for the Sentinel-2 problem. The relative performance of these different features are shown in Part IV, in Chapter 10.

From the other sets of features that were discussed, one high-dimensional feature, the EMP, is also retained for the experiments on the SPOT-7 data set, whick make up Chapter 10, and will be compared amongst other methods to the mean, variance, and edge density.

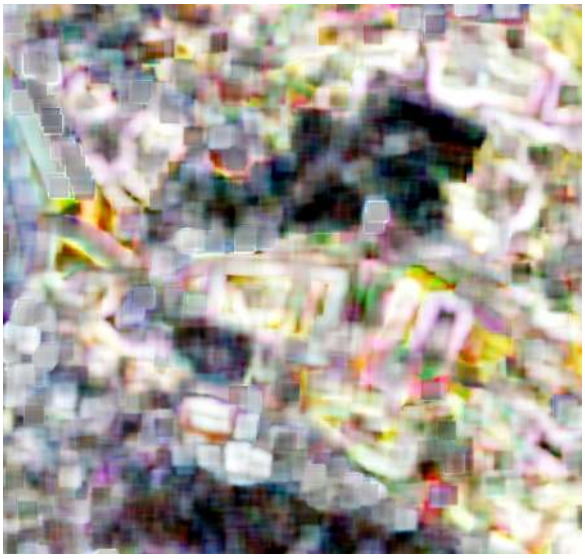
The other features that were mentioned were not retained for experimentation because they require a large number of features, and are therefore not suited for high-dimensional imagery (Haralick's textures, SIFT, SURF).

With this in mind, there is a group of contextual features that remains to be studied. These are methods that apply a transformation, in general a form of dimensional reduction, to the pixel values before calculating statistics upon them. The nature of this transformation and the choice of statistic features makes up the methodological contributions in Part III, Chapter 8.

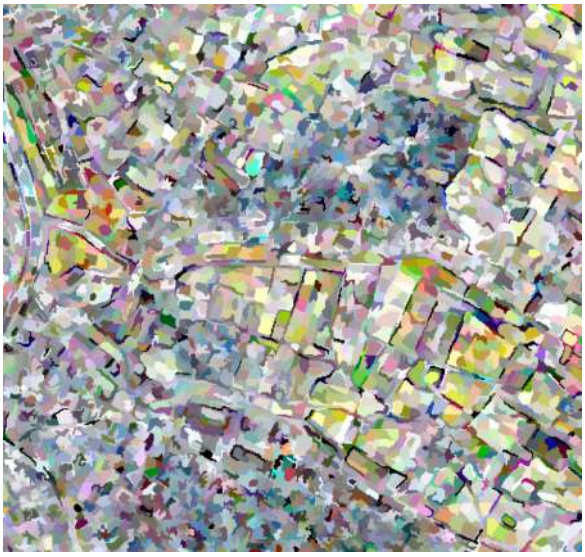
(a) RGB bands of the first date of the time series, over a discontinous urban area.



(b) Moran's index in a sliding window of 7x7.



(c) Moran's index in a sliding window of 15x15.



(d) Moran's index in a superpixel of average size 7x7.

(e) Moran's index in a superpixel of average size 15x15.

Figure 5.4: Examples of the spatial autocorrelation feature, Moran's index, in sliding windows and superpixels of varying sizes. The discontinous urban area, marked by a close-knit texture, shows low autocorrelation values compared to the surrounding fields which are relatively homogeneous.

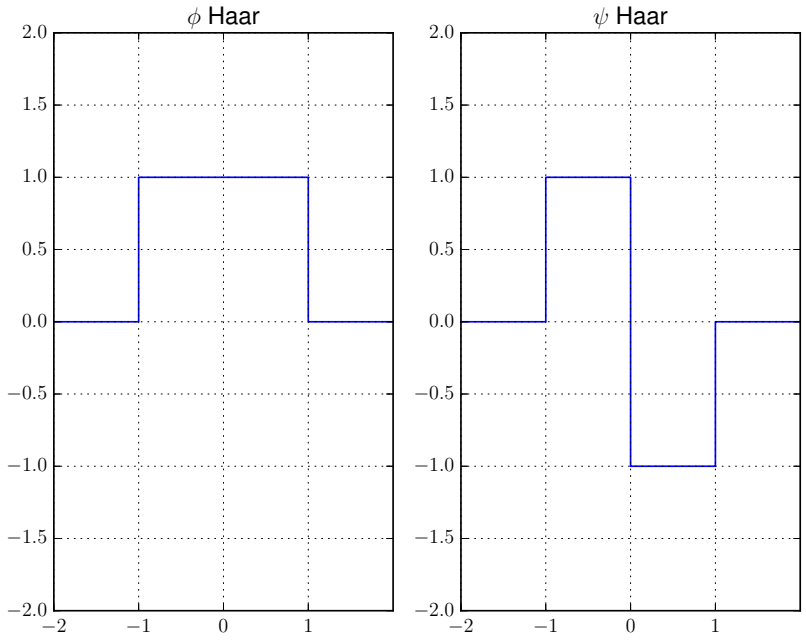


Figure 5.5: 1 dimensional Haar wavelets. ϕ is an averaging filter, whereas ψ translates a local gradient. The two dimensional filters are the four cross products of these two functions.

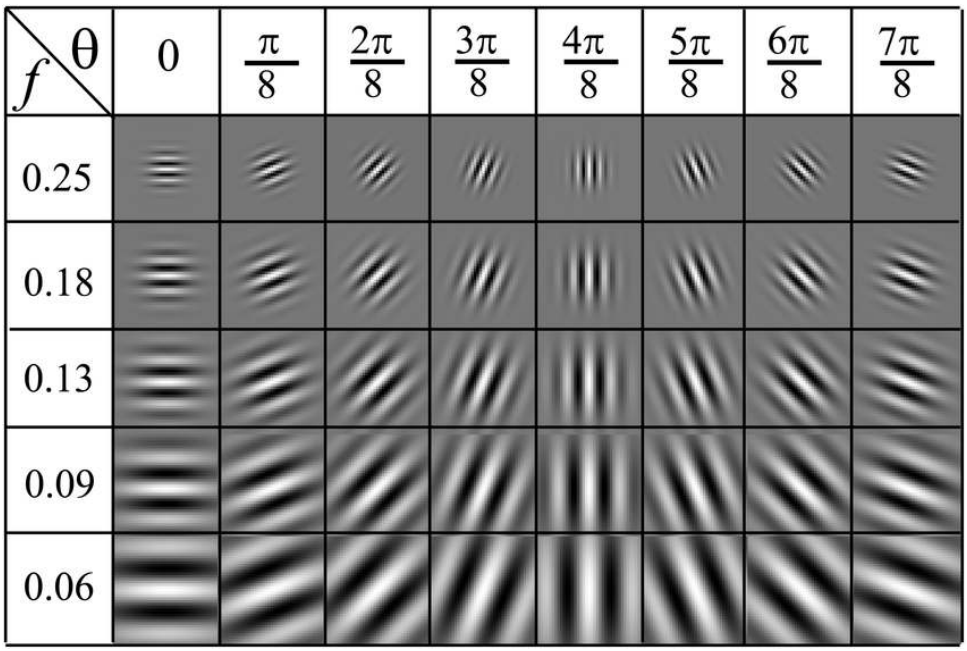


Figure 5.6: A bank of Gabor filters, commonly used in image analysis. Each filter is characterized by a spatial frequency f, and an orientation θ. This provides a description of the texture in different directions, and at different scales. The Gaussian component causes the edges of the higher frequency filters to be ignored. In this example, σ is taken inversely proportional to the frequency [Haghighat et al., 2015].

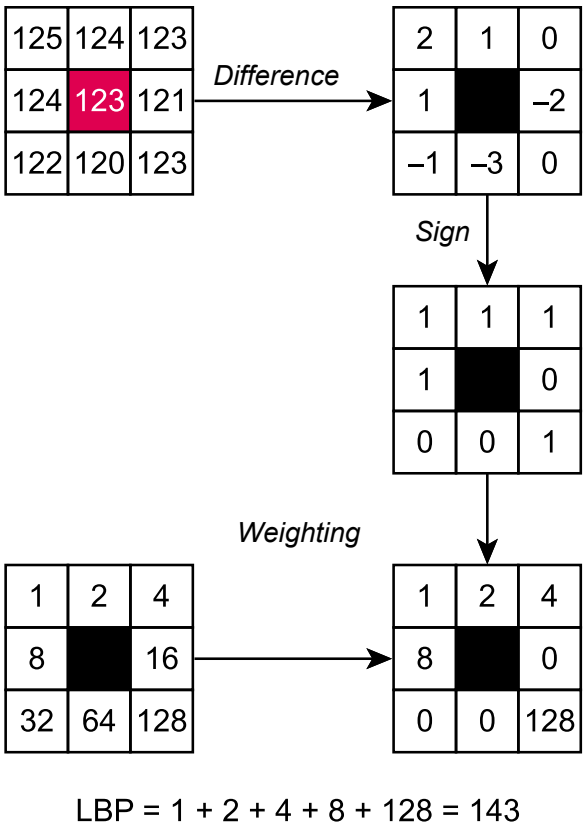


Figure 5.7: The local binary pattern (LBP) is obtained by thresholding the surrounding pixels using the central pixel, outlined in red. This list of 1s and 0s is converted into a binary number by weighting the values with powers of 2.

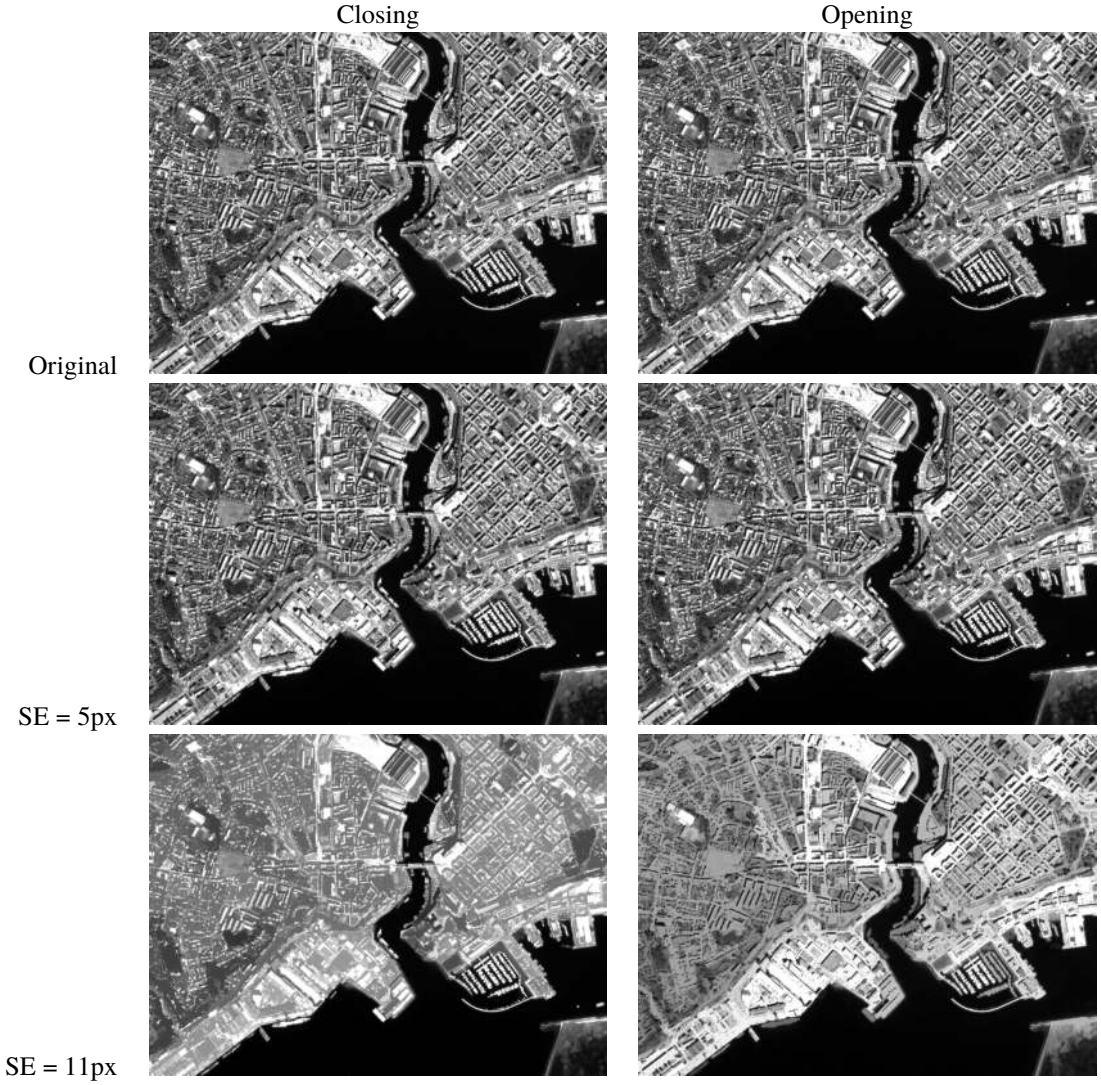
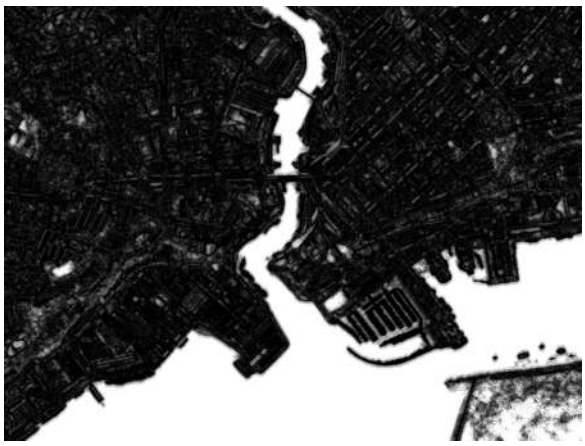


Figure 5.8: Extended Morphological Profile (EMP) on the blue band of a SPOT6/7 image over the city of Brest, for different structuring element (SE) sizes (side of the square SE). In the closing profile, objects that are darker than their surroundings are erased, whereas in the opening profile, objects that are brighter than their surroundings are smoothed.

(a) Maximum segment length. Provides a high value if the current object has a wide extent in at least one direction.

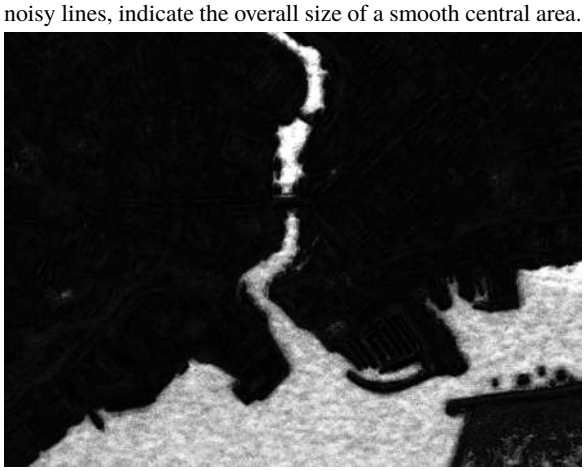


(b) Minimum segment length. Presents a low value if the current object is restricted in at least one direction.

(c) Mean of histogram. Indicates the overall size of the object.

(d) Weighted mean. The weights eliminate contribution from noisy lines, indicate the overall size of a smooth central area.

(e) Ratio of 5th shortest to 5th longest line. Is a measure of the compactness of the object: low for very thin, elongated objects like streets.



(f) Standard deviation of the histogram. It presents strong values for objects with an elongated shape.

Figure 5.9: The six components of the SFS feature, describe the shape of objects present in a multivariate image in a compact way.

6 Evaluation of land cover maps

"It seems to me that the natural world is the greatest source of excitement; the greatest source of visual beauty; the greatest source of intellectual interest. It is the greatest source of so much in life that makes life worth living."

– David Attenborough

Evaluating the quality of a contextual classification must be done with care, as some ways of including context alter the geometry of the output map. Indeed, the validation data only covers a sparse area of the image, and only covers certain areas of the map. Therefore, it does not describe cartographic quality notions, such as shape, connectivity, or homogeneity. These are necessary, as the final goal is after all to produce a map, that is meant to be used for cartographic purposes.

The evaluation of a classification result only describes how well each individual pixel is recognized, but does not integrate any spatial notions, such as a corner or the border between two objects.

Moreover, there are natural biases that exist within the testing data, which is made up of labeled polygons. Indeed, these contain far more pixels in the central area than in the edges, or than the areas in the corners. This means that the pixels in central areas of a polygon have more weight in the evaluation.

From a cartographic perspective, these pixels near borders of elements define the shape and size of these objects, and are therefore very important.

The desire to create land cover maps that are both thematically and cartographically accurate pushes towards the design of new metrics to take the latter quality in account. The efforts made to evaluate this notion, which is called *geometric quality* or *geometric precision* here, are presented in the following sections. These are an extended version of the work published in [Derksen et al., 2019a].

6.1 Class accuracy metrics

The usual statistical performance indicators, Overall Accuracy, Cohen's $kappa$, and F-score (shown below), are naturally biased towards the most common samples in the validation data set. This means that high spatial frequency elements, such as corners and fine elements, are usually poorly represented in the validation. The further implication is that errors in such areas have a low influence on the overall statistical performance indicators. In other words, the deterioration of high spatial frequency areas can be overshadowed by other effects, such as the smoothing of noisy pixels in homogeneous areas, which usually increases the classic statistical accuracy scores.

- **Overall Accuracy** (OA) is the percentage of correctly classified elements in the test set. This score is frequently used as it represents the average performance of the classifier, over the entire test data set.

$$OA = \frac{Ncorrect}{Ntotal}$$

- **Cohen's Kappa** (κ) also takes into account the expected accuracy EA, which is the accuracy that any random classifier could expect through pure chance. It accounts for unbalanced data sets when a low number of classes is used. However, as the number of classes increases, the EA gets closer to 0 and the value of κ approaches the OA.

$$\kappa = (OA - EA)/(1 - EA)$$

- **Recall** is the ratio of elements correctly attributed to a class, compared to the number of elements that truly belong to the class. In other words, for a sample labeled i, the probability that the classifier will indeed predict i.

$$Recall_i = \frac{Ncorrect_i}{Ntrue_i}$$

- **Precision** is the ratio of elements correctly attributed to a class, compared to the total number of elements that the classifier has predicted of that class. In other words, for a sample that is predicted as i, the probability that its true label is i.

$$Precision_i = \frac{Ncorrect_i}{Npredicted_i}$$

- **F-score** is the harmonic mean of *precision* and *recall*. The F-score will be close to 1 if and only if the class is neither under-estimated (low recall) or over-estimated (low precision).

$$F - score_i = \frac{2Precision_i \times Recall_i}{Precision_i + Recall_i}$$

6.2 Standard geometric quality metrics

One way to evaluate the quality of the geometry of a classification map is to split the validation set in several subsets, where each subset contains pixels of a certain geometric category, such as corners, edges, or central areas, as is done in [Bruzzone and Carlin, 2006], and later in [Huang et al., 2008]. This allows a specific measurement of the deterioration of the various geometric entities in the image, but requires dense reference data to categorize the validation labels as corners, edges, etc. Another commonly used metric, the Intersection over Union (IoU) [Everingham et al., 2010] also requires dense reference data to calculate the areas of intersection and union. Moreover, it is subject to the same biases as Overall Accuracy and κ, as it measures an average error on the target object or segment. The more sophisticated Overall Geometric Accuracy (OGA) proposed by [Möller et al., 2014] also uses the areas of intersection and union, in combination with the position of the center of gravity of the reference and target objects. However, using such metrics is only possible if the validation data set is dense, in other words, if every pixel of the validation area is labeled. Indeed, without this information, there is no way to split the validation data into geometric categories, or to extract the reference objects.

Unfortunately, dense validation data is not available in most practical land cover problems. Indeed, there are many cases where training data is manually collected in the field, or comes from a combination of existing data bases, which are all incomplete, or for which certain classes are out of date. A small, dense validation set could be manually constructed, but this would limit the metric to a reduced region, and would be a very time-consuming process. Figure 1.9a illustrates the sparse reference data used later in the experimental section, which is similar to the database used in [Inglada et al., 2017] for time series mapping over France. The validation data contains polygons that unfortunately do not contain a full description of the geometry. First of all, the polygons have been eroded, to limit the negative impact of spatial co-registration errors between different images at dates. Second of all, the edges and corners of the polygons can not be used as reference geometry, because there is no guarantee that each polygon edge truly separates elements of two different classes.

6.3 Pixel Based Corner Match

In this section, a new metric that aims to quantify the geometric precision of a contextual method, with respect to a pixel-based method is presented. This metric relies on the output of a pixel-based classifier to extract sharp corners, which are compared to the corners from a contextual classification map. This is based on the assumption that the pixel-based classification map respects the high spatial frequency areas, and the target geometry. Indeed, a pixel-based classification map can be sensitive to noise and to errors in context dependent classes, but it should preserve the corners and fine elements in the image. On the other hand, context-based classifiers can alter the geometry of the result. An example of this phenomenon is given in Section 6.3.3, in which many of the sharp corners originally present in the pixel-based classification are smoothed when using a contextual method. For these reasons, the Pixel-Based Corner Match (PBCM) is based on corner detection alone, with the pixel-based classification map used as a reference. The validity of this hypothesis is discussed further in Section 6.3.5.

The image itself should not be used to detect the reference corners, because highly textured areas contain many corners which should not be present in the target classification maps. In other words, the corners that should be

preserved by a contextual classifier are the ones at the intersections of the different classes, which are not necessarily the same as the corners in the actual image.

Using successive steps of line detection and corner detection, the objective is to calculate the percentage of corners in the target classification that are situated near at least one corner in the pixel-based classification. The notion of proximity is given by a radius parameter, which is taken to be very small (1 pixel). This gives a quantitative indication of how many corners were displaced or lost, when using a contextual method.

It is important to note that the metric is intended to be used in a relative manner, in other words, to compare the geometry of results from various possible choices of spatial support, feature, or parameters on a given problem. Indeed, the absolute values of the corner matching must not be interpreted directly, as they depend strongly on the parameters of the corner detection, which should be calibrated according to the type of imagery, and to the target classes. The absolute values also may depend on other unknown factors, such as the level of noise in the classification map.

6.3.1 Corner detection

Detecting the corners in a classification map can be done by extracting straight line segments in the map, and by calculating the position and angle of the intersection between pairs of lines. For this, first of all, the classification map is split into a set binary maps (one binary map for each class). Then, an unsupervised Line Segment Detector (LSD), based on [Von Gioi et al., 2012], is applied to the map, generating a set of segments for each class. In order to find corners on the edges of areas of various classes, all of the segments of the different classes are merged together. A corner is detected if the angle of the two segments is within a certain range (30°-120°), and if their extremities are close enough. This is shown in Figure 6.1.

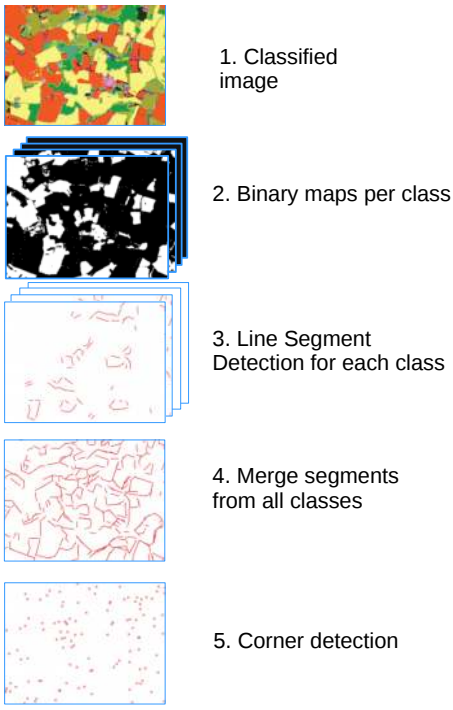


Figure 6.1: Corner extraction process. First, the image is split into binary maps for each class. Then, the Line Segment Detector is applied on each binary map. In order to extract corners from various classes, the segments are all merged together before the corner detection step.

6.3.2 Corner matching

After the corners have been extracted in both the target classification map and the reference classification map, the ratio of corners that match up in both maps to the number of corners in the target classification is used as a performance metric. In other words, let C_{ref} be the set of corners of the reference image (the pixel-based classification), and C_{test} be the set of corners of the classification map whose geometry is being measured. The

set of matching corners is defined in equation 6.1, where $dist(x, y)$ is the standard Euclidean distance, and t is a threshold parameter. This is also illustrated in Figure 6.2.

$$C_{match} = \{x \in C_{test} \mid \exists y \in C_{ref}, dist(x, y) \leq t\} \tag{6.1}$$

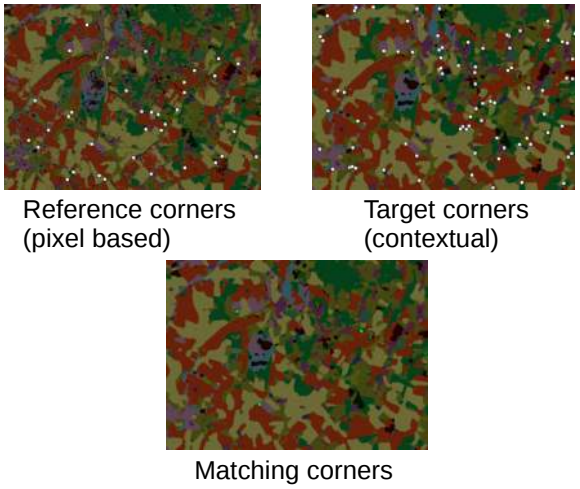


Figure 6.2: Corner matching. The corner detection is applied on a pixel-based classification, called the reference, and on a contextual classification, called the target. The PBCM is the ratio of matched corners to the number of corners in the target.

From here, the geometric precision metric $PBCM$ can be defined, as is shown in equation 6.2.

$$PBCM = \frac{Card(C_{match})}{Card(C_{test})} \tag{6.2}$$

A high ratio means that many of the corners detected in the target are also present in the pixel-based classification, and reversely, a low ratio means than many of the corners in the pixel-based classification have been lost. When comparing two pixel-based classifications generated with a different sampling of training data, and therefore a different Random Forest, an average matching ratio of 51.3% is measured, see Figure 6.4. This seemingly low number is due to imperfections in line and corner detection, which are sensitive to the label noise present in the pixel-based classification.

In order to increase the robustness of the metric, each target classification can be compared to several pixel-based results, which are generated by classifiers trained on various random samplings of the training data. This reduces the contribution of noise, in the same manner as a cross-validation scheme. Then, the average value and standard deviation of the metric can be calculated, in order to provide an indication of the confidence of the metric, when different sub-samplings of the training data are used.

The PBCM metric also has its limits, as it only measures the smoothing of corners, and not of other high spatial resolution features, such as fine elements. Furthermore, it is biased by the corners of the majority classes, in this case, the two crop classes (summer and winter crops), which account for the wide majority of corners detected in these maps. The geometry of other classes, such as the urban classes, which are unfortunately the most challenging to classify, might not be measured in this case. The metric might also overlook the geometry of minority classes, which do not generate many corners in the first place. However, it still can play the role of an indicative metric, as these biases are known and can be accounted for in the interpretation. Moreover, it would be possible to add weights to the different corners, according to the classes that form them, and in this way to reduce the biases linked to class proportions. However, this would be application dependent and is not developed further in this work.

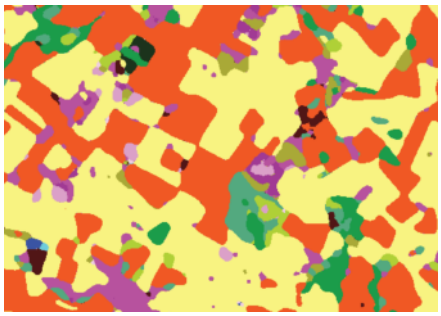
6.3.3 Impact of regularization

To demonstrate the pertinence of the metric, a majority vote filter, also known as regularization filter, is applied in a sliding window to the result of a pixel based classification. This common post-processing step consists in replacing the label of the central pixel of the sliding window by the most frequent label in the neighborhood. It is known to increase the statistical accuracy by removing isolated pixels in the final result. This is illustrated in figure 6.3. Figure 6.3a shows the result of a pixel based classification, which contains sharp corners, and a certain

amount of isolated pixels, that can be attributed to noise. As figure 6.3b demonstrates, this noise is largely reduced by the regularization filter. However, the corners are slightly smoothed. In figure 6.3c, a larger neighborhood of 11x11 pixels was chosen for the regularization filter, which has a heavy smoothing effect on the previously sharp corners.

(a) Pixel-based classification over a test area. The sharp corners are present at this level of detail.

(b) Regularization in a 3x3 sliding window. The geometry remains relatively well respected, although some corners are slightly smoothed out.



(c) Regularization in a 11x11 sliding window. The smoothing effect is clearly visible.

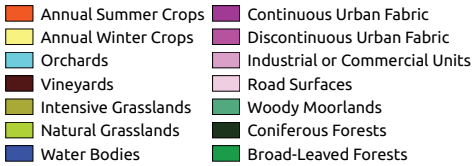


Figure 6.3: Regularization (majority vote filtering) in increasingly large sliding windows shows the smoothing of round corners.

The impact of the regularization on the statistical accuracy is shown in Figure 6.4. In this figure, the vertical axis shows the difference in overall accuracy with respect to the pixel-based classification, while the horizontal axis represents the PBCM. The labels above the points indicate the size of the sliding window, in pixels. Clearly, regularizing the classification result using a sliding window has a positive impact on the classification accuracy metric (Overall Accuracy). This remains true, even for very large sliding window sizes. In fact, the most accurate performances are achieved for the large windows (11x11, 13x13, 15x15), where the geometric deformation is very visible, as is shown in figure 6.3c. Figure 6.4 also shows that when applying a majority vote filter in a sliding window neighborhood, like in figure 6.3b, the PBCM decreases as the size of the filter increases. Indeed, the metric reaches 30% for a window of 5x5, and passes under 10% for a window of 11x11. These results give a first indication that the corner matching metric is indeed sensitive to a deterioration of the geometric quality, and allows for an initial quantitative evaluation of this effect. This also shows that measuring the Overall Accuracy or the Kappa alone is not sufficient to fully evaluate the quality of a map, and that a specific metric for evaluating the quality of the geometry is indeed necessary.

The ellipses around the points indicate the standard deviation of the metrics across the ten runs. Their relatively thin width shows that PBCM is consistent in its measurement, with regards to different samplings of training data for the pixel-based classifier.

6.3.4 Calibration of the metric

Extracting the corners involves several parameters that need to be calibrated to the type of imagery used. In particular, the Line Segment Detector depends on 7 parameters, which all have a significant influence on how well the line segments are extracted. The advised parameters given in [Von Gioi et al., 2012] have been selected for computer vision problems, and do not always provide coherent results when applied on binary maps at a 10m resolution. Indeed, at such a resolution, each desired segment is made up of a relatively small number of pixels,

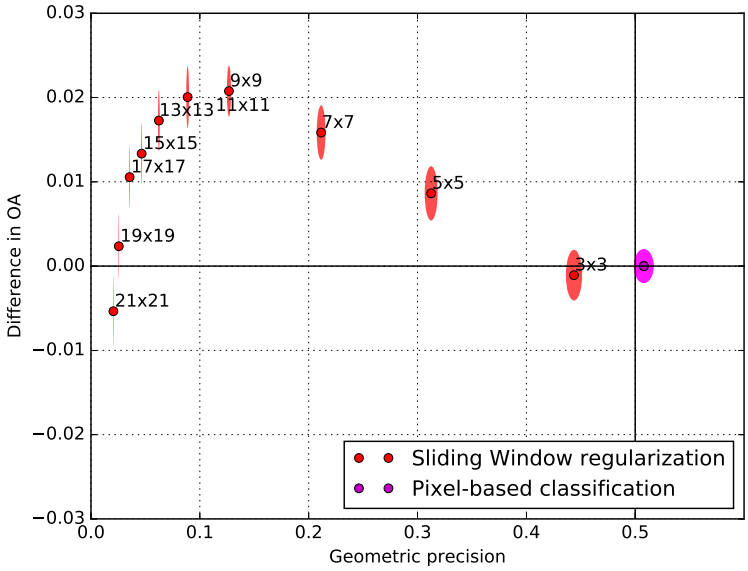


Figure 6.4: Impact of the sliding window majority vote regularization on the Overall Accuracy and geometric precision (PBCM). Each pixel is assigned the majority label in the sliding window. The size of the filter, in pixels, is shown next to the points. This kind of regularization increases the statistical classification scores, however, the image of the result shows that the corners are strongly smoothed out. This is confirmed by the PBCM metric. The axes of the ellipses show the standard deviation of both the Overall Accuracy and the PBCM, over the 10 runs with different subsets of training data.

when compared to computer vision images. Secondly, the contrast along the lines in binary maps is stronger than in natural images. For this reason, a calibration step is used before applying the metric. This involves maximizing the average number of matching corners between pairs of pixel-based classification maps, while minimizing the number of matching corners between a pixel-based classification map and a regularized classification map. In practice, the difference between the two is used as a cost function for a grid search optimization over the parameters, around their default values. Pixel-based results from several samplings of the training data are used to increase the robustness of the PBCM metric at each step of the calibration. The resulting values of the calibration are given in Tables 6.1 and 6.2. The graphs that justify the choice of certain of the parameters are shown in Appendix A, on page 183.

Table 6.1: Calibration parameters of the Line Segment Detector, as presented in [Von Gioi et al., 2012].

S	Σ	q	τ	$log(\epsilon)$	D	N bins
0.8	0.6	2	45°	0	0.7	1024

Table 6.2: Calibration parameters of the corner extraction and corner matching. The angle interval and extremity distance threshold define how a corner is extracted from two segments. The angle formed by the segments must be within the interval and the extremities must be at a distance smaller than the threshold. The matching threshold determines how much tolerance is taken when matching corners from two different classification maps.

Angle interval	Extremity distance threshold	Matching threshold
60°-120°	10m (1px)	10m (1px)

The details of this calibration are not provided in great depth here, however, it is possible to provide a visual validation of the line and corner detection stages by seeing how well these are performed. Figure 6.5 shows a small part of the classification map, along with the lines and corners that are detected on the Annual Summer Crop class (yellow). Indeed, these follow the principal outlines of the fields, and segments with intersecting ends indeed form corners. While this detection is not complete, it appears that some corners are missed, it seems that it is quite accurate, the corners that are detected correspond to true corners visible in the classification map.

The extent to which this metric is valid, in view of these limits, is discussed in the following section.

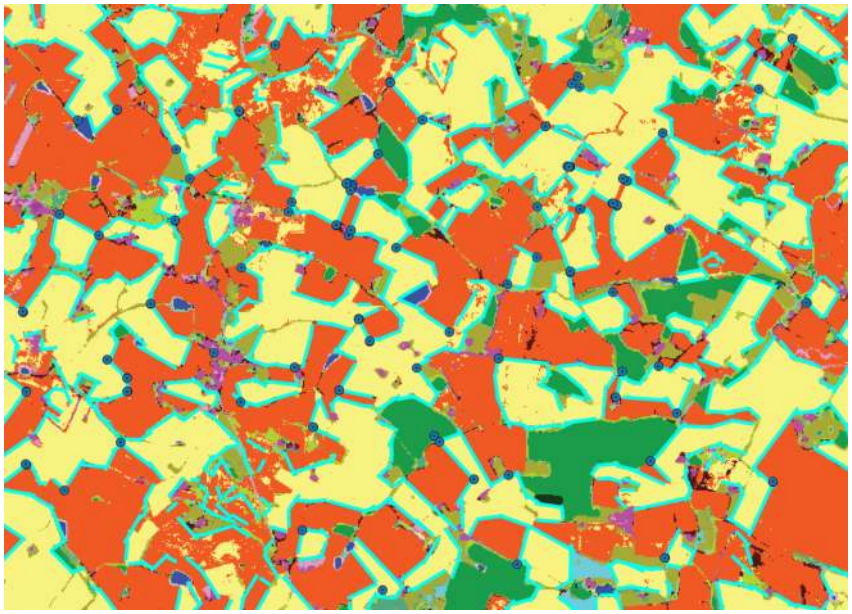


Figure 6.5: Extract of the line and corner detection steps on the Annual Summer Crops (yellow) class on a small area on the T31TCJ tile. The lines, shown in clear blue, seem to follow the edges of the fields. The corners, marked as dark blue points, are indeed detected at the intersection of segment ends. The corner detection is not perfect, as many corners are missed, however the ones that are detected seem to be coherent, visually speaking.

6.3.5 Further validation with dense reference data

The study of the cartographic quality of land cover maps is strongly limited by the lack of dense reference data. The reason that the PBCM uses the pixel-based classification for the corner detection is in fact to simulate the presence of dense reference data, or at least its geometric aspects.

In fact, there is an available source of dense reference data that covers large cities, such as Toulouse. The Urban Atlas classes fully describes these these cities, in the sense that each different area is labeled. The OSO reference data contains only an extract of these classes, as was mentioned in Part II, Table 3.1, on page 48. In fact, the other classes are meant to be encapsulated within the higher-level classes.

Analyzing the possibility of creating a dense data set with the four OSO urban classes was the focus of the work of an intern, Lucas Schwaab, who proposed this correspondence and evaluated the performance of the PBCM with regards to this data set.

This evaluation was done in two steps. First of all, the hypothesis that the pixel-based classification contains the true corners of the data set is put to the test.

The corner detector uses a pixel-based classification, which does not have a perfect geometry in itself. The question becomes, how well is the corner detector able to recognize the geometric elements of the reference data, using only a pixel-based classification ?

The results show that between the corners detected in this way, and the corners from a dense reference data set, a recall of around 85% is achieved, for a precision nearing 10-15%. In other words, this means that many of the corners that are detected are indeed corners from the reference. However, the detector misses a large percentage of the others.

This validates the fact that the PBCM does indeed detect salient elements in the maps. It also explains why such low values of PBCM are generally reached: the 15% of detected corners are not necessarily the same in every case.

This dense reference data was also used to validate the fact that the PBCM does indeed measure geometric degradation. This is done by correlating it with another metric encountered in literature [Bruzzone and Carlin, 2006, Huang et al., 2008], namely the class accuracy in pre-defined geometric areas, i.e. corners and edges of the testing data, that was mentioned earlier.

Figure 6.7 shows the PBCM plotted against the average F-Score of the four urban classes shown in equation (6.3), calculated in the corners present in the dense reference data. The points show the evolution of the scores when applying successive regularizations of increasing size, as was done earlier. This graph shows that this causes the PBCM to decrease immediately, whereas the average urban F-score increases slightly before decreasing. This indicates that perhaps the average urban F-score is sensitive to effects that increase the overall accuracy everywhere

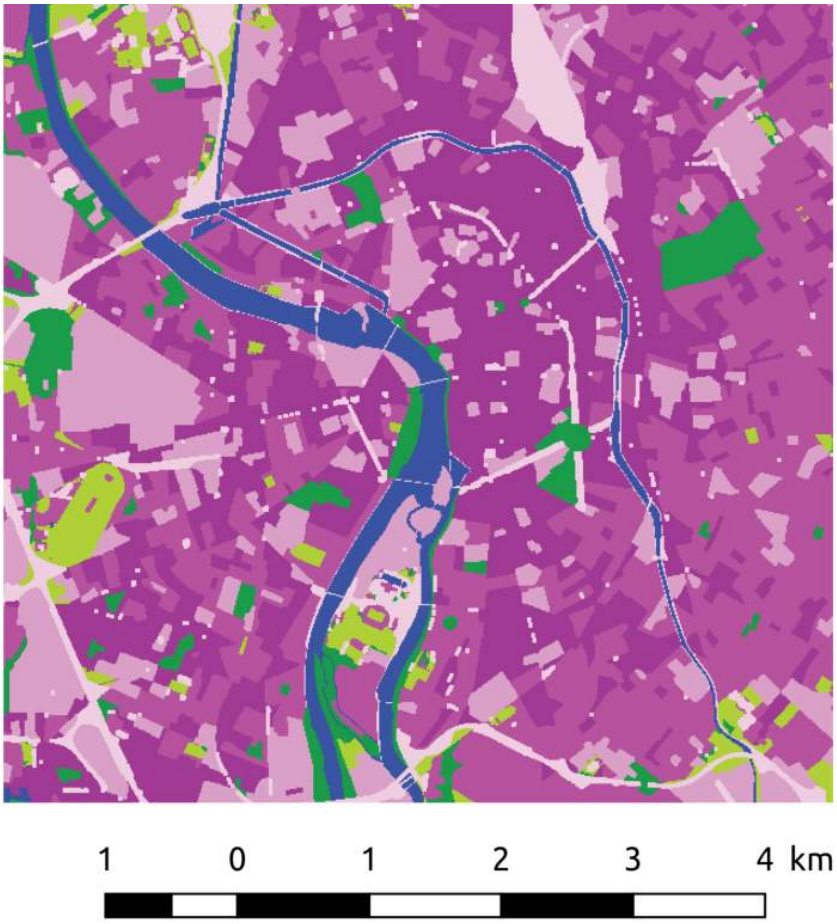


Figure 6.6: Image of the dense reference data set obtained on Toulouse using the full nomenclature of Urban Atlas, translated into the four OSO classes. The correspondence between UA classes and the OSO classes is given in Table 3.1, on page 48.

in the image, such as the smoothing of classification noise.

$$AUFS = \frac{FS_{DUF} + FS_{CUF} + FS_{ICU} + FS_{RSF}}{4} \tag{6.3}$$

$$RAUFS = \frac{AUFS_{corners}}{AUFS_{image}} \tag{6.4}$$

Figure 6.8 shows the PBCM plotted against the relative average urban F-score (RAUFS), which is the ratio of the score in the corners to the score everywhere in the image, shown in equation (6.4). This is done in order to mitigate the global improvement, to focus on the deterioration of the geometry alone. In this case, the score also decreases with increasing regularization size, and seems correlated to the PBCM, until the PBCM saturates.

This graph shows two things: first of all, the PBCM can be correlated to the relative urban f-scores, for small geometric degradations. Second of all, it cannot measure extremely strong degradations. Once no more corners are detectable in the image, none can be matched.

These results are encouraging with regards to the PBCM approach, and validate two of its fundamental aspects.

1. It indeed detects corners that are present in a dense reference data set.
2. The matching of these corners is a way to measure a local geometric degradation that is independent of effects that increase the accuracy score everywhere on the image.

However, the experiments in Part IV will question the validity of this metric, and suggestions regarding its further possible improvements are discussed in Part V.

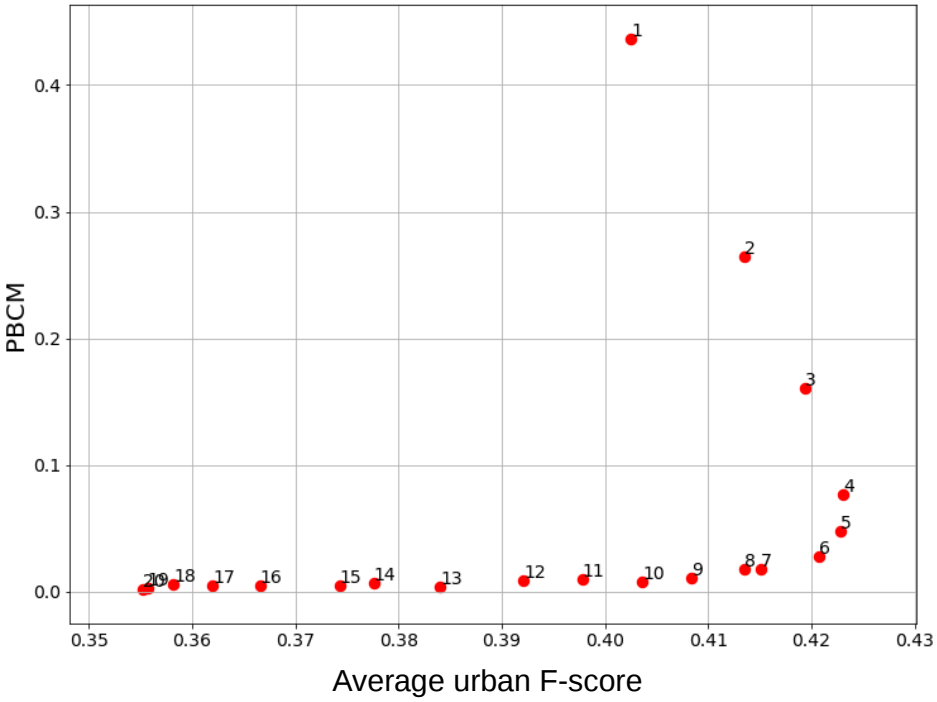


Figure 6.7: PBCM and Average urban F-score, which is the average of the F-scores of the four urban classes, shown in equation (6.3). The points represent different stages of regularization of increasing radius, which is shown in the labels. The Average Urban F-score has a tendency to increase for small regularization sizes, which indicates that it might be biased by global effects such as label smoothing.

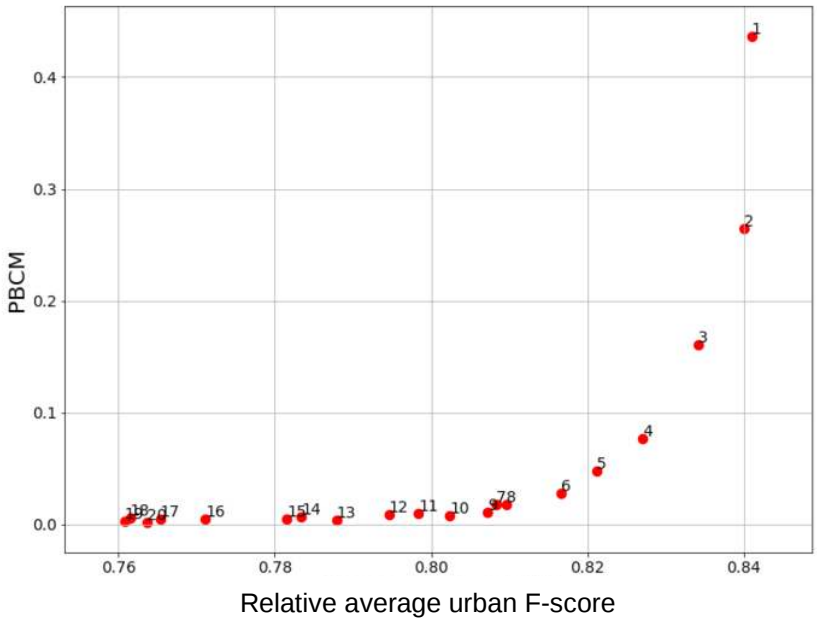


Figure 6.8: PBCM and relative average urban F-score (RAUFS), which is defined in equation (6.4). The relative Average Urban F-score decreases with increasing regularization windows, which indicate that it indeed measures a geometric degradation. The PBCM seems correlated to this metric for small radii (3-8 pixels), until it saturates as it is no longer able to detect corners in the area.

Part III

Advanced contextual classification

7 Scaling the spatial supports

"The question of whether a computer can think is no more interesting than the question of whether a submarine can swim."

– Edsger W. Dijkstra

In Section 1.4.3, it was emphasized that current land cover mapping problems are based on large volumes of data, such as time series of multi-spectral images, or mono-date hyperspectral images.

For this reason, a memory management scheme known as *piece-wise processing* must be implemented to enable any form of computation on such large images. In many cases, the image can be split into smaller regions, usually rectangular in shape, such that each region fits in the volatile memory. This is illustrated on in figure 7.1. This way, the processing can be done entirely within the RAM, without any time loss due to I/O operations with the storage unit. The final result is usually a stitching of the individually processed regions.

On a side note, this also allows different processing units to operate in parallel, by each dealing with a separate region of the data. Doing so decreases the total processing time, and is commonly done even when the entire image fits in the RAM.

Piece-wise processing works flawlessly for any pixel-based method, where the order in which the pixels are read has no importance H.owever, when contextual methods are used, the spatial arrangement of the pixels, and their relative position in the image becomes important. Indeed, the edges of tiles most likely intersect objects of the image. Piece-wise processing must be used with precaution, or errors will occur near the edges of the pieces.

This chapter addresses the issue of extracting spatial supports in large images, which are only truly an issue for supports that use an image segmentation method, such as objects, or superpixel.

Applying sliding window methods to very large images poses no particular problem, it only requires a slight adjustment to the previously defined piece-wise processing scheme. Using a square spatial support of $N \times N$ without precaution means pixels within a distance of $\frac{N}{2}$ pixels from the edge of the tile will contain a false contextual description, as some of the pixels in their neighborhood are on the other side of the edge. To solve this, each tile is *padded* with $\frac{N}{2}$ pixels, which overlap on the neighboring tiles. This *pad* or *margin* allows sliding window methods to be applied to very large images.

A small extra cost can be attributed to the supplementary areas that need to be read along the tile edges. This implies that the total length of the tile edges should be minimized, in order to optimize this process. This is done by using square-shaped tiles rather than long rectangular strips, as squares have a higher area to perimeter ratio. This is illustrated in figure 7.1.

7.1 Application of Mean Shift to large images

Segmentation on images with a great number of pixels can be difficult, due to the nature of piece-wise processing. For the same reasons as mentioned earlier, in Section 4.1, pixels near the edges of the tiles are sensitive to their presence, as they lack information from the pixels beyond the edge. If used without precaution, the tile edges draw a straight lines through the image, which have nothing to do with the objects in the image.

This issue, known as the *scaling* of segmentation methods, was recently addressed on the Mean Shift algorithm by [Michel et al., 2015] and on Region Merging algorithms by [Lassalle et al., 2015]. In this context, the word *scaling* means ensuring the scalability of algorithms, which is their property to be applied on large data sets. This can be declined into two properties.

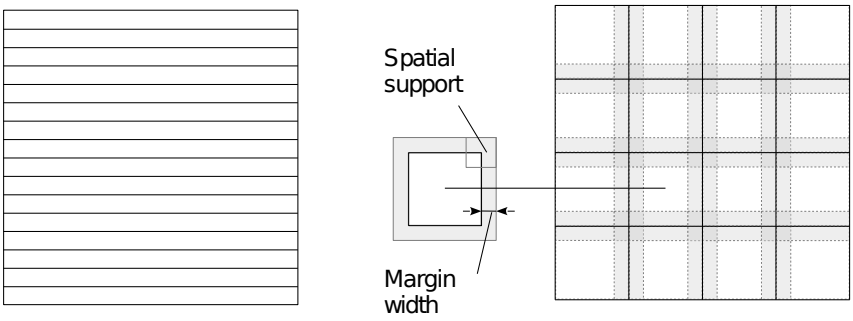


Figure 7.1: Two possible piece-wise processing configurations. On the left, the image is divided into 16 equal strips. This is often how pixel-based methods are applied to large images. When contextual information in a sliding window is required, a scheme like the one on the right image is used, in which each tile is padded by a margin of half the size of the window. Using square tiles like in the configuration on the right side visibly requires a lower total margin area, and is more optimal in that sense.

First of all, they must have a linear complexity with respect to the total number of pixels in the image. In other words, calling N the number of pixels in the image, and S(N) the number of operations necessary to perform a segmentation on that number of pixels, the function $\frac{S(N)}{N}$ should converge towards a fixed number as N increases. This notion is encapsulated in the O notation. In this case S(N) is said to be O(N), which is equivalent to saying that the segmentation method has a linear complexity. This provides a theoretical safeguard that the algorithm can be applied to large data sets without requiring an exponential number of operations. This property is verified for all of the algorithms studied here, so the complexity of each one will not be derived in detail.

The second property is that any data set should be processable with a finite amount of memory, which fits with the real world cases in which the RAM is smaller than the total size of the image.

The scaling up of segmentation algorithms is a challenging task, and one unifying framework suitable for all algorithms has not yet been found. Two main groups of methods to deal with this issue can be distinguished. The first group contains methods based on adding a post-processing step to fix the errors linked to tiling. Such methods operate on the tile edge regions after the segmentation step is finished. For example, in [Michel et al., 2012], segment fusion along the tile edges is based on a user-defined topological criteria such as the length of the contact surface between the segments to be merged. However, this does not completely eliminate errors on the edges as the edge can still appear in certain parts of the segments. Another idea from the same group of methods is to re-process the region edge areas after the algorithm is finished using a more sophisticated method than simple region fusion. In [Yang, 2011], the segments on the tile edges are merged by the Full Lambda Schedule Algorithm (FLSA), first on the vertical borders, and then on the horizontal ones. A similar idea is developed in [Happ et al., 2010], in which the segment border areas are processed separately at the end of each iteration of the algorithm. Unfortunately, reading and writing the image into disk memory at each iteration is very costly.

The other group of methods for dealing with the tile edges issue involves acting on the tiling itself before applying the segmentation algorithm. In [Korting et al., 2011], the image is split into non-rectangular tiles. The authors propose a tiling following gradients in the image. However, there is no guarantee that the actual segments would have followed this tiling, if the method had been applied on the whole image. Furthermore, image gradients are difficult to use in the case of very high dimensional data, such as a Sentinel-2 time series. Another idea is to pad each tile by a margin on all sides, instead of using non-overlapping tiles, following a similar logic as the one used for applying sliding windows to large images. By doing so, the hope is that segmentation results in the central area of the tile are less affected (or even unaffected) by the tile border. A certain rule can then be defined to choose how to stitch the margin areas together. [Michel et al., 2015] propose a method that guarantees precisely the same results with and without tile-wise processing, applied to the Mean Shift segmentation algorithm, that was mentioned in Part II, Chapter 4, Section 4.2.2.

For this, they define the concept of *stability*. A segmentation algorithm is considered stable if when applying it to any sub-region of an image, it produces the same result as if the image had been first segmented, then cropped to the sub-region.

Marginal stability has a looser constraint: the results must be identical within a central zone of the sub-region.

If an algorithm is shown to be marginally stable, and this margin can be calculated, the tiling with margins method (described above) is guaranteed to provide exactly the same result as a segmentation without tiling. In a further study [Lassalle et al., 2015], these concepts are applied to the Region Merging family of algorithms.

The following section describes how the Simple Linear Iterative Clustering (SLIC) algorithm can be adapted to process very large images, which is the subject of the work done in [Derksen et al., 2019b].

7.2 Scaling the SLIC superpixel algorithm

The main difficulty for applying superpixel segmentation to high-dimensional images is finding an efficient way to scale the segmentation algorithm. Indeed, if the image does not fit in the memory, it must be split into smaller pieces which can be dealt with individually (tile-wise processing). If no precautions are taken, the tile edges themselves will appear as segment edges in the final segmentation result. To counteract this effect, it is possible to initialize SLIC's segmentation on the entire image. The initialization is a square regular grid, and the initial labels can be calculated anywhere, even if the whole image is not loaded in memory. Then, the algorithm can run as usual. In this case, the tile edges will not directly appear in the final segmentation, but other kinds of anomalies will, as shown in figure 7.6c.

1. Sharp edges will appear along the tile edge zones.
2. A segment that is on two adjacent tiles will end up being twice as big as the average segment size.
3. Disjoint segments can be found along the edges, as the simple point constraint is no longer present between the two independent tiles.

The solution that is developed here is inspired by the work in [Michel et al., 2015], in which the authors take margins around each processing tile. Their objective was to create a tile-wise processing mechanism that guaranteed precisely the same segmentation result as if the image had been processed with no tiling. This was shown to work for algorithms that were stable. An image processing algorithm is considered stable if when applying it to any region of the image, the same result is obtained as if you had applied the algorithm on the whole image, and then extracted the result in that region. SLIC is not stable by this definition, because the result depends on the order in which the pixels are considered. If applying it to only a sub-region of the image, the pixels will not be scanned in the same order as on the entire image. Even the original version of SLIC is not stable, because the post-processing step also depends on which order the isolated regions are read. In fact, the scanning order is arbitrary, [Chang et al., 2013] scan pixels in a random order, but for processing speed reasons a lexicographical order is chosen for our application. The scanning order on the full image does not provide theoretically better results than scanning each region independently, because both orders are arbitrary. That is why in this study the *identical results* requirement from [Michel et al., 2015] and [Lassalle et al., 2015] was loosened, and replaced by a *similarity* requirement, which involves two aspects.

1. No segmentation errors should appear along tile edges.
2. The segmentations with and without tile-wise processing should show similar overall segmentation characteristics, as defined by the four criteria described in the next paragraph.

7.2.1 Segmentation quality criteria

In this part, a method to verify if the two segmentations (with and without tile-wise processing) have the same overall characteristics is developed. In [Achanta et al., 2012], superpixel quality is given by two measures, adherence to boundaries and boundary recall. These measures, like the Hoover metrics [Hoover et al., 1996], are based on dense validation data, in which all the pixels are labeled. The two segmentations could be compared to a reference segmentation, to ensure they have the same quality. Unfortunately, dense validation data is not available in our case. In [Lassalle et al., 2015], the Hoover metrics were used with the segmentation on the full image as reference data, because the objective was to achieve an identical segmentation. However, as mentioned previously, SLIC is not stable. Therefore, the differences between the segmentations with and without tile-wise processing are not due only to tiling, but are also linked to the arbitrary pixel scanning order. The Hoover metrics will be sensitive to these differences. For this reason, simpler unsupervised evaluation metrics have been chosen. Superpixels aim at being compact, homogeneous and similarly sized, so these three criteria will be measured. Compactness can be calculated in two ways. Firstly, based on the perimeter and area, [Zhang et al., 2008], as was shown earlier in equation (5.8), on page 81. This measure is always between 0 and 1, 1 being the value for a circle (maximal compactness). The perimeter-based compactness measure is sensitive to segments with rough edges, so an new

measure is proposed, based on the statistical spread of the position vectors (x, y) of the pixels of a segment. First the 2x2 symmetric covariance matrix of the positions x and y of the pixels in the segment is calculated. Next, the eigenvalues of this matrix are calculated by the formulas in equations (7.1a) and (7.1b).

$$S_1 = \sigma_x + \sigma_y + \sqrt{\sigma_x \sigma_y - \sigma_{xy}^2} \tag{7.1a}$$

$$S_2 = \sigma_x + \sigma_y - \sqrt{\sigma_x \sigma_y - \sigma_{xy}^2} \tag{7.1b}$$

The normalized ratio of the first eigenvalue to the second eigenvalue, equation (7.2), gives a measure of the statistical elongation of the segment. This metric is between 0 and 1, with 1 being the maximal compactness (minimal elongation). It is rotation invariant because the eigenvalues are rotation invariant, and scale invariant because it measures a normalized ratio.

$$C_s = \frac{2S_1 S_2}{S_1^2 + S_2^2} \tag{7.2}$$

Spectral homogeneity can be measured in several ways [Zhang et al., 2008]. The selected metric, taken from [Moore et al., 2008], is the explained variance, as shown in equation (7.3). It measures the variance retained in the image after replacing each pixel value by the mean value in the segment. By normalizing this value by the total variance in the image, a value between 0 and 1 is obtained, that is insensitive to the feature dimension and image content. A value of 1 indicates maximal homogeneity in the segments.

$$EV = \frac{1}{D} \sum_d \left(\frac{\sigma_{d,seg}}{\sigma_{d,im}} \right) \tag{7.3}$$

Finally, the similarity of segment sizes is measured by the relative standard deviation of the areas of the segments, i.e. the standard deviation divided by the mean segment size, equation (7.4). The higher this value, the more the segment sizes are spread out.

$$RSD_{size} = \frac{\sigma_{size}}{\mu_{size}} \tag{7.4}$$

These measures will be used to compare the characteristics of a segmentation with and without tile-wise processing.

7.2.2 Tile-wise processing procedure

The key of this tile-wise processing procedure lies in taking margins around certain tiles. A margin is defined as a padding of a given width, applied on the tile in all directions. The idea is that the overlapping parts of two neighboring tiles can be combined to eliminate errors in these areas.

The process starts by splitting the image into $Nx \times Ny$ tiles, along the X and Y directions. Margins do not need to be taken around all the tiles. Indeed, a tile edge is shared by two tiles, so only one of the two tiles needs to take a margin on the other, to fix the errors around their common edge. Margins are therefore taken only on one in two tiles. Visually, the tiling can be compared to a chessboard with alternating black and white squares. The tiles around which margins are taken will be referred to as *white tiles*, the others as *black tiles*. This is illustrated in Figure 7.2. White tiles take margins on the four surrounding black tiles, and the corners of two white tiles above them.

This process is described step-by-step on a 2x2 tiling example in Figure 7.3.

1. The first step consists in running the segmentation algorithm on all the black tiles. These tiles are treated independently, so they can even be processed in parallel.

2. Once all black tiles areas are segmented, the white tile processing can start. White tiles of the first row take margins on the surrounding black tiles. They load into memory not only the image in this area, but also the segmentation result provided by the surrounding tiles.

3. The key is then to freeze all the segments on the margin outer edges, meaning they will not be altered by the segmentation algorithm. Then, the remaining parts of the white tile are processed.

4. Because the outer edge segments have been frozen, the segments in that area will be coherent with the ones in the black tile, and will not exhibit any edge anomalies.

5. The white tile of the next row is processed. If all the white tiles are processed independently like the black tiles, errors will appear around the corners of the tiles. For this reason, each white tile takes margins not only on the surrounding black tiles, but also as on the white tiles of the row above. This means the white tiles can only be processed row by row, and the white tiles in one row can be processed in parallel.

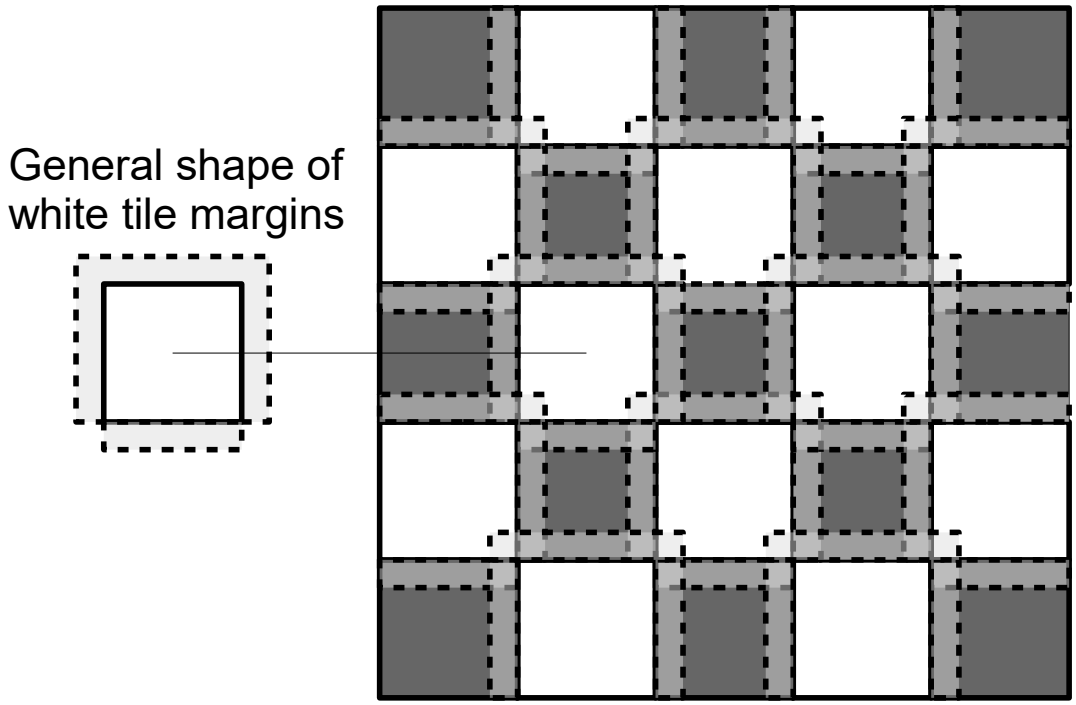


Figure 7.2: Chessboard margin strategy, white tiles take margins on the four surrounding black tiles and on the two white tiles of the row above

6. The processing is finished, and the frozen segments of the final white tile should fit well with the segments of all surrounding tiles.

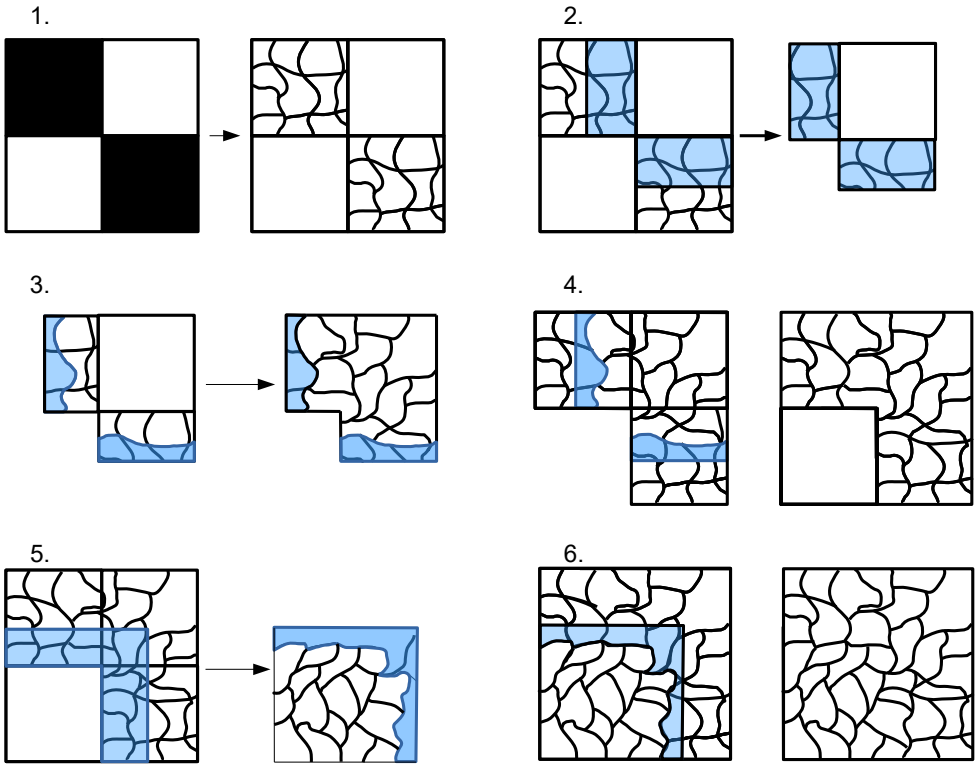


Figure 7.3: Step by step tile-wise segmentation process on a 2x2 example. 1. All the black tiles are segmented first. 2. White tiles take margins on surrounding black tiles. 3. Freeze the segments on margin edges. 4. No anomalies on the tile edges. 5. White tiles need to take a margin on the white tiles of the row above. 6. The segmentation is complete.

The condition for this mechanism to function correctly is that the margin must be wider than the maximal width of a segment. If a segment crosses the whole width of the margin, it will be frozen with the tile edge still intact, and a straight line will appear along the edge in the final result. For general segmentation methods, it is difficult to calculate a maximal segment width, as segments can have any size. Fortunately, superpixels are known to be compact, and their average width is an entry parameter of the algorithm. Therefore, it is sufficient to take a margin of a constant number of superpixel widths. Experimental results show that for 3 superpixel widths, no more edge anomalies appear in the final segmentation result.

7.2.3 Parallel processing

Tiling the image as independent pieces opens up the possibility of processing several pieces at the same time, to speed up computation. One property of the tiling algorithm is that all the black tiles can be processed in parallel. Furthermore, all the white tiles in one row can be processed in parallel, however the white rows must be processed in order. Therefore, the speed-up will not necessarily be a linear function of the number of processors, as the algorithm has sequential steps. It is possible to calculate the theoretical speed-up as a function of the tiling parameters. For example, say N_p processors have to perform N tasks. We define N_{eq} as the number of times N_p tasks have to be performed in order to finish all N tasks, in other words, the equivalent number of tasks to perform when several processors are involved. N_{eq} can be seen as the total computation time expressed in a unit of time equal to the time needed to accomplish one task. Equation (7.5) gives the formula for calculating N_{eq} as a function of N and N_p. In the following equations, $E(.)$ is the integer floor function.

$$N_{eq} = E(\frac{N-1}{N_p}) + 1 \tag{7.5}$$

The achievable speed-up s is $s = \frac{N}{N_{eq}}$. For example, if 5 processors wish to perform 12 tasks, they will first perform 5 in parallel, then the next 5 in parallel, then two processors will perform the last two tasks, while the others are idle. This could be the case for the black tiles in a 6x4 layout, or the white tiles of one row in a 24x24 layout. In

this case, $N_{eq} = 3$, and $s = \frac{12}{3} = 4$, meaning that using 5 processors only brings a speed up of 4. Estimating these speed-ups for the particular case of tiling with black and white squares can be done by calculating the total number of tasks for the different steps of the algorithm. In the first step of the algorithm, all the black tiles can be processed in parallel. The total number of black tiles is

$$N_{bl} = E(\frac{N_x N_y + 1}{2})$$

meaning N_{eq} for the first phase can be calculated using equation (7.5).

To calculate the achievable speed-up for the white tile processing step, it is necessary to first derive the number of white tiles in odd and even rows, the details of this calculation can be found in equation (7.7).

The total number of white tiles is

$$N_{wh} = E(\frac{N_x N_y}{2});$$

the number of white tiles in an even row is

$$N_{wh,even} = E(\frac{N_{wh}}{N_y});$$

the number of white tiles in an odd row is

$$N_{wh,odd} = E(\frac{N_{wh} + N_y - 1}{N_y});$$

the total number of even rows is

$$N_{even} = E(\frac{N_y + 1}{2});$$

the total number of odd rows is

$$N_{odd} = E(\frac{N_y}{2});$$

The $N_{wh,even}$ tiles of any even row can be processed in parallel. There is a total of N_{even} even rows, which must be processed sequentially. The equivalent number of tasks for white tiles on even rows is therefore

$$N_{eq,even} = (E(\frac{N_{wh,even} - 1}{N_p}) + 1)N_{even};$$

The same logic applies to odd rows, the equivalent number of tasks for white tiles on odd rows is

$$N_{eq,odd} = (E(\frac{N_{wh,odd} - 1}{N_p}) + 1)N_{odd};$$

The total equivalent number of tasks for all white tiles is

$$N_{eq,wh} = N_{eq,even} + N_{eq,odd};$$

The total equivalent number of tasks for all tiles is

$$N_{eq} = N_{eq,bl} + N_{eq,wh};$$

Combining all of the previous equations, the equivalent number of tasks for the entire process is:

$$\begin{aligned} N_{eq} = E(\frac{E(\frac{N_x N_y + 1}{2}) - 1}{N_p}) + 1 \\ + (E(\frac{E(\frac{E(\frac{N_x N_y}{2})}{N_y}) - 1}{N_p}) + 1)E(\frac{N_y + 1}{2}) \\ + (E(\frac{E(\frac{E(\frac{N_x N_y}{2}) + N_y - 1}{N_y}) - 1}{N_p}) + 1)E(\frac{N_y}{2}). \end{aligned} \quad (7.6)$$

$$s = \frac{N_x N_y}{N_{eq}} \quad (7.7)$$

Figure 7.4 presents the results of experiments on the memory reduction achievable with different numbers of processors, along with the theoretical speed-ups calculated using the formulas from equations (7.6) and (7.7). The dotted lines in Figure 7.4 represent the theoretical speed-up for a given number of processors, which is an upper bound for the actual speed-up. Memory reduction is the ratio of the size of the image to the size of a tile. Best performances are achieved in the upper-left region of the graph, where the speed-up is high for a low memory reduction. As the number of processors increases, the curves are nearer to this region.

Although the tiling mechanism does not allow for all tiles to be processed in parallel, Figure 7.4 shows that the maximal achievable speed-up does approach the number of processors when the memory reduction is high, i.e. when the number of tiles becomes large. This graph shows that having more processors does not always mean more speed-up. More processors become beneficial when the need in memory reduction is strong, that is to say when dealing with very large images, or when very little memory is available. The local maxima of the curves are points where the parallelization degree is the highest. For instance, the first local maximum of the 5 processors curve is placed at a memory reduction of 100, which is a 10x10 tiling, meaning each row has exactly five white tiles, so the processors are never idle. The saw-tooth behavior is due to the fact that some processors will be idle when the number of tasks is not a multiple of the number of processors.

The full lines in figure 7.4 show the measured memory reduction for different numbers of processors. The image used for these experiments is a 11000x11000 time series of Sentinel-2 images, sampled at 10m, with 33 dates each containing 10 spectral bands. The total size of this image is 75Gb, making it impossible to process without tiling. The experiments are run on a shared memory architecture, with Intel® Xeon® CPU E5-2650 v4 @ 2.20GHz processors that communicate using the Message Passing Interface (MPI) protocol. For each tiling layout, the computation time with several processors is divided by the computation time with one processor to obtain the speed-up value. As predicted, the measured speed-ups are always lower than the theoretical speed-ups, but clearly follow the same trends, and show the same local extrema, meaning the theoretical curves can be used to calculate the optimal memory reduction parameter for a given number of processors. The difference is stronger as the number of processors gets higher, because of the need for synchronization in a parallel computing environment.

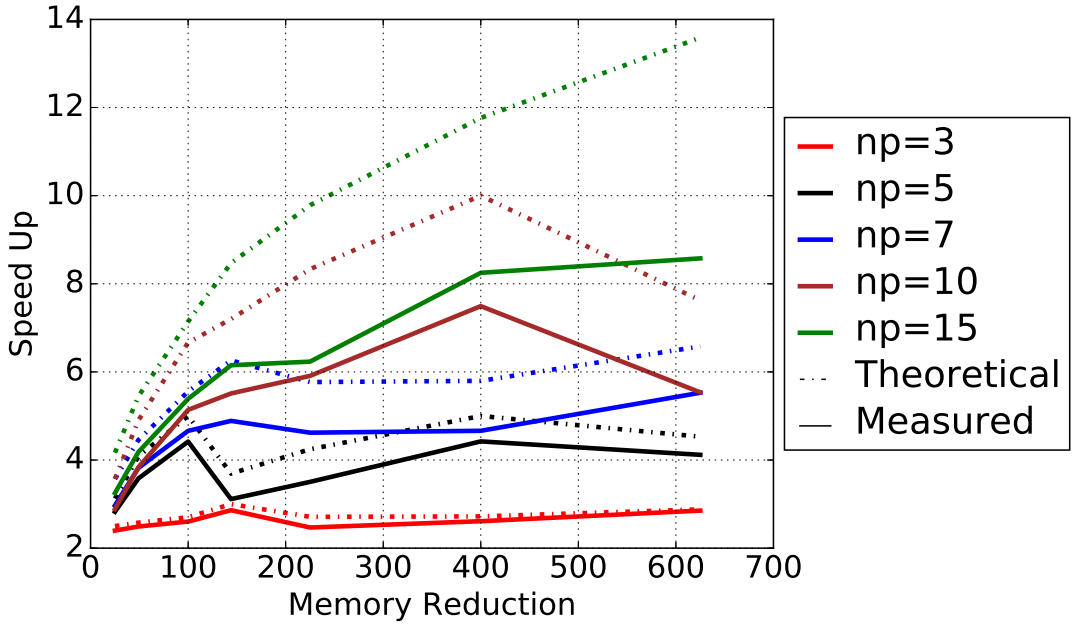


Figure 7.4: Memory reduction vs. Speed-up for different numbers of processors. Memory reduction is the ratio of the size of one tile to the size of the whole image. Solid lines represent the measured speed-ups, while the dotted lines represent the theoretical speed-up as given in equation (7.7). The measured speed-up is lower than the predicted speed up, but they exhibit the same trends.

A more complex, and perhaps more optimal scheme could be set up in which a white tile is processed as soon as the two diagonally adjacent white tiles in the row above it are finished. However, compared to the simpler row-by-row strategy, this would only become beneficial when processing with a very large number of tiles. This idea is not considered further, seeing as the proposed strategy provides sufficient scaling capabilities.

7.2.4 Estimating the optimal tiling parameters

Seeing as the final objective is to provide a fully automatic scalable method that works on any image without user interference, it is necessary to calculate the number of required regions for the processing to fit in memory. For this calculation, we will assume the following parameters are known:

1. The image I is of size $X \times Y$;
2. The margin value is $m = 3 \times SpW$ in pixels;
3. The available memory, T bits;
4. Input pixel size, p_{in} bits;
5. Output pixel size (label coded on an integer) p_{out} bits;

The tiling is entirely defined by two parameters, N_x and N_y, which represent the number of divisions along the X and Y axes of the image. For example, in the illustration in figure 7.2, $N_x = 5$ and $N_y = 5$. The problem can be formulated as finding the values of N_x and N_y that satisfy the memory constraint. The finer the tiling, the higher the total margin surface, which means extra memory overhead and in the end, longer processing times. Therefore, the objective is finding the largest tile size that fits in memory. The total margin m_{tot} can be seen as a cost function, and the problem as a constrained integer optimization problem.

- Variables: N_x, N_y
- Minimize:
$$m_{tot} = m(XN_y + YN_x)$$
- Constraint:
$$T > \alpha \frac{XY}{N_x N_y} + \beta\Big(\frac{X}{N_x} + \frac{Y}{N_y}\Big)$$

where α and β depend on the input parameters.

$$\alpha = 2p_{in}$$

$$\beta = (2p_{in} + p_{out})2m$$

The difficulty is finding an integral solution to this problem. Dedicated libraries could be used to find an exact solution, however in this particular case, as the constraint function is strictly decreasing with respect to the input variables, a valid approximate solution is the ceiling of the real solution. The real solution is simply the solution of a 2nd degree polynomial. This provides us with a solution that is always valid, and while not necessarily optimal, it is a good approximation of the optimal solution.

$$N_x^* = E(\frac{X\alpha}{\sqrt{\alpha T + \beta^2} - \beta}) + 1$$

$$N_y^* = E(\frac{Y\alpha}{\sqrt{\alpha T + \beta^2} - \beta}) + 1$$

7.2.5 Experimental results

The experimental validation is done in two parts:

1. Validating the tile-wise method by comparing the results with and without tile-wise processing, on a small 4000x4000 Pleiades image, with 4 spectral bands and a total size of 123MB, shown in 7.5.
2. Validating the scalability by testing the procedure on a large image: a 11000x11000 Sentinel-2 time series, with 330 features per pixel, and a total size of 75GB.

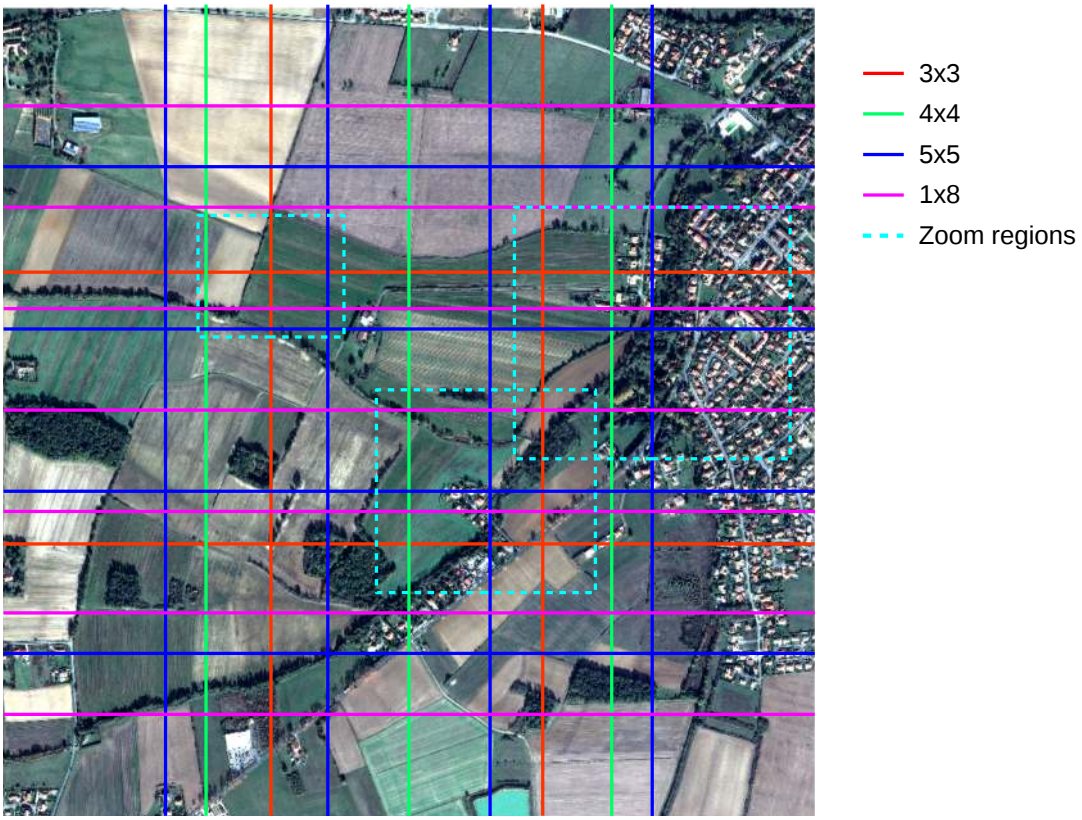


Figure 7.5: Different tiling layouts for the tests on the Pleiades image. The tile edges cover a variety of inhomogenous terrains. Some of the edges of the layouts cannot be seen in this figure as they are covered by others. Three regions were selected for the illustrations in Figure 7.6.

First, a visual comparison between SLIC with and without margins exhibits the nature of the anomalies on tile intersection areas. Figure 7.6 shows results of the method on three intersection zones, covering relatively homogeneous natural areas, as well as more heterogeneous urban areas. The layouts and the zoomed regions can be seen in figure 7.5. In a second experiment, many different layouts are tested to show the robustness of the method to a variety of terrains at the tile edges. The different layouts for the second experiment are shown in figure 7.5.

The zoomed regions and the tile intersections are shown in Figure 7.6a. Figure 7.6b shows the result of SLIC applied on the whole image, with no tile-wise processing. The image is small and fits in memory, so this result can be generated for validation purposes. Figure 7.6c contains the result of SLIC when applied tile-wise with no margins. The three types of anomalies cited earlier are found in this example.

1. Straight edges along the border zone
2. Abnormally large superpixels, especially on the intersection of all four tiles
3. A disjoint segment (marked with a star)

Figure 7.6d shows the segmentation result of SLIC applied with the tile-wise strategy explained earlier. None of the previously mentioned errors are seen in this result, and it is very similar to the result without tile-wise processing, shown in figure 7.6b. In figure 7.6e, the difference between tile-wise processing with no margins and applying the algorithm on the whole image, i.e. between figures 7.6b and 7.6c is shown. A pixel is colored white if the two labels are different. Figure 7.6f shows the difference between figures 7.6b and 7.6d. The segmentation result when using margins is clearly much closer to the original. However, as predicted earlier, there are still minor differences. Figure 7.6 shows that the method is robust to heterogeneous terrains such as urban areas. In fact, there are generally fewer differences in such areas, because the result of a superpixel segmentation is more sensitive to the initial superpixel grid in homogeneous areas, as can be seen in figure 7.6f.

To see if these differences have an impact on the overall segmentation statistics, the unsupervised segmentation criteria have been measured on the three cases above, for different values of the spatial width, with a distance weight of 1. In order for the tile edges to cover a variety of terrains, tiling layouts ranging from 2x2 to 5x5, as well as a rectangular layout, 1x8, were tested, as shown in figure 7.5. In these cases, the tiles size range from 2000x2000 to 800x800. In the 1x8 case, tiles are 4000x500. These results are shown in figure 7.7. Figures 7.7a, 7.7c, 7.7e and 7.7g show the results when processing without margins. The overall criteria clearly vary according to the tiling layout. Generally speaking, the differences are more important when the tiles are smaller (5x5 tiling layout), because the number of segments affected by the tile edges increases. Figure 7.7e shows that the explained variance is slightly lower when processing without margins, which indicates less homogeneous regions. The differences in RSD of segment size are explained by the abnormally large segments on tile edges. In some cases the values without margins are equal to the values without tile-wise processing, for instance, 2x2 tiling layout with a spatial width of 50. These points appear when the tile size is a multiple of the spatial width, meaning there are no segments that are initially on two tiles. The perimeter compacity increase is due to the straight borders on tile edges, as opposed to the rough edges which appear on most segments. Figures 7.7b, 7.7d, 7.7f and 7.7h show that when using the tile-wise processing strategy with margins, the overall characteristics are indistinguishable from when the algorithm is applied on the whole image. While the segmentation result is not precisely the same, the margin strategy guarantees that these two cases will be similar in terms of their global characteristics.

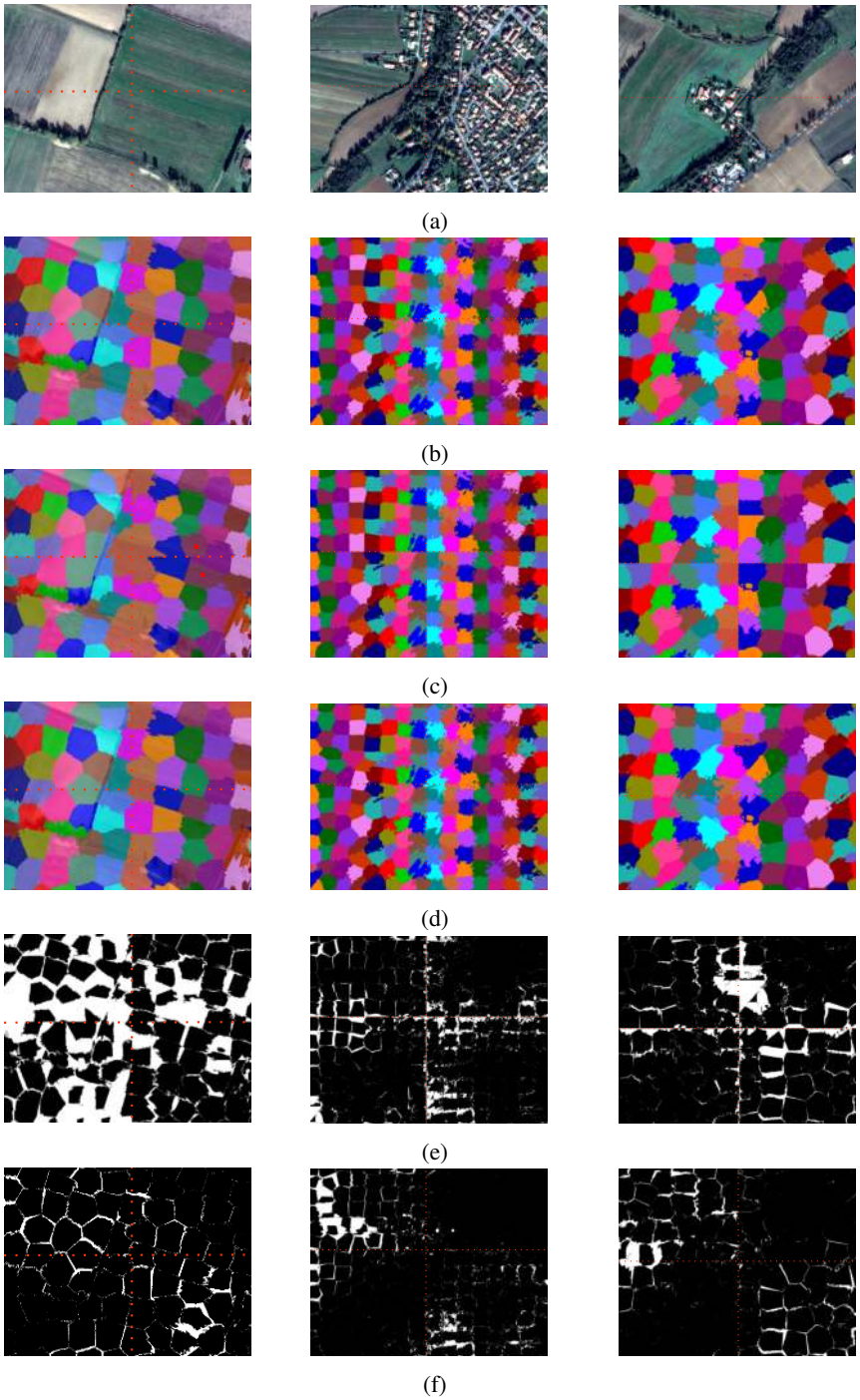


Figure 7.6: (7.6a) Zoom on the intersection regions between tiles. The intersection areas cover a variety of both agricultural and urban covers, to ensure the robustness of the method to heterogeneous terrains
(7.6b) Result of SLIC applied on the entire image with no tile-wise processing.
(7.6c) SLIC applied on each tile independently with no margins. Linear tile edges and abnormally large superpixels can be seen, as well as a disjointed segment marked with a star.
(7.6d) SLIC applied with tile-wise procedure, with margins of 3 superpixel widths. No more anomalies can be seen on the edge areas.
(7.6e) Difference between SLIC on entire image and tile-wise SLIC with no margins (a white pixel means the segments are different). There are many differences between the two segmentations, especially around the tile edges.
(7.6f) Difference between SLIC on entire image and SLIC with tile-wise processing and margins of 3 superpixel widths. There are still minor differences between the two segmentations. These are concentrated in smooth terrains, where small differences in the initial superpixel grid can cause large differences in the segment boundaries, as there are no strong gradients for them to adhere to. The differences are almost all eliminated in rougher terrains, such as built-up urban areas.

7.2.6 Overview and validation

To validate the tiling parameters of the segmentation, namely the number of tiles in the X and Y direction for a given problem, the SLIC algorithm is applied to a large Sentinel-2 time series. This image is the stacking of 33 dates of Sentinel-2 acquisitions, in 10 spectral channels, on a 110x110km region. All channels have been sampled at 10m. The total size of this image is around 75GB, which clearly exceeds the 16GB of available memory on the computer used for testing. In this case, the parameters recommended by the theoretical developments in Section 7.2.4 are a 10x10 tiling.

Table 7.1 shows the peak memory measured on a run of the SLIC algorithm. These measurements were made using the valgrind massif tool [Nethercote and Seward, 2007]. The optimal tiling calculation was set with a maximal memory of 2GB for validation purposes. The table shows a comparison of the case where the optimal tiling is used, and the case where bigger tiles are used. It shows that calculating the tiling parameters allows the algorithm to best respect the memory constraints.

Table 7.1: Memory profiling results on Sentinel-2 tile. The optimal tiling parameters allow for the peak memory to remain under the maximal indicated memory of 2GB, regardless of the superpixel width.

Superpixel width (px)	Optimal tiling	Tested tiling	Peak mem. (GB)
20	10x10	10x10	1.893
20	10x10	9x9	2.277
2	11x11	11x11	2.000
2	11x11	10x10	2.350

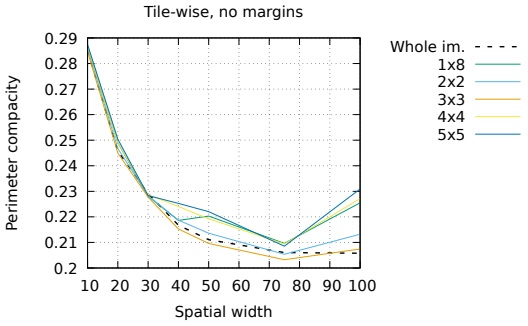
Figure 7.8 shows a small region (1300x1000) of the result the application of the tiled algorithm to an entire Sentinel-2 time series. The linear and round elements that can be seen in the water are linked to the temporal interpolation of cloudy images on other dates in the year. This effect is stronger above the water than on land, where the superpixels adhere to the natural boundaries, and manage to separate the different agricultural plots from urban cover, and from forests. Two explanations for this observation appear:

1. Water is generally darker than dry land in satellite images, which increases the impact of missed clouds on the Euclidean distance.

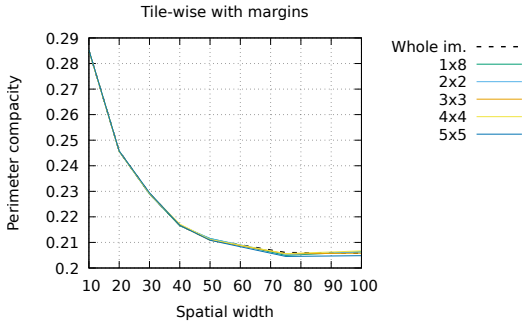
2. The number of missed clouds is greater over water, in general, due to imperfections in the cloud detection system.

Overall, it should be remembered that the proposed scaling strategy does not guarantee a result exactly the same as if the image had been processed in one piece. However, experimental tests using a modified version of SLIC on a Pleiades image show that the results are of the same quality in terms of four unsupervised segmentation criteria: perimeter compacity, statistical compacity, explained variance and relative standard deviation of segment sizes.

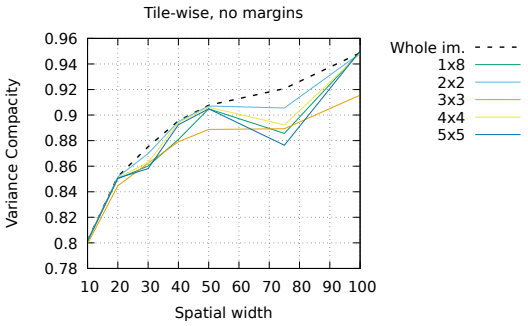
Finally, the scalability was validated by applying this method with SLIC on a real-size Sentinel-2 time series. This method allows us to process large images that were previously impossible to process on machines with limited memory. As the image is processed tile-wise, some tiles can be processed in parallel. The results presented in this chapter have shown that that although the speed-up is not necessarily a linear function of the number of processors, if the tiling is chosen wisely, a maximal speed-up can be obtained.



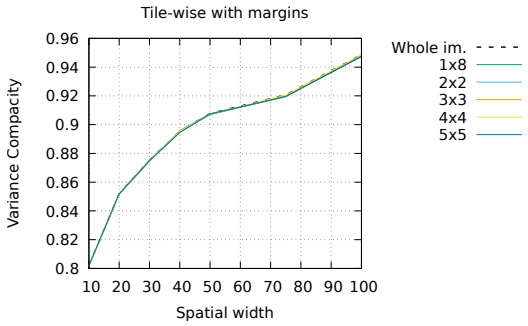
(a) Perimeter compacity, no margins



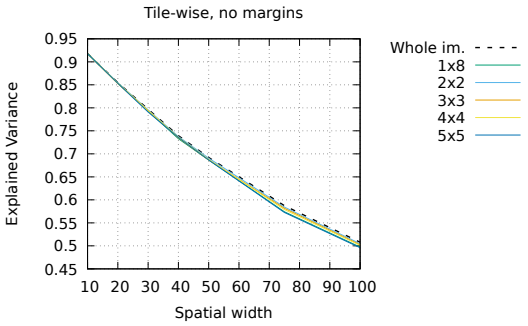
(b) Perimeter compacity, margin=3



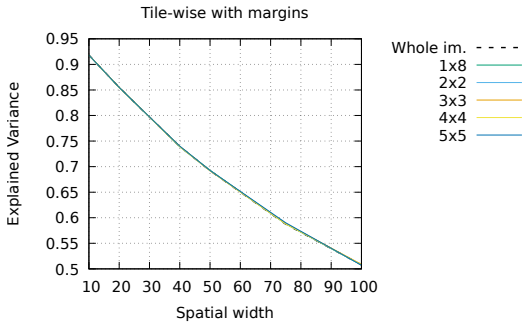
(c) Variance compacity, no margins



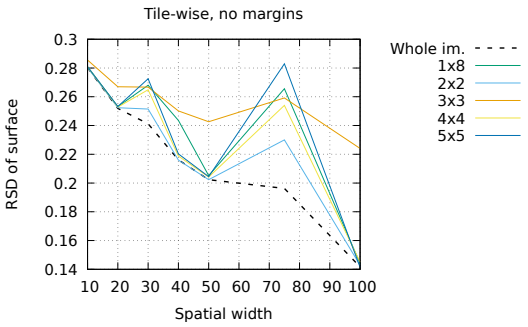
(d) Variance compacity, margin=3



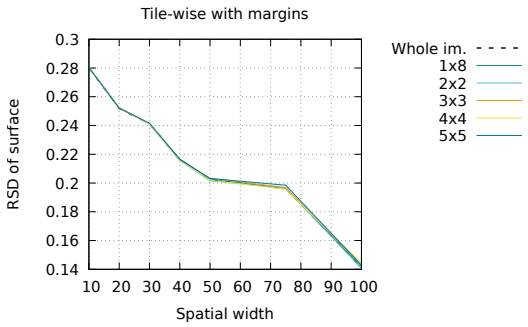
(e) Explained Variance, no margins



(f) Explained Variance, margin=3



(g) RSD of segment area, no margins



(h) RSD of segment area, margin=3

Figure 7.7: Comparison of segmentation statistics of SLIC applied to a SPOT6/7 image, for different tiling layouts, (2x2 to 5x5 and 1x8) and different superpixel sizes (spatial widths). If no margin is taken (left column), anomalies around tile edges change the statistics for different tiling layouts. When a margin is taken around the tiles (right column), the statistics become identical to when the image is processed without tiling, and no longer depend on the tiling layout.

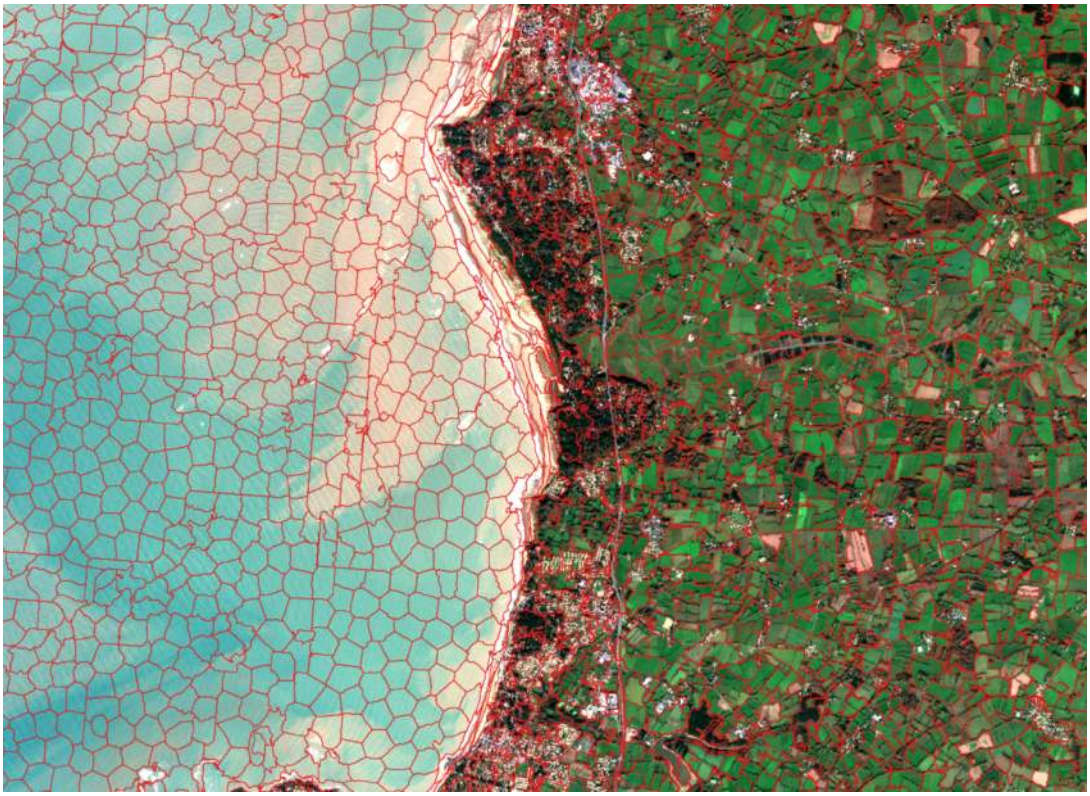


Figure 7.8: Demonstration of SLIC applied to a 11000x11000 Sentinel-2 time series of 33 dates zoomed on a small coastal area of 1300x1000. The background is the visible bands of the 1st date of the series.

8

Stacked contextual classification methods

"The solution often turns out more beautiful than the puzzle."

– Richard Dawkins

One of the *in fine* goals of this research work is to provide practical insights and tools for the inclusion of contextual information in a land cover mapping process, while respecting the constraints of operational production. This restricts the scope of the possible methods, for reasons that were mentioned in Part II, Chapter 5, and that are restated here.

1. The land cover mapping methods must be scalable, so that in the future, the techniques that are developed can be extended to wide areas. In other words, a global application should not pose any theoretical constraints. Supervised classification methods are a tool adapted to solve such a problem, as they are able to deal with large volumes of data in a reliable manner, given a sufficient quantity and quality of training data.

2. The use of high-dimensional images is key for describing many of the target classes, which also strongly limits the use of methods that are designed for low-dimensional images.

3. The sparsity of the training data is an aspect that cannot be overlooked, as it can prohibit certain methods which require nearby training points.

These factors drive the direction of the research towards light-weight classification methods, with relatively low-dimensional contextual descriptors, and that are suitable for training with sparse data.

The OSO land cover mapping problem [Inglada et al., 2017], is an interesting case study, as its class nomenclature contains several classes difficult to recognize on a pixel basis alone. If these context-dependent classes can be classified more accurately, a richer variety of land cover types can be targeted in the future. Moreover, it represents an operational land cover mapping case over a wide area based on high-dimensional images with sparse training data.

These *contextual relations* or *contextual dependencies* exist at several different scales, but a first simple distinction can be made between relations within a given object, and relations between groups of neighboring objects. These two kinds of relations, called *intra-object* and *inter-object* relations are shown in figure 8.1.

Short-range descriptors seek to characterize pixel relations inside an object, called *intra-object relations*. Features similar to those presented in Chapter 5 are often used to describe these relations. These features characterize notions like texture and auto-correlation, which are often calculated within objects. These features are useful in land cover mapping, in particular for agricultural areas with a high degree of variability. Such areas have well-defined outlines, but an inner texture that confuses some of the pixel-based classifiers. This can be due to differences in water access, to slopes, or simply to the size and sparsity of the vegetation. This is one of the sources of error in many image classification problems, and will be referred to as *intra-object classification noise*.

The second type of relation that can be qualified as intra-object is linked to the disposition of the pixels within an object, which forms the shape of the object. For example, roads and rivers are land cover classes that are characterized by an elongated shape. Compared to lakes and parking lots, the difference lies in the spatial arrangement of the pixels within the object, with respect to neighboring objects of different classes.

Secondly, there are relations between an object and its neighboring objects, called *inter-object relations*. These exist in many image classification problems, both in computer vision and land cover mapping. Inter-object relations can appear between certain classes which have a high probability of being near each other. For instance, industrial

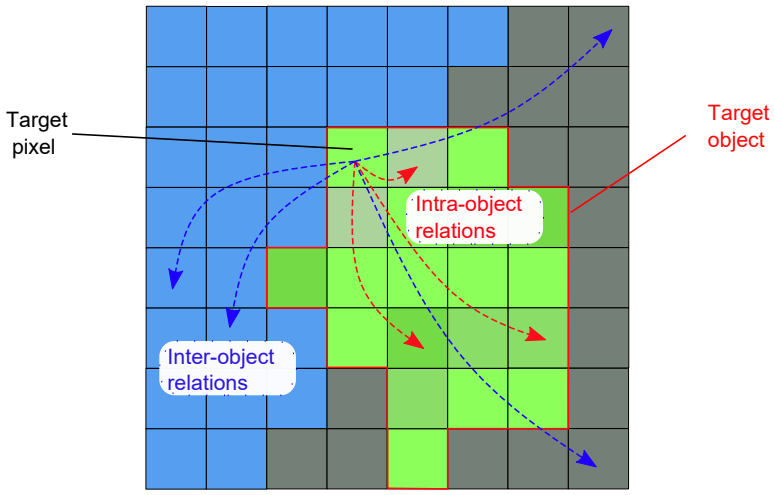


Figure 8.1: Intra-object relations are contextual dependencies within an object, whereas inter-object relations express links between pixels of nearby objects.

areas must have road networks and parking lots connecting them, to allow access for workers and clients. In many cases, beaches and dunes are situated near water bodies. These relations are challenging to model, as they can sometimes represent long-range dependencies. Therefore, they require using an extended spatial support covering a variety of different objects which can be difficult to describe with few features.

One possiblility is to model inter-object relations is to examine the shapes formed by a segmentation of the image. This is the idea behind qualitative spatial reasoning. This method involves applying a hierarchical image segmentation to extract a description of the shapes of the objects. [Inglada and Michel, 2009] integrated this process into a model which uses shape descriptors contained in a graph-based representation to perform object detection on VHSR imagery. However, these methods depend strongly on the quality of the segmentation, and the ability to accurately estimate the shapes of objects. This is questionable at a 10m spatial resolution, where discontinuous and continuous urban areas are marked by a close-knit texture that does not truly reflect the shapes of the underlying objects. For this reason, methods that rely solely on describing shapes are not considered further.

In fact, many of these inter-object relations translate notions that we can understand on a *semantic* level and not on a shape or on a feature level. In other words, relations between different class types rather than between shapes, or between the aspect of the neighboring pixels. For instance, saying that industrial areas must be near roads is a semantic relation. Saying that roads are elongated or that roads are surrounded by large orbjects is a shape relation. Finally, saying that roads are surrounded by green pixels would be a feature relation. Incorporating semantic relations into a classification scheme can be very promising for modeling inter-object relations.

However, there is a kind of Catch-22 paradox : how do we know that a certain land cover class is nearby, if the pixels have not yet been classified ? Indeed, if we wish to use semantic information from nearby pixels, they must first be labeled, which is not the general case of pixels in an image, when sparse training data is used. However, it is always possible to consider using an initial prediction of the class labels, albeit containing errors, to derive contextual features. This process can even be repeated any number of times, in an iterative manner. An approach that applies several successive steps of classification using the results of the previous step is known as a *stacked classification* approach.

The following section presents the literature, and theoretical background related to this group of methods. Then, a new stacked contextual classification method, which is the principal methodological contribution of this Ph.D. is presented, in Section 8.2. This classification scheme includes contextual information through local class histograms in superpixels, and is called the Histogram of Auto-Context Classes in Superpixels (HACCS) process.

8.1 Using the prediction of nearby pixels

One of the difficulties in applying contextual classification methods is the very large number of total features that are required to fully describe a large context. The number of pixels contained in a neighborhood is in quadratic

proportion to the scale of the desired context. In the case of VHSR imagery, a very large number of pixels can be required to capture long-range information. Moreover, in the case of multi-temporal images with several spectral bands, the spatial resolution is generally lower, but the number of features per pixel is high. In both cases, providing every single band of every single pixel in the neighborhood as an input to the classifier is questionable. Indeed, supervised classification in high-dimensional spaces is known to present certain difficulties. This phenomenon, the so called "curse of dimensionality", can be linked to the sparsity of training data points in very high dimensional spaces [Friedman, 1997]. The training points are insufficient to properly model the high dimensional class-conditional probability density functions, which can lead to a poor classification. In addition, a very large number of features implies heavy memory and computational requirements. In land cover mapping, the combination of temporal and spectral information is necessary in order to distinguish the basic classes, however, it is difficult to include these directly in a contextual classification scheme.

Rather than attempting to consider both the spectral/temporal and the spatial information simultaneously, many studies have attempted to consider them in a sequential way. One way of doing this is to use a machine learning method (supervised or unsupervised) to predict a set of labels for some or all of the pixels in the image, thereby taking into account the dimensionality of the pixels. These predictions contain confusions related to a lack of contextual information, but have the advantage of being very low-dimensional. The objective behind this idea is to contain the high dimensionality of the image by projecting it onto a lower dimensional label space. The labels in a spatial neighborhood can be used to take into account certain contextual dependencies, in particular long-range semantic ones.

Moreover, the pixel-based prediction is a source of contextual information available everywhere in the image, as opposed to the training data which comes in a sparse form. This is an important characteristic for supervised classification.

It is also interesting to note that recurrent errors can be accounted for in the classification model, as they may provide a unique contextual characterization for each pixel. For instance, the combination of any *artificial class* (even if it is wrongly labeled) and *vegetation class* in the neighborhood serves as a strong indicator of the presence of *discontinuous urban cover*.

The following sections name some of the previous works that have been done in this direction: to study how a prediction of the neighboring pixels can be used to enrich the amount of information available for making a decision regarding the class label of a pixel. Then, the common points and differences between the different methods are summarized, in an effort to justify the choices made in the design of the proposed HACCS method.

Figure 8.2 illustrates how a high dimensional image can be classified to provide a low dimensional contextual characterization of a pixel, using a spatial support.

8.1.1 Bag of Visual Words

At sub-metric spatial resolutions, features such as the Scale Invariant Feature Transform (SIFT) [Lowe, 1999], the Speeded-Up Robust Feature (SURF) [Bay et al., 2006], or more recently the Point-Wise Covariance of Oriented Gradients (PW-COG) [Pham et al., 2016], aim to describe context by characterizing the high spatial resolution features, namely, the sharp gradients and local extrema in the vicinity. This is achieved by extracting so-called *keypoints*, which are meant to characterize the points of interest in the image, and should help describe its content.

The context of a pixel can be described by statistical information regarding the keypoints in its surroundings. One way of exploiting these points is known as the Bag of Visual Words (BoVW) method [Yang and Newsam, 2010]. It consists in applying unsupervised clustering to the set of keypoints, in order to extract a dictionary of *visual words*, which represent different spatial features, like corners or a local extrema. Then, histograms of these visual words are calculated within a spatial support to be used as contextual features. The histogram can also be calculated on the entire image, for instance for image classification or for image matching [Russell et al., 2006]. An unsupervised approach is used because an extensive nomenclature to characterize each of the low level classes in the image is impossible to obtain. Another reason for using clustering is that keypoint features and texture features often present a very large dimension, and clustering reduces the dimension of the contextual feature to the size of the visual dictionary.

This method provides an unsupervised labeling of the high spatial frequency elements that surround the target pixel, but does not directly take into account semantic relations between neighboring objects.

8.1.2 Random Fields

Many supervised classification methods aim to determine the class label l of an unlabeled sample x, via the so-called *class posterior probability density function*, also known as *class-conditional probability density function* or *likelihood of class membership*, noted $p(l|x)$. For each of the labels in the class nomenclature, this posterior

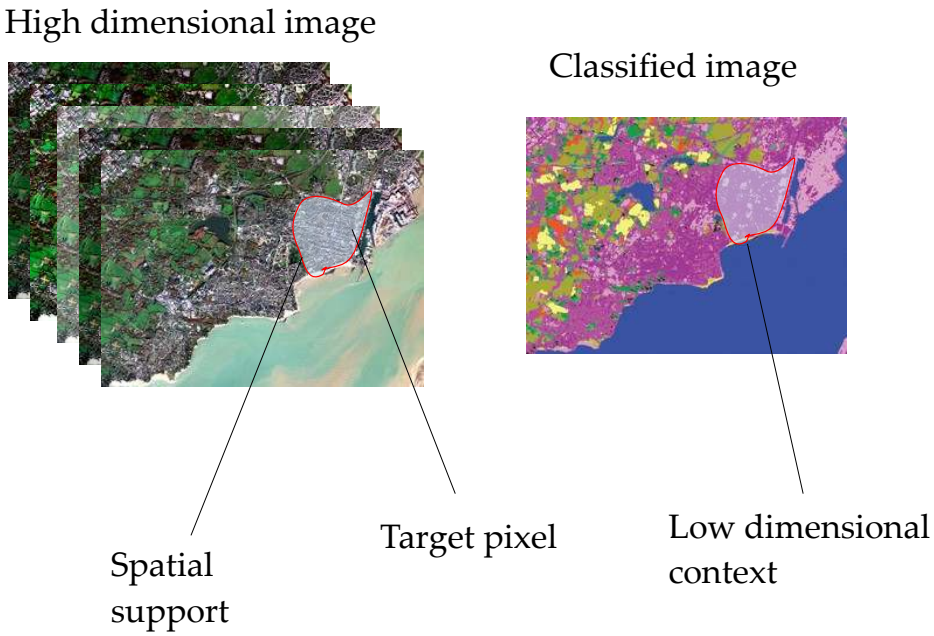


Figure 8.2: Illustration of the principle behind using a prediction of the neighboring pixels to provide a low dimensional contextual characterization of a pixel in a high-dimensional image.

probability indicates the likelihood that the sample x, given its features, carries the label l. If these functions can be estimated, the sample can then be classified with the label with the highest probability. The vector of class probabilities is sometimes called the *soft* or *fuzzy* label. Popular classifiers including the Support Vector Machine, as well as Neural Networks, and Random Forest are able to provide an estimation of the class posterior probabilities, which are often used as an indicator of the confidence of the classifier. For example, a very strong probability value for one label compared to the others indicates that the classifier is certain of its decision.

One popular group of methods that use the class-conditional probability density functions from a pixel-based classifier is known as *Random Field* methods, introduced as Markov Random Fields (MRF) by [Pieczynski, 1989]. Later works by [Melgani and Serpico, 2003] show how these models can be used to combine spatial, temporal, and spectral information. MRFs are based on the output of a soft pixel-based decision of both the pixel itself (spectral), and the prediction of the pixel at the same location, but at a nearby date (temporal). Moreover, the soft pixel-based classifications of nearby pixels in the image (spatial) are included in the model. The first step involves training pixel-based classifiers using an available data set at both dates. Then, these predictions enter into a model, which minimizes energy functions to enforce coherence, both within a spatial support, but also between dates. In other words, the model recognizes valid and invalid class transitions, using the transitions observed in the training data set. In the temporal domain, this can correct for many obvious errors linked to improbable or even impossible land cover changes. In the spatial domain, class transitions can be seen as the different possible shapes within the neighborhood, for instance, a strong edge between two classes, or an isolated pixel surrounded by different classes. However, these methods are designed for classifying mono-date images given the neighboring dates. This can help in classifying each date of a time series, but associating one label to an entire time series is a different issue altogether. Moreover, the 3×3 neighborhood proposed in the study is too small to take into account long-range inter-object semantic relations.

A more recent study by [Zhao et al., 2016] uses Conditional Random Fields (CRF) to include contextual information in a supervised classification scheme on VHSR imagery. The CRF takes three levels of information. Firstly, it incorporates the spectral information by using the unary potentials from an SVM classification of the pixel, similar to the approach designed in [Moser and Serpico, 2013]. The spatial information as well as the class conditional probability density of the neighboring pixels, are inputs to the model. Moreover, the class conditional probability density of nearby training samples in the image area are required, in an effort to include long-range information. This method performs well in areas with relatively dense training samples, but the authors do not address the case where no training samples are available. Naturally, these areas have higher error rates than areas near training data, because they can be different in terms of class behavior and content.

8.1.3 Stacked classifiers

Generally speaking, classifiers that use the result of a previous classification as features for a new classification are part of a group called *stacked classifiers* or *cascade classifiers*. The fundamental idea behind this approach is very interesting: teaching a classifier to recognize and correct errors of a previous classification. This can be done in several ways. The most basic idea involves using the predicted label as a feature, but the class-conditional probability densities, or other features from the classifier can be used. For instance, if the first classifier is a Random Forest, the vector formed by the predictions of all of the trees can be used as a feature for the next classifier. If the classifier is a neural network, it is possible to use the output of intermediate layers as features, as is done in [Hu et al., 2015].

A study by [Cohen, 2005] uses the predicted class labels of "nearby" points in the data set, to improve the prediction accuracy. Naturally, the notion of proximity depends on the problem at hand. In their natural language processing problem, the authors use the textual distance (number of words separating two words) to define the nearby points.

In image classification problems, pixels belonging to the same spatial support can be considered as nearby. For example, [Munoz et al., 2010] propose to use segments from a hierarchical segmentation method, similar to the object segmentation methods presented in Part II, Section 4.2, rather than square spatial supports. This provides more accurate results than the CRF approach on a computer vision problem, without even taking into consideration the probable improvement in geometric quality.

In another computer vision study, [Larios et al., 2011] use a Random Forest to classify a set of SIFT-like keypoints, and a histogram of the label scores from the various trees in the RF are then used as features by a SVM classifier, to classify the entire image. This resembles the Bag of Visual Words approach, with the difference that a supervised classification method is used instead of an unsupervised method to form the visual dictionary.

Applying this method to a dense classification problem rather than an image classification problem would require computing the histogram in a spatial support. However, this is quite complex to do on high-dimensional images, due to the large number of features generated by SIFT keypoints, and seems more relevant on low-dimensional images.

8.1.4 Semantic Texton Forests

Another approach that involves using the result of a previous dense classification of the image is the Semantic Texton Forest (STF), introduced by [Shotton et al., 2008] for computer vision problems. The STF method can be separated in two steps.

Firstly, a great number (millions) of features are used to generate the splits of the Random Forest. These features are simple functions of raw image pixels within a sliding window around the target pixels: either the raw value of a single random pixel, or the sum, difference, or absolute difference of a random pair of pixels. The optimization of the purity criteria guarantees the use of relevant features that best split the training data. Random pixels and pairs of pixels in the neighborhood provide a contextual characterization, in a way similar to how the CNNs consider the different combinations of neighboring pixels through convolutions. These notions are defined in Chapter 9.

This step ressembles a model-based approach in the sense that it provides the entire context, in a sliding window, to the classifier. Indeed, there is no computation of particular contextual features. The classifier recieves only the pixel values, and some basic combinations of these. This very important step of the STF will be referred to here as the Basic Semantic Texton Forest (B-STF), and is discussed further in Section 8.3.

Secondly, for each pixel, two more contextual features are calculated in a neighborhood:

1. The histogram of the tree responses.

2. The histogram of the split nodes.

The *tree response* is defined as the vector of labels provided by the forest, as each tree provides one label. This implicitly contains the information of the decision that the previous stage of classification has reached, which following the Random Forest logic is the most common label in the forest. Moreover, given that each tree is trained on a different sampling of the training points, certain patterns can emerge in the tree responses, which might characterize a particular class behavior.

The list of split nodes provides information regarding exactly which pixel features or pairs of pixels were the most relevant in the given neighborhood.

Semantic Texton Forests were used by [Fröhlich et al., 2012] for building facade recognition. In their *combined Random Forest* approach, object spatial supports from a Mean Shift segmentation are used to compute the mean class-conditional probabilities, along with a large number of other features similar as the ones used by the STF. This approach was also used by [Shapovalov et al., 2013], for point cloud classification.

It is worth mentioning that using a very large number of random features in this way was first done in a computer vision application by [Lepetit and Fua, 2006]. Their proposed approach involved selecting features from millions of random pixel pairs to calculate the split nodes of a Random Forest, in order to classify keypoints in an image.

Another recent study by [Hänsch and Hellwich, 2018] considers the application of stacked Random Forests to SAR images. This study investigates the simultaneous use of two types of contextual features. First of all, the distance between one two or four random rectangles in the neighborhood is used as a feature, based on distances specific to SAR imagery. The number, position, size, operator, and distance measurement are all randomized during training. Secondly, the distance between one, two, or four random rectangles in the neighborhood are once again randomized during training and used as features. For example, the Histogram Intersection, city block distance, Euclidean distance, Kullback-Liebler distance, Bhattacharyya distance are all potential distances that can be used during training. Another innovative aspect of this approach, compared to the previous versions of the STF, is the fact that the authors stack several stages of classification, rather than only applying this process once.

Methods originally designed for low dimensional problems are rarely directly applicable to large, high-dimensional satellite images or time series of images. Indeed, techniques that involve randomly selecting one of millions of possible contextual features at each split pose several practical issues in terms of implementation. These practical issues would need to be solved in order to evaluate the performance of these methods. While considered possible in theory, this is beyond the immediate scope of this Ph.D. work, seeing as much simpler methods can be evaluated first. Nonetheless, ideas for possible work-arounds that would make this algorithm suitable to the current classification framework are discussed in the perspectives, in Chapter V, Section 12.4.

However, on low dimensional imagery, a basic version of the STF can be tested without much difficulty, as is explained in Section 8.3.

One very interesting idea proposed by the inventors of the Semantic Texton Forests is the second step of the algorithm. Namely, the way in which this method uses the decision of a previous stage of classification, in an adaptive neighborhood, to include contextual information. This is contained within the vector of tree responses in [Shotton et al., 2008], and is present as the mean vector of class posterior probabilities in [Fröhlich et al., 2012]. Such low-dimensional features are viable for application on large, high-dimensional images. This idea is the basis of the approach that was designed and tested during this Ph.D.

8.1.5 Auto-Context

Auto-context was designed for an application on a two-class dense classification computer vision problem [Tu, 2008, Tu and Bai, 2010]. Here, each pixel is classified using a number of contextual and non-contextual features, such as color and texture. While the initial classification is coherent, it lacks fine geometrical details in corners, and has the tendency to blur out sharp elements. This is possibly caused by the use of sliding window neighborhoods for calculating the contextual features, which can smooth out the fine details of the classification. In their study, the authors use successive iterations of pixel classification using the same classifier and training data, but add supplementary features based on the predictions of the previous iteration. Specifically, the mean of the vector of class probabilities provided by the classifier, in several regions surrounding the pixel is used. As there are only two classes, this represents a very low total number of extra features, which allows a large number of different neighborhoods to be considered simultaneously. Selecting these neighborhoods carefully allows different aspects of the problem to be learned by the classifier, like spatial relations between classes, or relative positional information, which can be very useful. Applying Auto-context several times refines the details of the geometry and improves the quality of the output classification. The study shows that a large number of iterations is not necessary, as no more notable changes occur are made after 5-6 iterations. This makes the process light and fast, and applicable to classification over wide areas.

The Auto-Context method addresses a similar problem as the one encountered in land cover mapping, namely, the difficulty of classifying context-sensitive areas while preserving the sharp corners and fine elements, on a very large dimensional data set.

8.1.6 Summary of the literature

The methods cited in the previous sections can be described by five properties, given in the following list.

1. The density of the initial prediction: certain methods predict the class or cluster of keypoints, whereas others use every single pixel in the image.

2. Supervised or unsupervised prediction: some methods prefer the use of clustering rather than supervised classification for the initial prediction.

3. The contextual features, or lack thereof, used for the initial prediction: the first prediction can be pixel-based, or can already be based on the use of pre-selected contextual descriptors.

4. The number of iterations: certain methods stop at the second prediction, while others use successive predictions to improve the classification in an iterative way.

5. Adaptive spatial support. The definition of the context can be a sliding window, or an adaptive spatial support.

Table 8.1 shows the characteristics of the different methods that were cited previously. Clearly, the nomenclature of the different methods is not very well defined, as they originate from a variety of backgrounds and applications. None of the methods possess all five of the characteristics. Table 8.1 also shows that the the HACCS method, which is described in the following section, can in some cases bear all five of these properties.

Table 8.1: Five main properties of methods that use a label prediction to include contextual information in a classification scheme. Certain methods with the same name are in fact very different, which is why such a table is necessary. The Histogram of Auto-Context Classes in Superpixels method, which is described in section 8.2, has the potential to verify all five of the properties.

Method	Dense	Supervised	Initial context	Iterative	Adaptive
MRF, CRF	X	X		X	
BoVW			X		
Stacked ([Munoz et al., 2010])	X	X		X	X
Stacked ([Larios et al., 2011])		X	X	X	
STF ([Fröhlich et al., 2012])	X	X	X		X
STF ([Shotton et al., 2008])	X	X	X		
Auto-Context	X	X	X	X	
HACCS	X	X	optional	X	X

8.2 Histogram of Auto-Context Classes in Superpixels

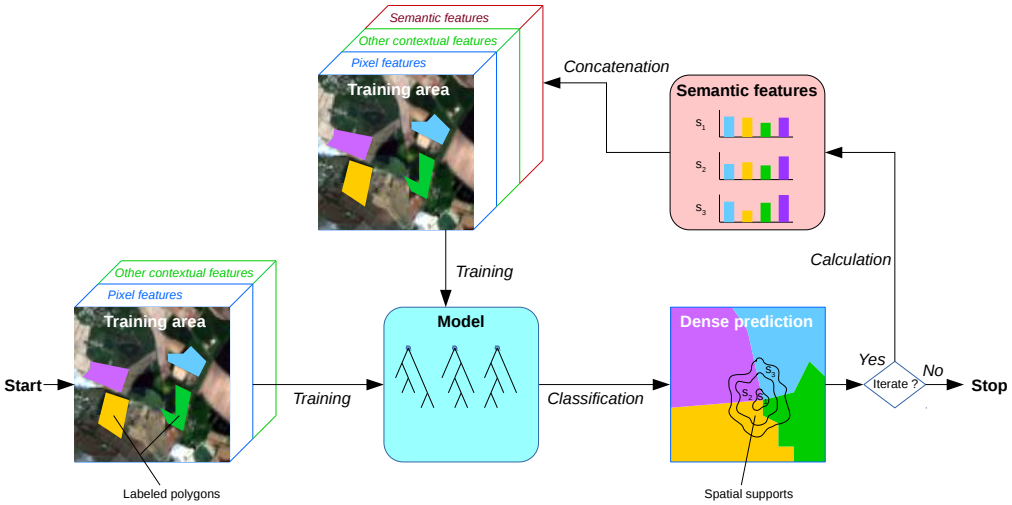


Figure 8.3: The complete Histogram of Auto-Context Classes in Superpixels process. A dense labeling of all the pixels is used to compute the histogram of the different classes. This histogram provides a contextual characterization, which serves as a supplementary feature for the next classification step, to refine the decision based on nearby contextual cues. The initial dense prediction can also originate from a different classification system, such as a D-CNN.

Figure 8.3 shows the different steps of the HACCS process. By using the pixel features, optionally in combination with other contextual features similar to the ones mentioned in Part II, Section 5, an initial classification model is trained. This model is then used to achieve a first dense classification of the image. Once all of the pixels

in the image are labeled, the histogram of these predictions is calculated in each superpixel neighborhood, and used as a contextual feature. This process can be iterated any number of times, using the prediction of the previous classification step to re-estimate the histograms.

8.2.1 Principle of HACCS

HACCS was designed to verify all five of the properties mentioned in Table 8.1, but was mainly inspired by Auto-Context.

Like Auto-Context, the HACCS method, condenses the initial features of the image, which can be high-dimensional in many cases, into a low-dimensional space by using a supervised classifier.

However, there are several differences between the Auto-Context method as presented in [Tu, 2008, Tu and Bai, 2010] and the HACCS process presented here.

First of all, a normalized histogram of the predicted labels is used rather than the output probability vector of a soft labeling classifier, in a similar way to the histogram of clusters used in the BoVW features. This is a choice that was made for computational economy, as a one-dimensional label map is more compact than an map containing the full vector of class-conditional probabilities. This implies that the entire process requires lower disk access time, which is beneficial for mapping wide areas. Considering the histogram rather than the mean vector of probabilities is basically equivalent to considering that the classifier has a 100% confidence on all of its predictions. However, the choice of the histogram as a feature could very well be replaced by the mean of the class-conditional probabilities as is done in the original Auto-Context studies.

Then, rather than taking sliding window neighborhoods as in [Cohen, 2005, Jiang and Tu, 2009, Fröhlich et al., 2013], or more recently in [Jampani et al., 2015, Huynh et al., 2016], superpixels are used as spatial support to calculate the class histograms, in a similar way to how mean-shift objects are used in [Fröhlich et al., 2012]. This encourages the conservation of high spatial frequency elements. As was stated in Part II, superpixel segments also allow for diverse pixels to be grouped in the same segment, even in very textured areas, and for the strong gradients between homogeneous areas to be respected.

The size of the grid at the first iteration of the superpixel algorithm (SLIC) is a parameter which is set manually, and which conditions the average size of the superpixels in the final segmentation. This parameter is referred to as the *scale* of the superpixels. This parameter defines the *characteristic diameter* of the segments, in other words, the square root of the average area of the segments. This gives an indication of the average distance at which contextual information is considered.

The choice of which scale or scales to use depends on the application, in particular, which context-dependent classes are targeted, as well as the spatial resolution of the images. Initial experiments on the HACCS process indicate that using histograms in several scales of superpixels is beneficial compared to using only one. Indeed, it seems reasonable that a multi-scale description of the neighborhood would be useful for characterizing context-dependent land cover classes, which depend on both intra-object and inter-object relations.

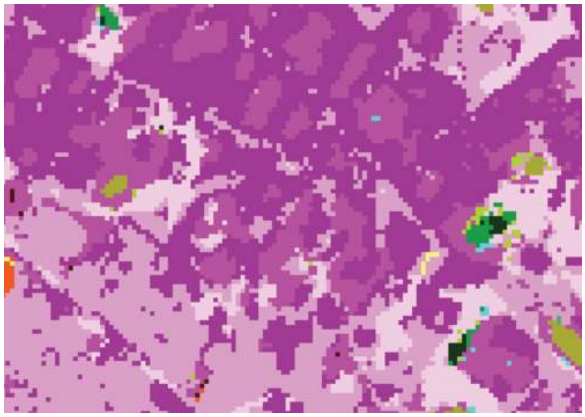
8.2.2 Illustrations

Figure 8.4 shows the evolution of three components of the histogram feature throughout the successive iterations of HACCS. In this example, three of the urban classes are mapped to the RGB channels of the image: Continuous Urban Fabric, Discontinuous Urban Fabric and Industrial and Commercial Units.

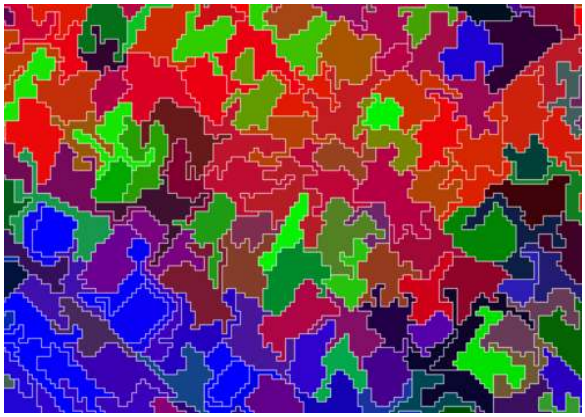
After several iterations, the edges of the images objects appear in the histogram features, as these become more and more homogeneous. This indicates that intra-object label heterogeneity is most likely being reduced at this scale. On the other hand, many of the superpixels still contain mixed labels after several iterations, because the decision is made at the pixel level, and not at the segment level as in OBIA. This means the method is robust to segmentation errors, particularly to under-segmentation, at least to a certain degree.

One important consideration regarding the HACCS process is the relative sparsity of the training data. In some areas of the image, many of the pixels are labeled, whereas in others, no pixels are labeled at all. Reference data is seldom collected in a homogeneous fashion, with certain areas often being prioritized over others. For instance, the Urban Atlas only contains cities with more than 100,000 inhabitants. One characteristic of supervised classifiers is that they often show very low training error rates. In other words, they are accurate when it comes to recognizing samples that are part of the training data. This implies that areas that are densely labeled usually have lower error rates than areas that are sparsely labeled. This is interesting, as the objective of the HACCS process, and of stacked contextual classification in general, is to use this pixel-based prediction to provide low dimensional contextual features. If an area of the image used for training is densely labeled, this means that the spatial arrangement of the predicted classes will be coherent with reality. Simply put, these dense training areas show the classifier what the

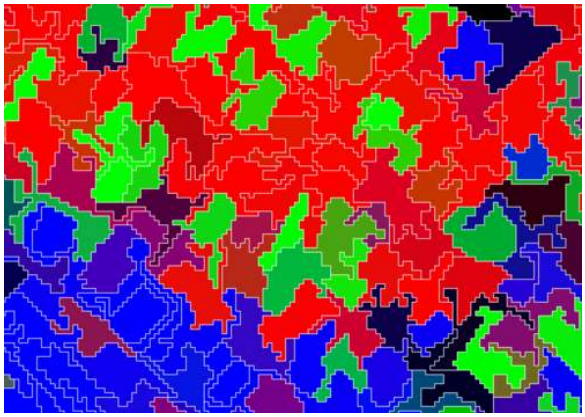
(a) First date of the Sentinel-2 time series overlaid with the SLIC segmentation at a scale of 10.



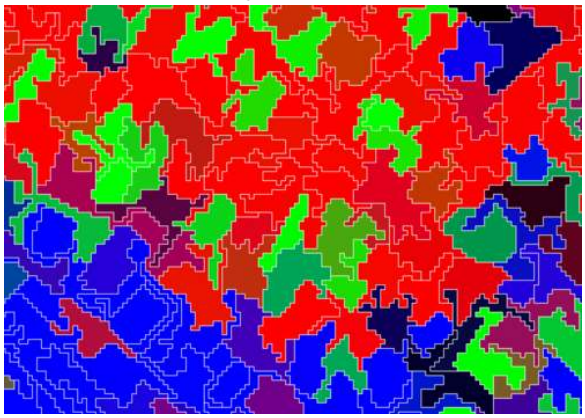
(b) Pixel-based classification of the area used for the estimation of the first histograms (figure 8.4c)



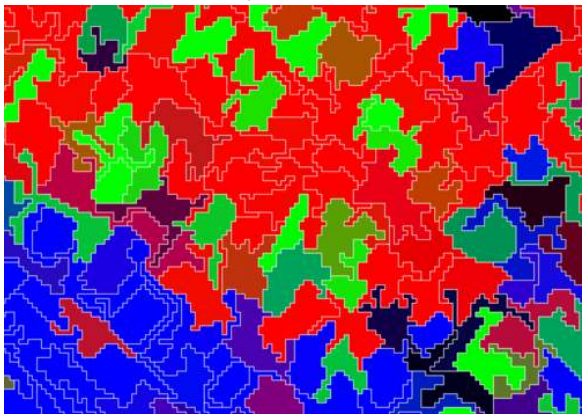
(c) Histogram at iteration 0.



(d) Histogram at iteration 1.



(e) Histogram at iteration 2.



(f) Histogram at iteration 3.

Figure 8.4: Visualization of three components of the histogram feature, over successive iterations of HACCS. The Red Green and Blue channels show the proportions of three urban classes: Continuous Urban Fabric, Discontinuous Urban Fabric and Industrial and Commercial Units. In the first iteration, the pixel-based errors, such as intra-object classification noise are visible in the histograms. Colors that are not Red Green or Blue indicate a mix of classes in the superpixel. With successive iterations, some superpixels become more and more pure, while others remain mixed. This shows that HACCS is not equivalent to a simple majority vote in the superpixels, and allows for superpixels with mixed classes.

context of each pixel should look like. Areas with contextual dependencies will therefore be well described in the training data, even though they are being generated by a pixel-based classifier.

Figure 8.5 shows how at the first iteration, unlabeled pixels that are in the dense training area will be associated with a contextual characterization that is coherent with reality. Then, during the second iteration, these pixels, which are now labeled with a higher degree of accuracy, will provide relevant contextual features for the pixels

that were originally labeled in the training data. This means that pixels in areas with little or no training data will receive the information from the dense training areas after two iterations.

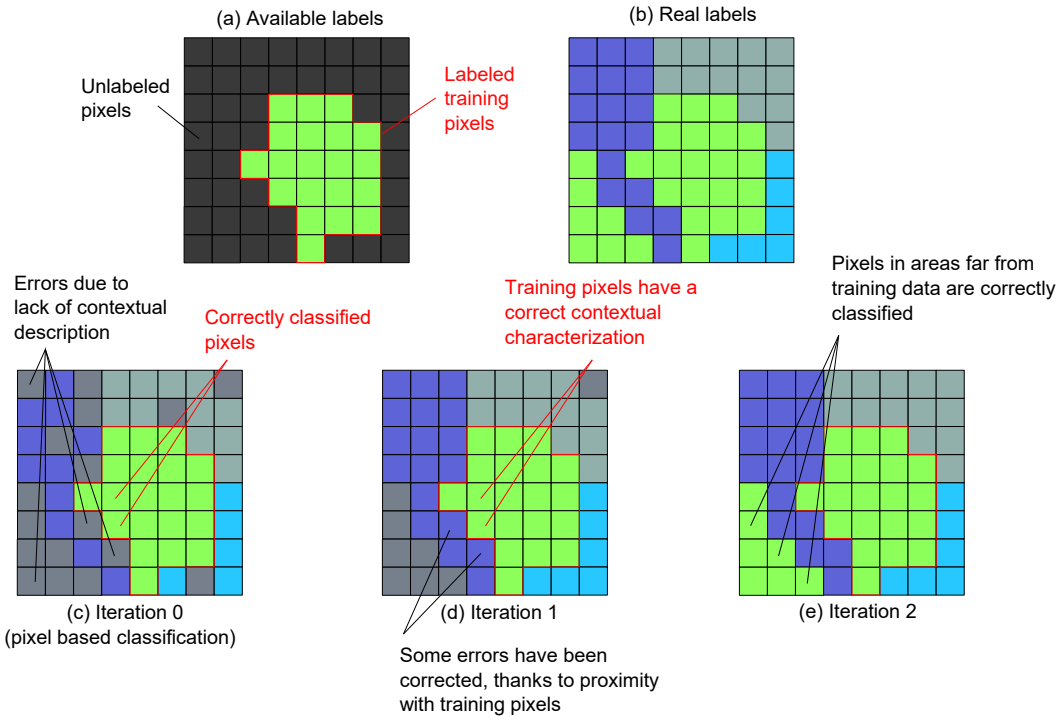


Figure 8.5: Illustration of the importance of successive iterations for any contextual characterization based on a prediction of nearby pixels.
(a) In this example, labels are only available in the green area in the center.
(b) The true labels, which are used for the illustration.
(c) Iteration 0 represents the pixel-based classification. Several errors can be found in the areas that are not covered by the training data.
(d) Iteration 1. Pixels nearby the training areas profit from the proximity of correctly classified pixels to find a relevant contextual characterization. However, pixels far from the training samples remain incorrectly classified, as their contextual features contained errors from iteration 0.
(e) After two iterations, the contextual characterization of the green training samples is correct, which allows the pixels in the bottom left of the example image to be classified correctly.

Being the main methodological contribution of this work, the HACCS process is evaluated in depth on two classification problems in Part IV. Indeed, HACCS was originally designed for land cover mapping using high-dimensional images over wide areas, but it is also evaluated on a very different classification problem, on a low-dimensional VHSR problem, on a relatively small area. The principal method with which HACCS is compared here is the object of a great number of recent studies : Deep Learning with Convolutional Neural Networks (CNN). These methods are nowadays considered as the state of the art of contextual classification. The following chapter provides the definitions and vocabulary of these methods, as well as the description of the main CNN approaches that have recently been used for land cover mapping. Moreover, some of the theoretical advantages and drawbacks of these methods are stated, which should help in understanding the results of these methods that are presented in Part IV.

8.3 Basic Semantic Texton Forest

The STF algorithm, as defined by [Shotton et al., 2008], is divided into two main classification steps. The first step operates a classification of each pixel using a model-based approach in a sliding window. The second step uses

contextual features in an adaptive spatial support, which can be image-based or semantic, as features for another classification. This section questions whether or not the first step of this algorithm, called the Basic Semantic Texton Forest or B-STF here, can be used on satellite imagery for land cover mapping.

In practice, the limit of the first step is the number of features that current classifiers can recieve at once. The entire context represents a large number of features, and this effect can only be amplified when using higher spatial resolutions or images with more features.

On multi-temporal multi-spectral data such as Sentinel-2, the use of a 3x3 sliding window would be prohibitive, as it would already multiply by 9 the number of features. While the authors of the STF studies claim that a large number of features is not an issue, it nonetheless has heavy implications on the way supervised classification is performed in practice.

The current software implementation for supervised classification with Random Forest involves pre-computing all of the features for each training sample. If the number of features is beyond a certain limit, the set of training samples no longer fits within the RAM of the machine, which slows down the training significantly.

However, in theory, all of the available features are not necessarily used by every single tree. Each tree uses a random set of features, whose size is conditioned by the depth of the tree. Naturally, if a low number of features are present in the data set, there is a high chance that each tree will split according to all of the features, and most likely more than once. However in the case where a very large number of total features are available, the number of features used in practice by a tree is far smaller than the number of possible features. This is particularly true when no bagging is used. Therefore, it could be imaginable to perform the random sampling of features beforehand, and training each tree separately using only the necessary features. These would require a significant reworking of the software involved, which while theoretically possible, was not chosen as a focus for this Ph.D.

The application of the B-STF to the Sentinel-2 data set being impossible in practice for the time being, it can nonetheless be used on the SPOT-7 data set, in a simplified setting.

The STF was designed for low-dimensional imagery, namely, RGB images in a Computer Vision setting. For this reason, the algorithm may be adapted for use on mono-date mosaics at a VHSR, which also have a low number of features per pixel. For a first-order evaluation of this method, the combinations of pixel values can be discarded as possible features. This should allow for a consequent neighborhood size to be taken into account.

The authors of [Shotton et al., 2008] analyze the behaviour of the *mean patch* in a leaf. This is defined for one tree of the forest, as the training samples of each leaf are averaged together. This mean patch can be associated with the class label of the majority of the training samples in the corresponding leaf. Visualizing these patches shows how the tree learns to recognize different elements of the classes, and especially, whether the spatial features allow for a clustering of the elements according to their structure, or their average pixel values.

This same idea, applied on the SPOT-7 data set, is shown in Figure 8.6. In this representation, the columns show leaves that contain a majority of the same class label. Each row contains the mean patch of a leaf, visualized with the R band switched for the NIR band. This exhibits how areas with a high NIR reflectance, which therefore appear in red, can be labeled as crops or vegetation. It also appears that the crop patches are sometimes captured in their bare soil stages, for instance, patches numbered 2, 4, 10, 11, 12, 17 and 18. The remaining crop patches have a higher NIR reflectance than vegetation, which is indicative of dense vegetation, and therefore in many cases a of intensive agriculture. Indeed, the vegetation patches (trees, hedges) generally appear darker, as they contain mixed of soil, and sometimes shadows due to their vertical structure.

This figure shows that a form of spatial clustering is indeed achieved by the B-STF algorithm. This phenomenon is particularly visible in the Roads and Urban classes. Certain of the patches labeled as roads contain bright linear shapes, for example, 1, 5, 8, 17 and 19. Urban patches contain brighter pixels, spread across the patch, and in some cases (1, 8, 12, 17, 18) show the presence of a North/North-Western shadow. Interestingly, the patch number 9 of the Roads class shows the opposite gradient, with the central pixel in the shadowed area. This indicates that the B-STF is able to assimilate such geometric notions from the data set. To what extent this brings precision to the final classification is discussed further in Part IV, in Chapter 11.

8.4 Overview with regards to operational land cover mapping

This chapter presented insights regarding ways to provide a compact description of the context. This comes down to imagining a transformation upon the pixel values, which projects them on a lower dimensional space. Among the many possible methods to do so, one group was studied with particular attention. The so-called stacked contextual classification methods use a prediction, in other words a form of labeling, supervised or unsupervised, of the nearby pixels to categorize them into groups. These neighborhing group labels can then be used in various ways, for instance, as an input to a model, or in many simpler approaches, as base elements for a statistical characterization. A recurring statistical feature used is the histogram of group labels, which is way of representing the

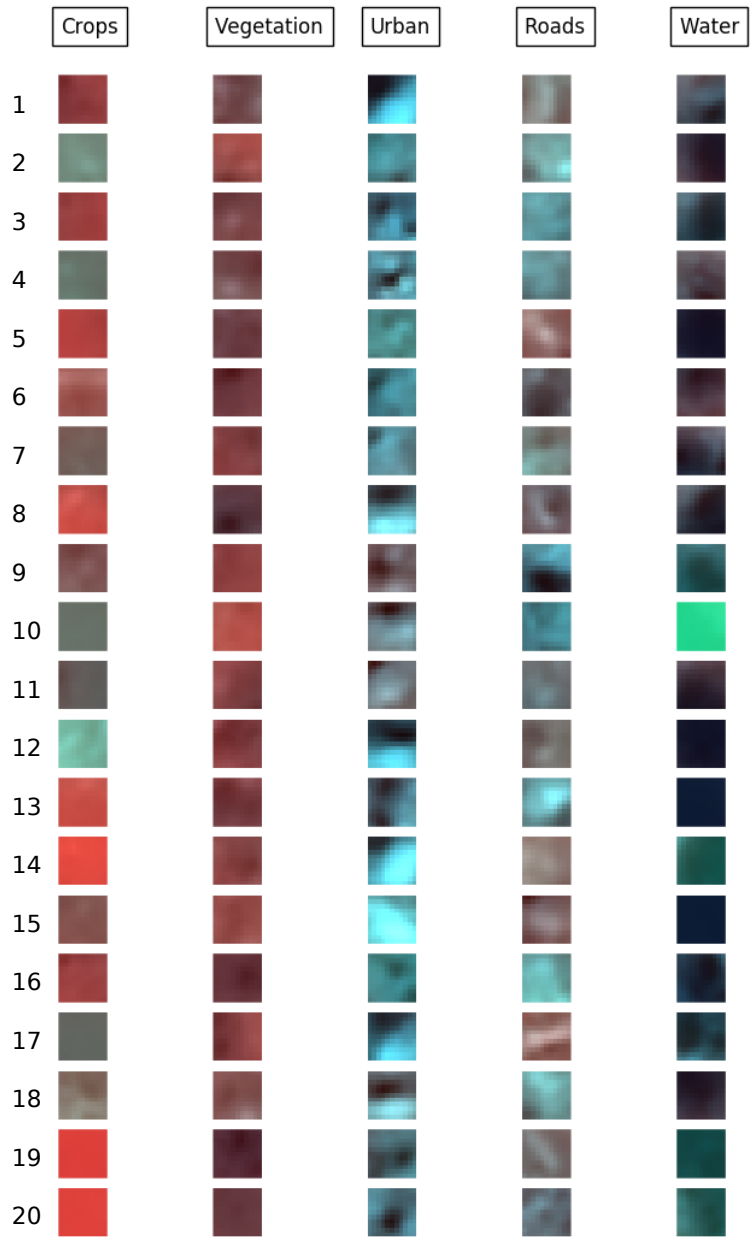


Figure 8.6: Mean patch in the leaves of a tree in the B-STF was trained on SPOT-7 data. Each column shows the leaves that belong to a certain class of the data set, in the sense that they contain a majority of elements of that class. These are shown with the NIR band instead of the R band, in order to show the impact of the IR information for the vegetative classes.

proportions of the local groups in the spatial support.

The question of how to select the labeling procedure is not a simple one, as both supervised and unsupervised methods have attractive aspects. Nonetheless, the direction chosen in this chapter was to investigate supervised methods, and particularly ones using the same nomenclature as the target nomenclature. This choice was made so that the contextual classification may be iterable, which opens many possibilities and offers a higher degree of flexibility for integration within an existing process. The ability to integrate any previous classification, from any

model or training data, and train a new model that integrates spatial information into it is appealing.

The other issue, which concerns the choice of spatial support, was also considered in this chapter. Some methods, in particular model-based ones, are restricted to the use of sliding windows. However, statistics such as the histogram of class labels, can be calculated in any spatial support. The superpixel, adjancency layers, and sliding windows, are all therefore valid candidates for experiments.

This brings forth a new method, which was named the Histogram of Auto-Context Classes in Superpixels (HACCS). This process is initialized using a existing dense classification of the image. If none is available, it is possible to use a pixel-based prediction. Then, it involves calculating the normalized histogram of the class labels in the superpixels, which serve as features alongside the pixel features, to perform a new classification. This can, and should be repeated several times until the result converges.

From this literature study emerged a model-based classification method that could be applicable to VHSR images (SPOT-7): the Basic Semantic Texton Forest (B-STF). This approach is quite straightforward. It involves using the extensive list of pixel values, over the entire spatial support, as features for a supervised classification. This method was evaluated on the SPOT-7 data set, and the classification results of this evaluation are shown in Part IV, Chapter 11. The analysis of the mean patches in the trees of this B-STF show that each tree performs a form of labeled clustering, according to recurring spatial patterns, such as gradients, and fine elements. The N/NW shadow pattern observed in these patches questions the assumption that rotation invariance is a desired property in this classification problem, as oriented shadows are present in the image. It is important to remember that the issue of shadows is not as impactful in time series as in mono-date images, due to the fact that the length of the shadows can evolve over the year. This is part of the reason why dates near the summer period are often highly regarded by supervised classifiers.

The B-STF is far less complex than state of the art model-based approaches that are developed mostly for computer vision purposes nowadays. Practically speaking the B-STF remains strongly limited in terms of neighborhood size, as it involves one stage of classification that should not integrate too many features at once. The possible application of this method to high-dimensional images, such as Sentinel-2, was not resolved during this Ph.D. and is discussed further in Part V, Chapter 12.4. In order to integrate a larger neighborhood size, the information must be structured in a way before the decision stage. This is the idea behind Convolutional Neural Networks, which are the focus of the following chapter.

9 Deep Learning on images with Convolutional Neural Networks

"Machine learning is the science of getting computers to learn without being explicitly programmed."
– Sebastian Thrun

Today, a very popular approach to tackle the issue of context-dependent classes is to use a Convolutional Neural Network [Maggiori et al., 2017], which is a supervised classification method that aims to learn both the feature extraction and the decision steps in an end-to-end manner, from the training data that has been provided. In order to provide an in-depth analysis of the results of these networks, it is important to understand precisely how they include the pixel context in the decision making process. To this end, Section 9.1 provides the theoretical background of CNNs, from the basic neural networks, that are applicable on any classification problem, to their convolutional counterparts which are meant to be applied on images. Then, Section 9.2 presents some of the studies that have applied these methods to land cover mapping, as well as their main benefits and limits.

9.1 What is Deep Learning ?

The term *Deep Learning* actually refers to the use of deep neural networks, which are simply neural networks with a large number of layers. The exact amount of layers that qualify a network as deep is not always clear, and some studies with only one or two hidden layers are already called "deep" by their authors. The depth of the network conditions the complexity of the problems it is able to apprehend, but making deeper networks can also increase the training time, and amount of training data needed.

Section 9.1.1 provides the basic definitions, vocabulary, and notions useful for understanding neural networks. Then, Section 9.1.2 shows how these are adapted for imagery applications, and provides some discussion regarding their theoretical advantages and drawbacks.

9.1.1 The Neural Network, a connected group of simple neurons

A neural network is formed by an assembly of neurons, which are elementary functions that encode a weighted sum of the inputs, followed by an *activation function*, as is shown in 9.2. The three most common activation functions are listed below and shown in Figure 9.1. These functions have at least one parameter, the bias b which shifts the center point of the function along the x axis.

1. The binary threshold, outputs 1 if the net input is greater than the bias, 0 otherwise.

2. A smooth sigmoid function. Similar to the binary threshold, but with smoother edges near the threshold. Often, it is a logistic function of the form

$$\frac{1}{1+e^{-ax+b}}$$

   where a and b respectively define the smoothness and bias of the threshold.

3. A rectified linear unit (ReLU), which is directly proportional to the net input if the net input is greater than the bias, and 0 otherwise.

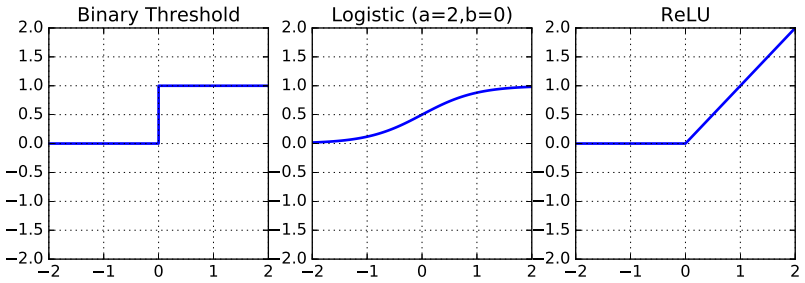


Figure 9.1: Common activation functions used in neural networks : the binary threshold, the logistic function, and the rectified linear unit (ReLU). These are all shown with a null biais, in other words, centered at x=0.

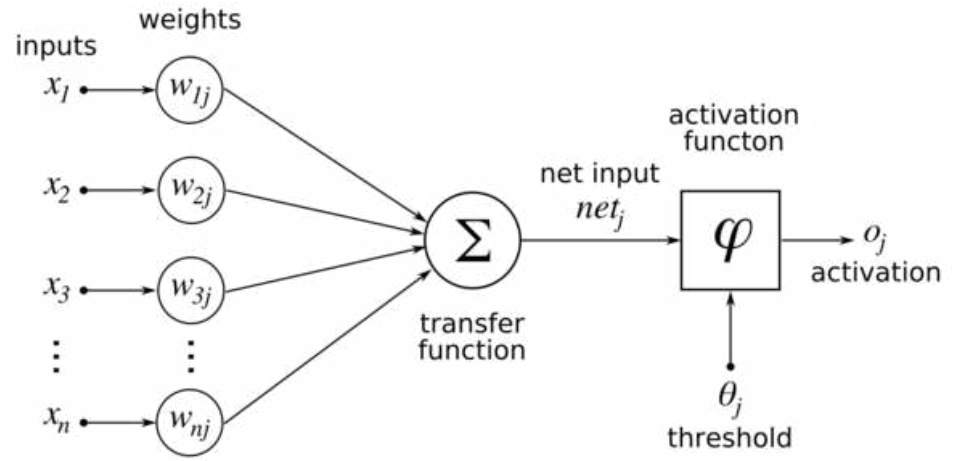


Figure 9.2: Each neuron of a neural network performs the weighted sum of its inputs before passing it through an activation function, which is often a threshold-like function.
Attribution: Perceptron. Mitchell, Machine Learning

The individual neuron can be taught to perform a binary classification with a linear decision function using an algorithm called the *perceptron*, proposed by [Rosenblatt, 1958], which iteratively updates the values of its weights and biais, in order to minimize the number of misclassified points. However, in more complex classification problems, the optimal decision boundaries are rarely linear, and there are generally more than two classes. This is addressed by connecting several neurons together in successive layers, to form a *multi-layer perceptron* (MLP), an example of which is shown in Figure 9.3. The first layer or *input layer* is directly connected to the data. The output of this layer is then connected to the input of the following layer. These intermediate layers are known as the *hidden layers* of the network. In a fully connected network, each neuron of each hidden layer takes as an input every single output of the previous layer. The total number of hidden layers is commonly called the *depth* of the network, which is the origin of the term *deep learning*. Adding a greater number of layers allows the network to apprehend more complex class distributions, and to exhibit non-linear decision boundaries. On the other hand, this implies that if a large number of hidden layers with many neurons are used, the total number of weights and biases increases, which makes the optimization of the network more challenging.

Neural networks can be used for both regression and classification purposes. For regression networks, which aim to create a model of real-valued multivariate data, the number of neurons should be equal to the number of variables to estimate. These networks can be trained to estimate non-linear functions, given a set of measurements of the inputs and the outputs.

For a classification network, the final layer or *output layer* should contain one neuron for each class in the training data set. The outputs of this layer can be seen as the class-conditional probability density functions of the different classes. In order to determine the class of an input data point, the class of the neuron with the maximal output is taken. This is known as a *soft-max* operation.

Neural networks are trained by repeatedly passing the entire training data set through the network, and optimizing the weights in such a way that a *cost function* or *loss function* is minimized. For regression networks, this is often the mean squared error between the training samples and the estimated samples. For classification problems, this cost function is the total classification error, in other words, how many of the samples the network

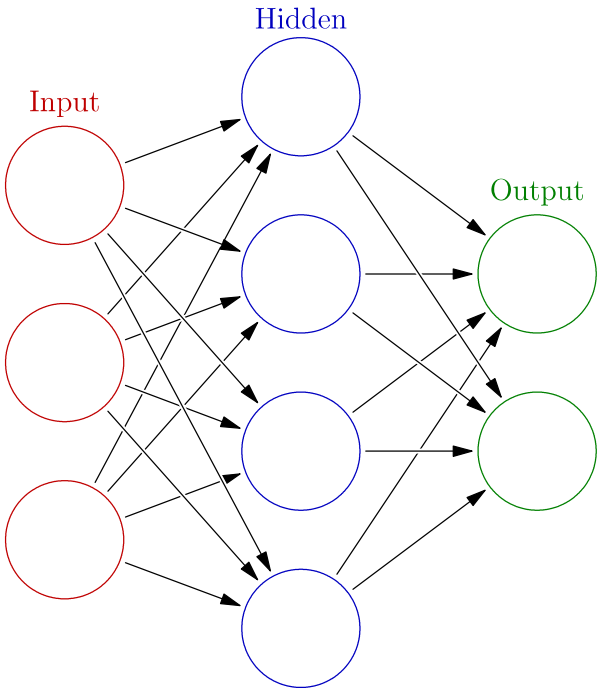


Figure 9.3: Simple schematic of a multi-layer perceptron. The input layer reads the data, and each neuron is connected to every neuron of the next hidden layer. There can be any number of hidden layers, with any number of neurons, but the quantity of weights to optimize limits the total size of the network.
Source: www.todglosser.ca

has misclassified. When training a classification network, a labeled sample has a fixed output layer, with a 1 on the neuron of its corresponding class, and 0s everywhere else. The optimization algorithm then adjusts the weights to make the prediction of the network fit with the desired output.

The challenge of training neural networks resides in optimizing the very large number of parameters (weights and biases) in the network, which poses both theoretical and practical issues. Optimization in very large dimensional spaces incurs a higher risk of converging towards local optima. Neural networks are also sensitive to overfitting: if a very complex network is used only on a few points, there is a risk that the network will simply memorize the inputs and not seek to generalize. Moreover, the computational aspects can make training very slow, especially when deep networks are used.

For this reason, the number of inputs in the network is generally speaking a limiting factor. Indeed, the higher the dimension of the inputs, the more neurons each layer should have. This effect is even stronger on complex multi-class classification problems where a deep network is useful to accurately model the non-linear decision boundaries.

For applications in imagery, the number of features depends both on the dimension of the image and on the spatial context that is desired. For problems like computer vision, where thousands of pixels all participate in forming an image and are all required to determine the output, some modifications need to be made to the structure of the network for this model to be usable. For application on time series, the Long Short Term Memory network [Hochreiter and Schmidhuber, 1997] is sometimes used, as it models the recursive aspect of temporal signals.

The following section explains how these basic neural networks were adapted for applications in imagery, through the use of an architecture known as the *Convolutional Neural Network* or CNN.

9.1.2 Convolutional Neural Networks

The particularity of the CNN approach is to consider a square area of pixels, called a *patch*, as input to the neural network. In other words, the network receives the value of every single feature of every single pixel of the area as an input. This very large amount of information is structured as a square grid of pixels, which allows for the network to be designed in such a way that the spatial arrangement of the pixels is taken into account. The trick behind Convolutional Neural Networks is reducing the density of the network, compared to the standard neural network model. Indeed, if each neuron of each hidden layer is connected to every single neuron of the previous layer, the

number of weights would be too large for an entire patch to be considered at once. To this end, [Fukushima et al., 1983] adapt the structure of the network using principles taken from the visual system of humans and animals. Our brains have a complex visual system with millions of neurons, but the authors realized that certain neurons were structured to form *convolutions*, in other words, basic operations (multiplications and additions) regarding a small area of the visual field. These neurons show a strong activation when given specific geometrical patterns, such as gradients, corners, and local extrema. The outputs of these *convolutional layers* is then fed into more complex, fully connected neural systems, of which our understanding is limited.

In a CNN, each neuron of a hidden layer is only connected to a small part of the previous layer. This allows for the spatial structure of the input to be conserved during the initial layers, and for these initial layers to perform elementary feature detection tasks. Moreover, it is a way to deal with the very large amount of information present in a patch of pixels.

Between each convolutional layer, a *pooling* operation is introduced, which conserves a factor like the maximum value (max-pooling) or the average (mean-pooling) in 2x2 areas of the previous layer. This allows the spatial dimension of the layers to be progressively reduced, and is a way for the network to achieve a degree of translation invariance. A schematic of pooling and convolutional layers is given in Figure 9.4.

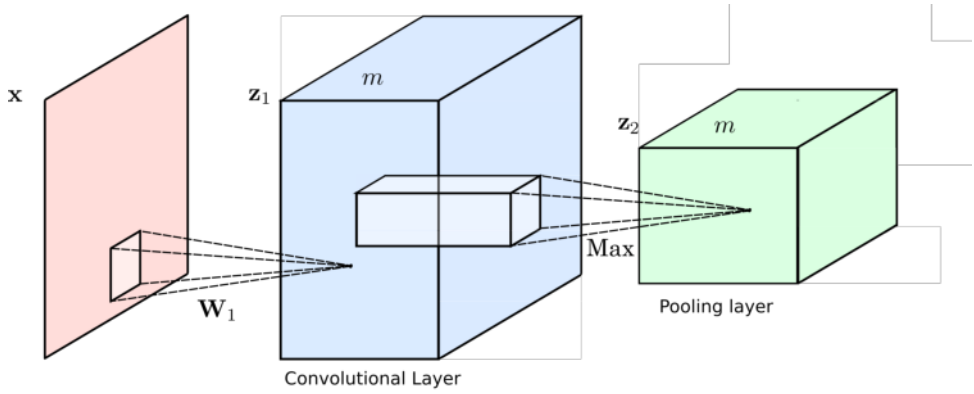


Figure 9.4: Example of convolution and pooling layers of an input image, x. The convolution weights W_1 define the type of convolution and are optimized during training. A bank of m convolutional filters is used, in order to potentially extract different types of convolutions in a given area. The pooling layer effectively reduces the extent of the image, by applying a max filter to the result of the convolutional layer. This is the base element of any convolutional neural network, and can be applied several times in succession.

After several steps of convolution and pooling, the input is reduced to an *intermediate representation* which has a lower dimension than the patch. Then, there can be two possible approaches: either the network associates one class label to the entire input, or it attempts to classify every single pixel of the input.

1. If the objective is to associate one label to the entire input, this intermediate representation can then be connected to a standard multi-layer perceptron which learns how to translate this representation into a class label. Figure 9.5 shows an example of such a network. This is the case of neural networks like ImageNet, proposed by [Krizhevsky et al., 2012].

2. If the aim is to classify every single pixel of the input, the intermediate representation must be restored to its original spatial dimension through the inverse operators of the ones used during the pooling and convolutional layers, known as unpooling and deconvolutional layers. An example of such a network is shown in Figure 9.8. This is the method employed by networks such as SegNet [Badrinarayanan et al., 2015] and U-net [Ronneberger et al., 2015].

Convolutional Neural Networks were originally developed to classify relatively small images in the field of computer vision, and began to show strong performance indicators on problems like optical character recognition [Simard et al., 2003, Chellapilla et al., 2006b, Chellapilla et al., 2006a], as well as image classification [Krizhevsky et al., 2012]. By using techniques like data augmentation, dropout regularization [Hinton et al., 2012], and batch normalization, these neural networks provide unprecedented results on these challenging problems.

Many studies show that the recognition rate of context-dependent classes is improved by the use of D-CNNs, when compared to pixel-based classification. However, a deeper analysis suggests that such methods have difficulty providing geometrically precise results. Indeed, several recent studies attempt to include a degree of geometric information into the classification process to counteract undesirable effects like the smoothing of sharp corners

and the removal of small elements. For instance, in [Marmanis et al., 2018], the authors combine a regular CNN with an edge detecting CNN like the Holistically-Nested Edge Detection network [Xie and Tu, 2015] in order to improve the classification performance in these sensitive areas, however the authors do not address the case where no dense reference data is available for training. More recent works include efforts to implicitly regularize the classification result using a distance transform [Audebert et al., 2019].

Analyzing a CNN after it has been trained can provide important insights regarding the nature of the spatial features that are used by the network to separate the different classes. Very often, the first layers of the network learn to extract the base elements of the geometry at a very local scale: oriented gradients, local maxima, sharp corners, and repeated patterns. This can be visualized by observing the values of the weights in these layers, or by following how the different areas of the network are activated. Often, a clustering is applied to the space of weights to distinguish groups of filters. Interestingly, these initial convolution filters strongly resemble some of the contextual features that are presented in Part II, Chapter 5, such as 2-D wavelets or Gabor Filters. Some say that this allows for a more optimal selection of these filters, as they are extracted in a data dependent manner, and are therefore appropriate for the problem at hand. Others argue that it is wasteful to learning these features again for every single problem, as we already know that they are relevant for distinguishing context-dependent classes. Compared to handcrafted contextual features, this incurs a supplementary dependence on the data, which increases the need for large volumes of high-quality training data.

There are also interesting common points between the concepts behind Convolutional Neural Networks and the methods presented in Chapter 8, in particular the Semantic Texton Forest of [Shotton et al., 2008]. Indeed, the idea of the STF is to train an ensemble of decision trees according to pixel values, or differences between pixel values, randomly selected in a square neighborhood around the target pixel. In other words, both the CNN and the STF consider the entire patch of pixels directly as an input to the classification method. By analyzing the way that the trees split the data, it seems that during its initial training step, the STF uses similar basic geometric operations to the ones found in the first layers of a CNN. Examples of this are shown in Part IV, Chapter 11, in the application of a basic version of the STF method to the SPOT7 data set. In fact [Richmond et al., 2015] shows that such a decision forest can be mapped onto a CNN, and inversely a CNN can be mapped to a decision forest. This offers many interesting possibilities, mainly because decision forests are usually many orders of magnitude faster to train than CNNs. This could allow for a rapid initial training on large volumes of data, followed by a fine-tuning of the network for specific areas or applications.

Convolutional Neural Networks have been used in several studies to perform land cover mapping tasks. They are presented here as they form the state-of-the-art for contextual classification. However, they also have certain constraints and limits which are important to take into account if these architectures are to be used for an operational land cover mapping application.

9.2 Deep Learning for land cover mapping

The following sections detail two types of Deep Convolutional Neural Network that were recently used to classify land cover: patch-based networks and fully-convolutional networks.

9.2.1 Patch-based network

The patch-based methods involve taking a standard CNN architecture like ImageNet [Krizhevsky et al., 2012], and considering that the output label, which was originally meant to describe the entire patch, is assigned only to the central pixel of the patch. This way, all of the pixels in the image can be labeled by applying the network to a window around each pixel, like a sliding window. Furthermore, this is adapted for sparsely labeled training data, where each training sample is a labeled pixel. This allows for existing network architectures originally used for image classification to be re-used to achieve a dense classification, and therefore allows neural networks to be applied to land cover mapping [Penatti et al., 2015, Marmanis et al., 2016, Audebert et al., 2016a]. Recently, a patch-based network was successfully applied on a 5-class land cover mapping problem, including context-dependent classes such as roads and urban cover, which are commonly confused by pixel-based classifiers [Postadjian et al., 2017]. Figure 9.5 shows the relatively simple network architecture, which is composed of three stages of convolution and max-pooling, followed by a fully connected layer.

For training this patch-based D-CNN, 1000 patches per class are randomly sampled throughout the labeled polygons. Examples of such training patches are shown in Figure 9.6.

The application of this network to SPOT7 strongly increases the classification accuracy of the context-dependent classes, as is shown later in Part IV, in Chapter 11. However, the level of detail around the edges of the objects is very often blurry, and presents a number of noisy decisions, particularly around the class boundaries. This is illustrated in figure 9.7b, which shows the classification result of the D-CNN on an urban area.

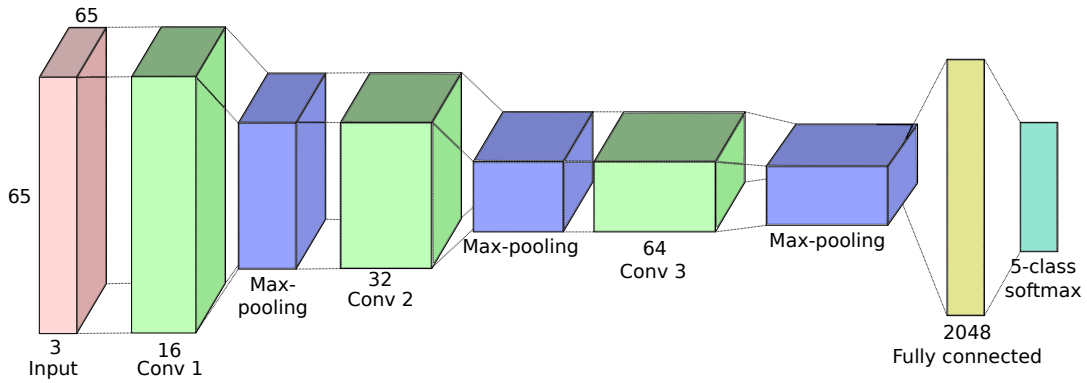


Figure 9.5: Architecture of the patch-based network, identical to the work in [Postadjian et al., 2017]. The first layer intakes a neighborhood of 65x65 pixels around the central pixel. Then, this relatively shallow network contains three stages of convolution and max-pooling, followed by a fully connected layer.

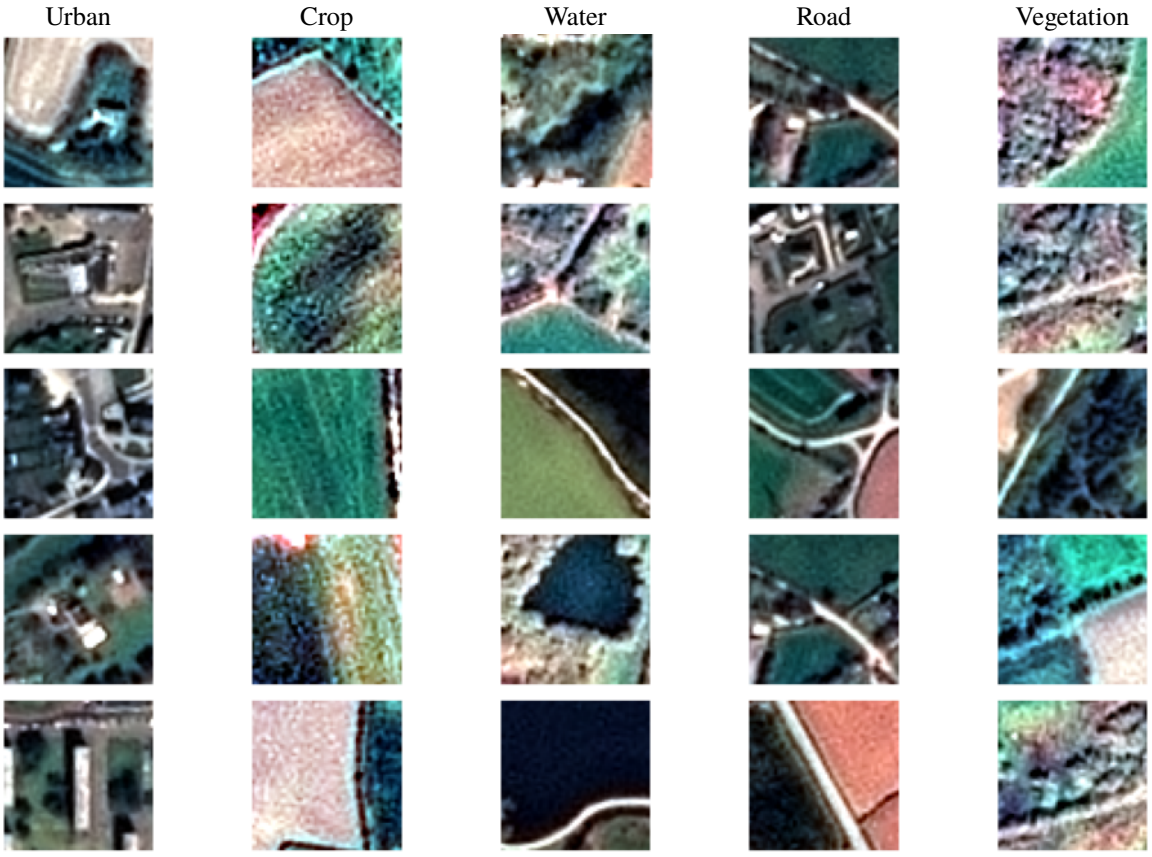
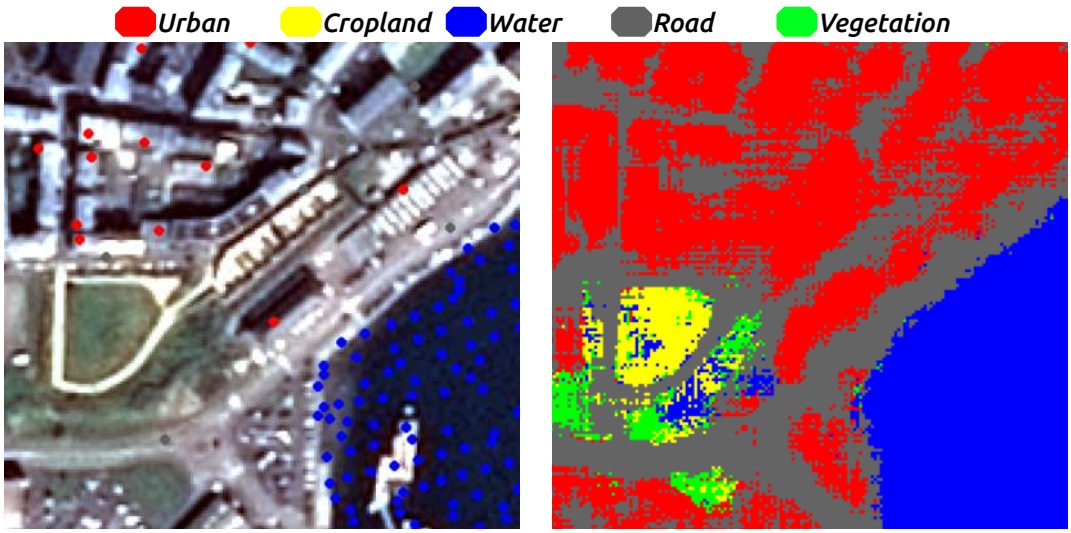


Figure 9.6: Images of labeled patches used by the patch-based Convolutional Neural Network

Indeed, the training data that was used for this classification presents numerous flaws. First of all, a strong concentration of training pixels in the center of the objects means that an extensive description of the class boundaries is missing. This is representative of a land cover mapping problem over wide areas, in which the only available training data sets are sparsely labeled. Secondly, there are errors in the training labels, which are due to the fact that these training data sets are often out of date, and hence do not take into account the land cover changes that may have occurred. However, most standard classifiers are robust to a low amount of label noise [Pelletier et al., 2017].

Thirdly, this might be an effect of the translation invariance that is implied by the use of a patch-based network. To understand this, consider a piece of information that arrives to the final decision layer, in some form or another, and that describes the presence of a road somewhere in the patch. Because this information has gone through pooling steps, The precise localization of the road with regards to the patch can be uncertain. If the central pixel

lies next to the road by a few pixels, the decision label might indicate "road" thinking that the road passed through the central pixel. This suggests that the patch-based network might be insufficient to precisely model the spatial distribution of the classes.



(a) Extract of training area with sparsely labeled training points.

(b) Result of the patch-based D-CNN, the boundaries between the different classes are blurry and present salt and pepper noise.

Figure 9.7: Illustration of the issues with sparse training data, when applying a patch-based D-CNN on VHSR optical SPOT7 data.

9.2.2 Fully Convolutional Networks

Another way of using Deep Convolutional Neural Networks to achieve dense classification involves combining a series of convolutional and pooling layers to a series of deconvolution and unpooling layers, to assign a label to each pixel in the patch. For this reason, it will be referred to as the *fully-convolutional network* here. This is the idea behind networks like Seg-Net, [Badrinarayanan et al., 2015, Audebert et al., 2016b, Audebert et al., 2018] and U-Net [Ronneberger et al., 2015, Maggiori et al., 2017]. This way, entire patches of the image can be classified without the need to pass through the whole image, which means that fully convolutional neural networks are usually faster than their patch-based counterparts. These methods are usually applied in conjunction with dense training data, but several recent studies have shown that this architecture can also be applied to problems with sparse training data. In [Stoian et al., 2019], the authors propose a Fine Grained U-net architecture (FG-Unet), which is a slightly modified version of the classic U-net architecture, particularly to deal with the issue of sparse training data. Usually, the loss function is calculated by comparing the output of the last layer of the network, which contains the label predictions, to the densely labeled training patch. When the training data is sparsely labeled, the adapted loss function ignores the unlabeled points, as is shown in equation 9.1. Moreover, a weighted average is made over the various classes, in order to provide more weight to classes with a low number of training points, to deal with imbalances in the class priors.

$$\mathcal{L} = \sum_{pixels} \sum_{k=1}^{class\_nb} weight_k \cdot y_{true_k} \cdot \log\left(y_{pred_k}\right) \tag{9.1}$$

Secondly, a weight-sharing scheme is employed in order to limit the size of the initial layers. Indeed, the large number of dimensions in the input data is necessary, however, if no precautions are taken, it can create very large networks that are more difficult to optimize. For this reason, the structure of the network is changed such that instead of taking all of the dates in the time series, the entry layers of the network focus on three consecutive dates. During training, the entire time series is passed through these layers, in an attempt to learn the non-time dependent aspects of the problem, at least for the initial convolutional layers. Evidently, the weight-sharing scheme is not applied to the deeper layers. Indeed, they receive the response from the first layers, but applied to all of the dates separately. In this way, the deeper layers can learn the temporal aspects of the problem. This is found to be more

efficient than the LSTM network, which was in fact recently applied to a similar land cover mapping problem by [Ienco et al., 2017].

A third adaptation to the U-Net architecture is made, in order to deal with the issues of geometrical degradation in the output maps. This consists in connecting a series of 1x1 convolutional layers to the fully-connected layers in the deepest part of the network. In other words, this is equivalent to adding a pixel-based classification of the time series at full resolution to aid in the decision.

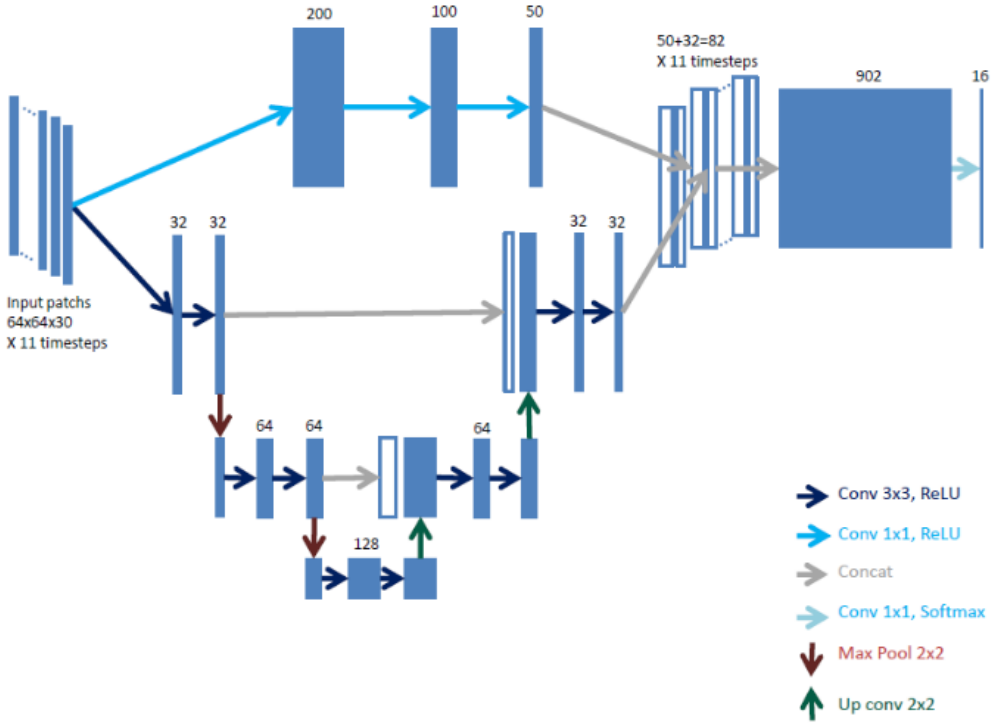


Figure 9.8: Fully convolutional architecture used on the Sentinel-2 data set in [Stoian et al., 2019]. Inspired by the U-Net architecture, it contains several convolution and max-pooling stages, followed by deconvolution and unpooling, to generate a dense prediction of the patch. A 3-date weight sharing scheme, as well as a 1x1 convolution stage were used to adapt this problem to use with time series, and sparsely labeled data.

With these adaptations, the quality of the classification of context-dependent areas, particularly of urban classes, is greatly improved. On the other hand, some fine details remain absent from the output maps, as is shown in figure 9.10b.

9.2.3 Issues with sparse data

Generally speaking, fully-connected CNNs are challenging to use for land cover map production, especially over large territories, because the reference data is only available in a sparse form. In other words, unlike many applications of fully-connected CNNs to imagery, not every pixel of the training area is labeled. This is a very often the case in land cover mapping, as the reference data comes from a combination of existing geodatabases [Inglada et al., 2017], which each contain certain classes of the desired nomenclature. This means that the fine details of the geometry, such as edges and sharp corners, and the explicit spatial relations between various classes are absent from the training data. This might make training difficult, because the fully-connected CNN attempts to learn the feature extraction step from the data itself.

Geometric degradation, the smoothing of sharp corners and of small elements in the classification map, is a recurrent observation in several recent studies evaluating the use of Deep Learning architectures to achieve dense classification with a sparse data set. In [Kussul et al., 2017], a patch network is used to classify crops using a combination of Landsat-8 and Sentinel-1 time series. While the fully-connected CNN approach outperforms the pixel-based RF, the authors note that some small objects are misclassified, and the sharp corners appear rounded in the final result. Similar misclassifications are also observed in hyper-spectral image classification, when using a patch-based network based on Auto-Encoder features [Chen et al., 2014].

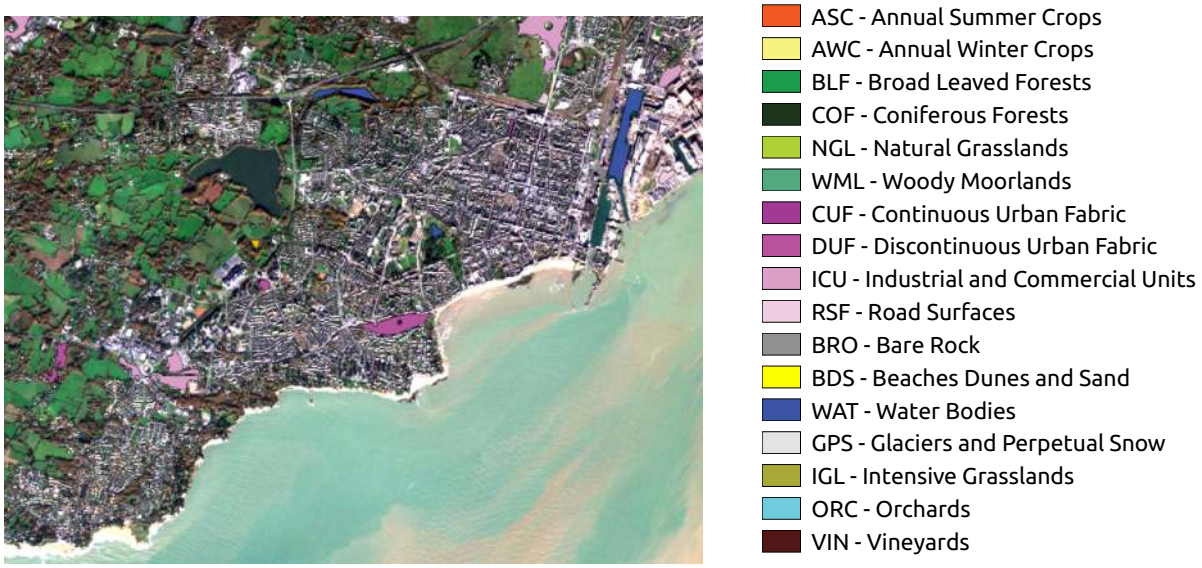


Figure 9.9: Sparsely labeled training samples over the city of Saint-Nazaire. Background: RGB bands of the first date of the time series (January 2016).

This phenomenon is rarely considered by studies that apply fully-connected CNNs with dense training data on toy problems, but is a real issue for land cover mapping. Indeed, when dense training data is available, the use of fully convolutional networks can provide land cover and land use mapping with high context-dependent class recognition rates and spatial precision, as is shown in [Kampffmeyer et al., 2016, Audebert et al., 2016b].

The fully-connected CNN aims to optimize both the feature extraction step, and the classification step, in an end-to-end fashion. In other words, the training data drives both the selection of which contextual features to use, and how to use them properly to achieve a precise classification. This is the case for both the patch network and the fully convolutional network, presented previously.

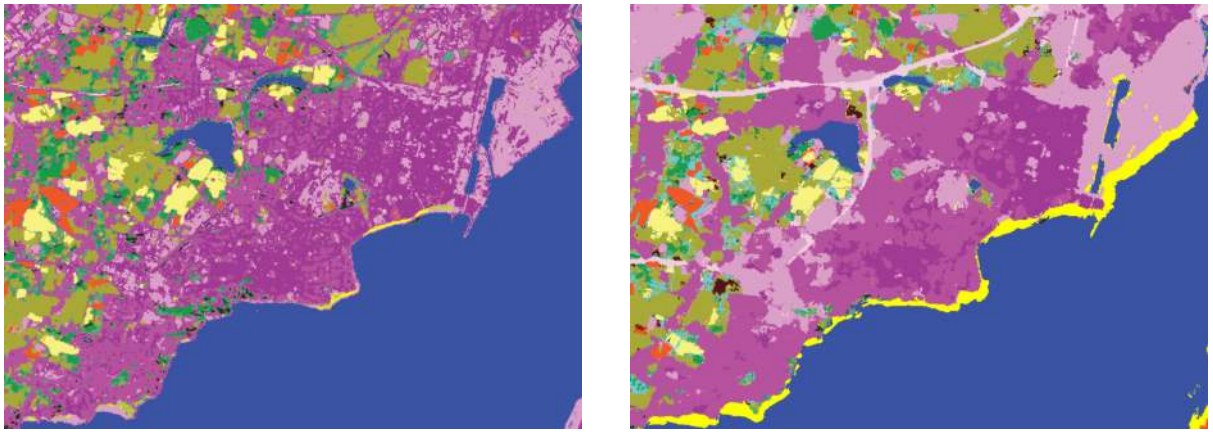
However, by basing the feature extraction step entirely on the training data, problems may appear when this data does not contain a sufficient quantity of points to correctly characterize certain elements of the problem. Indeed, the success of any data-driven method like fully-connected CNN is very dependent on the quality of the training data, in other words, how well the training data represents the desired output. This leads to the common conception that training a Deep Neural Network requires a large amount of training data, which true in most cases, especially for complex problems. For this reason, in practice, several applications using D-CNN increase the number of training samples by applying *data augmentation* techniques such as rotations and other such transformations. However, having a large amount of training data does not always mean that the Deep Learning approach will be successful. Indeed, the training data must be sufficiently rich to cover the most important aspects of the classification problem in the first place.

Unfortunately, in the case of operational land cover mapping, the training data is very rarely densely distributed across the image, because it usually comes from several different sources, which combine on-ground measurements with human photo interpretation [Postadjian et al., 2017, Inglada et al., 2017]. This implies that the fine grained-geometry, i.e. the specific spatial arrangement of the classes is absent from the training data set.

Generally speaking, if a supervised classifier has never been trained on a labeled pixel that is near the boundary between two classes, it might produce a result with an edge displaced towards one of the two classes. In practice, this often translates in a degradation of the high spatial resolution elements in the output classification. Sharp corners are rounded, and fine elements are either thickened, or that disappear entirely.

Moreover, the labeled points are generally concentrated in the center of the objects, and rarely on the edges or in the corners. Fully-connected CNNs may encounter issues linked to an insufficient amount of training points in sharp corners and along the object boundaries, which can reduce the spatial precision of the labeling. Because the high-level decision layers are situated after several pooling layers which reduce the spatial resolution, a geometrically precise decision can be difficult to obtain. For this reason, fully convolutional networks like U-Net introduce *skip connections*, which pass full resolution information to the deep layers of the network. However, pixel features are not as valuable as a training class label for describing spatial relations between objects. The ability of such networks is to create classification maps with an accurate geometry is therefore a subject of discussion in this work.

The results of the patch-based network presented in [Postadjian et al., 2017] and the FG-Unet from [Stoian



(a) Pixel-based result

(b) FG-Unet

Figure 9.10: Results of the classification of the fully-connected CNN method, FG-Unet, over an urban area (Saint-Nazaire). The pixel based result contains a strong degree of intra-object classification noise as well as a poor characterization of the different levels of urban density. The FG-Unet result has a stronger discrimination power for urban classes, but the geometry of the result is questionable at places, for instance in the harbor area.

et al., 2019] are analyzed in greater depth in Part IV where their geometric and thematic precision is evaluated, and compared with stacked contextual classification methods such as Semantic Texton Forests and Histogram of Auto-Context Classes in Superpixels.

Part IV

Results

10

Multispectral time series experiments on Sentinel-2 images

"What's a process? A process is like a magical spirit that lives in the computer and does something. What directs a process is a pattern of rules called a procedure. Procedures are the spells. The programming language is the language for casting the spells."

– Harold Abelson, Structure and Interpretation of Computer Programs, 1986

This chapter presents the results of the experiments that were performed on the 17-class land cover mapping of Sentinel-2 time series, which was described in Part I and is close to the one described by [Inglada et al., 2017].

The data set is based on multi-spectral time series of the year 2016, containing 33 dates of 10 band optical images at a 10m spatial resolution. A total of 12 tiles of approximately 110x110km covering a variety of landscapes across France were chosen for the evaluation. The geographic layout of the tiles is shown in figure 2.4, on page 43. The tiles are spread across the territory to cover different areas, in an effort to be representative of the land cover mapping problem of the entire country.

The training data comes from various sources that were listed in Section 3.1.1, and covers a variety of agricultural, natural, and artificial classes. The number of training samples taken for each tile is shown in Table 10.1. Each tile contains quite different class proportions, which implies the evaluation is performed on a variety of different situations, in order to address the particularities of some of the minority classes that are only present in certain regions. For instance, the Bare Rock (BRO) and Glaciers and Permanent Snows (GPS) are only present in mountainous areas, while Beaches, Dunes and Sand plains (BDS) can only be found at the coasts.

Table 10.1: Number of samples (pixels) taken for training on the various tiles. Up to 15,000 pixels are taken for each class. The T31TCJ tile contains the same number of samples for all of the classes present in the area, namely, all of the classes except bare rock, beaches and snow.

Name	T30TWT	T30TXQ	T30TYN	T30UXV	T31TDJ	T31TDN	T31TEL	T31TGK	T31UDQ	T31UDS	T32ULU	T31TCJ
Index	1	2	3	4	5	6	7	8	9	10	11	12
ASC	15000	15000	15000	15000	15000	15000	15000	2576	15000	15000	15000	15000
AWC	15000	2484	9271	15000	15000	15000	15000	14149	15000	15000	15000	15000
BLF	15000	15000	15000	15000	15000	15000	15000	15000	15000	15000	15000	15000
COF	15000	15000	15000	14575	15000	15000	15000	15000	15000	1541	15000	15000
NGL	6496	0	15000	0	15000	732	15000	15000	1220	1377	15000	15000
WML	15000	15000	15000	7975	15000	15000	15000	15000	15000	8468	8641	15000
CUF	12262	15000	3271	14154	1841	2247	15000	373	15000	15000	15000	15000
DUF	15000	15000	15000	15000	15000	15000	15000	13739	15000	15000	15000	15000
ICU	15000	15000	14706	15000	15000	15000	15000	5679	15000	15000	15000	15000
RSF	3761	15000	1674	2307	1029	2803	15000	1214	15000	12900	9203	15000
BRO	0	0	15000	406	0	0	80	15000	0	0	0	0
BDS	3729	15000	0	3811	0	1687	0	14315	0	3778	0	0
WAT	15000	15000	12511	15000	15000	15000	15000	15000	15000	15000	15000	15000
GPS	0	0	1978	0	0	0	0	15000	0	0	0	0
IGL	15000	15000	15000	15000	15000	15000	15000	15000	15000	15000	15000	15000
ORC	1499	345	37	2205	2171	809	202	6122	4935	578	695	15000
VIN	4406	15000	89	0	15000	2784	917	78	0	0	3433	15000

Many of the illustrations and detailed analysis are made on on the tile *T31TCJ*, which contains the city of Toulouse. Figures 10.1 and 10.2 respectively show the RGB bands of the first date of the time series over this area, as well as the reference data used for the training and validation. This area was chosen among the others as it covers large urban agglomerations as well as a variety of forests and agricultural lands. The other area of interest that is used for visual map analysis in Section 10.3.2 is the city center and harbor area of Saint Nazaire. Figure 10.3 shows this area within the T30TWT tile. The juxtaposition of continuous, discontinuous and industrial areas along with sharp geometrical features along the water make it an interesting study area.

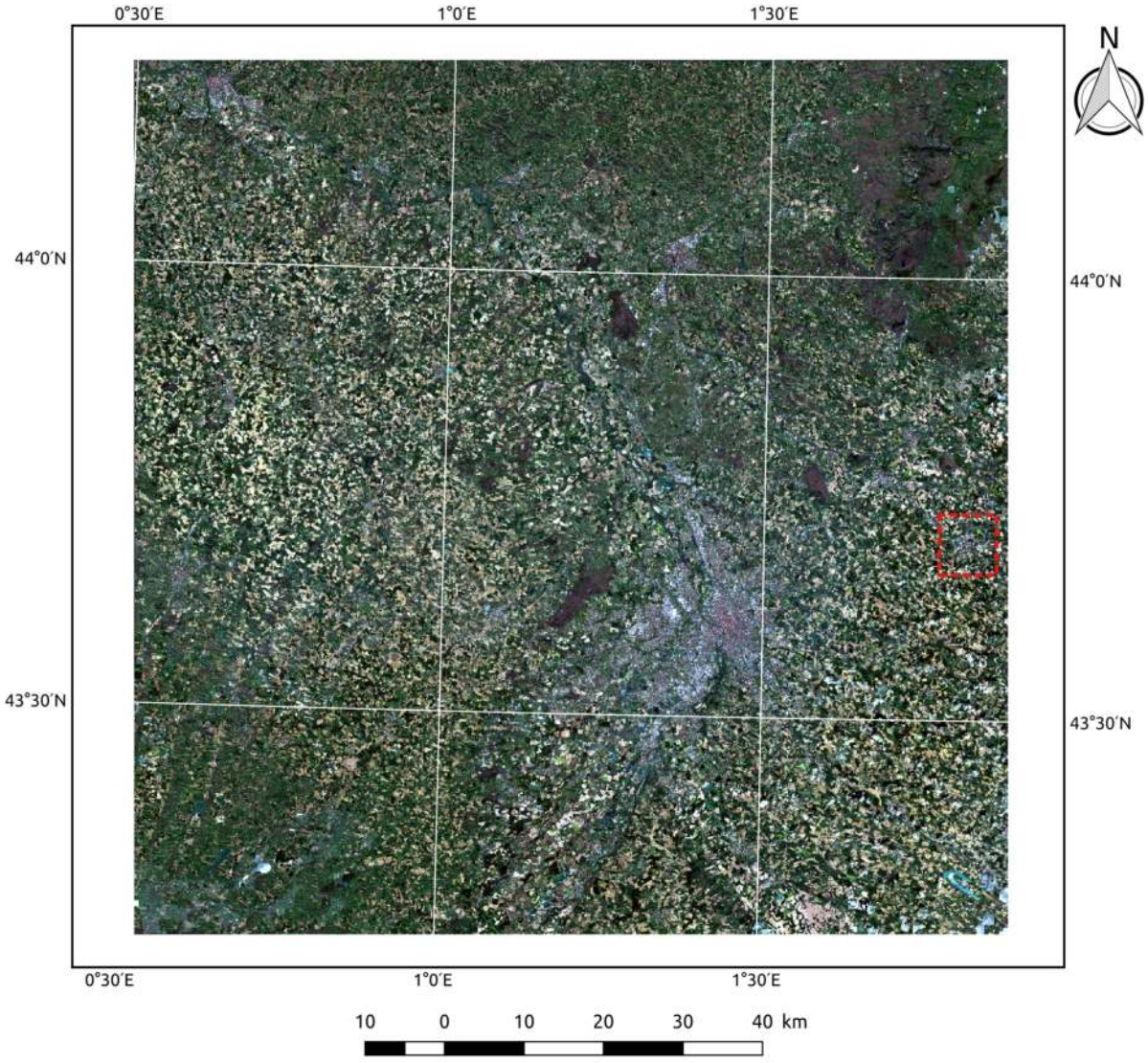


Figure 10.1: The area chosen for the detailed experiments is the 110x110km tile containing the city of Toulouse in the lower right side of the image (T31TCJ in figure 2.4). The image shows RGB bands of the first date of the Sentinel-2 time series. The red dotted line represents the small city of Lavaur, which is used for the illustrations in Figure 10.5.

10.1 Experimental setup

The experiments on the Sentinel-2 data set are presented as two consecutive sets of experiments. The first deals with *image-based* contextual features, that is to say derived directly from the surrounding pixel values. The second studies *semantic* contextual features that are based on a previous classification of the surrounding pixels. The notion of surrounding is kept as a variable in all of the experiments. Different shapes and sizes of spatial supports are compared on their ability to improve the pixel-based classification in terms of class accuracy and geometric precision.

The set of experiments on image-based contextual features presented in Section 10.2 has two main objectives. First of all, to study the impact of the mean, variance, edge density, and shape features, which were defined in Part II, in Chapter 5 on the classification. The other features, such as Morphological Profiles or Haralick Textures are not evaluated here for reasons that were mentioned earlier, in particular, the fact that they not directly applicable to high-dimensional data.

Here, the presence of pixel information is an important factor, as some methods such as OBIA recommend discarding it and replacing it with object features alone. This pixel information, a time series of multi-spectral surface reflectances, is also combined with three spectral indices; the Normalized Differential Vegetation Index

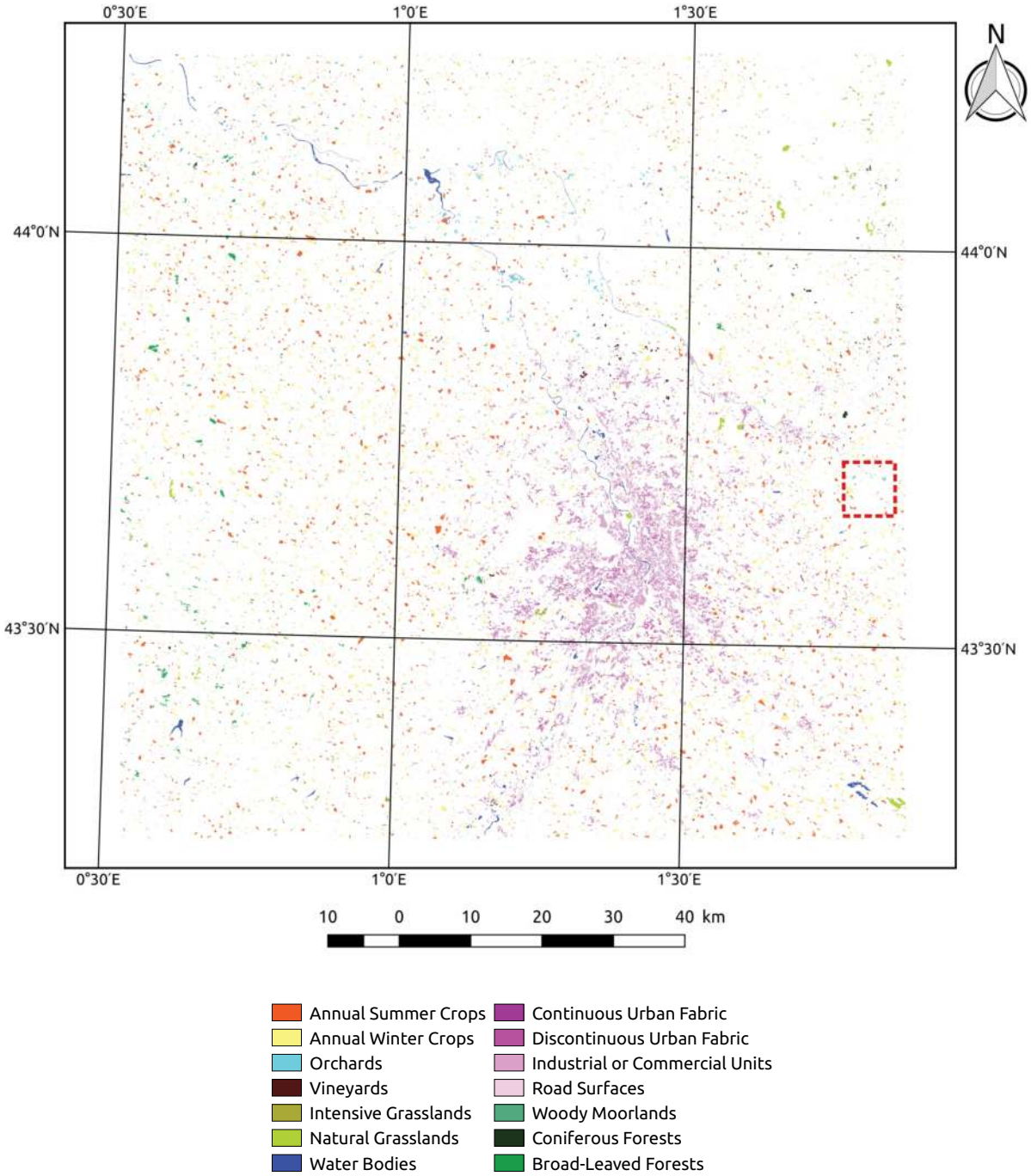


Figure 10.2: Reference polygons, later split into the training and validation sets.

(NDVI), the Normalized Differential Water Index (NDWI) and Brightness, which gives each pixel a total dimension of 489 features. The second objective of these experiments is to evaluate the impact of the shape and size of the spatial support, and the impact of the scale at which context can be considered by a given feature. For this, sliding windows, superpixels, and objects segments from a Mean Shift segmentation are evaluated.

In this experiment, the geometric evaluation is important as certain ways of including context run the risk of smoothing the corners and displacing the edges of context-dependent classes.

The second set of experiments has two sides. First of all, an evaluation of the histogram of classes as a contextual feature in different spatial supports is made, in order to establish a viable set of parameters for application on wider data sets. Here, the number of selected scales can become a parameter, as the histogram of classes represents a low number of features, and contextual information can therefore be accumulated on more than one spatial support. The spatial supports chosen for this evaluation are the sliding windows, superpixels, and adjacency layers based on a superpixel segmentation. The second objective is to compare the semantic contextual features to a Deep Learning

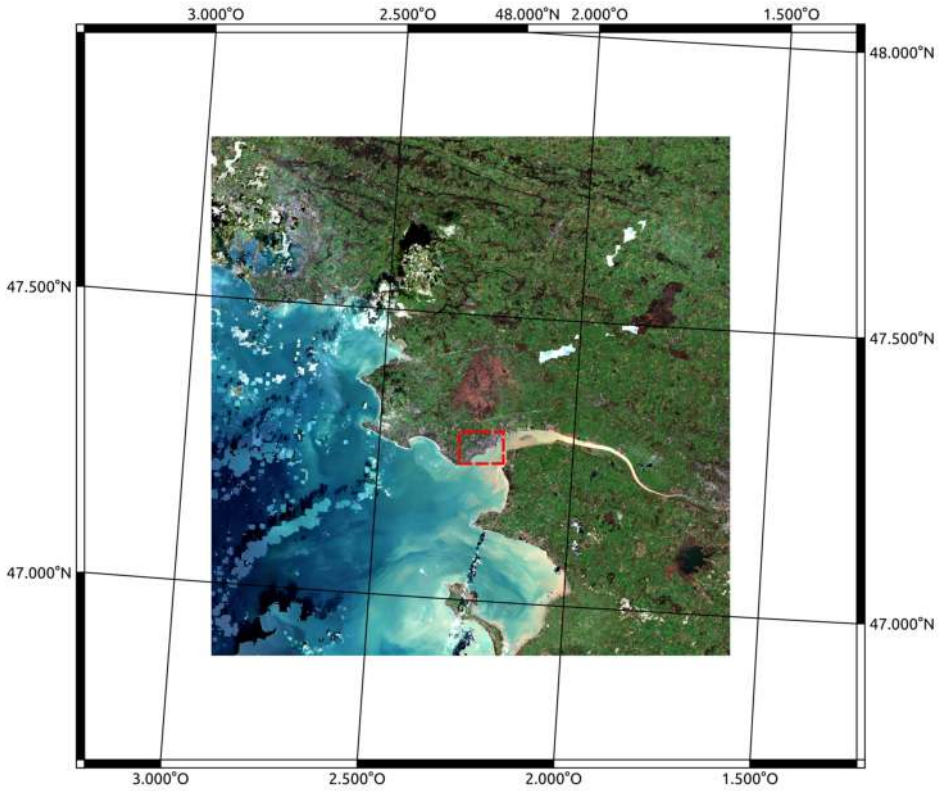


Figure 10.3: Image of the first date of the Sentinel-2 time series over the T30TWT tile in Brittany. The red dotted rectangle shows the location of the city of Saint-Nazaire, which is the focus of figures 10.10 and 10.13. The cloud detection and removal, which are described in Part I, are not perfect at each individual date which causes some clouds to remain in certain images of the time series. However, thanks to the high temporal repetitivity of Sentinel-2, the impact of these detection errors is minimal.

method, namely, FG-Unet, trained on the same data.

Each set of experiments is divided into two parts : detailed experiments on T31TCJ alone, and on the other tiles of the data set. Generally speaking, the experiments on T31TCJ evaluate certain sets of parameters, such as the selection of which spatial supports to use, and their size. Individual class performance is also analyzed in detail on this area. In the end, the methods that provide the best results are then tested on all of the tiles. The cross-tile experiments provide a validation of the selected methods in different situations, which enriches the analysis to other classes that are not present in T31TCJ. These are also used in the comparison with Deep Learning, as T31TCJ was not a part of the study led by [Stoian et al., 2019].

The classifier used for the evaluation is a Random Forest [Breiman, 2001] with 100 trees, and a maximal depth of 25. In addition, ten random samplings of the data are made for training and validation, and the value of the standard deviation across these runs is provided as an indicator of the confidence of the OA, Kappa, and PBCM metrics.

Finally, in the training data, a maximum of 15,000 samples is taken for each of the 17 classes, as Table 10.1 shows. This is done in order to balance the class prior probabilities to a certain degree, as the number of samples for the majority classes such as annual crops (ASC, AWC, IGL) and water are several orders of magnitude higher than the minority classes. This is also done to limit the computational burden of training, as considering a large number of samples increases the training time for the Random Forest method (as is the case for most supervised classification methods).

In the following sections, it is important to note that the validation scores are not always calculated with the same number of samples. On T31TCJ, a maximum of 15,000 different samples were used for the validation, which provides an estimation of the OA and Kappa that is not heavily influenced by the majority classes. On the multi-tile data set, in order to be coherent with the validation of the FG-Unet results made by [Stoian et al., 2019], validation scores using the full extent of the testing data are shown. The class distributions of the testing data on the 11 tiles is shown in Appendix B, Table B.13, on page 191. In turn, this provides a more realistic estimation of the OA in the

sense that it contains many more points, although the majority classes have a stronger influence on the score than the minority classes. Fortunately, the F-scores are relatively independent of the class prior density distributions, and are therefore frequently used in the analysis.

10.2 Results of image-based contextual features

These experimental results are obtained by including contextual features based on the values of nearby pixels, such as the ones presented in Part II, Chapter 5. These features are calculated in a spatial support around the pixel that is being classified, that can be either a sliding window, a superpixel, or an object. First, a detailed analysis of the results on the tile T31TCJ is provided in Section 10.2.1. This includes the class scores of the four main candidate methods, as well as graphs showing their Overall Accuracy and Pixel Based Corner Match (PBCM). Next, in Section 10.2.2, the results on the other tiles are shown, which gives an indication of the performance of the various methods in different land cover mapping situations, each with unique class proportions and class variability.

10.2.1 Experiments on T31TCJ

In Figure 10.4, the different shapes and feature choices are compared according to two criteria:

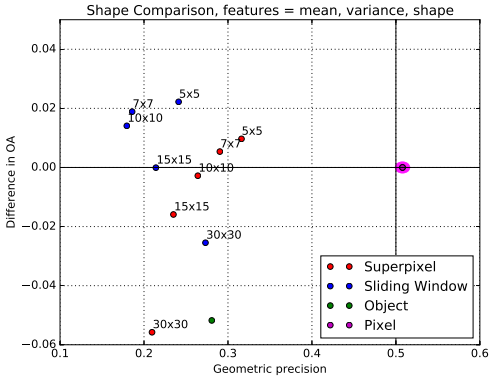
1. How much overall accuracy they bring to the classification,
2. The geometric precision of the result, compared to a pixel-based classification, using the PBCM metric.

The horizontal axis shows the difference between the Overall Accuracy of the contextual method and of the pixel-based method, which is also given in the second column of Table 10.2. The vertical axis represents the ratio of matching corners between the pixel-based classification and the contextual classification. To obtain a reference value for the pixel-based classification, the PBCM metric is calculated on pixel-based results that are generated using different sub-samples of the training set. The labels above the points represent the scale factor, which is the diameter of the window for a sliding window, expressed in pixels, or the square root of the average size for a superpixel.

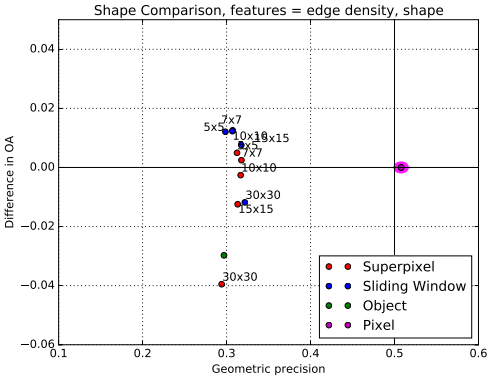
Next, Table 10.2 shows the detailed per-class results for different combinations of features and spatial supports. Only the scale providing the a high OA and PBCM, according to Figure 10.4, is shown for each method.

Tables B.1 - B.8, in Appendix B, (pages 187 - 189), show the F-scores of all of the different scales and combination of features over T31TCJ. The table shown here is a condensed version of these experiments, which only shows the combinations with the higest quality scores. These experients show that improvements are made for textured classes: the four urban classes, as well as orchards and vineyards. Only the crop classes seem to suffer slightly from the inclusion of context in this way. Features describing the texture or in this case the variability are mainly irrelevant for homogeneous classes. However, their recognition rate remains sufficiently high. The best performance is achieved for sliding windows in combination with the edge density feature. The sliding window also has the best F-scores for the four urban classes. The second best combination of spatial support and contextual feature is the superpixel with the edge density; many of the F-scores are similar to those provided by the sliding window, and the recognition of orchards is more accurate.

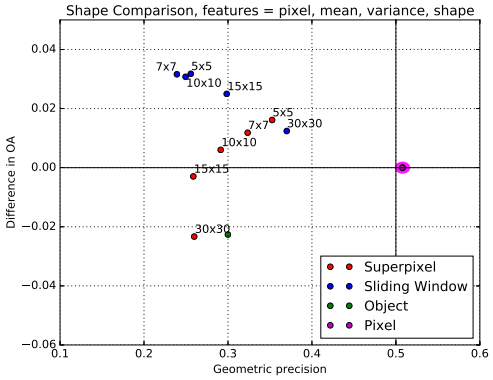
Figure 10.5 shows an extract of the classification maps, generated with different combinations of spatial support shape and feature choice, on an urban area with a small dense center and its surroundings.



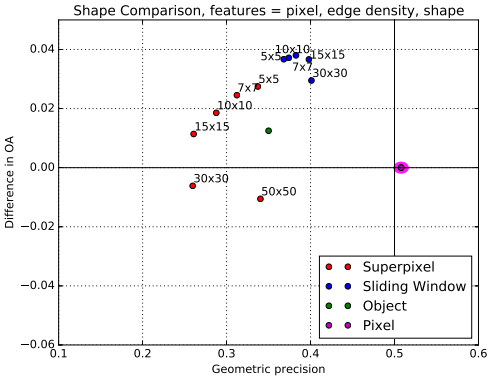
(a) Local statistics features. Without the pixel information, both the increment in classification accuracy and geometric precision of the result are poor.



(b) Edge density features. The edge density alone is not sufficient to generate a result with a high improvement in classification accuracy and geometric precision, no matter which spatial support is chosen.



(c) Pixel and local statistics features. The 5x5 superpixel offers the most interesting trade-off between both of the criteria. The sliding window seems sensitive to geometric deterioration with this feature choice.



(d) Pixel and edge density features. Here, the 15x15 sliding window provides the best results, both in terms of geometric precision and of overall accuracy.

Figure 10.4: Differences in overall classification accuracy compared to the pixel based classifiers, plotted against the PBCM, for the different combinations of features and spatial supports. Whenever relevant, the shape features were included. For a given feature, sliding windows have a higher OA than superpixels, which in turn have a higher OA than objects. Moreover, for a given spatial support, the result using the edge density feature has a higher OA than the one using a local statistics feature. The only case that has a lower overall accuracy than the pixel-based classification is the combination of objects with local statistics features. The use of the edge density feature allows for higher values of the geometric precision for sliding windows and objects, but not for superpixels. The choice of scale has a important impact on both the OA and the PBCM, regardless of the spatial support and feature choice.

Figure 10.4 shows that the presence of pixel information is key for providing maps with both a high classification accuracy and a sharp geometry. Interestingly, this is especially the case when a structured feature like the edge density is used. This shows that the pixel information and the edge density are in a way complimentary; each of the two features needs the other in order to provide an accurate classification. The graphs show that the edge density feature systematically boasts a higher geometric precision, for both superpixels and sliding windows, relative to the local statistic features. This might be due to the fact that the edge density is a structured feature, which takes into account local variations, and not only the overall variability in the spatial support. While this result may seem intuitive, the PBCM metric provides quantitative evidence to this conclusion.

The following paragraphs further the analysis of the results from each of the three different spatial supports: sliding windows, objects, and superpixels.

Table 10.2: Overall and per-class accuracy (F-score), for the best feature/support combinations on the tile T31TCJ. In the feature descriptions, P indicates the presence of pixel information, while LS stands for local statistics, and ED for edge density. In the spatial support descriptions, SW stands for sliding window, SP for superpixel and O for object. The bold numbers indicate the method achieving the highest value for each metric. The sliding window with edge density provides the highest values of F-score for 6 of the 14 classes present in this tile, in particular for the urban classes. Including context in this way decreases the recognition rates of two of the crop classes (ASC, AWC), compared to the pixel-based classification, indicating that these classes are relatively context-independent.

Spatial supp. Feature Scale	P	SW P+LS 5	SW P+ED 15	SP P+LS 5	SP P+ED 5	O P+LS	O P+ED
Overall Acc.	73.7%±0.21	76.9%±0.23	**77.4**%±0.23	75.0%±0.25	76.8%±0.22	71.3%±0.20	74.9%±0.13
Kappa	71.7%±0.27	75.1%±0.24	**75.6**%±0.25	73.1%±0.27	75.0%±0.24	69.1%±0.21	73.0%±0.12
ASC	**0.914**	0.902	0.900	0.890	0.891	0.891	0.884
AWC	**0.909**	0.899	0.900	0.897	0.900	0.891	0.888
BLF	0.831	0.831	**0.843**	0.812	0.841	0.804	0.837
COF	0.806	0.819	**0.841**	0.798	0.832	0.792	0.822
NGL	0.322	0.350	**0.343**	0.323	0.340	0.172	0.218
WOM	0.473	**0.481**	0.469	0.459	0.474	0.341	0.418
CUF	0.604	**0.687**	0.678	0.622	0.663	0.629	0.636
DUF	0.576	0.658	**0.690**	0.631	0.671	0.503	0.641
ICU	0.592	0.671	**0.690**	0.642	0.671	0.599	0.652
RSF	0.838	0.882	**0.899**	0.856	0.880	0.806	0.857
WAT	0.989	**0.990**	0.989	0.988	0.988	0.989	0.988
IGL	0.677	0.693	0.696	0.678	**0.698**	0.676	0.662
ORC	0.824	0.859	0.857	0.863	**0.864**	0.841	0.863
VIN	0.833	**0.868**	0.855	0.863	0.864	0.795	0.846
PBCM	51.3%±0.96	25.6%±1.14	**38.26**%±0.55	34.43%±0.4	33.06%±0.38	29.6%±0.31	35.6%±0.46

Sliding window

Figures 10.5c and 10.5d show the result of the classification when including respectively local statistics, and edge density features, in a sliding window neighborhood of size 11x11. Visually, it appears that the amount of noise is reduced, compared to the pixel based classification result. However, in the case of the local statistics features, some of the corners seem quite rounded, and fine elements like the river are deformed, and at some places even lost. Using a structured texture feature like edge density, this effect is largely reduced. On the other hand, round-shaped artifacts appear in the urban area, due to the isotropic nature of the square neighborhood.

Figure 10.4 confirms several of these observations. In particular, figure 10.4c shows that using scales in the order of 7-15 pixels, the sliding window neighborhoods can provide an increase in classification accuracy, but at the cost of a deterioration of the geometric precision. When the window size is very large (30×30), the geometric precision increases, but the statistical accuracy decreases. It appears that when the window size is too large, the contextual information is mostly discarded by the classifier. This explains why when increasing the window size, the geometric precision increases, because the classification gets closer to the pixel-based prediction. Furthermore, figures 10.4a and 10.4b show that both the classification accuracy and geometric precision of the sliding window neighborhood result are quite low when the pixel information is not included. Finally, figure 10.4d shows that using edge density features in combination with pixel features provides the best results, both in terms of classification accuracy and geometry.

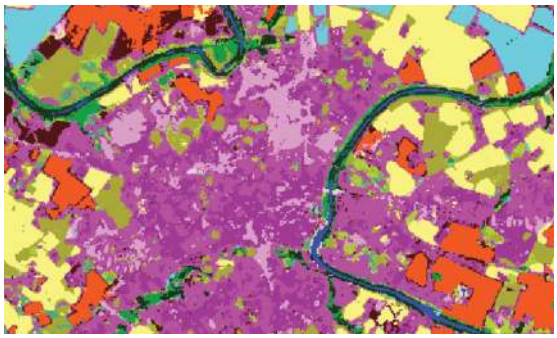
Mean Shift Object

The result using features calculated in a object from a Mean Shift segmentation is shown in figure 10.5g. The high-frequency elements are conserved, but the noise smoothing effect in the urban area is clearly less present than when using superpixels. This is due to the fact that segmentation methods like Mean Shift create very small segments in urban areas, because of the high spectral variability (over-segmentation). Calculating contextual features in objects does therefore not bring much information when compared to pixel-based classification. This is confirmed by figures 10.4a and 10.4d, as the increment in classification accuracy is quite limited, regardless of the feature choice.

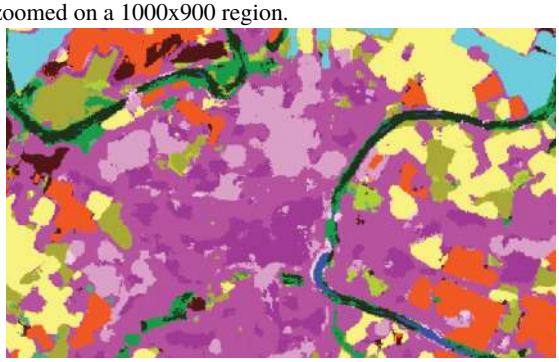
Superpixel

When using superpixel features with an initial segment size of 11×11, figures 10.5e, and 10.5f the noise seems filtered out and the high spatial frequency elements remain in the final result. The class edges in the urban areas have the tendency to follow the superpixel edges, which adhere to strong gradients in the image. Figure 10.4 shows that when using a superpixel as a spatial support, the pixel-based information once again has a positive effect, increasing both the classification accuracy and geometric precision. Furthermore, with the local statistics features,

(a) RGB bands of the summer date of the Sentinel-2 data set, zoomed on a 1000x900 region.

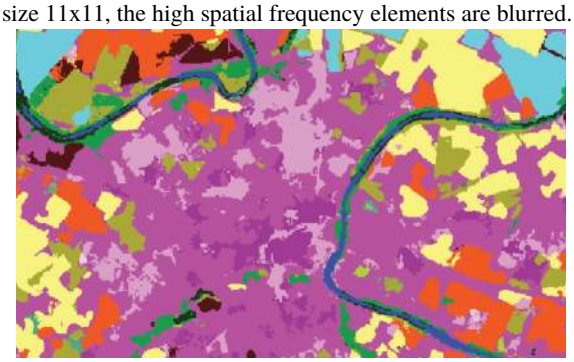


(b) Pixel-based classification by Random Forest. The main sources of errors are noise, and the lack of context.

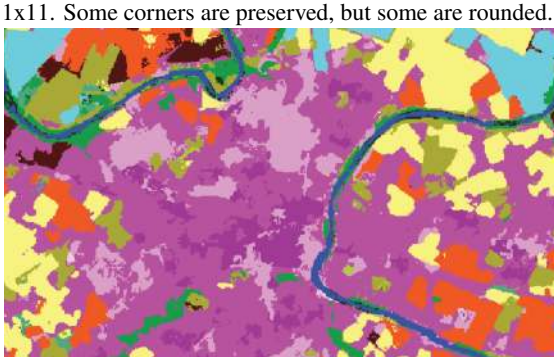


(c) Pixel, mean and variance features, in a sliding window of size 11x11, the high spatial frequency elements are blurred.

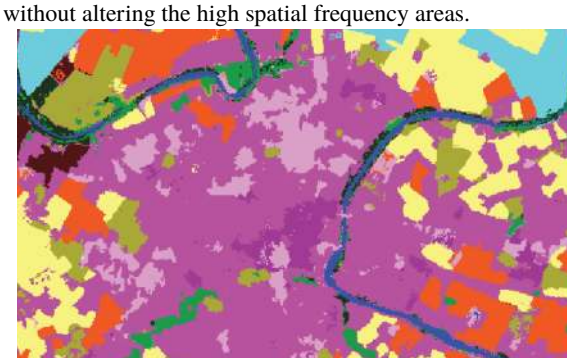
(d) Pixel and edge density features, in a sliding window of size 11x11. Some corners are preserved, but some are rounded.



(e) Pixel, mean and variance features, in a superpixel of average size 100. Intra-object classification noise is reduced without altering the high spatial frequency areas.



(f) Pixel and edge density features, in a superpixel of average size 100. The Overall Accuracy shows that the urban area has a finer characterization.



(g) Pixel, mean and variance features, in a Mean Shift object. Similar to the pixel-based result, due to over-segmentation.

(h) Pixel and edge density features, in a Mean Shift object. The urban classes are more precise than with local statistics.

Figure 10.5: Results of different combinations of spatial support shapes and feature choice.

figures 10.4a and 10.4c show that superpixels offer the best trade-off between the two evaluation criteria, although the optimal superpixel size is 5x5, which implies that this choice of feature and support is only relevant at smaller scales, for capturing a local context. When using the edge density and pixel information, the OA increases, and the PBCM is decent, but they remain slightly lower than when using sliding windows.

10.2.2 Experiments on the 11 tiles

Table 10.3: Average F-scores over the 11 tiles, of contextual classification using either sliding windows or superpixels as spatial supports, while maintaining the pixel values. The use of local statistics features provides a lower degree of geometric accuracy, whereas edge density features have higher values of both OA and PBCM. For a given feature, superpixels offer slightly higher values of PBCM than sliding windows. The classes that benefit the most from the inclusion of contextual information are the urban classes, which show improvements of 0.1-0.15 in F-score compared to the pixel-based classification.

Method Spatial Support Scale	Pixel	P+LS SW 15	P+ED SW 15	P+LS SP 10	P+ED SP 10
Overall Accuracy	86.1%±0.12	**88.5%** ±0.11	88.2%±0.11	88.1%±0.14	88.2%±0.10
Kappa	80.6%±0.17	**83.8%** ±0.16	83.6%±0.15	83.5%±0.16	83.6%±0.13
ASC	0.929	0.926	**0.932**	0.928	0.928
AWC	**0.903**	0.893	0.898	0.902	0.892
BLF	0.843	**0.877**	0.860	0.867	0.860
COF	0.868	**0.896**	0.877	0.882	0.876
NGL	0.317	**0.335**	0.333	0.319	0.311
WML	0.423	0.453	0.455	**0.455**	0.451
CUF	0.330	**0.459**	0.436	0.419	0.429
DUF	0.713	0.789	**0.815**	0.782	0.806
ICU	0.556	0.651	**0.709**	0.672	0.699
RSF	0.510	0.599	**0.690**	0.631	**0.690**
BRO	0.430	0.444	0.436	**0.463**	0.442
BDS	0.471	0.486	**0.521**	0.473	0.501
WAT	0.959	0.959	**0.963**	0.959	0.962
GPS	0.517	**0.527**	0.526	0.516	0.521
IGL	0.768	0.789	**0.790**	0.783	0.788
ORC	0.189	0.205	**0.262**	0.210	0.238
VIN	0.463	**0.502**	0.494	0.496	0.500
PBCM	44.1%±0.87	20.7%±0.51	28.3%±0.61	23.4%±0.35	**29.0%** ±0.55

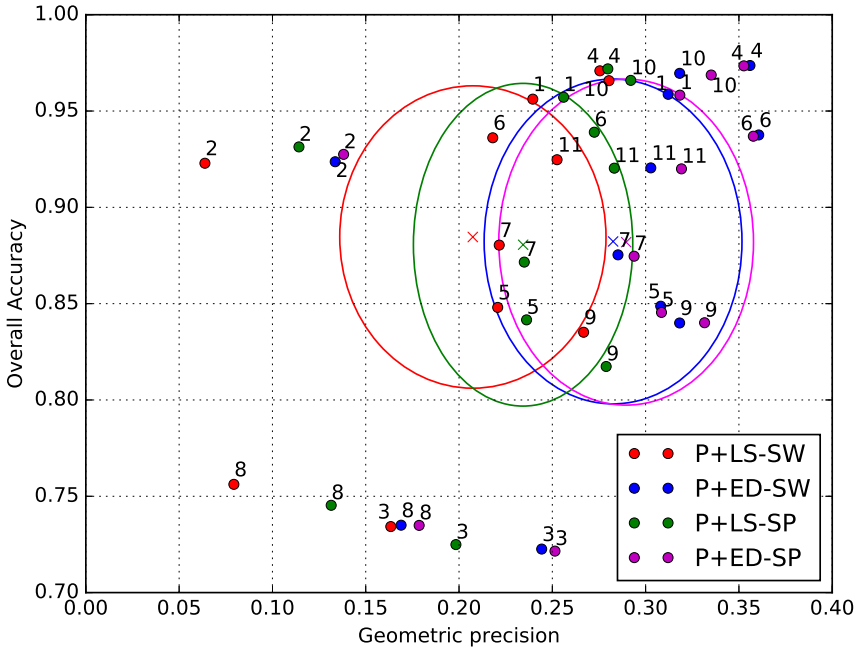


Figure 10.6: Overall Accuracy plotted against geometric precision for the tiles numbered 1-11 in the experimental data set (see Table 10.1 for the equivalent tile names). The centers of the ellipses, symbolized by crosses, are placed on the coordinates of the average score over the 11 tiles. The size of the ellipses represents the standard deviation.

Figure 10.6 shows the Overall Accuracy and geometric precision scores for the 11 tiles present in the evaluation data set. Each point represents the performance on one tile. The scores show some variability, due to the unique class proportions of each tile. Tiles 2 and 8 (T30TYN and T30TGK) show a far lower absolute PBCM than the other tiles, no matter which method is considered. This is linked to the low number of corner rich crop classes in these tiles, which cover mountainous areas, respectively, the Alps and the Pyrenees. However, the absolute value of the PBCM metric is not very important, as it is meant to be used as a relative metric to compare different methods.

The ellipses in figure 10.6 show the mean and standard deviation of the scores, over the 11 tiles. The center of the ellipse is placed on the mean value, while the length of the semi-minor and semi-major axes are equal to the standard deviation of the two scores.

It is interesting to note that the relative values of the scores for a given method are similar from one tile to another. This shows that the relative PBCM metric is quite robust to diversity in tile content, in other words, it provides a reliable indication of the geometric degradation of a contextual classification method.

In terms of Overall Accuracy, the sliding window provides the highest performances, especially in combination with the edge density feature, which confirms the conclusions that were drawn on T31TCJ in Section 10.2.1.

The results show that the choice of spatial support and the choice of features cannot be made independently. Indeed, in the case of the local statistics features (mean and variance), the sliding windows have a strong tendency to deteriorate the geometry of the output map, particularly by smoothing out sharp corners and erasing fine elements. This phenomenon could be linked to the fact that the local statistics features are unstructured, meaning that they do not depend on the arrangement of pixels in the spatial support, and are therefore akin to a low-pass filter. It is interesting to note that when these features are combined with a superpixel support, a modest improvement in classification accuracy is reached, while maintaining a higher level of geometric precision. It is possible that the averaging nature of unstructured features implies that they run the risk of smoothing the output geometry, and therefore should be applied in combination with an image segmentation technique.

However, when using the edge density feature, the conclusion is quite different. Sliding windows provide the highest classification improvement, while generating a geometrically precise map, at least in the corners of the majority crop classes. Meanwhile, with these same features, superpixels seem to capture less valuable contextual information than sliding windows, although they also preserve the geometry, and do improve the context-dependent classes. This could be explained by the structured aspect of the edge density feature, which is based on the average of a local gradient, and therefore depends on the spatial arrangement of the pixels in the context, making corners and fine elements easier to characterize. This feature being more similar to a high-pass filter, is therefore adapted for use with a sliding window.

10.2.3 Overview of the results

These experiments demonstrate the performance of 3 different spatial supports (sliding windows, superpixels, objects), in combination with 2 types of contextual features (local statistics, edge density), in their capacity to improve upon the quality of a pixel-based classification. The objective is to demonstrate how the choice of contextual feature and spatial support can be linked.

The first main conclusion of these results is that pixel information is paramount for achieving a classification with both a precise geometry and a precise overall accuracy. Replacing the pixel information entirely with a contextual feature in any of the three spatial supports provides inferior performance to when the pixel and contextual features are combined. This is true for large sizes of spatial support, but more importantly, even for small spatial supports. Discarding the pixel information has a negative impact both on the OA metric and the PBCM metric. Indeed, pixel features are defined at the finest spatial resolution, and therefore provide many of the geometric details. They also contain the high-dimensional time series, and in this way provide extremely valuable information for classification. This might not be the case at finer spatial resolutions or on different land cover mapping problems, but with a 10m spatial resolution for fine land cover classes like roads and isolated buildings, it seems as though the pixel information should not be replaced.

Second of all, it is clear that the edge density feature is a better descriptor of local variability than the variance, or than the mean and variance combined. Regardless of the spatial support choice, this feature provides the highest values of OA. Generally speaking, it would seem that the use of an unstructured feature is beneficial for the both the class accuracy and the geometry of the result.

Finally, on the subject of spatial supports, it seems as though the sliding window provides the highest OA, but can deteriorate the geometry if combined with an unstructured feature, or if the pixel information is not included. In second place, the superpixel offers slightly lower OA compared to the sliding window, but with a generally higher PBCM. However, it offers a compromise for unstructured features, providing a lower OA with a more precise geometry. This is a sign that unstructured features must be used with care, as they present the highest risk of geometric degradation. Object segments are unable to match the performance of the superpixels or sliding windows on this problem. Indeed, the influence of the shape features (area, perimeter) is insufficient to differentiate between the target context-dependent classes.

However, there is a slight contradiction in these results. The PBCM of the classification map resulting from the combination of pixel information and edge density in sliding windows is quite high, which should indicate that the corners are well recognized with respect to a pixel-based classification. However, this opposes the visual analysis of the urban area, figure 10.5d, which suggests that the sliding window methods have the tendency to deteriorate

the geometry of the urban classes.

This apparent discrepancy is the first evidence that the geometric precision might not be the same in all parts of the image. It is indeed possible that a contextual method generates maps with a precise geometry for certain classes, but with poorly estimated borders for other classes. It is important to remember that crop classes are well recognized by the pixel-based classifier, in other words, they are relatively *context-independent*. This implies that their geometry is well defined by the pixel information alone, regardless of the contextual feature choice. When pixel and contextual features are combined, the classifier should learn to use the pixel-based information for context-independent classes, and the contextual information for the context-dependent classes. For example, to recognize a crop, the time series of surface reflectances is sufficient, whereas to recognize an urban density class, information such as the edge density in a wider neighborhood is required. The shape of this neighborhood therefore only has an influence on the geometry of the context-dependent classes.

In fact, the corners of the annual crop classes make up for the majority of the corners in the tile T31TCJ. This is due to both to their rectangular or polygonal shape, and to their abundance in the study area. In this tile, the number of corners contained in context-dependent classes is significantly lower than the number of corners formed by elements of context-independent classes. This implies that the PBCM evaluates the geometric precision of the classes that form the majority of the corners, which in this case are formed by context-independent classes, and are therefore not influenced by the shape of the spatial support.

This implies that the value of the PBCM should be considered as an indicator of the overall geometry, and that a result with a relatively high value of PBCM can still be subject to geometric degradation. Similarly, two results with similar values of PBCM can have very different geometric qualities in certain parts of the image. This implies that the analysis of the geometry cannot be simplified down to one overall metric such as the PBCM, and must be made in conjunction with a visual analysis of the minority, context-dependent areas.

Image-based contextual features provide significant improvements compared to the pixel-based classification on context-dependent classes in the studied areas. However, they also have two main limitations.

The first limitation of image-based features is the maximal area that they can consider without deteriorating the OA or PBCM of the result, which physically equates to an area of $100-150m^2$ (a scale of 10-15 pixels), regardless of the choice in spatial support. Much of the behavior of the context is not captured within such an area. In fact, the smallest objects of the training data occupy a MMU of $25m^2$ [Montero et al., 2014], which means that such features can describe the context of at most 4 of the smallest objects. Many of the objects in the training data are much larger than the MMU, and cannot be described within such a small area.

The second limitation is that these features lack a multi-scale description, which as was mentioned before is an important aspect for the variety of target land cover classes, and complex relations they can have. This is due to the high dimension of the initial data set, which is multiplied at each use of feature or scale. When only one scale of context is used, there is a trade-off between local and long-range information, which are both valuable to describe intra-object and inter-object relations.

In fact, the first limitation, the inability to consider long-range information, can be seen as a consequence of the second limitation, which is that only one scale is considered. The use of high-dimensional imagery imposes that only one scale can be used, if such direct image-based features are chosen as contextual descriptors.

Otherwise, the dimension of the features can be reduced. This can be done in several ways, some of which were described in Part III, Chapter 8. In order to keep the general framework of high-dimensional imagery intact, methods such as feature selection are avoided. Section 8.2.1 pointed towards the use of a unstructured contextual feature to be calculated in a spatial support: the histogram of local classes from a previous prediction. This low-dimensional feature can therefore be used in several multi-scale spatial supports. The following section shows that multi-scale features are in fact able to benefit from this trade-off, and to consider both local aspects, and long-range information from a far wider area, ranging up to $1km^2$.

10.3 Results of semantic contextual features

This section provides the results of the semantic contextual features that were found as viable for application on such a high-dimensional data set in Part III. These experiments provide the base of the design of the Histogram of Auto-Context Classes in Superpixels (HACCS) process, as well as the justifications for many of the choices that were made, such as the use of superpixels rather than sliding windows.

10.3.1 Experiments on T31TCJ

The objective of this initial set of experimentations on T31TCJ is to compare spatial supports of different shapes and sizes in their ability to improve the classification rates of the target context-dependent classes. The contextual

feature is the local histogram of the classes from the previous prediction. All of the results are shown after 3 iterations. Three types of spatial support are evaluated here:

1. Sliding windows;
2. Multi-scale superpixels;
3. Adjacency layers.

Indeed, object segments are not evaluated in this set of experiments, following the conclusions of the experiments based on other unstructured features, in Section 10.2.

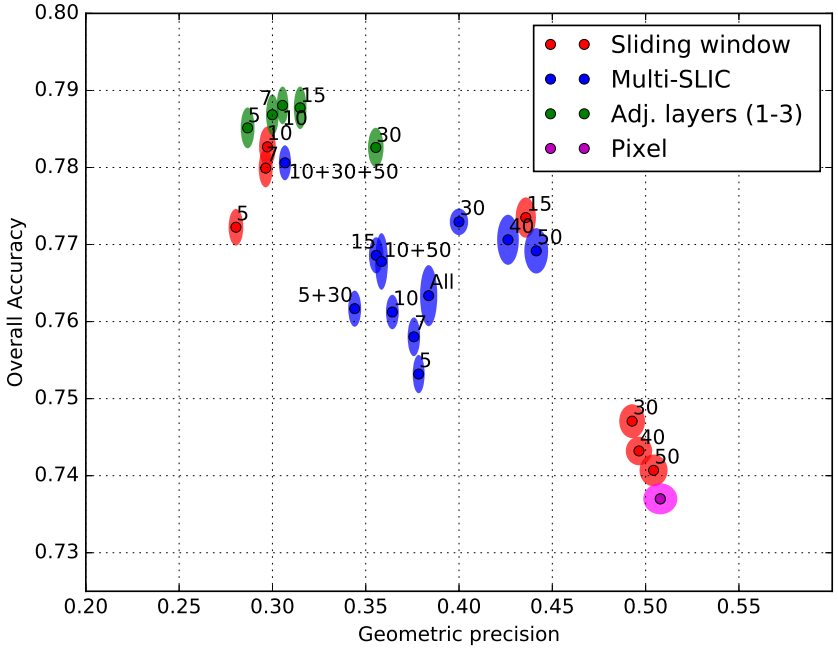


Figure 10.7: Overall Accuracy plotted against geometric precision for different choices of spatial support, after 3 iterations of the HACCS process over the T31TCJ tile. The colored ellipses surrounding each point represent the standard deviation obtained across 10 runs with different samplings of the training data. The numbers next to the points indicate the scale of the spatial support, in other words the diameter for sliding window, the superpixel size parameter for SLIC, and the size of the base superpixel for the adjacency layers. Adjacency layers provide the strongest results in Overall Accuracy, and with acceptable geometric precision if the appropriate base superpixel size is well chosen.

Figure 10.7 shows the OA and PBCM after 3 iterations of the HACCS process with different scale parameters for the spatial support. This is compared to the pixel-based classification.

First of all, sliding windows show performances with varying degrees of geometric precision, which shows that they depend strongly on the choice of window size. If the window size is too large, the contextual information is discarded by the Random Forest, and the result becomes similar to the pixel-based result. A window size between 7 and 15 pixels provides a significant improvement in Overall Accuracy. If the window size is too small, i.e. 5x5 pixels, the geometry is altered with respect to the pixel-based classification, for no real boost in overall accuracy. This is confimed by Tables B.9 - B.12 in Appendix B, on pages 190 - 191.

Superpixels have overall higher degrees of geometric precision, and provide more consistent results. Using only one scale of superpixel, the OA is highest for a value of 30. When compared to the result using a sliding window of equivalent size (30x30 pixels), it appears that superpixels do provide a finer long-range characterization. However, in order to reach an OA similar to the best sliding windows, it is necessary to include several scales of superpixels, (10 30 and 50). On the other hand, when using all available scales of superpixels (5, 7, 10, 15, 30, 40 and 50) the result is not better than certain of those scale choices alone.

Finally, the spatial support that consistently provides the highest OA is the use of 3 adjacency layers of superpixels. In Figure 10.8, the numbers next to the points indicate the size of the base superpixel for the layers. Both small and large superpixel bases provide a higher overall accuracy than all of the other methods. This does come at a cost in geometric accuracy, which is highest for small spatial support sizes.

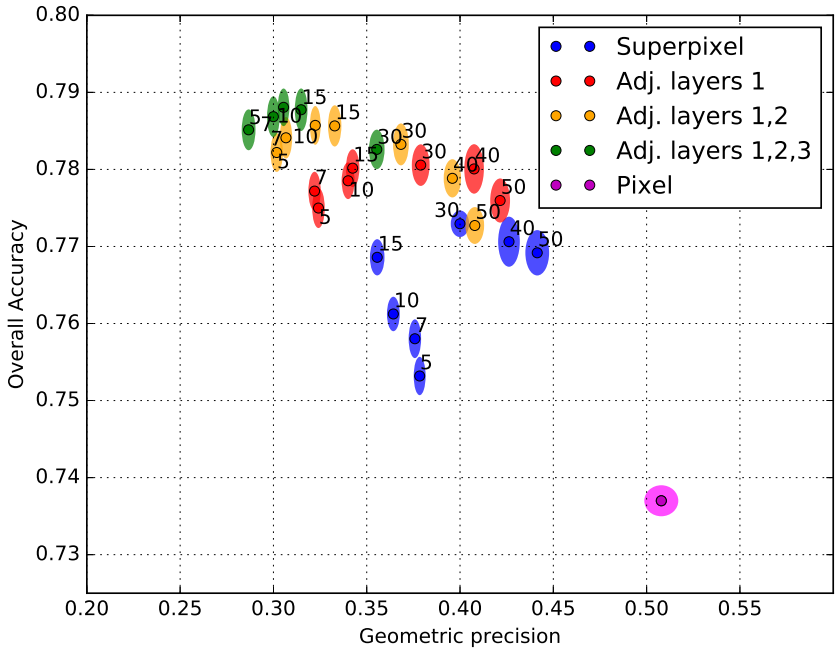


Figure 10.8: Overall Accuracy plotted against geometric precision for different choices of base superpixel sizes, and number of layers, for the adjacency layer spatial support, following the same conventions as Figure 10.7.

Figure 10.8 shows the classification scores of the different base superpixel scales, used with 1 to 3 adjacency layers. First of all, this graph shows that adding more layers improves the classification accuracy, but deteriorates the geometry of the result.

Secondly, for a given number of layers, there seems to be an optimal base scale, that provides the highest overall accuracy. This optimal scale depends on the number of scales used. Using 3 layers, the optimal base scale is around 10, when using 2 layers it lies near a value of 15, and when using 1 layer, around 30. The average diameter D covered by an adjacency containing L layers can be calculated as $D = b(2L + 1)$, where b is the scale of the base superpixel.

Simply put, the adjacency layer diameters that provide the highest OA cover an physical area of nearly the same size, respectively, for 1, 2 and 3 layers, 90, 75 and 70 pixels, which means an area between $700m^2$ to $1km^2$ around the central pixel. It can be added that taking a larger number of layers with finer superpixels provides a higher OA and a lower PBCM. Naturally, this comes at the cost of a higher number of features, but these remain low-dimensional compared to the time series of reflectance features.

First of all, this indicates that there may be a maximal area at which local histogram features are able to capture relevant information. Indeed, beyond a certain area, the overall accuracy decreases, meaning that the contextual feature is no longer as relevant.

Using a high number of adjacency layers within a given area provides a finer description of the content of this area. Nonetheless, this may lead to a geometric degradation. In particular, this manifests as a "leaking" of urban areas into non-urban areas. Areas of vegetation nearing urban areas contain urban classes in their histogram, and are therefore likely to be classified as discontinuous urban areas.

The high performance of the adjacency layers is also shown in Table 10.4, where the F-scores of the different classes are listed for the best performing spatial supports. Indeed, using 3 adjacency layers with a base superpixel of 10 provides the higest recognition rates of the four urban classes, as well as equivalent performance to sliding windows and multi-scale superpixels in the other classes. The PBCM of this method is also similar to the other high-performing methods. The tables of the F-scores for the different scale choices of these methods are given in the appendices, in Tables B.9-B.12, on page 190. In particular, Table B.12 shows that the base superpixel value of 10 is not the highest performing for all classes, but that the classification accuracy and F-scores are very sensitive to this parameter.

Figure 10.9 shows the *variable importance* (VI) associated to the pixel-based features before the iterations, and to the semantic features after three iterations for different superpixel sizes and combinations. VI is defined by [Breiman, 2001] as the loss in average classification error (out-of-bag error) when the variable in question is replaced with a random permutation of its elements. This provides an indication of the most useful features for the

Table 10.4: F-scores of the highest performing spatial supports, using the local class histogram feature. In this table, SW indicates a sliding window neighborhood, with the scale parameter being the side of the square window. SP indicates that one or several superpixels of corresponding scales are used. AL-3 indicates that 3 succesive adjacency layers are used, based on superpixels of a given size. The adjacency layer provides the highest overall accuracy, and the highest F-Scores for the urban classes, and is in close contention for first place on the other classes as well.

Spatial Support	Pixel	SW	SP	SP	AL-3
Scale		10	30	10 30 50	10
Overall Acc.	73.7%±0.21	78.3% ±0.26	77.3%±0.17	78.1% ±0.23	**78.8%** ±0.24
Kappa	71.7%±0.27	76.6% ±0.28	75.5%±0.19	76.4% ±0.24	**77.2%** ±0.26
ASC	0.914	0.923	**0.927**	0.922	0.921
AWC	0.909	0.916	0.917	0.916	**0.918**
BLF	0.831	**0.853**	0.848	0.849	0.852
COF	0.806	0.842	0.838	0.842	**0.854**
NGL	0.322	0.342	**0.359**	0.333	0.347
WOM	0.473	0.531	0.514	0.518	**0.538**
CUF	0.604	0.704	0.667	0.708	**0.713**
DUF	0.576	0.662	0.626	0.651	**0.670**
ICU	0.592	0.676	0.643	0.665	**0.679**
RSF	0.838	0.889	0.882	0.890	**0.910**
WAT	0.989	**0.990**	0.989	0.989	0.989
IGL	0.677	**0.719**	0.696	0.713	0.715
ORC	0.824	0.876	0.878	0.880	**0.878**
VIN	0.833	**0.883**	**0.883**	0.878	0.878
PBCM	51.3%±0.96	29.7%±0.45	**40.0%**±0.49	30.7%±0.33	30.5%±0.31

classification, in general. However, variables with a low importance can still be valuable for classifying minority elements in the data set, which therefore have a low impact on the average classification error. This figure shows that the semantic contextual features used during the HACCS process have a high importance compared to the pixel-based classification, and the different classes have different importance which depends both on the scale itself, and the choice of scales.

Figure 10.9a shows the case when only pixel features are used. It appears that the summer and spring dates, as well as certain dates in the winter, are considered as very important. Moreover, the 9th spectral band (SWIR, 2202.4nm) seems to be more relevant than some of the other bands.

Figures 10.9b - 10.9d, show the case where only one scale of superpixels is used as a spatial support for HACCS. Here, differences can be observed for different scale sizes. When the superpixel describes a very local neighborhood, the histogram of the vegetation classes seems to be a very important indicator. This is possibly a sign that the histogram features are used to smooth out intra-object classification noise, such as patches of bare soil in the agricultural fields. Inversely, when the scale is large, the histograms of the crop classes are no longer as useful, however, urban classes seem to be important to the classifier. It is interesting to note that at a large scale, the contextual features are still useful, and considered important, by the classifier, when only one scale is available.

When a small and large scales are combined together, as is shown in figures 10.9e and 10.9f, the local features (scales 5-10) appear to be considered as more important than the long-range features. Among the urban classes, it seems that the presence of the Road Surfaces class provides useful local contextual information.

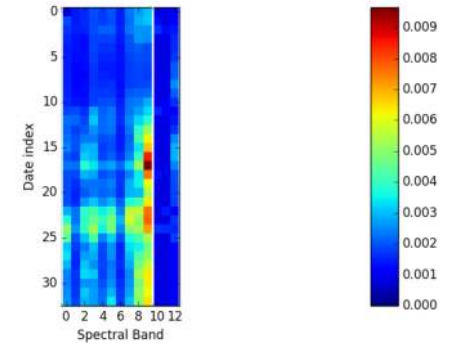
The following section presents the results of the application of adjacency layers and multi-scale superpixels to the 11 tiles of the Sentinel-2 data set, and their comparison with the Deep Learning method, FG-Unet, that was presented in Part III.

10.3.2 Experiments on the 11 tiles

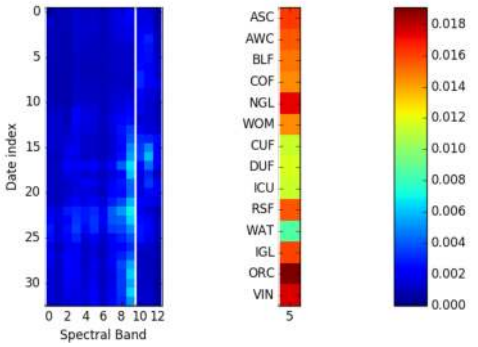
The graph in Figure 10.12 plots the Overall Accuracy against the PBCM, for the different methods and for each different tile. The ellipses show the mean and standard deviation of the two scores calculated across the eleven tiles. This is done to provide an indication of the average performance of the classification over a wide area, as well as the robustness of the different methods to differences in class behavior and class proportions. Each ellipse is centered on the mean of the performance indicator across the eleven tiles, and the semi-axes are equal to the standard deviation. Figure 10.12 shows that the multi-scale HACCS method provides similar results to the FG-Unet approach in terms of Overall Accuracy, but with a geometric precision that is closer to that of the pixel-based classification.

Moreover, it is interesting to note that the relative position of the points for each tile is similar for the different methods. Indeed, tiles that show relatively low overall accuracy for one method show a relatively low overall accuracy for all methods. In other words, this implies that the *inter-tile* differences are linked mostly to variations in class proportions and class behavior, and not to differences in how they are classified by the various methods.

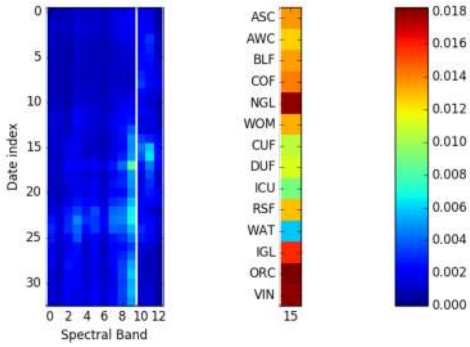
The impact of the iterations on the overall accuracy and geometric precision is shown in Figure 10.11, which plots these two metrics across three iterations of HACCS, for the eleven tiles. The arrows show the rapid progress



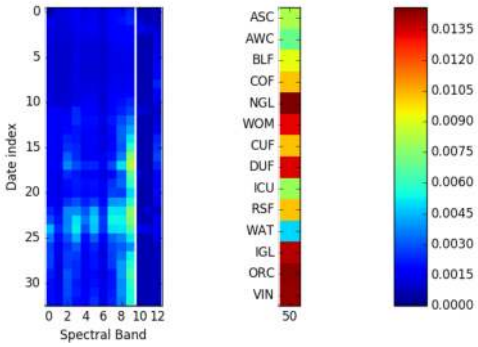
(a) Pixel-based.



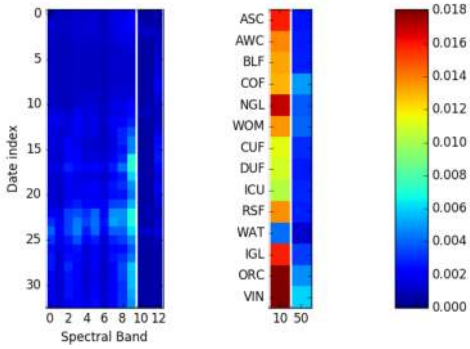
(b) Superpixel scale 5.



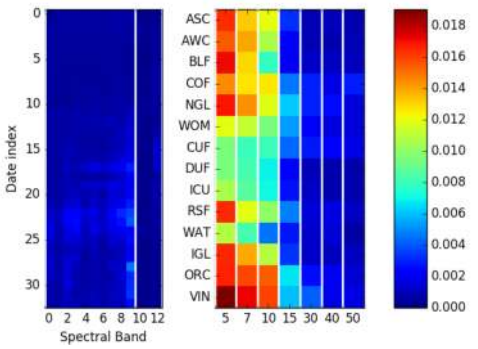
(c) Superpixel scale 15.



(d) Superpixel scale 50.



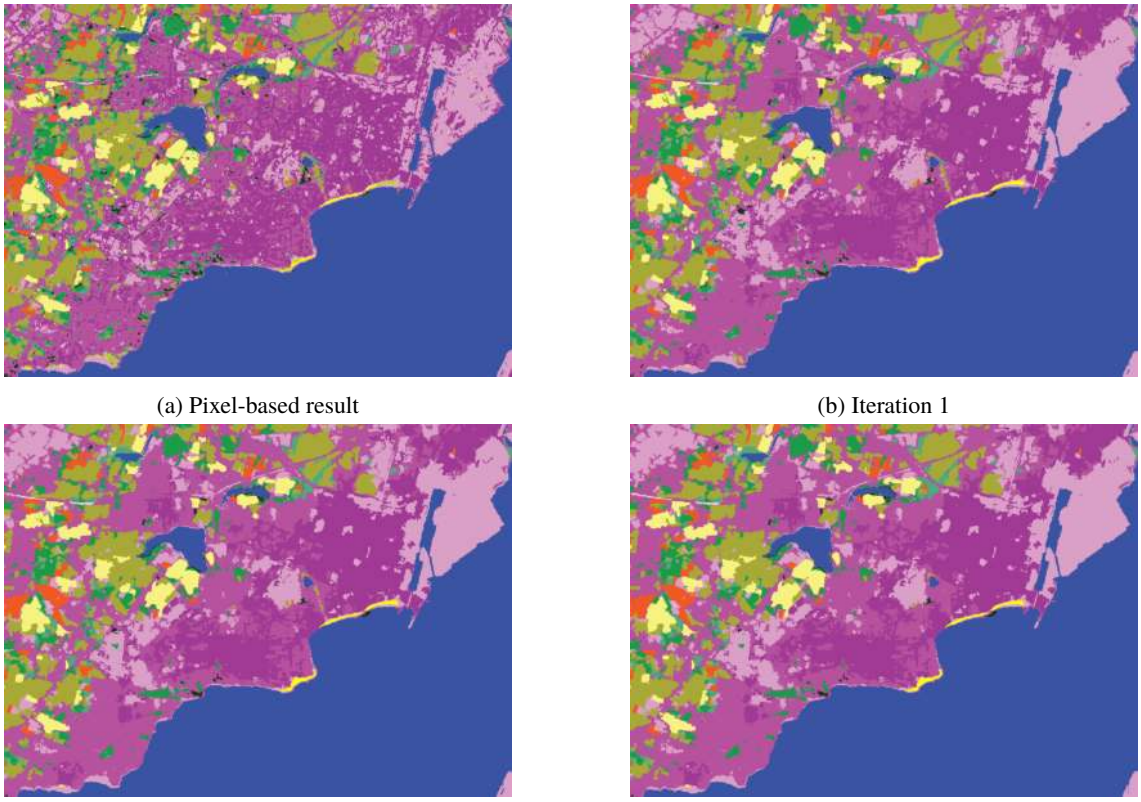
(e) Superpixels scales 10 and 50.



(f) Superpixels scales 5, 7, 10, 15, 30, 40, and 50.

Figure 10.9: Evolution of the RF variable importance for different superpixel scales and combination of scales, after 3 iterations of HACCS. In each figure, the importance of the pixel features is shown on the left side, organized by date index and spectral band. A red color indicates a feature with a high importance. The dates cover a period of the 2016 year, starting in January. The importance of the histogram of the local classes features in superpixels are shown on the right side of each feature, and are organized according to the scale of feature and the associated class. It appears that the semantic contextual features are indeed considered important by the RF, and that when more than one scale is provided, the local scales provide more important contextual information.

of the performance indicators, which converge within very few (2-3) iterations. This is also visible in Figure 10.10; the classification results in figures 10.10c and 10.10d are almost identical. Generally speaking, the HACCS process has the effect of increasing the overall accuracy of the classification with respect to the pixel-based classification, regardless of the tile on which it is applied. This also comes at the cost of a decrease in PBCM, as some of the corners are lost or displaced. This figure also shows that tiles with a lower initial overall accuracy show stronger improvements than tiles that are already relatively well classified. This is explained by the fact that these difficult tiles most likely contain larger proportions of context-dependent classes, making them prone to errors when using a pixel-based approach. For example, tiles 31TYN and 31TGK, respectively indexed 3 and 8 in Table 10.1 cover mountainous areas, and contain significant proportions of the *Bare Rock* class, which is very often confused with urban cover. This is also the case for tile 31UDQ which covers the extended urban area of Paris.



(a) Pixel-based result (b) Iteration 1

(c) Iteration 2 (d) Iteration 3

Figure 10.10: Evolution of the multi-scale HACCS classification result throughout three iterations. At each iteration, the result of the previous classification is used to re-estimate the histograms in several scales of superpixels, which are used as contextual features. Most of the differences are observed at iteration 1, when contextual information is included for the first time. After a few iterations, the classification result shows very few changes between successive iterations.

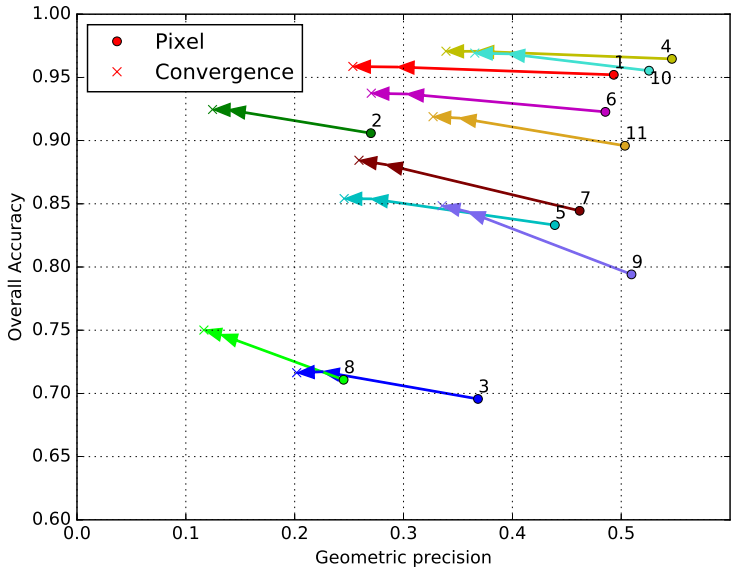


Figure 10.11: Iterations of HACCS with superpixels at scales 10, 30 and 50 as spatial supports. The arrows represent successive iterations, which start from the pixel-based classification, and reach the convergence point once no significant change is detected. The numbers and colors represent the different tiles, once again following the definitions from table 10.1.

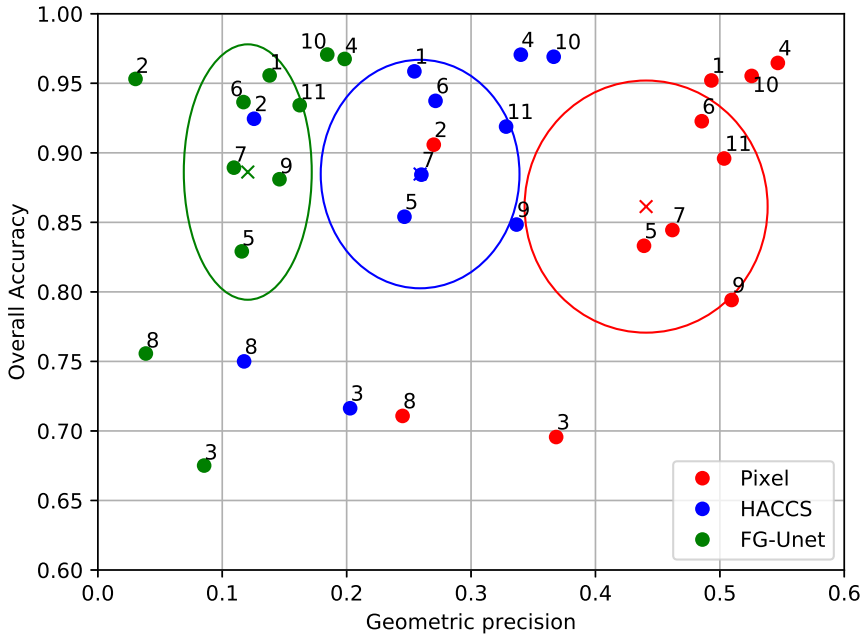


Figure 10.12: Statistic accuracy and geometric accuracy of the pixel based classification, of HACCS in multi-scale superpixels and in adjacency layers, as well as the the FG-Unet (CNN) on the 11 Sentinel-2 tiles. The number labels designate the tile that was used for training and testing, the correspondence is given in Table 10.1.

Next, the per-class performances can be analyzed using the F-score metric. The average OA, Kappa, and F-scores over the 11 tiles of the experimental data set are shown in Table 10.5. The only classes where the pixel-based prediction is generally more accurate are the Winter Crop and Water Bodies classes, however, the difference is quite small, and these classes are not a primary focus as they already exhibit very high recognition rates.

In terms of thematic accuracy, the two contextual methods, HACCS and FG-Unet, show a relatively equivalent performance, with neither method strongly outperforming the other. However, it is worth noting that the CNN architecture provides generally more precise results on the urban fabric classes: continuous urban fabric, discontinuous urban fabric and roads, whereas the HACCS method has a high F-score for the I.C.U. and Road Surfaces classes, as well as the vegetation classes, like coniferous forests, grasslands (both natural and intensive), orchards, and vineyards. On the two classes where the pixel-based method provides the best performance, Annual Winter Crops and Water, HACCS provides a slightly more accurate result in average. Overall, HACCS has higher F-scores than the D-CNN on 13 out of the 17 classes.

Of the four classes where HACCS has weaker results than FG-Unet, the differences are relatively low for three of the classes: 1.5% for CUF and DUF, less than 1% for the BDS. The results of the GPS (glaciers and permanent snow) class are not well represented in this evaluation, as they are only present in 2 of the 11 tiles, and very scarcely in T30TYN. As a matter of fact, the T31TGK tile is the only one to contain a representative quantity of GPS samples for training and validation. On this mountainous area, HACCS with adjacency layers and FG-Unet have an F-score of respectively 0.859 and 0.918, in others words, a difference of around 6%. This means that the GPS class is the only one where the FG-Unet truly outperforms HACCS. However, this would benefit from a further validation on different snowy areas.

For a visual analysis of the results, Figure 10.13 shows the classification maps generated by the two contextual approaches: FG-Unet and HACCS, compared to the pixel-based Random Forest over the harbor area of Saint-Nazaire. This area is located in the T30TWT tile, in the red dotted rectangle shown in Figure 10.3. Visually speaking, the HACCS method provides a higher degree of geometric accuracy than the CNN method, as it encourages label homogeneity in superpixels, which represent parts of physical entities that are present in the original time series. This translates as sharp corners, and other fine details in the classification result. This is coherent with the patterns that are statistically observed over the entire data set.

Table 10.5: Average performance across the 11 tiles. The image-based features are also shown here, in order to compare their performance to the semantic features. HACCS with adjacency layers has the best OA and κ, with the FG-Unet method coming in second. There is not one method with the highest values for all of the classes. FG-Unet is able to recognize urban density classes (CUF, DUF) with more accuracy, but is weaker than the Edge Density feature on roads and industrial & commercial units (RSF, ICU).

Method	Pixel	P+ED	P+ED	HACCS	HACCS	FG-Unet
Spatial Support		SW	SP	Adj. Layer	Multi-SP	
Scale		15	10	10 + 3 layers	10+30+50	
Overall Accuracy	86.1%±0.12	88.2%±0.11	88.2%±0.10	**89.1%** ±0.15	88.5%±0.17	88.6%
κ	80.6%±0.17	83.6%±0.15	83.6%±0.13	**84.7%** ±0.20	83.9%±0.23	84.2%
ASC	0.929	0.932	0.928	0.937	**0.940**	0.926
AWC	**0.903**	0.898	0.892	0.895	0.899	0.893
BLF	0.843	0.860	0.860	**0.886**	0.879	0.882
COF	0.868	0.877	0.876	**0.904**	0.899	0.843
NGL	0.321	0.333	0.311	**0.339**	0.332	0.244
WML	0.423	0.455	0.451	**0.477**	0.469	0.461
CUF	0.330	0.436	0.429	0.485	0.463	**0.499**
DUF	0.713	0.815	0.806	0.819	0.798	**0.835**
ICU	0.556	**0.709**	0.699	0.697	0.671	0.660
RSF	0.509	**0.690**	**0.690**	0.673	0.648	0.666
BRO	0.430	0.436	0.442	0.441	**0.448**	0.423
BDS	0.469	0.521	0.501	0.598	0.551	**0.606**
WAT	0.959	**0.963**	0.962	0.956	0.954	0.946
GPS	0.517	0.526	0.521	0.528	0.516	**0.718**
IGL	0.768	0.790	0.788	**0.798**	0.794	0.761
ORC	0.189	**0.262**	0.238	0.235	0.249	0.214
VIN	0.464	0.494	0.500	0.524	**0.579**	0.494
PBCM	44.2%±0.24	28.3%±0.61	**29.0%** ±0.55	23.7% ±0.42	26.1%±0.31	11.8%

10.3.3 Overview of the results

This first set of experiments on T31TCJ aimed to compare different spatial support types for including semantic contextual information under the form of a local histogram feature. To this end, 3 types of spatial support were compared :

1. Sliding windows
2. Superpixels at several different scales
3. Adjacency layers formed on a superpixel grid.

Overall, the adjacency layers provided the results with the highest OA, under the condition they were parametrized appropriately. They benefit most from a large number of layers, although this must be combined with a relatively fine superpixel grid. The optimal set of parameters for different numbers of layers and superpixel grids consistently represented an area of $700m^2$ to $1km^2$ around the central pixel, indicating that these features benefit from long-range information, but not beyond a certain limit. This might be the maximal scale at which class histograms are relevant contextual descriptors. However, these features tend to poorly estimate the borders between urban and non-urban areas, by shifting them towards the non-urban areas. This translated as decrease of the PBCM metric.

In second place, the sliding windows provided results with a relatively high class accuracy, but have other negative influences on the geometric quality of the result. Visually, the maps contained rounded corners with certain smooth details erased. Compared to adjacency layers, they provided a better estimation of the edges of objects, but a poorer estimation of the corners and small elements.

Finally, the experiments on the 11-tile data set show that the HACCS process, in combination with adjacency layers, provided a similar performance to the CNN architecture in terms of Overall Accuracy. Moreover, the HACCS process produced maps with a higher geometric accuracy than the FG-Unet method. This result was consistent across 11 Sentinel-2 tiles, which each cover an area of around ten thousand square kilometers, meaning that the proposed method is relatively robust to differences in class proportion and land cover class behavior.

10.4 Conclusions

The classification problem, presented in Section 10.1, is a land cover mapping problem over 110×110km tiles covering varous parts of the French territory. The use of 10m spatial resolution Sentinel-2 time series enables a variety of natural, agricultural, and artificial classes to be recognized using supervised classification methods. Training data for these methods almost always comes in a sparse form, which is why this case study is representative of a real world land cover mapping problem. Moreover, high-dimensional time series are rarely used in combination with contextual features, due to limitations in the total number of features that can be simultaneously considered by a supervised classifier.

In particular the analysis of image-based features presented in Section 10.2, suggests that edge density feature allows for the highest values of OA with sliding windows, while maintaining a similar PBCM to the superpixels. The geometric analysis on these cases shows that context-dependent classes and context-independent classes can have a very different geometric quality, which is not necessarily reflected in the PBCM. Visually speaking, the superpixel results provide a finer geometry in urban areas, particularly with the edge density feature.

However, image-based features remain limited by their inability to consider multi-scale aspects, which restrains the maximal size at which they can descibe context. This leads to the results of the experiments on semantic contextual features, which is presented in 10.3. In particular, this section shows that one way of reducing this high-dimensionality is to use a supervised classification method. Classification can be seen as a projection of the high-dimensional data on a low dimensional label space. The result of the classification of nearby pixels provides valuable cues towards the correct classification of a given pixel. Spatial information is integrated by calculating the histogram of classes from a previous prediction in a certain spatial support containing the pixel.

The main advantage of using the histogram is that it does not require many features for common classification problems. Indeed, the number of features is conditioned by the size of the class nomenclature and the number of scales, and not by the type of imagery, making it adapted for classifying images presenting a large number of original pixel features.

These results push towards the design of the HACCS process, which can be described as follows. First, a supervised pixel-based classifier is trained and applied to the entire image. Then, the histogram of classes is calculated in superpixel segments, which are extracted from the image. These histograms serve as features for the following classification. This process should then be repeated several times, in order for the correct contextual description to be reached in all areas of the image.

The configuration that provides the highest overall accuracy is the use of superpixels or adjacency layers of superpixels as spatial support, iterated several times. After 3 iterations, no more significant changes are observerd.

Nonetheless, this configuration does not provide the highest values of PBCM, compared to sliding windows. Although the result is visually satisfying in many areas, with the presence of sharp corners and some fine elements preserved, there is a tendency of the method to displace the borders of urban areas towards the surrounding non-urban areas. In general this manifests as an over estimation of the discontinuous urban areas class.

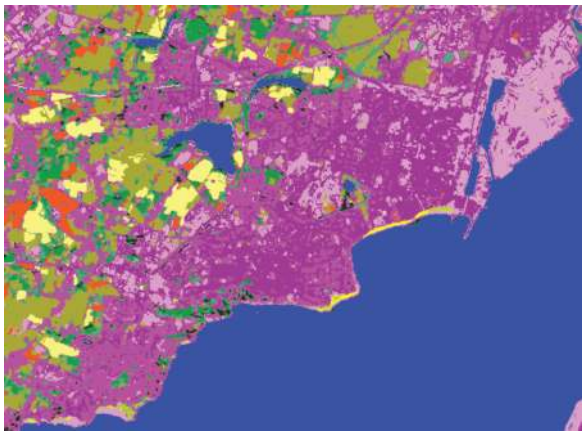
The experiments on all 11 tiles set show that the HACCS process provides a similar performance to the CNN architecture in terms of overall accuracy. Moreover, the HACCS process produces maps with a higher geometric accuracy, measured by how well the classification can restore sharp corners compared to a pixel-based classification. This result is consistent across 11 Sentinel-2 tiles, which each cover an area of around ten thousand square kilometers, meaning that the proposed method is relatively robust to differences in class proportion and land cover class behavior.

Finally, it is important to mention the total computation time that the various methods require in order to perform supervised training. While speed is not the primary criterion for evaluating the different methods, as this research work focuses on the quality of the obtained maps, it is interesting to compare the efficiency of the different methods, in achieving their goal. Table 10.6 shows the training time per CPU, which is indicative of the general efficiency of the methods. This measure does not consider the use of parallel computation, which can be used to improve the effective time of both of these methods. In the case of RF, the trees can be trained in parallel, which is not done here. For DL methods, the use of GPUs allows for massive speed-ups, as operations such as linear algebra and convolutions are highly optimized.

Table 10.6: Computation time per CPU of Random Forest, FG-Unet, and HACCS. These methods can be run in a parallel processing scheme to decrease the total computation time. It appears that the FG-Unet method is far less efficient than the Random Forest and HACCS.

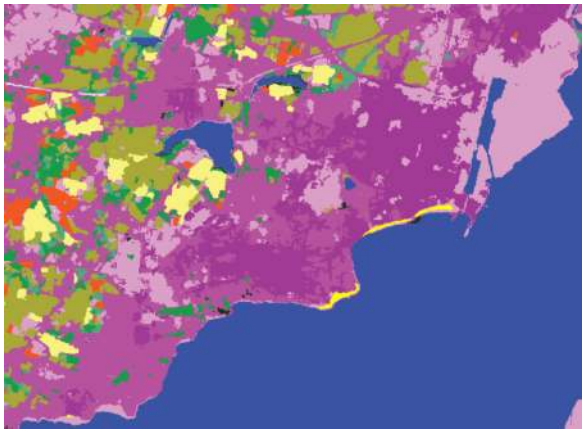
Method	Training time/CPU
RF	$\approx$ 25h
HACCS (3 iterations)	$\approx$ 80h
FG-Unet	$\approx$ 3300h

(a) Image.

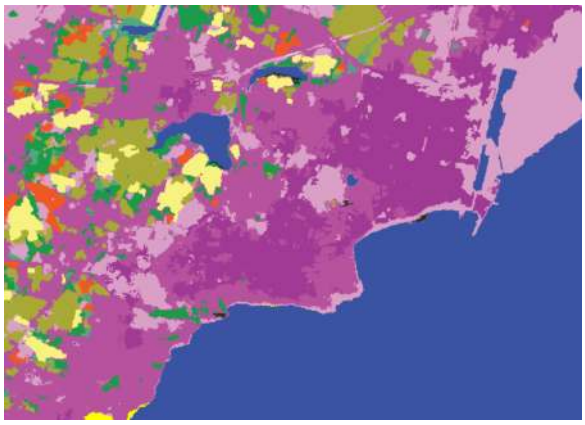


(b) Pixel-based result.

(c) FG-Unet.



(d) HACCS in superpixels of scale 10 30 and 50.



(e) HACCS in adjacency layers 1, 2, and 3, at a base scale of 10.

Figure 10.13: Classification results of the three methods over the urban area of Saint-Nazaire, (see figure 10.3), in T30TWT. The pixel-based result contains a strong degree of intra-object classification noise as well as a poor characterization of the different levels of urban density. The FG-Unet result has a stronger discrimination power for urban classes, but the geometry of the result is questionable at places, for instance in the harbor area. The Histogram of Auto-Context Classes in Superpixels (HACCS) result offers a result with similar class accuracy, but with more precisely outlined objects. The geometry of the adjacency layer result seems less precise than the multi-scale superpixels choice.

11

Mono-date VHSR experiments

"If intelligence is a cake, the bulk of the cake is unsupervised learning, the icing on the cake is supervised learning, and the cherry on the cake is reinforcement learning."

– Yann LeCun

This data set is based on Very High Spatial Resolution SPOT-7 optical imagery, which measures surface reflectance in the visual RGB bands, as well as the Near Infrared band (NIR), at a spatial resolution of 6m. This multi-spectral image is combined with a Panchromatic (brightness) image at 1.50m spatial resolution, which produces a *pan-sharpened* RGB-NIR image with a spatial resolution of 1.50m.

Figure 2.6, which can be found in Part 1, page 45 shows an illustration of the full extent of the data set, which covers an area of $16.5km \times 16.5km$ (11000x11000 pixels). This area contains the city of Brest, which is used as the area of interest for the visual analysis of the results.

The data set contains 5 classes: Urban cover, Roads, Water, Vegetation, and Crops, which come from the National Topo Data Base (BD Topo) and the Land Parcel Information Registry, (Registre Parcellaire Graphique, RPG) which are described in Part I, in Section 3.1.1. This data set is identical to the one proposed in the experimental section of [Postadjian et al., 2017]. The challenging classes in this problem are the buildings (Urban Cover), the roads, and the water, due to common confusions between water and shadow, which both appear dark, and between the tones of gray that make up certain roofs and streets. In this situation, contextual information is key, as the shape, size, and texture of the objects is more relevant than the color of each individual pixel. In essence, this problem is not very different from the ones encountered in Computer Vision, as the number of features per pixel is low, and the spatial resolution is far higher than the size of the objects in the image. In fact, the CNN model was originally developed for this type of applications.

An illustration of different scales of superpixel segmentation that are calculated from the SPOT-7 image is given in Chapter 4, Figure 4.5.

11.1 Experimental setup

The aim of this set of experiments is not to demonstrate a method perfectly adapted to VHSR data. Rather, it is to evaluate the validity of the HACCS method in a context different than the one in which it was designed. This is a way to study how applicable the HACCS method is to different types of data with unique temporal, spectral, and spatial characteristics. This explains why this set of experiments is not as extensive as the ones performed on the Sentinel-2 data set, that were presented in Chapter 10. For instance, the object segments, that provided the weakest results, are not evaluated here. Moreover, many of the possible parameter choices, in particular the combinations of scales and spatial supports, are not all shown here; in all of the experiments, the class histogram features are calculated in superpixels at scales 5, 10, 20, 30, 40 and 50. This choice is justified in the complimentary results presented in Appendix C, on page 193, which show in more detail the impact of the choice of scale on the classification result for the different methods.

This experimental setup compares five different initial classification results, and evaluates the application of HACCS on them. Indeed, the histograms of local classes can be calculated based on any previous dense classification of the image, which makes HACCS applicable to any available result. The five methods are listed here.

1. The pixel-based classification. Applying HACCS here is the most basic scenario, and is essentially identical to what is done on the Sentinel-2 data set.

2. The Local Statistic features, which in this scenario include the sample mean, variance, and edge density, defined in Part II, Chapter 5.

3. The Extended Morphological Profiles (EMP), also defined in Chapter 5. These describe context using several morphological operations on the image, with a structuring element of increasingly large size. Here, 5 structuring element sizes, ranging from 7 pixels to 15 pixels are used to calculate EMPs of the Brightness and Normalized Differential Vegetation Index (NDVI) over the area.

4. A model-based approach, the Basic Semantic Texton Forest (B-STF), presented in Part III, Chapter 8. This method involves training a RF classifier using the features from all of the pixels in a square neighborhood (sliding window). In the results that are shown, a 17×17 window was used.

5. The patch-based D-CNN designed by [Postadjian et al., 2017].

The objectives of this set of experiments can be summarized as follows. First of all, the idea is evaluate the performance of HACCS in a context that is different from the one it was originally designed for, namely, in a low-dimensional setting. Second of all, the aim is to compare this performance to two existing model-based methods from literature, the Basic Semantic Texton Forest (B-STF) and the Convolutional Neural Network. Finally, we wish to see to what extent applying HACCS to the result of a contextual classification can improve the quality of the land cover map.

For training and validation of the supervised classification method, exactly 2000 training samples are randomly selected for each class. Like in the experiments in Chapter 10, the Random Forest classifier is applied with a total of 100 trees.

11.2 Results

The experimental results are represented under three forms.

First of all, Figure 11.1 compares the semantic and geometrical accuracy of the various methods by plotting the OA against the PBCM. Next, Table 11.1 provides the numerical values of the classification accuracy scores: Overall Accuracy, Kappa, as well as the per-class F-scores of the different methods. Finally, Figure 11.2 shows the classification maps of the central area of the city of Brest.

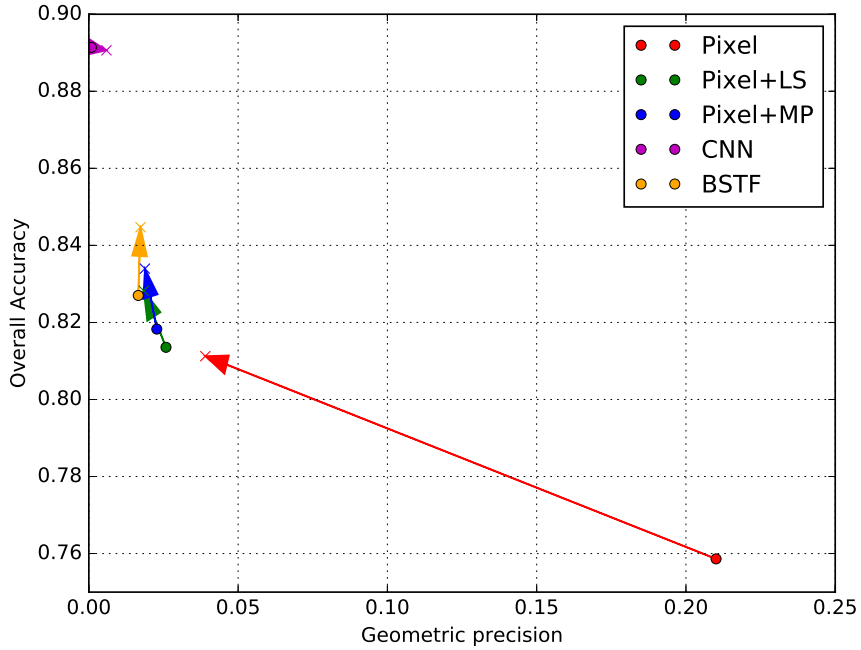


Figure 11.1: Geometric precision and Overall Accuracy of the different methods compared in Table 11.1. The solid points show the *iteration 0*, in other words, the scores of the method used to generate the classification map for the first class histograms. The arrows show the evolution after 4 iterations of HACCS, with a cross indicating the convergence point.

Figure 11.1 shows how the geometric precision, as measured by the PBCM, evolves with the overall accuracy when applying the HACCS process. The solid point at the root of the arrow shows the scores of the initial classification, the *iteration 0* of the HACCS process, which was used for the first estimation of the histograms. The head of the arrow is positioned according to the scores after four iterations of the HACCS process, with a cross at the convergence point. For the pixel, local statistics, and Morphological Profiles and B-STF, applying the HACCS iterative process improves the overall accuracy, while diminishing or maintaining the geometric precision of the result. On the other hand, applying iterations of HACCS on the result from the patch-based CNN slightly decreases the overall accuracy, but restores some of the sharp corners, which are measured by the PBCM.

Table 11.1: Class accuracy (OA, κ, and F-scores) and geometric accuracy (PBCM) of the various methods on the SPOT-7 data set, expressed in percent units, along with the 1σ error intervals. For the HACCS results, superpixel scales of 5, 10, 20, 30, 40 and 50 are used, with 4 iterations. In this table, P stands for Pixel, LS stands for Local Statistics (mean, variance, and edge density), MP for Morphological Profiles, and B-STF for Basic Semantic Texton Forest.

Method name	OA (%)	κ (%)	Urban	Crop	Water	Roads	Veg.	PBCM (%)
P	75.9%±0.16	69.8%±0.20	0.666	0.836	0.865	0.675	0.758	21.0%±1.49
P+HACCS	81.1%±0.17	76.4%±0.21	0.750	0.880	0.915	0.694	0.819	**3.9%** ±0.55
P+LS	81.4%±0.10	76.7%±0.13	0.762	0.889	0.898	0.699	0.826	2.6%±0.47
P+LS+HACCS	82.8%±0.10	78.5%±0.13	0.770	0.901	0.918	0.707	0.846	1.8%±0.22
P+MP	81.8%±0.14	77.3%±0.18	0.771	0.887	0.904	0.709	0.828	2.3%±0.52
P+MP+HACCS	83.4%±0.11	79.2%±0.13	0.782	0.900	0.918	0.722	0.850	1.9%±0.18
P+B-STF	82.7%±0.12	78.4%±0.13	0.809	0.869	0.903	0.753	0.807	1.7%±0.15
P+B-STF+HACCS	84.5%±0.15	80.6%±0.17	0.811	0.896	0.927	0.754	0.838	1.7%±0.12
CNN	**89.1%** ±0.06	**86.4%** ±0.08	**0.862**	0.932	**0.961**	**0.813**	0.886	0.1%±0.04
CNN+HACCS	**89.1%** ±0.09	**86.3%** ±0.12	0.857	**0.936**	0.959	0.808	**0.891**	0.6%±0.08

Table 11.1 shows that including contextual information improves the F-score of all of the classes compared to the pixel-based classification, for all of the methods that are evaluated here. Secondly, it can be noted that the HACCS iterations never make the F-score values decrease, except for the CNN result, where a slight decrease is observed for 3 of the 5 classes.

The best performing method overall is the CNN, which is relatively equivalent in terms of overall performance scores to the CNN+HACCS, although the latter has a slightly higher PBCM. In second place, the Basic Semantic Texton Forest, particularly combined with HACCS, provides a remarkable improvement over the pixel-based classification ($\approx$ +10% in OA), reaching approximately two thirds of the improvement achieved by the CNN ($\approx$ +15% in OA). This improvement is consistent on the 5 classes; the P+B-STF+HACCS method provides F-scores that are between 0.03 and 0.06 lower than the CNN method.

It should be noted that the low values of PBCM are due to the noise present in the pixel-based classification which makes corners more difficult to detect, regardless of the calibration parameters of the line detector. This effect is due to the application of this metric to very high spatial resolution imagery, in which the pixel-based classification is more subject to such errors than in high resolution imagery. For this reason, a visual analysis of the result is presented in Figure 11.2.

Figure 11.2a shows the result of a pixel-based Random Forest on the central urban area, which illustrates the importance of contextual information in this problem. Indeed, with only the pixel information available, there are several confusions between the contextual classes mentioned earlier: Urban cover, Roads, and Water.

In Figure 11.2b, the multi-scale class histogram features are directly calculated from the pixel based classification, as is done on the high-dimensional Sentinel-2 time series. The HACCS process allows for a better discrimination between the context dependent classes mentioned above. Indeed, the presence of water and vegetation classes in the urban area is somewhat reduced when compared to the pixel-based classification. However, this result is not entirely satisfactory, as many confusions remain, especially between roads and buildings in the city center.

The classification result in figure 11.2c is generated using standard local statistics: sample mean, variance, and edge density [Trias Sanz, 2006]. These image based contextual features are calculated in the same spatial supports as the class histograms. While the use of local statistics allows for the confusions between roads and water to be greatly diminished, there still remain confusions between roads and urban cover. This is certainly progress, but it would be desirable to observe the details of the network of roads and streets in the classification result. The impact of applying the HACCS process to this result is shown in Figure 11.2d. In this case, the HACCS process provides a smoother result, although many confusions remain between roads, buildings, and water.

Figure 11.2e shows the classified urban area, based on both the pixel and the EMP features. Figure 11.2f shows the obtained classification map when using the histogram of the classes from the pixel and EMP feature result. In this image, some of the streets are recognized, and the fine geometrical elements, such as the harbor and the bridges are correctly restored. However, the streets that are recognized are disconnected and spread out through the urban area. This would not be sufficient for determining the precise network of streets.

The following experimentation uses a Basic Semantic Texton Forest, which randomly selects a pixel value, or

the mean of the spectral bands of a pixel, in a neighborhood of 21x21 pixels. Before the application of HACCS, figure 11.2g shows that many of the streets are well recognized and connected, although they are slightly blurry and retain confusions with water. Moreover, fine details such as the harbor area are well conserved. Figure 11.2h shows the result after the HACCS iterations, which has significantly fewer confusions between streets and water. However, large areas of shadows remain confused with water, which indicates that the size of the 21x21 neighborhood for the B-STF may be insufficient. Larger neighborhood sizes were not evaluated due to the elevated number of features to be considered by the RF that this would imply. Moreover, the harbor area and fine details such as the bridges are blurred out.

Finally, HACCS was also applied to the result of the CNN. This is done to evaluate whether or not HACCS can improve the quality of a classification that is already very well classified. It appears that the result after HACCS does not restore much of the high spatial frequency geometry, such as the fine details of the harbor. On the other hand, it has a smoothing effect everywhere in the image, which removes isolated pixels and other kinds of label pixel noise. Moreover, the borders between objects have a relatively clear definition, with some of the corners being restored. This is illustrated in Figure 11.3, through two zooms on the classification results in areas marked in Figure 2.6 which can be found on page 45. The left column shows the results in a discontinuous urban setting, which contains a mix of vegetation, roads, and small residential buildings.

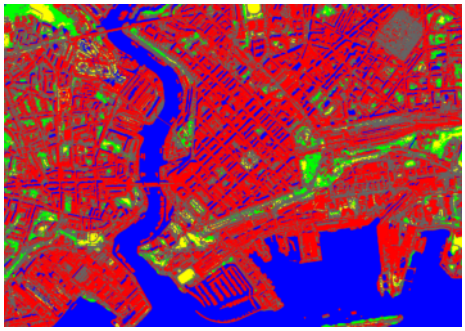
11.3 Conclusions

Four groups of context-based approaches are evaluated on this problem: local statistics features, which contain sample mean, variance, and edge density, Extended Morphological Profiles, the Basic Semantic Texton Forest that was presented in Part III, and the patch-based D-CNN.

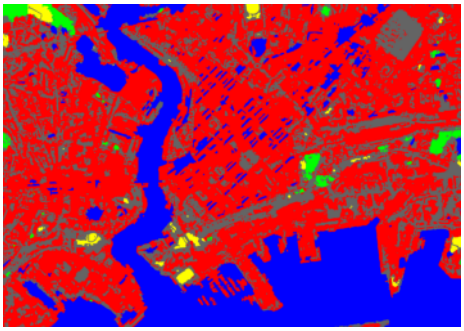
When starting with the pixel-based classification, using several iterations of HACCS in multiple superpixel scales allows for a consistent improvement of the precision of each class. However, the result lacks in fine definition in the central urban area, where roads and buildings are heavily confused. When the histograms are based on a classification result that is already generated with contextual features, the thematic quality of the classification (class accuracy) is improved, however, some of the corners present in the pixel-based classification are displaced or lost.

When comparing these results to the classification map generated by the patch-based CNN from [Postadjian et al., 2017], it appears that the handcrafted features (local statistics, EMP, B-STF, and their results after 4 iterations of HACCS), provide lower values of overall accuracy on this problem. Indeed, very high spatial resolution problems with a low number of image features are similar to the Computer Vision problems for which CNNs were originally designed. On the other hand, the handcrafted features generally show higher degree of geometric precision, which is measured to a certain extent by the PBCM, and is especially visible in the classification results. In other words, they contain well localized sharp corners and borders that are not polluted with classification noise, but that can contain the irregularities of the superpixel segmentation.

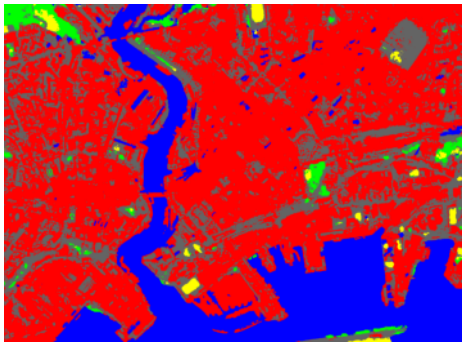
Overall, the results presented in Section 11.2 show that the HACCS process does generate a pertinent contextual description, even for low-dimensional VHSR images. Alone, HACCS provides similar results to other contextual features like local statistics and Extended Morphological Profiles. When combining the HACCS process with the standard contextual features, the classification accuracy after a few iterations is improved. However, on this type of mono-date high spatial resolution imagery, the patch-based CNN does provide an overall more thematically accurate classification result, albeit with a low geometrical precision score.



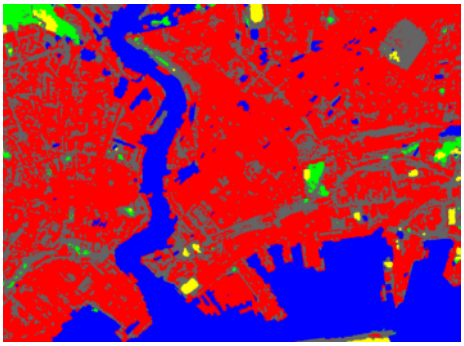
(a) Pixel-based classification.



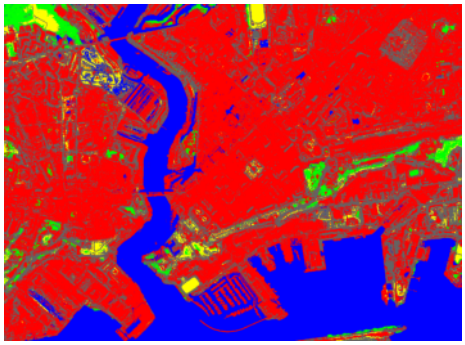
(b) Pixel + HACCS



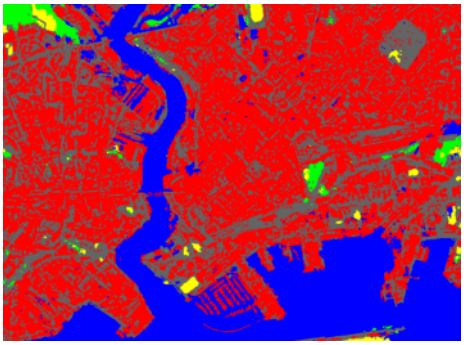
(c) Pixel + LS



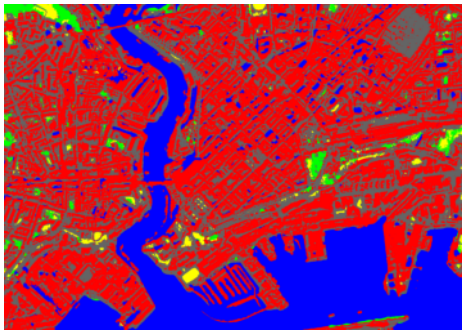
(d) Pixel + LS + HACCS



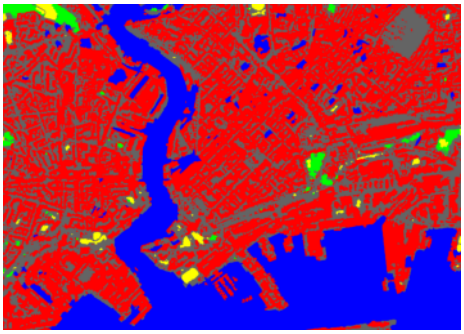
(e) Pixel + MP



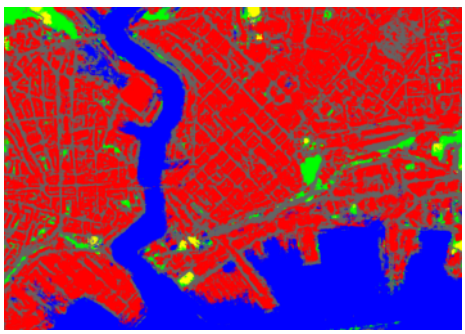
(f) Pixel + MP + HACCS



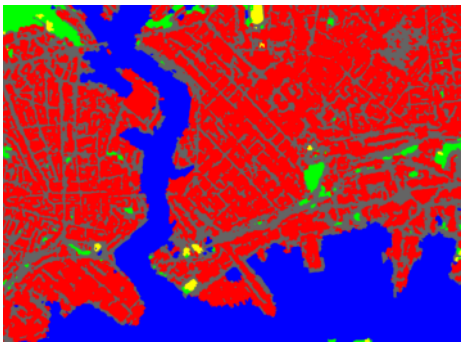
(g) Pixel + B-STF



(h) Pixel + B-STF + HACCS



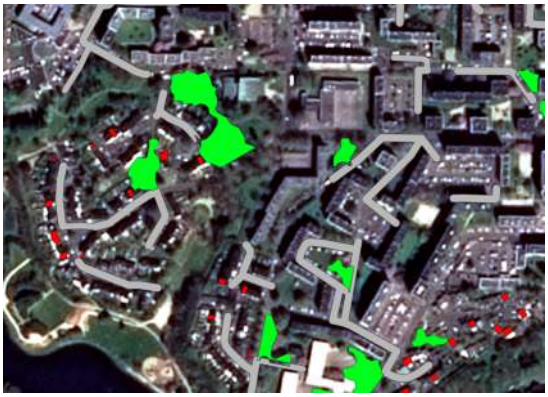
(i) Patch-based CNN



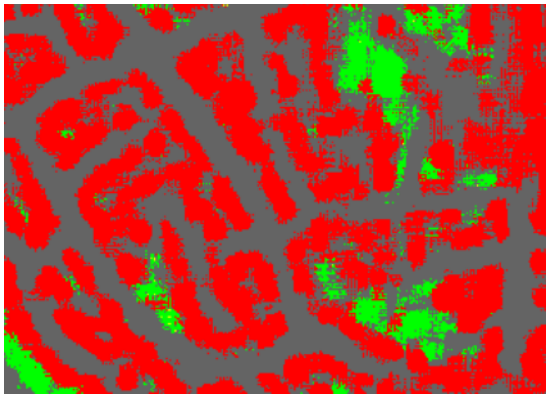
(j) Patch-based CNN + HACCS

Figure 11.2: Classification results over the city of Brest (France). The left column shows the classification result of various methods before the application of the HACCS process, and the right column shows the results with the inclusion of class histograms, after four iterations.

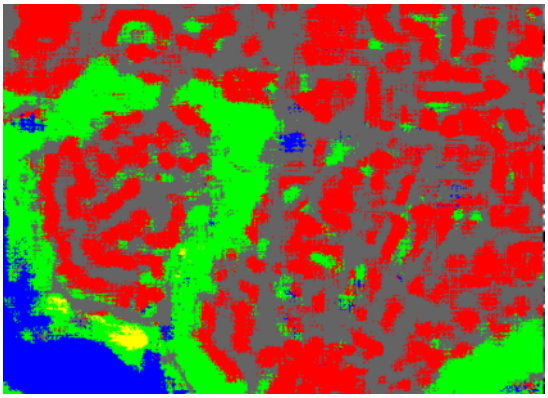
(a) Discontinuous urban setting.



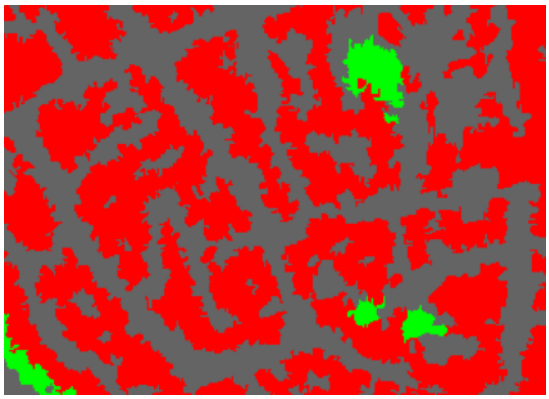
(b) Mix of tall residential buildings and vegetation.



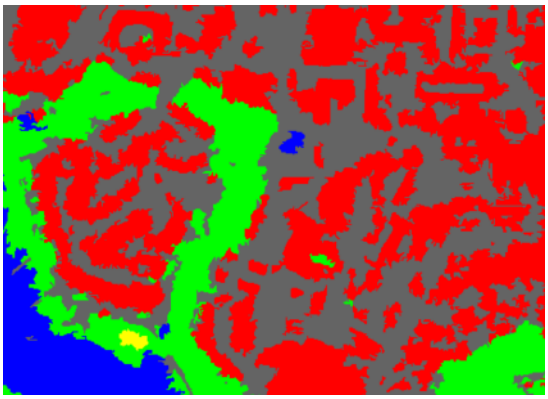
(c) Patch-based CNN.



(d) Patch-based CNN.



(e) CNN + HACCS



(f) CNN + HACCS

Figure 11.3: Zoom on the results of the patch-based CNN, before and after application of HACCS. Before, the maps contain isolated pixels of vegetation and rounded corners, and the outline of the buildings is not well captured. After 4 iterations of HACCS, the pixel classification noise seems reduced, which allows for a more precise outlining of the roads and buildings. However, HACCS does not restore missing elements of the geometry, or necessarily provide smooth linear borders.

Part V

Conclusion

12 Conclusion

"The most exciting phrase to hear in science, the one that heralds new discoveries, is not 'Eureka!', but 'That's funny ...' "

– Isaac Asimov

Land cover mapping of large areas is a complex task, due to the multitude of classes that make up our environment. Nonetheless, using time series of satellite images, combined with a regular collection of reference data in a sparse form, supervised classifiers can be trained to recognize and categorize many elements of our surroundings. This has been done with varying precision in the past, as some classes show lower recognition rates than others [Inglada et al., 2017]. The basis of this work is to understand how to improve the way in which these classifier automatically recognize certain land cover elements.

These sections present the conclusions of the efforts made in this direction, which can be summarized in three main points. Any research work must begin with an understanding of the problem at hand. Section 12.1 of this chapter presents the first conclusions regarding the importance of including context in high-resolution image classification. These also set in place the constraints which define the scope of methods that can be evaluated, and the requirements of these methods. The subsequent part of this work deals with the identification of potential solutions for including context. This is done while taking into account the theoretical limits of each method, and providing ways to overcome certain of them. Section 12.2 recalls the justification behind the selection of methods retained for the experiments. Finally, the conclusions of the experiments on both the Sentinel-2 and SPOT-7 data sets is shown in Section 12.3. This is done by analyzing which conclusions are common to both data sets, and therefore more general, and which are specific to each data set.

12.1 The importance of contextual information

When adressing a classification problem, it is advisable to think about the different dimensions of information that are potentially available in the data set. In the case of the classification of time series of multi-spectral images, the pixel features alone contain the spectral and temporal dimensions. The third potential dimension of information that is investigated in this work is often called the *spatial* dimension. This dimension is based on the arrangement of the pixels that form the image itself, and can be very important in certain image classification problems.

Chapter 1 presented the OSO map, which has been produced on a yearly basis since 2017 using time series of Sentinel-2 images. These maps present high confusion rates in urban areas as well as between classes such as orchards, vineyards, and to a lesser extent, crops. In Section 1.4.2, an explanation for these confusions is proposed.

Simply put, the classes recognizable at the pixel scale are not identical or strictly included in the classes present in the target nomenclature. This discrepancy is caused both by the spatial resolution of the image, and the choice of target land cover classes. This translates as three conditions which may worsen the impact of confusions linked to a lack of contextual description.

1. The spatial resolution of the image is fine enough that the pixels are far smaller than the objects in the nomenclature.
2. The classes described in the nomenclature are in fact made up of heterogeneous parts.
3. The classes have inter-object contextual dependencies, for instance, certain target classes can only appear near others, or in certain arrangements.

These three conditions are met in the case of the OSO land cover mapping problem.

First of all, the 10m×10m pixel size used in Sentinel-2 images is indeed far smaller than the size of most of the target objects described in the reference data, which are more around the size of 500×500m (the MMU of CLC). Second of all, the nomenclature contains classes such as Continuous and Discontinuous Urban Fabric (CUF & DUF), which are made up of buildings, roads, and a certain degree of vegetation, all of which have different spectral and temporal properties. Thirdly, other classes that are difficult to classify include beaches and bare rocks, which exhibit inter-object dependencies. For instance, a beach must be situated near a body of water, and bare rocks and mineral surfaces are more likely to be found in mountainous areas.

If a precise classification of the target context-dependent classes is to be reached, the spectral, temporal, and spatial information must all enter equally into the decision process.

This is a justification for the need to further investigate methods that can take into account the spatial dimension of images, without forgetting about the constraints of land cover mapping at high/very high spatial resolutions, over wide areas. The choice of which method or methods to use depends on a great number of factors, the most important of which are listed here.

1. The type of imagery used, i.e. the pixel features and their spatial, spectral, and temporal resolutions.
2. The reference data, in particular the class nomenclature, and the contextual dependencies it may contain. Another interesting aspect is the relative density of available training points or polygons.
3. The computational power available, as some contextual classification methods can be very costly.

12.2 Different ways of including context

The inclusion of context first requires a precise definition of which pixels are a part of the context of a given pixel p. Three intuitive definitions of a context are provided here.

1. Pixels within a certain radius of p.
2. Pixels in a connected group of pixels similar to p.
3. Pixels in a connected group of pixels both similar and nearby to p.

Mixed feature-spatial relations such as "similar and nearby" or "similar and connected" can be expressed through image segmentation methods. These algorithms divide an image into connected groups of pixels, based on certain criteria regarding their relative position and their features. This is done without associating any form of class label to these areas. Segmentation can be seen as a kind of unsupervised classification of the image. Part II Chapter 4 provides more specific definitions for the three ideas mentioned above, which are listed here.

1. Sliding window, a square group of pixels surrounding the target pixel. This type of spatial support poses no theoretical issue for application on large images.
2. Object segments are adaptive to the nearby content, and only group similar pixels together. Their application to large images has been studied by [Lassalle et al., 2015].
3. Superpixels are segments designed to be both spatially and spectrally homogeneous, in other words, with limited intra-segment variations, and a compact shape.

This chapter also proposes a new shape of context that is based on a segmentation of the image. Adjacency layers are introduced in Chapter 4. The first adjacency layer is made up of the segments surrounding the segment containing the pixel, the second layer is made up of the segments surrounding the first layer, and so on. In order to preserve rotation invariance, and the comparability of scales of information, these are evaluated based on a superpixel segmentation, and therefore can be applied to high-dimensional imagery as well.

The issue of extracting spatial supports in an efficient way on high-dimensional imagery is detailed in Part III, Chapter 7. In particular, the methodology for scaling the SLIC superpixel algorithm is an original contribution, and is presented in Section 7.2.

Once a definition of the context has been selected, the next step involves deciding how to best interpret the large amount of information present in these neighboring pixels. Let us define $N(p) = Card(C(p))$ as the number of pixels in the context of a pixel p, and $F(p)$ the number of features that describes each pixel. The total number of features, in other words the quantity of raw information in the context of each pixel is N(p)F(p). In the case of high-dimensional features, the number F(p) is large. Moreover, if we consider a relevant context at a given spatial

area, say, $1km^2$, the number $N(p)$ increases quadratically as the spatial resolution becomes finer. It is important to keep in mind that these two numbers, $F(p)$ and $N(p)$, are bound to increase for classification problems in the future. The evolution of the latest satellites continues to be towards higher spatial resolution, with a ever greater number of features.

Let us consider the idea of providing this raw contextual information directly to the supervised classifier. This is defined in Chapter 5 as a model-based approach.

The first implication is that for this to work, the dimension of the features, $N(p)F(p)$ must be constant. In other words, $C(p)$ must always contain the same number of pixels. Moreover, these pixels must describe the same aspect of the target pixel for the supervised classification to work. This tends towards the selection of a sliding window.

An extremely high dimensional feature space can have a negative impact on supervised classifiers, as the number of training samples required to accurately model the class-conditional probability density functions increases with the dimension of the features [Friedman, 1997]. Moreover, the total number of features is conditioned in practice by the memory available on the computer that runs the training.

For these reasons, many studies have attempted to define sets of contextual features, These are functions that take as an input a group of pixels known as the *spatial support*, and describe certain aspects of these pixels. The fundamental idea is to compress this high dimension $N(p)F(p)$ into a more manageable dimension. A very simple example of such a feature is the sample mean, which takes a set of pixels and computes the average pixel. For an input area with a total dimension $N(p)F(p)$, the dimension of the mean feature is of the form $KF(p)$ with $K = 1$. As the dimension no longer depends on $N(p)$, such contextual features are viable for high spatial resolution imagery.

Different possible choices for contextual features are studied both in Parts II, and in Part III.

Chapter 5 presents image-based contextual features, which directly use the values of the neighboring pixels. The main issue with these methods resides in representing the high-dimensionality of the pixel features in combination with a multi-scale description. For instance, the sample mean and variance, and edge density all describe a spatial support in the same number of features as the pixels it contains ($K = 1$). These features can therefore be retained for experiments on the Sentinel-2 data set, which has 489 pixel features, but will be limited to one scale of features. In many cases, image-based features describe a spatial support in a higher dimension than the pixels. For example, the Morphological Profiles [Dalla Mura et al., 2010] require multiple features per pixel: for each spatial support, a dimension of $KF(p)$ with K in the order of 10-100. This is the reason why these methods are retained for the experiments on the SPOT7 images only, which have 4 pixel features.

Seeing as the number of initial pixel features is so high in time series, one idea is to consider the pixel-based and spatial information in two separate stages. For instance, to perform some kind of dimension reduction on the neighboring pixels in order to compute features in lower dimensional spaces. In Chapter 8, different methods to reduce the large dimension of the images are considered. Among methods previously employed on similar issues, one group of methods stands out. Stacked contextual classification methods, which integrate spatial information using a form of prediction of the pixels in the neighborhood. The class membership functions, also known as the soft class labels, can be used as an input to a probabilist model, as is done in Conditional Random Fields [Moser and Serpico, 2013]. Another idea is to compute simple statistics on these membership functions, which is the principle behind Auto-Context [Tu, 2008]. These methods use a *semantic* context which is calculated based on the predicted labels of the neighboring pixels, rather than the features of the pixels themselves.

This is the basis of a new process for integrating contextual information into high-dimensional image classification problems. The idea is to use the pixel-based prediction of the neighboring pixels to calculate a histogram, which should represent the proportions of the classes in the neighborhood. Then, following an iterative process, this can be repeated any number of times. One of the questions that arises is which spatial support or spatial supports to choose in combination with such a feature. Seeing as superpixels offer an interesting compromise between objects and sliding windows, they are considered a candidate alongside the other two.

12.3 Overview of the experimental results

The choice of the superpixel as spatial support in the HACCS method is one of the main issues of discussion in the experimental sections, which make up Part IV. These evaluations are divided into two data sets, one based on multi-spectral Sentinel-2 time series at a 10m spatial resolution, and the other using a mono-date 1.5m, SPOT7 image with a lower spectral and temporal resolution.

To evaluate the performance of a land cover map, two groups of metrics are employed. As in any classification problem, metrics designed to evaluate the accuracy of the prediction, OA, κ, and F-score, are used. However, these do not take into account the spatial arrangement of the pixels in the result, which is specific to applications on

images.

Chapter 6, Section 6.3 presents the Pixel Based Corner Match (PBCM), a new metric for measuring the geometric precision of a context-based classification map, in the absence of dense testing data. One conclusion of the initial classification experiments is that the visual aspect of the result can be strongly altered by some of the contextual features. The PBCM metric is therefore used to demonstrate the ability of the different combinations of spatial support and contextual feature to improve class recognition while maintaining a precise geometry.

This metric uses the output of a pixel-based classification map to simulate dense testing data, under the assumption that the corners formed by the edges between different classes are relatively well respected by the pixel-based classifier. By matching these corners with the corners in the contextual classification map, the degradation of these high spatial frequency elements can be quantified. Experiments using regularization (majority vote in a sliding window) show that this metric provides a quantitative indication of the amount of smoothing and loss of corners that occurs when using such a post-processing step. Indeed, when increasing the size of the regularization window, the metric decreases significantly. However, it is important to keep in mind that it is far from perfect, as it only measures the degradation of corners, and not of fine elements. Secondly, it is strongly biased by the classes that contain the most corners, in the Sentinel-2 case, the summer and winter crops. This metric is therefore meant to be used in a multi-criterion evaluation of various contextual classification methods.

There are common conclusions that can be drawn on both of these data sets, which address the issue of land cover mapping at a relatively high spatial resolution (1-10m). The differences between the data sets also cause diverging conclusions and questions which are discussed later in this section.

First of all, it appears that maintaining the pixel features is key both for geometry and for class accuracy. It cannot be replaced by purely object information on this problem. Doing so results in a strong deterioration of the geometric precision or of the overall accuracy, regardless of the choice in contextual features and spatial support.

The unstructured local statistics features (mean and variance) have a generally lower performance than the structured edge density. Indeed, the combination of pixel features with the edge density, calculated in a sliding window provides a stronger OA than with mean and variance, on both problems. On the other hand, superpixels provide results with a high geometric accuracy.

On the subject of semantic features, it is repeatedly observed that applying HACCS on a given classification improves the class accuracy during iterations. The only exception to this is cases where the classes are very poorly recognized in the first place. This is often linked to a lack of training data in the area, which might therefore be improved when applied on larger scales.

However, it is also noted that in some cases, HACCS decreases the geometric accuracy of the previous classification. This particularly occurs when using adjacency layers as spatial supports. This translates as a classification of areas of vegetation near urban areas as discontinuous urban

Moreover, the use of the same feature, the histogram of local classes, in sliding windows decreases the geometric accuracy of context-dependent classes. While this is not particularly measured in the PBCM, it is observed in the classification maps.

Finally, in these two very different land cover classification experiments, the use of Convolutional Neural Networks seems to provide lower degrees of geometric accuracy than superpixel based methods with handcrafted features. This may be due to the sparsity of the training data, which is insufficient to describe the geometry of the objects during training. In other words, the neural network is not discouraged from generating results with smooth corners, as these corners are absent from the training data, and therefore from the loss function. On the other hand, superpixel-based methods extract the geometry of the objects directly from the image, which preserves the geometry. Secondly, in high-dimensional feature spaces, such as the time series used in the Sentinel-2 experiment, it appears the HACCS process provides results with equivalent levels of class accuracy as the FG-Unet architecture [Stoian et al., 2019]. Moreover, the HACCS results systematically present a higher rate of geometric accuracy, with a finer localization of sharp corners. It also is worth mentioning that the HACCS process is computationally lighter than neural networks, as they only involve the training of a Random Forest for each iteration, which is very fast in practice.

Next, the experiments also underline differences between the two data sets, and how the properties of the data (spatial, temporal, and spectral resolutions) has an impact not only on the scope of possible methods, but also on the relative performance of the different methods.

Methods that are not directly applicable to high-dimensional imagery, such as Extended Morphological Profiles (E-MP) and Basic Semantic Texton Forests (B-STF) are therefore only applied to the SPOT-7 data set. In this context, the application of HACCS required careful parametrization, as several of the results during initial testing phases showed insatisfactory levels of accuracy. In particular, the use of local statistics and edge density features, even in superpixels, showed inferior performance to methods using a sliding window as spatial support, such as the B-STF.

Another important difference is the fact that adjacency layers do not provide stronger results than superpixels

on the SPOT-7 problem, whereas on the Sentinel-2 these spatial supports show the most significant improvements. Indeed, the effect of geometric degradation that was observed on the Sentinel-2 problem is amplified on this problem.

This is in part due to the balance between the number of pixel and contextual features. In all supervised classifiers, there is an initial assumption that all features may carry some degree of information regarding the problem at hand. For instance, the Random Forest model performs node splits using sets of features selected randomly from all of the features. For this reason, an aspect of the classification problem that is only represented in a low number of features, can be overriden by another group of more numerous features. In this case, the pixel features are in an inferior number, yet they contain the finest details of the geometry. Achieving a classification with both a fine geometry and a high accuracy therefore requires a balance between the pixel and contextual features.

However, this is not the only reason behind the geometric degradation linked to the use of adjacency layers. In fact, there is a fundamental difference between a multi-scale description using superpixels and adjacency layers, which is that superpixels are adaptive at all scales. On the other hand, while the base superpixels of an adjacency layer are adaptive, the successive adjacency layers capture pixels from neighboring objects. While this is the way in which they are able to capture long-range information regarding their surroundings, this form of inadaptivity can lead a contextual feature to smooth out the geometric elements of a map, in particular, to displace class edges.

In short, when a contextual feature is used in an unadaptive neighborhood (sliding window, adjacency layer), the risk of geometric deterioration increases. This is one of the factors that pushes towards the choice of multi-scale superpixels as spatial supports in the HACCS process.

The final difference between the sets of experiments is that HACCS with Multi-Scale Superpixels provides a lower Overall Accuracy (OA) than the Deep Learning approach on SPOT-7 data, whereas on the Sentinel-2 data set, the two methods have comparable class accuracy. Indeed, when using images with characteristics similar to the ones encountered in Computer Vision problems, i.e. a very high spatial resolution, and a low number of features per pixel, the CNNs show the strongest levels of class accuracy. On the other hand, the geometry of the result is questionable, with the presence of speckle-like classification noise near object boundaries, and smooth corners. Handcrafted features in superpixels, like HACCS, allow for higher levels of geometric precision on this problem.

These experimental results confirm several of the intuitive ideas suggested in the early chapters on the subject of context-based classification on high-dimensional images. The following list groups up the main ones.

1. Contextual features should be used in conjunction with pixel information.
2. Including several scales of context is generally preferable to only one scale.
3. The use of image segments as spatial supports provides results with a higher degree of geometric detail than sliding windows, especially in areas that contain context-dependent classes.
4. Structured image-based features (edge density) provide results with a higher level of geometric detail than unstructured ones (mean, variance).

With these first general conclusions in mind, there are also conclusions regarding notions that would have not been readily forseeable.

1. Semantic features, in particular, the histogram of classes in a local area provide low-dimensional contextual information that is valuable for many, if not all of the target classes.
2. Using several iterations of the HACCS process is beneficial for the classification accuracy score, and generally decreases the geometric precision, as contextual information is integrated. The method converges after 3-4 iterations.
3. When using both pixel-based and contextual features, the balance between the number of features of each source can be an important factor when using a Random Forest model. If the quantity of contextual features heavily outweighs the quantity of pixel features, there is a risk of geometric degradation in the map.
4. On the Sentinel-2 problem, adjacency layers provide the highest levels of overall accuracy, with a result that is visually and geometrically acceptable in many areas. However, these spatial supports have a tendency to overestimate certain context-dependent areas, such as urban areas.
5. Model-based methods, in particular, the B-STF (Section 8.3), the patch-based network of [Postadjian et al., 2017] and FG-UNet [Stoian et al., 2019], provide results with a lower geometric accuracy than HACCS. On the Sentinel-2 problem HACCS and the Deep Learning method provide results with a similar class accuracy.

The results also raise several questions regarding the nature of how contextual information and pixel information can be used together to achieve a correct classification.

1. Superpixel segmentation, like all image segmentation, is far from perfect. The impact of the parameters and of the quality of the segmentation, so to speak, was never truly discussed in an absolute manner. How does the quality of the superpixel segmentation influence the classification result ?

2. The geometric quality is a notion that is more complex than initially expected. Could a small set of significant metrics entirely represent the variety of geometric aspects and relations between the different classes ?

3. The Sentinel-2 data set covers a variety of different areas, but each one is relatively small, compared to the entire country. Do any of the conclusions change on land cover mapping problems that cover larger areas ?

4. The impact of contextual features, and how to select them, is a subject of discussion all throughout this work. However, the idea of leaving the design of contextual features to an appropriate classification model is an interesting idea. Using what is already known, would there be a way to create a model-based approach that is geometrically aware, despite the training data being sparse ?

5. Many of the proposed methods, B-STF, HACCS, are not particularly specific to land cover mapping. To what extent could they be applied on different dense image classification problems altogether ?

6. Much of the effort that is made on the theoretical side aims at creating an intermediate representation between the pixels and the objects that contain them, in an effort to untangle the contextual dependences which result from an inadequate characterization of the classes at their lowest level of description. A hierarchy of classes can be present in many other classification problems, such as semantic trees in text recognition, for example. Could any of the the concepts studied here be extended to problems entirely outside of image classification ?

The following section presents thoughts and insights on how to answer these questions. The first part fixes short-term objectives that could be made to improve the methods proposed in this work, and further our understanding of the results they produce. Then, the extension of these methods to different types of problems, which could be done on a long term, is discussed.

12.4 Perspectives

The first short-term objective could be to further work on the SLIC algorithm itself. Indeed, one observation is that certain errors remain in the segmentation due to the nature of temporal interpolation on images. Efforts could be made in two distinct directions to improve the quality of this superpixel segmentation. First of all, in order to reduce the sensitivity of the segmentation to missed cloud detection, it would be possible to define a feature distance that is less sensitive to such strong gradients than the Euclidean distance. Alternatively, a distance could be defined on the space of acquired dates and images, regardless of the interpolation on pre-defined dates. This would require some theoretical analysis. Another interesting direction could be to incorporate semantic features into unsupervised segmentation, in the same way as they are incorporated into the supervised classification model. For instance, using a single pixel-based classification, the histogram of classes in the current superpixel could be calculated throughout the SLIC process, at each iteration, starting with a square grid. The histogram could be updated on the fly in the same way as the centroid is updated during the iterations. This would imply the use of a histogram-specific definition for the distance used in the SLIC algorithm, and an adjusted weight, to balance it out with the spectral and spatial distances. This would be a way to perform a supervised superpixel segmentation of an area using sparse training data. In fact, if such contextual cues can be integrated into a superpixel segmentation, they might also have a positive impact on other segmentation algorithms. A further development could be to update the classification result at each iteration of SLIC using one or more iterations of HACCS, in an effort to optimize both the superpixel segmentation and the classification in a conjoint manner.

Secondly, it would be interesting to improve the PBCM by making it a per-class metric. In the experiments, it is repeatedly observed that when combining pixel and contextual information together, the context-independent classes (crops) do not use the contextual information, because the pixel information is sufficient. Therefore they are not subject to the same degree of geometric degradation. The regularization case, on which the PBCM is calibrated, does not represent the actual geometric degradation in a case where both context-dependent and context-independent classes have very different degrees of geometric accuracy. For this, the corner detection step could be performed on each set of lines that was detected on a given class. Alternatively, it could be interesting to incorporate weights to the different corners, according to the classes that form them. For instance, the weights might be inversely proportional to the class frequencies, to balance the contribution of the different classes to the metric, in the case of very unbalanced class proportions.

Finally, these methods should be validated on larger regions for the conclusions to be confirmed. A validation over an entire eco-climatic area, and eventually, over the country could confirm or question several of the conclusions made here.

A few imaginable improvements could also be made on the subject of HACCS. For example, rather than using the hard class labels to compute a histogram, perhaps the class-conditional probabilities as estimated by the classifier could provide more valuable features. Another idea could be to integrate the local structure of each surrounding pixel in the feature, in a similar way to how the edge density or auto-correlation features describe structured texture. These could define an interesting notion, the one of semantic texture, in other words, the spatial arrangement of estimated class labels, and not of pixels themselves.

Next, on the topic of model-based approaches, one relatively simple method that was brought under consideration here is the STF. On the SPOT-7 data set, the B-STF provides higher OA than other contextual features such as local statistics, edge density, and E-MP. An interesting question remains: how to apply this process to higher dimension images, such as time series ? The current limit is mainly practical, as training in the extremely high dimensional space imposed by the use of the raw contextual pixel values is not imaginable, if large contexts are to be considered. There is a possiblility of integrating a large number of features into a Random Forest model, under the condition that the feature boosting process is set aside. Indeed, the model proposed by [Fröhlich et al., 2012] could be applied to large-dimensional images but it would require a significant amount of changes to the classification models themselves. However, this would allow for other temporal and spectral features to be incorporated in the same way as spatial features in the STF. For instance, temporal features could include the mean, max, and variance of the time series of each pixel. In terms of spectral features, it would be possible to imagine any combination of spectral bands under a non-linear form, like the NDVI and NDWI.

In further studies, it would be interesting to evaluate the effect of the density of training data on the geometry of the classification map. Indeed, many contextual models are designed for cases with dense reference data.

In fact, this raises another interesting point. Would it be possible to create a training data set with both labeled and unlabeled polygons, in which the labeled polygons contain the training area, and the unlabled polygons contain geometric cues regarding other pixels in the image ? This could lead to the design of a model capable of integrating information from such a mixed data set.

The issue of contextual dependencies is not specific to time series or to high/very spatial resolution imagery, it can also appear in problems using hyperspectral, radar, or other types of imagery. In particular, it would be interesting to study whether these results can be extended to hyperspectral images, which also present a very high number of image features, and have sometimes been used in land cover mapping. In hyperspectral imagery, most of the pixel features are valuable for recognizing the different classes, as each material that makes up the pixel has a unique spectral signature. This is in a way similar to how the different dates in a time series are useful to recognize the temporal evolution of certain plant species. However, if the spatial resolution is smaller than the size of the base components of the objects that make up the target nomenclature, the pixel features might be insufficient to characterize them. Many of the issues that are raised here are also present in hyperspectral image classification problems, which makes one wonder if the solutions evaluated on time series, can also be transposed to other types of images.

Conclusion en français

La classification de l'occupation des sols est une tâche complexe à cause de la variété de classes qui composent notre environnement et de la grande étendue que représentent les zones à cartographier. Néanmoins, l'utilisation d'imagerie satellitaire, combinée avec une collecte régulière de données de référence permet d'entraîner des algorithmes de classification supervisée à reconnaître et à catégoriser les pixels de ces classes. La précision de cette classification n'est pas parfaite pour toutes les éléments du territoire, comme le montre [Inglada et al., 2017]. La base de ce travail est alors de comprendre comment améliorer la façon dont les classes d'occupation des sols sont reconnues par le classifieur.

L'effort de recherche réalisé durant la thèse peut être synthétisé en trois points principaux.

1. Tout travail de recherche commence par la compréhension du problème. Ainsi, la Section 12.1 présente-t-elle les premières conclusions au sujet de l'importance du contexte dans la classification d'imagerie à haute résolution spatiale.

2. Ce premier point met en place les contraintes qui définissent l'ensemble des méthodes qui peuvent être d'intérêt pour le problème de l'occupation des sols en particulier, et permet de définir les spécifications de chacune d'entre elles. La Section 12.2 rappelle la justification du choix des méthodes retenues pour les expériences.

3. Les conclusions au sujet des observations réalisées sur les données expérimentales (Sentinel-2 et SPOT-7) sont synthétisées dans la Section 12.3. Pour cela, les conclusions communes aux deux jeux de données sont présentées en premier, car elles sont plus générales. Ensuite, les conclusions spécifiques à chaque jeu de données sont fournies.

Cette conclusion se focalise sur ce troisième point, les deux premiers étant abordés dans l'introduction et détaillés dans le manuscrit.

Les résultats expérimentaux confirment plusieurs des idées intuitives qui sont suggérées dans les premiers chapitres, au sujet de la classification contextuelle d'imagerie à haute dimension.

1. Les primitives contextuelles doivent être utilisées en conjonction avec l'information pixel.

2. L'inclusion explicite de plusieurs échelles de primitives est généralement préférable à l'inclusion d'une seule échelle.

3. L'utilisation de segments comme supports spatiaux fournit généralement un degré de précision géométrique plus élevé que l'utilisation de fenêtres glissantes, particulièrement dans les zones contenant des classes à dépendance contextuelle.

4. Les primitives structurées (densité de contours) fournissent une carte avec une géométrie plus fiable que les primitives non-structurées (moyenne, variance).

Gardant ces premières idées en tête, il y a également certaines conclusions qui n'auraient pas été facilement prévisibles.

1. Les primitives sémantiques, notamment l'histogramme des classes dans un voisinage local fournit une description contextuelle de faible dimension qui est bénéfique pour la plupart, sinon pour toutes les classes.

2. Les itérations successives de ce procédé, nommé HACCS, augmentent le taux de reconnaissance des classes, mais ont également tendance à dégrader la géométrie du résultat au fur et à mesure que l'information contextuelle est ajoutée. En général, les scores convergent après 3 à 4 itérations.

3. Lorsqu'une combinaison de primitives pixel et contextuelles est utilisée, l'équilibre entre le nombre de primitives de chaque source peut être un facteur important pour le modèle Random Forest. Si la quantité de primitives contextuelles est largement supérieure à la quantité de primitives pixel, il y a un risque de dégradation géométrique de la carte.

4. Sur le problème Sentinel-2, les couronnes d'adjacence fournissent les plus hauts scores de précision globale (Overall Accuracy, OA), avec un résultat qui est visuellement et géométriquement acceptable à de nombreux endroits. Cependant, ces supports spatiaux ont tendance à surestimer certaines classes à dépendance contextuelle, ce qui se traduit par exemple par un mauvais placement des bordures entre l'urbain diffus et la végétation.

5. Les approches de *modélisation*, en particulier le B-STF (Section 8.3), le réseau de neurones "patch-based" [Postadjian et al., 2017] et le réseau FG-UNet [Stoian et al., 2019], fournissent des résultats avec une précision géométrique dégradée par rapport à HACCS. Sur le problème Sentinel-2, HACCS et les D-CNN ont des scores de précision globale (OA) similaires.

Ces résultats nous mènent à questionner comment utiliser ensemble l'information contextuelle et l'information pixel afin d'arriver à une classification correcte.

1. La segmentation superpixel, comme tous les algorithmes de segmentation, n'est pas parfaite. L'impact des paramètres et la qualité de la segmentation dans l'absolu n'a jamais été évaluée autrement que visuellement. Comment la qualité de la segmentation superpixel influence-t-elle le résultat de la classification ?

2. La qualité géométrique est une notion plus complexe qu'on ne pourrait le penser. Dans quelle mesure peut-on concevoir un ensemble de métriques qui représenterait la variété des aspects géométriques ? Et comment peut-on s'assurer qu'ils ne mesurent que la dégradation géométrique et non d'autres effets globaux ?

3. Le jeu de données Sentinel-2 couvre une diversité de régions différentes, mais chaque tuile est relativement petite, comparée à la taille du pays. Les conclusions établies-ici sont-elles également valides sur l'étendue complète du problème ?

4. L'impact des primitives contextuelles, et les critères de sélection de primitives à utiliser est un sujet de discussion poursuivi tout au long du manuscrit. Toutefois, l'idée de laisser une part de ce travail au classifieur est d'intérêt, et même dans l'esprit de l'apprentissage automatique. Est-il alors possible de créer une approche de modélisation qui incorpore des notions géométriques, et qui puisse apprendre avec des données de référence éparses ?

5. Les méthodes proposées, comme B-STF et HACCS ne sont pas spécifiques à la cartographie de l'occupation des sols. Comment se comportent-t-elles sur d'autres problèmes de classification dense ?

6. L'effort au niveau théorique vise à créer une représentation intermédiaire entre les pixels et les objets qui les contiennent, afin de démêler les dépendances contextuelles qui résultent de la caractérisation inadéquate de ces classes au niveau de description le plus réduit: le pixel. Des classes à dépendance contextuelle présentant une forme hiérarchique peuvent être présentes dans d'autres problèmes de classification, comme les arbres sémantiques dans la reconnaissance de textes. Les concepts étudiés ici peuvent-ils être étendus à d'autres problèmes entièrement en dehors de la classification d'images ?

Les paragraphes suivants présentent quelques éléments de réponses ou idées au sujet de ces questions. Ces perspectives sont divisées en deux parties. En premier, il s'agit des développements possibles sur le court terme pour améliorer les outils, et le procédé expérimental mis en place. Ensuite, la généralisation des méthodes étudiées à d'autres problèmes, qui pourrait être faite sur le long terme, est discutée.

Le premier objectif à court terme serait d'améliorer l'algorithme SLIC lui même. En effet, une observation récurrente est que certaines erreurs demeurent dans la segmentation, dû à l'interpolation temporelle des images. Cela peut se traduire par deux modifications envisageables sur la distance utilisée par SLIC. Il serait possible de réduire la sensibilité de la méthode aux nuages non-détectés, par exemple, en prenant une distance autre que la distance Euclidienne, et qui accentuerait moins l'influence de ces pixels très clairs. Alternativement, une distance pourrait être définie sur l'espace des dates et des images, qui serait donc calculable sans interpolation temporelle. Cela demanderait un travail d'analyse théorique.

Alternativement, il serait intéressant d'inclure de l'information sémantique dans la segmentation, de la même façon que celle-ci est incorporée dans la classification. Par exemple, en utilisant une classification pixel, l'histogramme des classes dans le superpixel pourrait être calculé au cours des itérations de SLIC, en commençant par un maillage régulier. L'histogramme pourrait alors être mis-à-jour à la volée, de la même façon que les centroïdes sont mis à jour dans la version courante de SLIC. Cela impliquerait l'utilisation d'une distance spécifique aux histogrammes, ainsi qu'un poids associé à cette distance, équilibré vis-à-vis des deux autres distances. Ce serait une façon de réaliser une segmentation superpixel supervisée d'une zone à l'aide de données de référence éparses. De même, si l'information sémantique contextuelle peut être ajoutée à SLIC, on peut envisager de l'inclure dans d'autres algorithmes de segmentation.

Ensuite, ces méthodes devraient être validées sur de plus larges étendues. Une évaluation sur une région éco-climatique, voire sur la France entière, pourrait confirmer ou questionner les conclusions tirées ici.

Les méthodes présentées ici sont prometteuses pour les problèmes de classification d'imagerie hyperspectrale, qui présentent également un très grand nombre de primitives, et sont utilisées pour l'occupation des sols. Dans l'hyperspectral comme dans les séries temporelles, la majorité des primitives sont valables pour représenter certains aspects du problème. Une description relativement légère du contexte, à partir d'histogrammes de classes, serait alors d'intérêt pour des applications de ce type.

Sur le sujet des approches par modélisation, une méthode relativement simple mise en lumière dans ce manuscrit est le Semantic Texton Forest (STF), et sa version basique, le Basic STF (B-STF). Sur le jeu de données SPOT-7, Le B-STF méthode fournit une meilleure OA que les primitives contextuelles évaluées, à savoir, les statistiques locales (moyenne et variance), la densité de contours, et les profils morphologiques. Une question demeure: peut-on appliquer ce même processus aux images à haute dimension, comme les séries temporelles ? En effet, il y a des limites d'ordre matériel (temps de calcul et mémoire) qui empêchent les méthodes de classification supervisée actuelles de prendre en compte un trop grand nombre de primitives à la fois. Il faudrait alors retravailler le modèle du Random Forest et son implémentation courante, pour permettre l'évaluation de ce genre de méthodes sur le cas Sentinel-2.

Sur le long terme, il serait intéressant d'évaluer l'effet de la densité de la donnée d'apprentissage sur la géométrie de la carte de classification, selon la méthode envisagée. En effet, il existe des zones où une donnée dense est disponible, même si elle contient des erreurs. L'idée serait alors de voir s'il vaut mieux accorder la priorité à la justesse des étiquettes d'apprentissage, et les prendre de façon éparse, ou tolérer quelques erreurs, mais garder une plus grande part d'information géométrique.

Cela lève une autre question, qui fait lieu de perspective finale. Serait-il possible de concevoir un modèle ou une approche de classification qui utilise une donnée d'apprentissage constituée à la fois de polygones étiquetés et de polygones non-étiquetés ? Il serait alors possible de bénéficier de données d'apprentissage qui ne sont pas à jour au niveau des classes, ou qui décrivent une nomenclature un peu différente, mais qui contiennent une géométrie valable. Par exemple, les surfaces agricoles ont une géométrie très stable, même si la classe de culture change tous les ans.

Part VI

Appendices

A

Calibration of the PBCM

This appendix extends the calibration of the PBCM on the Sentinel-2 data set, that was addressed in Chapter 6, in Section 6.3.4, on page 91. The calibration of a process involving 10 parameters is not easy to visualize, therefore the choice was made here to only show the variations of the corner extraction and matching steps. First of all, the matching threshold is shown along the X axis, in meters. This is the parameter of the final stage of the PBCM, it defines the distance at which two corners match. Second of all, the extremity distance threshold, d, also expressed in meters, is the minimal distance at which two segment extremities must be to form a corner. Finally, the angle interval, is shown slightly differently here, through a wideness parameter called a. Equation (A.1) shows how to calculate the angular interval $[a_{min}, a_{max}]$ using a.

$$[a_{min}, a_{max}] = [\frac{\pi}{2} - \frac{\pi}{3}a, \frac{\pi}{2} + \frac{\pi}{3}a] \tag{A.1}$$

The parameter a expresses how narrow the interval is centered around 90°, with a value of 1 representing the 30°- 150°, interval and for instance 0.5 representing the interval 60°-120°.

The six graphs presented here, in Figures A.1 and A.2 show the impact of the calibration parameters of the corner matching and corner extraction steps, that is to say, with the line segment detection parameters maintained constant. The top graph shows the result of matching a pixel-based classification with another pixel-based classification, generated with a different sampling of the training data. This value should be high, as these two classifications have similar geometry. The middle graph shows the matching between a pixel-based classification and a regularized classification, which should therefore generate a low PBCM, as the corners dissapear or are displaced. Finally, the bottom graph shows the difference between the pixel-based matching, and regularized matching scores, in other words, the difference between the top and middle graphs. This is the value that should be maximized for an appropriate calibration of the metric.

The lowest graph on the large regularization case, Figure A.2, pushes towards a corner and detection and matching step with a strict matching threshold of 10m (1 pixel), and a rather loose extremity distance threshold of 10m (1 pixel), combined with an average angular threshold factor of 0.5, in other words, an angular interval of [60°,120°]. It seems beneficial for the difference score to detect large numbers of corners, but to be more strict on the matching criterion, than to detect fewer corners but to let them match easily.

These graphs are shown to illustrate how the calibration process works, and how an optimization scheme can be set up and used, once the appropriate cost function is set up. The degradation of corners in a highly regularized image is clearly visible, everywhere in the image. The corners are usually well represented in different pixel-based classification results, which makes the difference between these two cases what the PBCM should measure. Therefore, maximizing the difference between the values on pixel-based classifications and strongly regularized cases allows to find correct parameters without needing to manually calibrate them individually, which would be nearly impossible to do.

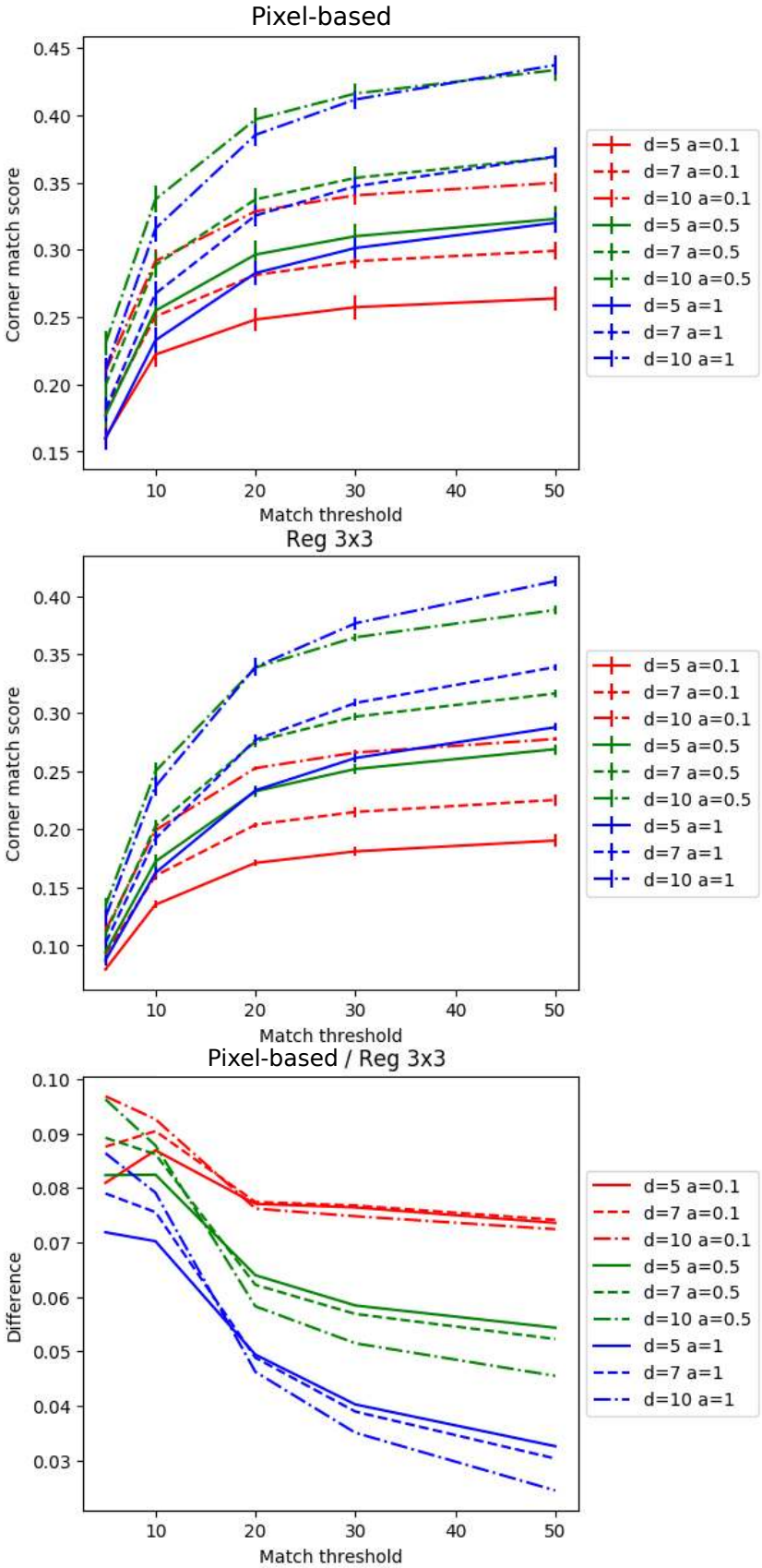


Figure A.1: Calibration graphs of the small regularization window case, 3×3 pixels. The X axis represents the matching threshold (in meters), which defines the maximal distance at which two detected corners should be in order to be considered as matching. The two parameters that control the corner detection step, d and a, are respectively the extremity distance threshold and angle factor (defined in equation (A.1)). The Y axis in the top two graphs shows the effective matching score, which should be high in the upper graph, and low in the middle graph. The Y axis in the lowest graph shows the difference between the matching score in the upper and middle graph, in other words between the pixel-based and regularized case. For a small regularization, detecting fewer corners by using a narrow angular interval is recommended by the calibration. In all three cases, using a loose extremity matching factor (/d/=10m) provides a higher difference in PBCM between the pixel-based and regularized maps. Moreover, using a low matching threshold also increases this difference.

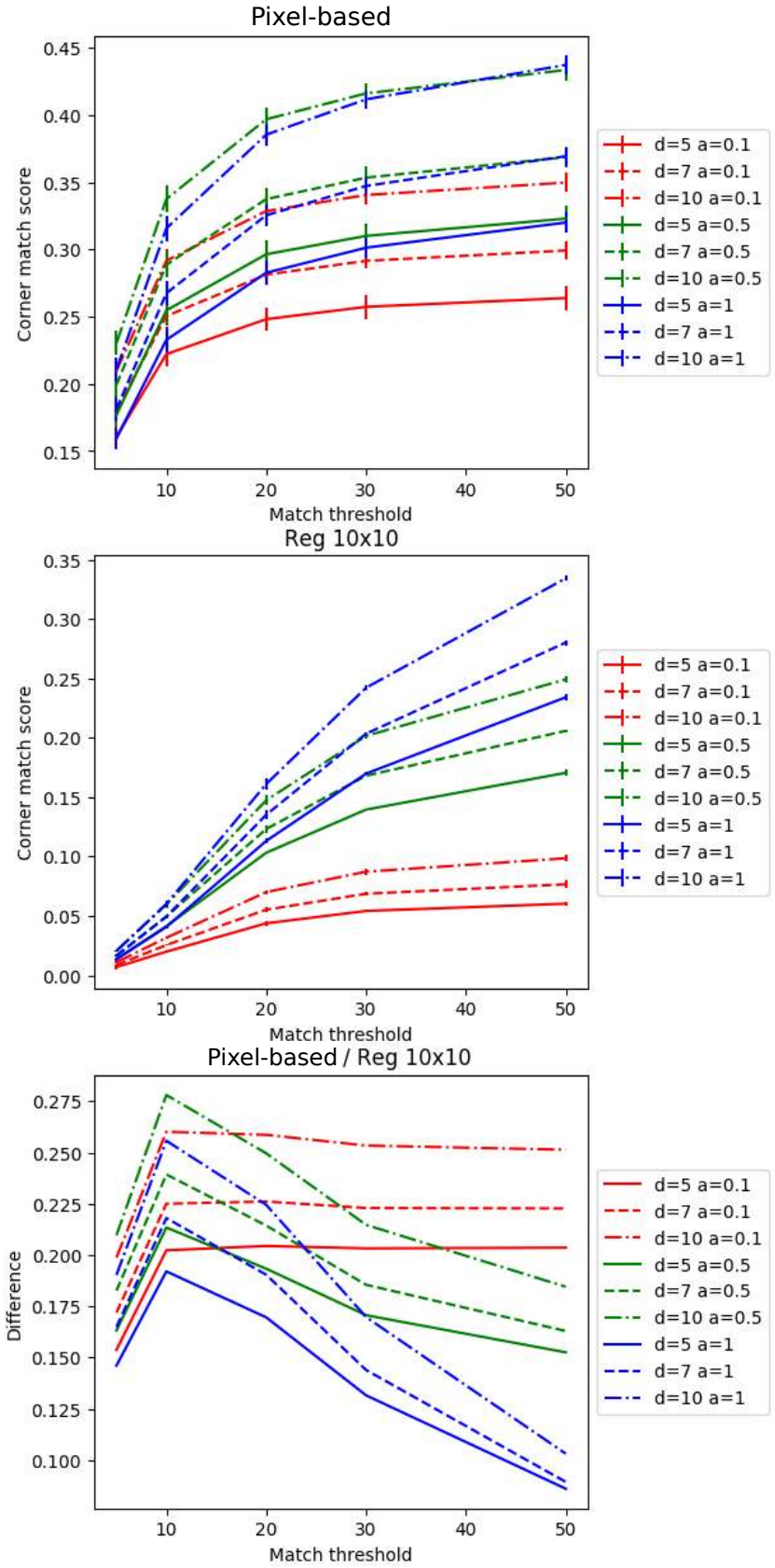


Figure A.2: Calibration graphs of the large regularization window case, 10×10 pixels, following the same convention as A.2. In this case, an average angular interval is preferable over a narrow one. This is once again best combined with a loose extremity matching factor, and a strict corner matching factor. It seems that the highest difference between the pixel and regularized case can be observed when a larger number of corners are detected, but the metric is strict about matching them. This is better than to detect fewer corners, therefore being more certain of their pertinence, but to be less restrictive on the definition of a match.

B

Sentinel-2 experiments

The following tables are complimentary to the experimental results presented in Chapter 10, in particular Sections 10.2.1 and 10.3.1. Tables B.1 - B.8 show the impact of the scale of spatial support on the F-scores of the various classes, for local statistics and edge density features. Tables B.9 - B.12 show how the shape and size of the spatial support has an impact on the F-scores, for the class histogram feature. The interpretation of these tables is dicussed on pages 145 and 152.

Table B.1: F-scores of the different classes on T31TCJ, using a superpixel spatial support with Local Statistics features only (no pixel information). In this case, the smallest spatial support provides the highest OA and PBCM.

Spatial supp.		SP	SP	SP	SP	SP
Feature	P	LS	LS	LS	LS	LS
Scale		5	7	10	15	30
Overall Acc.	73.7%±0.20	**74.7%**±0.25	74.2%±0.29	73.4%±0.29	72.1%±0.28	68.1%±0.27
Kappa	71.7%±0.22	**72.7%**±0.27	72.2%±0.31	71.4%±0.31	69.9%±0.30	65.6%±0.29
ASC	**0.915**	0.894	0.887	0.878	0.868	0.851
AWC	**0.909**	0.894	0.889	0.880	0.874	0.866
BLF	**0.831**	0.807	0.802	0.796	0.786	0.776
COF	0.806	0.804	0.806	**0.812**	0.812	0.766
NGL	**0.322**	0.314	0.285	0.251	0.227	0.212
WOM	**0.473**	0.439	0.427	0.384	0.374	0.296
CUF	0.604	**0.622**	0.619	0.620	0.622	0.612
DUF	0.576	**0.624**	0.620	0.609	0.575	0.491
ICU	0.592	0.638	**0.641**	0.635	0.611	0.560
RSF	0.838	**0.845**	0.836	0.821	0.791	0.731
WAT	**0.989**	0.988	0.988	0.987	0.987	0.979
IGL	**0.677**	0.672	0.660	0.655	0.647	0.614
ORC	0.824	0.855	**0.857**	0.846	0.843	0.793
VIN	0.833	**0.857**	0.851	0.841	0.813	0.752
PBCM	50.8%±0.91	**31.6%**	29.0%	26.4%	23.5%	21.0%

Table B.2: F-scores of the different classes on T31TCJ, using a superpixel spatial support with Pixel and Local Statistics features. The F-scores and PBCM are significantly higher than when no pixel information is used (Table B.1), and once again, the smallest spatial support provides the highest results.

Spatial supp.		SP	SP	SP	SP	SP
Feature	P	P+LS	P+LS	P+LS	P+LS	P+LS
Scale		5	7	10	15	30
Overall Acc.	73.7%±0.20	**75.3%**±0.23	74.9%±0.28	74.3%±0.28	73.4%±0.28	71.4%±0.27
Kappa	71.7%±0.22	**73.4%**±0.25	72.9%±0.30	72.3%±0.31	71.3%±0.30	69.1%±0.29
ASC	**0.915**	0.899	0.893	0.885	0.878	0.872
AWC	**0.909**	0.899	0.897	0.889	0.884	0.890
BLF	**0.831**	0.817	0.809	0.803	0.800	0.804
COF	0.806	0.812	0.804	0.811	**0.815**	0.807
NGL	0.322	**0.332**	0.307	0.280	0.248	0.225
WOM	**0.473**	0.454	0.439	0.411	0.400	0.333
CUF	0.604	**0.631**	0.627	0.624	0.625	0.620
DUF	0.576	**0.629**	0.626	0.619	0.597	0.532
ICU	0.592	0.642	**0.647**	0.645	0.629	0.614
RSF	0.838	**0.857**	0.855	0.845	0.833	0.842
WAT	0.989	**0.989**	0.989	0.988	0.989	0.987
IGL	0.677	**0.680**	0.670	0.662	0.655	0.634
ORC	0.824	**0.858**	0.858	0.853	0.850	0.814
VIN	0.833	**0.864**	0.858	0.848	0.822	0.768
PBCM	50.8%±0.91	**35.3%**	32.3%	29.1%	25.9%	26.0%

Table B.3: F-scores of the different classes on T31TCJ, using a superpixel spatial support with Edge Density features only (no pixel information). The performance is hardly better than the pixel-based classification. A local scale of information seems to be best described with this choice of spatial support and feature.

Spatial supp.		SP	SP	SP	SP	SP
Feature	P	ED	ED	ED	ED	ED
Scale		5	7	10	15	30
Overall Acc.	73.7%±0.20	**74.2%**±0.16	73.9%±0.17	73.4%±0.17	72.5%±0.18	69.7%±0.21
Kappa	71.7%±0.22	**72.2%**±0.17	71.9%±0.18	71.4%±0.18	70.3%±0.19	67.4%±0.23
ASC	**0.915**	0.888	0.885	0.879	0.872	0.864
AWC	**0.909**	0.889	0.888	0.884	0.877	0.863
BLF	**0.831**	0.823	0.818	0.817	0.802	0.782
COF	0.806	**0.824**	0.821	0.823	0.799	0.768
NGL	**0.322**	0.299	0.288	0.255	0.227	0.180
WOM	**0.473**	0.417	0.408	0.392	0.388	0.352
CUF	0.604	**0.628**	0.625	0.621	0.616	0.601
DUF	0.576	0.642	0.647	**0.651**	0.644	0.587
ICU	0.592	0.644	**0.646**	0.641	0.631	0.572
RSF	0.838	**0.855**	0.850	0.838	0.832	0.770
WAT	**0.989**	0.986	0.986	0.986	0.987	0.986
IGL	**0.677**	0.661	0.655	0.647	0.634	0.604
ORC	**0.824**	0.819	0.813	0.808	0.794	0.783
VIN	**0.833**	0.820	0.816	0.805	0.783	0.756
PBCM	50.8%±0.91	31.2%	**31.8%**	31.7%	31.3%	29.4%

Table B.4: F-scores of the different classes on T31TCJ, using a superpixel spatial support with Pixel and Edge Density features. The F-scores and PBCM are significantly higher than when no pixel information is used (Table B.3). Once again, the local scale is preferable.

Spatial supp.		SP	SP	SP	SP	SP
Feature	P	P+ED	P+ED	P+ED	P+ED	P+ED
Scale		5	7	10	15	30
Overall Acc.	73.7%±0.20	**75.3%**±0.23	74.9%±0.28	74.3%±0.28	73.4%±0.28	71.4%±0.27
Kappa	71.7%±0.22	**73.4%**±0.25	72.9%±0.30	72.3%±0.31	71.3%±0.30	69.1%±0.29
ASC	**0.915**	0.899	0.893	0.885	0.878	0.872
AWC	**0.909**	0.899	0.897	0.889	0.884	0.890
BLF	**0.831**	0.817	0.809	0.803	0.800	0.804
COF	0.806	0.812	0.804	0.811	**0.815**	0.807
NGL	0.322	**0.332**	0.307	0.280	0.248	0.225
WOM	**0.473**	0.454	0.439	0.411	0.400	0.333
CUF	0.604	**0.631**	0.627	0.624	0.625	0.620
DUF	0.576	**0.629**	0.626	0.619	0.597	0.532
ICU	0.592	0.642	**0.647**	0.645	0.629	0.614
RSF	0.838	**0.857**	0.855	0.845	0.833	0.842
WAT	0.989	**0.989**	0.989	0.988	0.989	0.987
IGL	0.677	**0.680**	0.670	0.662	0.655	0.634
ORC	0.824	**0.858**	0.858	0.853	0.850	0.814
VIN	0.833	**0.864**	0.858	0.848	0.822	0.768
PBCM	50.8%±0.91	**35.3%**	32.3%	29.1%	25.9%	26.0%

Table B.5: F-scores of the different classes on T31TCJ, using a sliding window spatial support with Local Statistics features only (no pixel information). The scale of 5 provides the highest value of OA.

Spatial supp.		SW	SW	SW	SW	SW
Feature	P	LS	LS	LS	LS	LS
Scale		5	7	10	15	30
Overall Acc.	73.7%±0.20	**76.4%**±0.23	76.2%±0.27	75.6%±0.23	74.8%±0.26	73.1%±0.27
Kappa	71.7%±0.22	**74.6%**±0.24	74.3%±0.29	73.7%±0.24	72.9%±0.28	71.0%±0.29
ASC	**0.915**	0.888	0.882	0.873	0.863	0.859
AWC	**0.909**	0.898	0.895	0.890	0.881	0.885
BLF	**0.831**	0.830	0.825	0.819	0.821	0.817
COF	0.806	0.826	0.822	0.824	**0.831**	0.829
NGL	0.322	**0.332**	0.319	0.291	0.273	0.233
WOM	**0.473**	0.465	0.455	0.444	0.435	0.386
CUF	0.604	**0.652**	0.648	0.633	0.625	0.619
DUF	0.576	0.667	**0.668**	0.663	0.650	0.607
ICU	0.592	**0.669**	0.669	0.660	0.642	0.607
RSF	0.838	**0.874**	0.868	0.858	0.847	0.833
WAT	**0.989**	0.988	0.988	0.988	0.988	0.987
IGL	0.677	**0.692**	0.687	0.681	0.670	0.640
ORC	0.824	0.861	**0.861**	0.857	0.860	0.857
VIN	0.833	**0.868**	0.863	0.855	0.842	0.811
PBCM	**50.8%**±0.91	33.7%	31.2%	28.8%	26.1%	26.0%

Table B.6: F-scores of the different classes on T31TCJ, using a sliding window spatial support with Pixel and Local Statistics features. The F-scores and PBCM are significantly higher than when no pixel information is used (Table B.5), once again, in a local spatial support.

Spatial supp.		SW	SW	SW	SW	SW
Feature	P	P+LS	P+LS	P+LS	P+LS	P+LS
Scale	Pixel	5	7	10	15	30
Overall Acc.	73.7%±0.20	**76.9%**±0.23	76.9%±0.25	76.8%±0.24	76.2%±0.25	74.9%±0.23
Kappa	71.7%±0.22	**75.1%**±0.25	75.1%±0.27	75.0%±0.26	74.3%±0.27	73.0%±0.25
ASC	**0.915**	0.902	0.902	0.902	0.903	0.908
AWC	**0.909**	0.899	0.898	0.898	0.899	0.902
BLF	0.831	**0.831**	0.822	0.819	0.820	0.827
COF	0.806	0.819	0.818	0.820	0.827	**0.837**
NGL	0.322	**0.350**	0.338	0.328	0.312	0.302
WOM	0.473	**0.481**	0.475	0.461	0.433	0.402
CUF	0.604	0.687	0.699	**0.705**	0.704	0.679
DUF	0.576	0.658	0.666	**0.669**	0.655	0.601
ICU	0.592	0.671	0.677	**0.682**	0.680	0.659
RSF	0.838	0.882	0.882	**0.884**	0.881	0.870
WAT	0.989	**0.990**	0.989	0.989	0.989	0.989
IGL	0.677	**0.693**	0.692	0.687	0.673	0.659
ORC	0.824	**0.859**	0.852	0.848	0.839	0.827
VIN	0.833	**0.868**	0.862	0.857	0.838	0.819
PBCM	50.8%±0.91	25.6%	23.9%	25.0%	29.8%	**37.0%**

Table B.7: F-scores of the different classes on T31TCJ, using a sliding window spatial support with Edge Density features only (no pixel information). The performance is hardly better than the pixel-based classification.

Spatial supp.		SW	SW	SW	SW	SW
Feature	P	ED	ED	ED	ED	ED
Scale		5	7	10	15	30
Overall Acc.	73.7%±0.20	74.9%±0.16	**75.0%**±0.16	74.9%±0.16	74.5%±0.20	72.5%±0.16
Kappa	71.7%±0.22	73.0%±0.17	**73.0%**±0.17	73.0%±0.17	72.5%±0.21	70.4%±0.17
ASC	**0.915**	0.892	0.892	0.891	0.892	0.887
AWC	**0.909**	0.891	0.891	0.889	0.886	0.880
BLF	**0.831**	0.829	0.829	0.828	0.822	0.796
COF	0.806	0.828	0.829	**0.830**	0.823	0.777
NGL	**0.322**	0.309	0.302	0.299	0.296	0.256
WOM	**0.473**	0.432	0.426	0.424	0.403	0.367
CUF	0.604	0.644	0.649	0.653	**0.653**	0.637
DUF	0.576	0.651	0.657	0.663	**0.664**	0.635
ICU	0.592	0.652	0.657	**0.659**	0.655	0.633
RSF	0.838	0.869	**0.871**	0.869	0.860	0.841
WAT	**0.989**	0.987	0.987	0.987	0.988	0.988
IGL	**0.677**	0.670	0.668	0.665	0.655	0.637
ORC	0.824	**0.826**	0.823	0.822	0.814	0.792
VIN	**0.833**	0.825	0.824	0.818	0.808	0.782
PBCM	50.8%±0.91	29.9%	30.7%	30.7%	31.8%	**32.2%**

Table B.8: F-scores of the different classes on T31TCJ, using a sliding window spatial support with Pixel and Edge Density features. The F-scores and PBCM are significantly higher than when no pixel information is used (Table B.7). Compared to the other combinations of spatial supports and features, in this case, a larger context can be taken into account, which reflects on the higher classification scores.

Spatial supp.		SW	SW	SW	SW	SW
Feature	P	P+ED	P+ED	P+ED	P+ED	P+ED
Scale		5	7	10	15	30
Overall Acc.	73.7%±0.20	77.4%±0.21	77.4%±0.22	**77.5%**±0.22	77.4%±0.24	76.6%±0.23
Kappa	71.7%±0.22	75.6%±0.23	75.7%±0.24	**75.8%**±0.24	75.6%±0.26	74.8%±0.25
ASC	**0.915**	0.901	0.900	0.901	0.900	0.900
AWC	**0.909**	0.903	0.902	0.901	0.900	0.898
BLF	0.831	0.843	0.842	**0.845**	0.843	0.840
COF	0.806	0.837	0.837	0.841	**0.841**	0.837
NGL	0.322	**0.356**	0.349	0.351	0.343	0.335
WOM	0.473	**0.481**	0.477	0.475	0.469	0.454
CUF	0.604	0.669	0.674	0.676	**0.678**	0.669
DUF	0.576	0.673	0.680	0.684	**0.690**	0.671
ICU	0.592	0.680	0.686	0.689	**0.690**	0.675
RSF	0.838	0.894	0.897	0.897	**0.899**	0.897
WAT	0.989	0.988	0.989	**0.989**	0.989	0.989
IGL	0.677	**0.705**	0.704	0.702	0.696	0.687
ORC	0.824	**0.863**	0.862	0.862	0.857	0.852
VIN	0.833	**0.870**	0.867	0.864	0.855	0.839
PBCM	50.8%±0.91	35.9%	36.4%	37.2%	38.6%	**39.0%**

Table B.9: F-scores of the local histogram of classes feature, using a sliding window of varying scale (side of the window in pixels).

Scale	Pix	5	7	10	15	30	40	50
Overall Acc.	73.7%±0.21	77.2%±0.24	78.0%±0.25	**78.3%** ±0.26	77.3%±0.26	74.7%±0.22	74.3%±0.19	74.1%±0.21
Kappa	71.7%±0.27	75.5%±0.26	76.3%±0.27	**76.6%** ±0.28	75.6%±0.28	72.7%±0.24	72.3%±0.20	72.1%±0.22
ASC	0.914	0.918	0.921	**0.923**	0.917	0.911	0.910	0.909
AWC	0.909	0.914	0.915	**0.916**	0.911	0.907	0.907	0.907
BLF	0.831	0.850	**0.853**	0.853	0.848	0.838	0.836	0.831
COF	0.806	0.835	0.840	**0.842**	0.834	0.819	0.815	0.808
NGL	0.322	**0.349**	0.347	0.342	0.342	0.303	0.296	0.299
WOM	0.473	0.514	0.526	**0.531**	0.528	0.492	0.486	0.480
CUF	0.604	0.682	0.698	**0.704**	0.684	0.624	0.621	0.621
DUF	0.576	0.638	0.653	0.662	**0.668**	0.599	0.592	0.587
ICU	0.592	0.664	0.673	**0.676**	0.657	0.603	0.601	0.600
RSF	0.838	0.880	0.888	**0.889**	0.874	0.848	0.844	0.841
WAT	0.989	0.989	0.990	**0.990**	0.989	0.989	0.989	0.989
IGL	0.677	0.714	0.719	**0.719**	0.704	0.678	0.676	0.677
ORC	0.824	0.861	0.871	**0.876**	0.850	0.837	0.831	0.828
VIN	0.833	0.869	0.878	**0.883**	0.866	0.843	0.840	0.837
PBCM	51.3%±0.96	28.0%±0.39	29.6%±0.38	29.7%±0.45	43.6%±0.54	49.3%±0.69	49.6%±0.71	**50.4%** ±0.75

Table B.10: F-scores of the local histogram of classes feature, using one scale of superpixel. The bold values indicate the higest value across both mono-scale and mutli-scale superpixels (see table B.11)

Scale	Pix	5	7	10	15	30	40	50
Overall Acc.	73.7%±0.21	75.3%±0.25	75.8%±0.25	76.1%±0.22	76.9%±0.24	77.3%±0.17	77.1%±0.32	76.9%±0.29
Kappa	71.7%±0.27	73.4%±0.27	73.9%±0.27	74.3%±0.24	75.1%±0.25	75.5%±0.19	75.3%±0.35	75.1%±0.32
ASC	0.914	0.918	0.921	0.921	0.927	**0.927**	0.924	0.918
AWC	0.909	0.915	0.915	0.914	0.915	**0.917**	0.911	0.910
BLF	0.831	0.839	0.838	0.840	0.845	0.848	0.842	0.841
COF	0.806	0.823	0.832	0.826	0.835	0.838	**0.844**	0.839
NGL	0.322	0.297	0.310	0.331	0.340	**0.359**	0.324	0.329
WOM	0.473	0.496	0.494	0.510	0.512	0.514	**0.530**	0.522
CUF	0.604	0.654	0.667	0.668	0.673	0.667	0.668	0.661
DUF	0.576	0.597	0.605	0.604	0.620	0.626	0.619	0.616
ICU	0.592	0.623	0.629	0.639	0.638	0.643	0.640	0.636
RSF	0.838	0.859	0.863	0.867	0.870	0.882	0.869	0.872
WAT	0.989	**0.990**	0.988	0.989	0.989	0.989	0.989	0.989
IGL	0.677	0.696	0.699	0.702	0.711	0.696	0.697	0.701
ORC	0.824	0.842	0.849	0.850	0.861	0.878	0.884	**0.886**
VIN	0.833	0.853	0.858	0.860	0.877	**0.883**	0.880	0.876
PBCM	51.3%±0.96	37.8%±0.32	37.6%±0.33	36.4%±0.34	35.6%±0.39	40.0%±0.49	42.6%±0.58	**44.1%** ±0.64

Table B.11: F-scores of the local histogram of classes feature, using multi-scale superpixels (HACCS). The bold values indicate the highest value across both mono-scale and mutli-scale superpixel choice (see table B.10)

Scale	Pix	5 30	10 50	10 30 50	5 7 10 15 30 40 50
Overall Acc.	73.7%±0.21	76.2%±0.23	76.8%±0.37	**78.1%** ±0.23	76.3%±0.40
Kappa	71.7%±0.27	74.3%±0.25	75.0%±0.40	**76.4%** ±0.24	74.5%±0.43
ASC	0.914	0.919	0.917	0.922	0.918
AWC	0.909	0.913	0.909	0.916	0.916
BLF	0.831	0.837	0.840	**0.849**	0.841
COF	0.806	0.827	0.834	0.842	0.822
NGL	0.322	0.325	0.307	0.333	0.276
WOM	0.473	0.509	0.504	0.518	0.510
CUF	0.604	0.678	0.692	**0.708**	0.692
DUF	0.576	0.604	0.642	**0.651**	0.624
ICU	0.592	0.641	0.647	**0.665**	0.650
RSF	0.838	0.866	0.870	**0.890**	0.869
WAT	0.989	0.989	0.988	0.989	0.988
IGL	0.677	0.701	0.704	**0.713**	0.710
ORC	0.824	0.849	0.865	0.880	0.848
VIN	0.833	0.859	0.862	0.878	0.856
PBCM	51.3%±0.96	34.4%±0.35	35.8%±0.34	30.7%±0.33	38.4%±0.46

Table B.12: F-scores of the local histogram of classes feature, using multi-scale superpixels (HACCS) using adjacency layers 0123 with varying base superpixel size.

Base SP size	Pix	5	7	10	15	30
Overall Acc.	73.7%±0.21	78.5%±0.27	78.7%±0.26	**78.8%** ±0.24	78.8%±0.28	78.3%±0.26
Kappa	71.7%±0.27	76.8%±0.29	77.0%±0.28	**77.2%** ±0.26	77.1%±0.30	76.6%±0.28
ASC	0.914	0.920	0.920	0.921	**0.925**	0.924
AWC	0.909	0.918	**0.919**	0.918	0.918	0.916
BLF	0.831	0.849	**0.852**	0.852	0.852	0.849
COF	0.806	0.841	0.846	0.854	0.854	**0.865**
NGL	0.322	0.338	0.344	**0.347**	0.343	0.313
WOM	0.473	0.523	0.529	**0.538**	0.532	0.511
CUF	0.604	**0.718**	0.716	0.713	0.705	0.691
DUF	0.576	**0.677**	0.674	0.670	0.661	0.645
ICU	0.592	**0.681**	0.681	0.679	0.675	0.661
RSF	0.838	0.905	0.908	0.910	**0.915**	0.905
WAT	0.989	**0.990**	0.990	0.989	0.989	0.989
IGL	0.677	0.711	0.715	0.715	**0.716**	0.701
ORC	0.824	0.875	0.875	0.878	0.882	**0.896**
VIN	0.833	0.872	0.876	0.878	0.885	**0.897**
PBCM	51.3%±0.96	28.7%±0.37	30.0%±0.34	30.5%±0.31	31.5%±0.34	**35.5%** ±0.42

Table B.13: Class distribution of the 11 tiles in the data set used in Sections 10.2.2 and 10.3.2.

Name	T30TWT	T30TXQ	T30TYN	T30UXV	T31TDJ	T31TDN	T31TEL	T31TGK	T31UDQ	T31UDS	T32ULU
Index	1	2	3	4	5	6	7	8	9	10	11
ASC	186084	112870	39577	111140	61063	66908	127364	2576	109349	32778	114279
AWC	170839	2484	9271	169424	132590	368659	186646	14149	1079411	151941	74096
BLF	65003	92320	137546	47169	143008	438458	352358	143659	429562	56820	296402
COF	86055	3406794	123960	14575	209091	99486	1571267	864432	102111	1541	985948
NGL	6496	0	335427	0	58593	732	31962	825736	1220	1377	24063
WML	112019	102667	184257	7975	64156	20671	111814	307237	27488	8468	8641
CUF	12262	34133	3271	14154	1841	2247	39067	373	287388	15081	15947
DUF	197925	368733	25559	50932	25543	31550	233471	13739	1035603	82494	107867
ICU	178134	281722	14706	50771	20237	39447	148008	5679	841308	108103	79574
RSF	3761	22572	1674	2307	1029	2803	19327	1214	67095	12900	9203
BRO	0	0	231491	406	0	0	80	77756	0	0	0
BDS	3729	112848	0	3811	0	1687	0	14315	0	3778	0
WAT	4650221	6458981	12511	1959898	58971	80282	46231	34065	84697	1894379	44049
GPS	0	0	1978	0	0	0	0	54664	0	0	0
IGL	249625	22112	109417	301560	87286	89923	761126	117134	150013	40845	156124
ORC	1499	345	37	2205	2171	809	202	6122	4935	578	695
VIN	4406	52594	89	0	17135	2784	917	78	0	0	3433

C

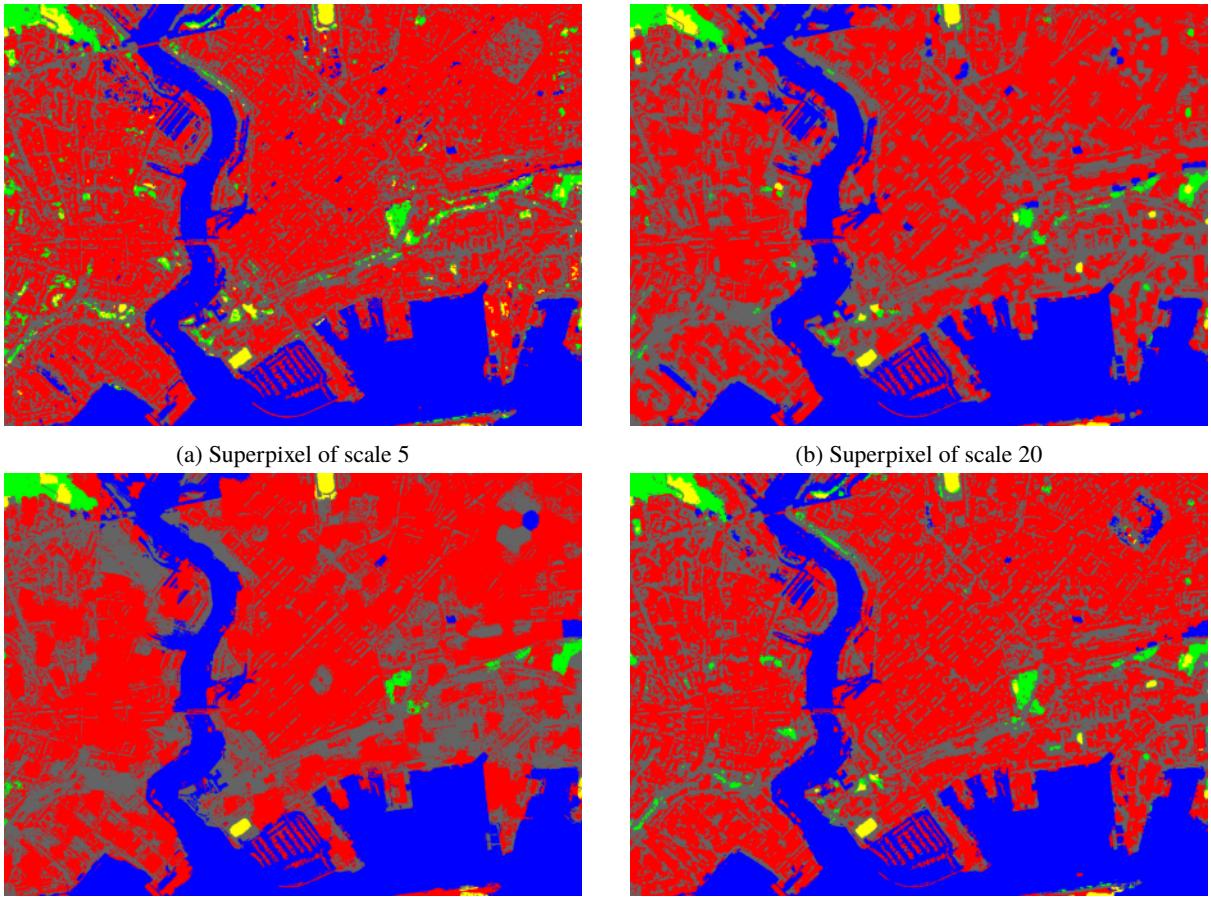
SPOT-7 experiments

Here, complimentary results to the ones shown in Chapter 11 are given. In Section 11.1, on page 161, it was mentioned that the justification of the choice of the scale of superpixel, in the results shown on the SPOT-7 classification problem were not necessarily of primary importance regarding the subject matter. This appendix presents the classification results obtained when including Extended Morphological Profiles into the classification, and then calculating HACCS in one or more superpixel. The main objective of this appendix is to compare the difference between the use of one and several superpixels, and if only one can be used, which size should it have. To this end, Figure C.1 shows a zoom of the classification results on the central urban area of Brest.

Next, Table C.1 shows the OA and F-scores of classification results shown in C.1. The same effect is observed: if only one scale is used, it is preferable to use the a local scale, and if multiple scales can be used, the accuracy of the map is higher than when only one scale is used. This is the case for almost all of the classes, except for the water class which is relatively well recognized using this contextual classification method. These results show that multi-scale neighborhood is desirable in this scenario, and justify why Chapter 11 focuses only on the multi-scale spatial support.

Table C.1: Classification accuracy and F-score for different scales of the superpixel spatial support, on the 5 class SPOT-7 classification problem. When one scale is used, the small scales provide higher scores, although the result is hardly better than the pixel-based classification. Using all of the scales is clearly beneficial over using only one scale.

Spatial support	Scale(s)	OA (%)	κ (%)	Urban	Crop	Water	Roads	Veg.
P		81.8%	77.3%	0.771	0.887	0.904	0.709	0.828
SP	5	82.0%	77.5%	0.780	0.874	0.924	0.711	0.813
SP	10	82.0%	77.5%	0.769	0.876	0.929	0.711	0.816
SP	20	81.4%	76.8%	0.746	0.882	0.929	0.693	0.823
SP	40	81.7%	77.1%	0.747	0.883	0.932	0.695	0.828
SP	50	80.9%	76.1%	0.737	0.879	**0.934**	0.670	0.826
SP	5+10+20+30+40+50	**83.4%**	**79.2%**	**0.784**	**0.895**	0.932	**0.721**	**0.839**



(a) Superpixel of scale 5

(b) Superpixel of scale 20

(c) Superpixel of scale 50

(d) Superpixel of scales 5+10+20+30+40+50

Figure C.1: Influence of the choice of spatial support size, and evaluation of the use of multiple scales. If the scale is too small, parts of the urban vegetation are classified as vegetation rather than urban. Moreover, the confusion between water and roads in the urban area is higher. When the scale is too large, the detailed structure of the streets and buildings is partially lost, and a far smoother result, with wide areas that bear identical class labels. Some of the larger roads are correctly labeled but the overall aspect of the result lacks in coherence in many places. When multiple scales are considered, the aspect of the map is similar to the aspect of the map of the smallest scale. In this case, including both short and long range contextual information is clearly beneficial regarding the visual quality of the maps.

Part VII

Bibliography

Bibliography

[Abrams, 2000] Abrams, M. (2000). The advanced spaceborne thermal emission and reflection radiometer (aster): data products for the high spatial resolution imager on nasa's terra platform. *international Journal of Remote sensing*, 21(5):847–859.

[Achanta et al., 2012] Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., and Süsstrunk, S. (2012). Slic superpixels compared to state-of-the-art superpixel methods. *IEEE transactions on pattern analysis and machine intelligence*, 34(11):2274–2282.

[Arnaud et al., 2003] Arnaud, A., Adam, N., Hanssen, R., Inglada, J., Duro, J., Closa, J., and Eineder, M. (2003). Asar ers interferometric phase continuity. In *IGARSS 2003. 2003 IEEE International Geoscience and Remote Sensing Symposium. Proceedings (IEEE Cat. No. 03CH37477)*, volume 2, pages 1133–1135. IEEE.

[Audebert et al., 2019] Audebert, N., Boulch, A., Le Saux, B., and Lefèvre, S. (2019). Distance transform regression for spatially-aware deep semantic segmentation. *Computer Vision and Image Understanding*, 189:102809.

[Audebert et al., 2016a] Audebert, N., Le Saux, B., and Lefevre, S. (2016a). 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)*, pages 5091–5094. IEEE.

[Audebert et al., 2016b] Audebert, N., Le Saux, B., and Lefèvre, S. (2016b). Semantic segmentation of earth observation data using multimodal and multi-scale deep networks. In *Asian Conference on Computer Vision*, pages 180–196. Springer.

[Audebert et al., 2018] Audebert, N., Le Saux, B., and Lefèvre, S. (2018). Beyond rgb: Very high resolution urban remote sensing with multimodal deep networks. *ISPRS Journal of Photogrammetry and Remote Sensing*, 140:20–32.

[Austin et al., 2017] Austin, K., Mosnier, A., Pirker, J., McCallum, I., Fritz, S., and Kasibhatla, P. (2017). Shifting patterns of oil palm driven deforestation in indonesia and implications for zero-deforestation commitments. *Land use policy*, 69:41–48.

[Baatz, 2000] Baatz, M. (2000). Multiresolution segmentation: an optimization approach for high quality multi-scale image segmentation. *Angewandte geographische informationsverarbeitung*, pages 12–23.

[Badrinarayanan et al., 2015] Badrinarayanan, V., Kendall, A., and Cipolla, R. (2015). Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *arXiv preprint arXiv:1511.00561*.

[Bay et al., 2006] Bay, H., Tuytelaars, T., and Van Gool, L. (2006). Surf: Speeded up robust features. In *European conference on computer vision*, pages 404–417. Springer.

[Belgiu et al., 2013] Belgiu, M., Lampoltshammer, T. J., Hofer, B., et al. (2013). *An extension of an ontology-based land cover designation approach for fuzzy rules*, volume 1. Verlag der Österreichischen Akademie der Wissenschaften.

[Benediktsson et al., 2005] Benediktsson, J. A., Palmason, J. A., and Sveinsson, J. R. (2005). Classification of hyperspectral data from urban areas based on extended morphological profiles. *IEEE Transactions on Geoscience and Remote Sensing*, 43(3):480–491.

[Bertrand, 1994] Bertrand, G. (1994). Simple points, topological numbers and geodesic neighborhoods in cubic grids. *Pattern recognition letters*, 15(10):1003–1011.

[Biesemans et al., 2007] Biesemans, J., Sterckx, S., Knaeps, E., Vreys, K., Adriaensen, S., Hooyberghs, J., Meuleman, K., Kempeneers, P., Deronde, B., Everaerts, J., et al. (2007). Image processing workflows for airborne remote sensing. In *Proceedings of the 5 th EARSeL Workshop on Imaging Spectroscopy, Bruges, Belgium*. Citeseer.

[Binaghi et al., 2003] Binaghi, E., Gallo, I., and Pepe, M. (2003). A cognitive pyramid for contextual classification of remote sensing images. *IEEE transactions on Geoscience and Remote Sensing*, 41(12):2906–2922.

[Blaschke, 2010] Blaschke, T. (2010). Object based image analysis for remote sensing. *ISPRS journal of photogrammetry and remote sensing*, 65(1):2–16.

[Bojinski et al., 2014] Bojinski, S., Verstraete, M., Peterson, T. C., Richter, C., Simmons, A., and Zemp, M. (2014). The concept of essential climate variables in support of climate research, applications, and policy. *Bulletin of the American Meteorological Society*, 95(9):1431–1443.

[Bossard et al., 2000] Bossard, M., Feranec, J., Otahel, J., et al. (2000). Corine land cover technical guide: Addendum 2000.

[Breiman, 1984] Breiman, L. (1984). *Classification and regression trees*. Routledge.

[Breiman, 2001] Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.

[Breiman et al., 2001] Breiman, L. et al. (2001). Statistical modeling: The two cultures. *Statistical science*, 16(3):199–231.

[Bruzzone and Carlin, 2006] Bruzzone, L. and Carlin, L. (2006). A multilevel context-based system for classification of very high spatial resolution images. *IEEE transactions on Geoscience and Remote Sensing*, 44(9):2587–2600.

[Cantelaube and Carles, 2014] Cantelaube, P. and Carles, M. (2014). Le registre parcellaire graphique: des données géographiques pour décrire la couverture du sol agricole. *Le Cahier des Techniques de l'INRA*, pages 58–64.

[Chai et al., 2017] Chai, D., Huang, Y., and Bao, Y. (2017). Irsl: Iterative refining superpixel lattice. *IEEE Geoscience and Remote Sensing Letters*, 14(3):344–348.

[Chang et al., 2013] Chang, J., Wei, D., and Fisher, J. W. (2013). A video representation using temporal superpixels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2051–2058.

[Chellapilla et al., 2006a] Chellapilla, K., Puri, S., and Simard, P. (2006a). High performance convolutional neural networks for document processing. In *Tenth International Workshop on Frontiers in Handwriting Recognition*. Suvisoft.

[Chellapilla et al., 2006b] Chellapilla, K., Shilman, M., and Simard, P. (2006b). Optimally combining a cascade of classifiers. In *Document Recognition and Retrieval XIII*, volume 6067, page 60670Q. International Society for Optics and Photonics.

[Chen and Guestrin, 2016] Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794. ACM.

[Chen et al., 2014] Chen, Y., Lin, Z., Zhao, X., Wang, G., and Gu, Y. (2014). Deep learning-based classification of hyperspectral data. *IEEE Journal of Selected topics in applied earth observations and remote sensing*, 7(6):2094–2107.

[Cheng et al., 2017] Cheng, G., Han, J., and Lu, X. (2017). Remote sensing image scene classification: Benchmark and state of the art. *Proceedings of the IEEE*, 105(10):1865–1883.

[Chevrel et al., 1981] Chevrel, M., Courtois, M., and Weill, G. (1981). The spot satellite remote sensing mission. *Photogrammetric Engineering and Remote Sensing*, 47:1163–1171.

[Cohen, 2005] Cohen, W. W. (2005). Stacked sequential learning. Technical report, CARNEGIE-MELLON UNIV PITTSBURGH PA SCHOOL OF COMPUTER SCIENCE.

[Comaniciu and Meer, 2002] Comaniciu, D. and Meer, P. (2002). Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 24(5):603–619.

[Comber et al., 2004] Comber, A., Fisher, P., and Wadsworth, R. (2004). Integrating land-cover data with different ontologies: identifying change from inconsistency. *International Journal of Geographical Information Science*, 18(7):691–708.

[Comber et al., 2005] Comber, A., Fisher, P., and Wadsworth, R. (2005). What is land cover? *Environment and Planning B: Planning and Design*, 32(2):199–209.

[Criminisi et al., 2012] Criminisi, A., Shotton, J., Konukoglu, E., et al. (2012). Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning. *Foundations and Trends® in Computer Graphics and Vision*, 7(2–3):81–227.

[Cui et al., 2017] Cui, Y., Chapel, L., and Lefèvre, S. (2017). Scalable bag of subpaths kernel for learning on hierarchical image representations and multi-source remote sensing data classification. *Remote Sensing*, 9(3):196.

[Dalal and Triggs, 2005] Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE.

[Dalla Mura et al., 2010] Dalla Mura, M., Villa, A., Benediktsson, J. A., Chanussot, J., and Bruzzone, L. (2010). Classification of hyperspectral images by using extended morphological attribute profiles and independent component analysis. *IEEE Geoscience and Remote Sensing Letters*, 8(3):542–546.

[Daubenmire, 1959] Daubenmire, R. F. (1959). Canopy coverage method of vegetation analysis. *Northwest Sci*, 33:39–64.

[Demarez et al., 2019] Demarez, V., Helen, F., Marais-Sicre, C., and Baup, F. (2019). In-season mapping of irrigated crops using landsat 8 and sentinel-1 time series. *Remote Sensing*, 11(2):118.

[Derksen et al., 2019a] Derksen, D., Inglada, J., and Michel, J. (2019a). A metric for evaluating the geometric quality of land cover maps generated with contextual features from high-dimensional satellite image time series without dense reference data. *Remote Sensing*, 11(16):1929.

[Derksen et al., 2019b] Derksen, D., Inglada, J., and Michel, J. (2019b). Scaling up slic superpixels using a tile-based approach. *IEEE Transactions on Geoscience and Remote Sensing*, PP:1–13.

[d'Oleire Oltmanns et al., 2014] d'Oleire Oltmanns, S., Marzolff, I., Tiede, D., and Blaschke, T. (2014). Detection of gully-affected areas by applying object-based image analysis (obia) in the region of taroudannt, morocco. *Remote Sensing*, 6(9):8287–8309.

[Drusch et al., 2012] Drusch, M., Del Bello, U., Carlier, S., Colin, O., Fernandez, V., Gascon, F., Hoersch, B., Isola, C., Laberinti, P., Martimort, P., et al. (2012). Sentinel-2: Esa's optical high-resolution mission for gmes operational services. *Remote sensing of Environment*, 120:25–36.

[Everingham et al., 2010] Everingham, M., Van Gool, L., Williams, C. K., Winn, J., and Zisserman, A. (2010). The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338.

[Fouilleux and Ansaloni, 2016] Fouilleux, E. and Ansaloni, M. (2016). *The common agricultural policy*. Oxford University Press Oxford.

[Friedman, 1997] Friedman, J. H. (1997). On bias, variance, 0/1-loss, and the curse-of-dimensionality. *Data mining and knowledge discovery*, 1(1):55–77.

[Fröhlich et al., 2013] Fröhlich, B., Bach, E., Walde, I., Hese, S., Schmullius, C., and Denzler, J. (2013). Land cover classification of satellite images using contextual information. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 3:W1.

[Fröhlich et al., 2012] Fröhlich, B., Rodner, E., and Denzler, J. (2012). Semantic segmentation with millions of features: Integrating multiple cues in a combined random forest approach. In *Asian conference on computer vision*, pages 218–231. Springer.

[Fukushima et al., 1983] Fukushima, K., Miyake, S., and Ito, T. (1983). Neocognitron: A neural network model for a mechanism of visual pattern recognition. *IEEE transactions on systems, man, and cybernetics*, (5):826–834.

[Gao, 1996] Gao, B.-C. (1996). Ndwi-a normalized difference water index for remote sensing of vegetation liquid water from space. *Remote sensing of environment*, 58(3):257–266.

[Gómez et al., 2016] Gómez, C., White, J. C., and Wulder, M. A. (2016). Optical remotely sensed time series data for land cover classification: A review. *ISPRS Journal of Photogrammetry and Remote Sensing*, 116:55–72.

[Goodchild, 1986] Goodchild, M. (1986). Spatial autocorrelation. concepts and techniques in modern geography 47. *Norwich, UK: Geo Books*.

[Griffiths et al., 2019] Griffiths, P., Nendel, C., and Hostert, P. (2019). Intra-annual reflectance composites from sentinel-2 and landsat for national-scale crop and land cover mapping. *Remote Sensing of Environment*, 220:135–151.

[Gupta and Hartley, 1997] Gupta, R. and Hartley, R. I. (1997). Linear pushbroom cameras. *IEEE Transactions on pattern analysis and machine intelligence*, 19(9):963–975.

[Hadria et al., 2009] Hadria, R., Duchemin, B., Baup, F., Le Toan, T., Bouvet, A., Dedieu, G., and Le Page, M. (2009). Combined use of optical and radar satellite data for the detection of tillage and irrigation operations: Case study in central morocco. *Agricultural water management*, 96(7):1120–1127.

[Haghighat et al., 2015] Haghighat, M., Zonouz, S., and Abdel-Mottaleb, M. (2015). Cloudid: Trustworthy cloud-based and cross-enterprise biometric identification. *Expert Systems with Applications*, 42(21):7905–7916.

[Hagolle et al., 2015] Hagolle, O., Sylvander, S., Huc, M., Claverie, M., Clesse, D., Dechoz, C., Lonjou, V., and Poulain, V. (2015). Spot-4 (take 5): Simulation of sentinel-2 time series on 45 large sites. *Remote sensing*, 7(9):12242–12264.

[Han et al., 2003] Han, X., Xu, C., and Prince, J. L. (2003). A topology preserving level set method for geometric deformable models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(6):755–768.

[Hänsch and Hellwich, 2018] Hänsch, R. and Hellwich, O. (2018). Classification of polsar images by stacked random forests. *ISPRS International Journal of Geo-Information*, 7(2):74.

[Happ et al., 2010] Happ, P., Ferreira, R. S., Bentes, C., Costa, G., and Feitosa, R. Q. (2010). Multiresolution segmentation: a parallel approach for high resolution image segmentation in multicore architectures. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 38(4):C7.

[Haralick, 1979] Haralick, R. M. (1979). Statistical and structural approaches to texture. *Proceedings of the IEEE*, 67(5):786–804.

[Harris and Browning, 2003] Harris, R. and Browning, R. (2003). Global monitoring for environment and security: Data policy considerations. *Space Policy*, 19(4):265–276.

[Hinton et al., 2012] Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*.

[Hochreiter and Schmidhuber, 1997] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

[Hoover et al., 1996] Hoover, A., Jean-Baptiste, G., Jiang, X., Flynn, P. J., Bunke, H., Goldgof, D. B., Bowyer, K., Eggert, D. W., Fitzgibbon, A., and Fisher, R. B. (1996). An experimental comparison of range image segmentation algorithms. *IEEE transactions on pattern analysis and machine intelligence*, 18(7):673–689.

[Hsu and Ding, 2013] Hsu, C.-Y. and Ding, J.-J. (2013). Efficient image segmentation algorithm using slic superpixels and boundary-focused region merging. In *Information, Communications and Signal Processing (ICICS) 2013 9th International Conference on*, pages 1–5. IEEE.

[Hu et al., 2015] Hu, F., Xia, G.-S., Hu, J., and Zhang, L. (2015). Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery. *Remote Sensing*, 7(11):14680–14707.

[Huang et al., 2007] Huang, X., Zhang, L., and Li, P. (2007). Classification and extraction of spatial features in urban areas using high-resolution multispectral imagery. *IEEE Geoscience and Remote Sensing Letters*, 4(2):260–264.

[Huang et al., 2008] Huang, X., Zhang, L., and Li, P. (2008). A multiscale feature fusion approach for classification of very high resolution satellite imagery based on wavelet transform. *International Journal of Remote Sensing*, 29(20):5923–5941.

[Huynh et al., 2016] Huynh, T., Gao, Y., Kang, J., Wang, L., Zhang, P., Lian, J., and Shen, D. (2016). Estimating ct image from mri data using structured random forest and auto-context model. *IEEE transactions on medical imaging*, 35(1):174.

[Ienco et al., 2017] Ienco, D., Gaetano, R., Dupaquier, C., and Maurel, P. (2017). Land cover classification via multitemporal spatial data by deep recurrent neural networks. *IEEE Geoscience and Remote Sensing Letters*, 14(10):1685–1689.

[Inglada et al., 2015] Inglada, J., Arias, M., Tardy, B., Hagolle, O., Valero, S., Morin, D., Dedieu, G., Sepulcre, G., Bontemps, S., Defourny, P., et al. (2015). Assessment of an operational system for crop type map production using high temporal and spatial resolution satellite optical imagery. *Remote Sensing*, 7(9):12356–12379.

[Inglada and Michel, 2009] Inglada, J. and Michel, J. (2009). Qualitative spatial reasoning for high-resolution remote sensing image analysis. *IEEE Transactions on Geoscience and Remote Sensing*, 47(2):599–612.

[Inglada et al., 2016] Inglada, J., Vincent, A., Arias, M., and Marais-Sicre, C. (2016). Improved early crop type identification by joint use of high temporal resolution sar and optical image time series. *Remote Sensing*, 8(5):362.

[Inglada et al., 2017] Inglada, J., Vincent, A., Arias, M., Tardy, B., Morin, D., and Rodes, I. (2017). Operational high resolution land cover map production at the country scale using satellite image time series. *Remote Sensing*, 9(1):95.

[Jampani et al., 2015] Jampani, V., Gadde, R., and Gehler, P. V. (2015). Efficient facade segmentation using auto-context. In *Applications of Computer Vision (WACV), 2015 IEEE Winter Conference on*, pages 1038–1045. IEEE.

[Jia et al., 2017] Jia, S., Hu, J., Zhu, J., Jia, X., and Li, Q. (2017). Three-dimensional local binary patterns for hyperspectral imagery classification. *IEEE Transactions on Geoscience and Remote Sensing*, 55(4):2399–2413.

[Jiang and Tu, 2009] Jiang, J. and Tu, Z. (2009). Efficient scale space auto-context for image segmentation and labeling. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1810–1817. IEEE.

[Joly et al., 2010] Joly, D., Brossard, T., Cardot, H., Cavailhes, J., Hilal, M., and Wavresky, P. (2010). Les types de climats en france, une construction spatiale. *Cybergeo: European Journal of Geography*.

[Joyce et al., 2009] Joyce, K. E., Belliss, S. E., Samsonov, S. V., McNeill, S. J., and Glassey, P. J. (2009). A review of the status of satellite remote sensing and image processing techniques for mapping natural hazards and disasters. *Progress in Physical Geography*, 33(2):183–207.

[Kampffmeyer et al., 2016] Kampffmeyer, M., Salberg, A.-B., and Jenssen, R. (2016). 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*, pages 1–9.

[Kang et al., 2014] Kang, X., Li, S., and Benediktsson, J. A. (2014). Spectral–spatial hyperspectral image classification with edge-preserving filtering. *IEEE transactions on geoscience and remote sensing*, 52(5):2666–2677.

[Kim et al., 2011] Kim, M., Warner, T. A., Madden, M., and Atkinson, D. S. (2011). Multi-scale geobia with very high spatial resolution digital aerial imagery: scale, texture and image objects. *International Journal of Remote Sensing*, 32(10):2825–2850.

[Korting et al., 2011] Korting, T. S., Castejon, E. F., and Fonseca, L. M. G. (2011). Divide and segment-an alternative for parallel segmentation. In *GeoInfo*, pages 97–104.

[Krizhevsky et al., 2012] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

[Kuffer et al., 2016] Kuffer, M., Pfeffer, K., and Sliuzas, R. (2016). Slums from space-15 years of slum mapping using remote sensing. *Remote Sensing*, 8(6):455.

[Kussul et al., 2017] Kussul, N., Lavreniuk, M., Skakun, S., and Shelestov, A. (2017). Deep learning classification of land cover and crop types using remote sensing data. *IEEE Geoscience and Remote Sensing Letters*, 14(5):778–782.

[Larios et al., 2011] Larios, N., Lin, J., Zhang, M., Lytle, D., Moldenke, A., Shapiro, L., and Dietterich, T. (2011). Stacked spatial-pyramid kernel: An object-class recognition method to combine scores from random trees. In *Applications of Computer Vision (WACV), 2011 IEEE Workshop on*, pages 329–335. IEEE.

[Lassalle et al., 2015] Lassalle, P., Inglada, J., Michel, J., Grizonnet, M., and Malik, J. (2015). A scalable tile-based framework for region-merging segmentation. *IEEE Transactions on Geoscience and Remote Sensing*, 53(10):5473–5485.

[Lepetit and Fua, 2006] Lepetit, V. and Fua, P. (2006). Keypoint recognition using randomized trees. *IEEE transactions on pattern analysis and machine intelligence*, 28(9):1465–1479.

[Leroy et al., 2013] Leroy, M., Kosuth, P., Hagolle, O., Cherchali, S., Maurel, P., and Desconnets, J. (2013). Theia land data center. In *ESA Living Planet Symposium. Edimburgh, UK. 9-13 Septembre 2013*.

[Liang et al., 2017] Liang, J., Zhou, J., Qian, Y., Wen, L., Bai, X., and Gao, Y. (2017). On the sampling strategy for evaluation of spectral-spatial methods in hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 55(2):862–880.

[Lim, 1990] Lim, J. S. (1990). Two-dimensional signal and image processing. *Englewood Cliffs, NJ, Prentice Hall, 1990, 710 p.*

[Liu et al., 2014] Liu, G., Xia, G.-S., Yang, W., and Zhang, L. (2014). Texture analysis with shape co-occurrence patterns. In *2014 22nd International Conference on Pattern Recognition*, pages 1627–1632. IEEE.

[Lowe, 1999] Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee.

[Maggiori et al., 2017] Maggiori, E., Tarabalka, Y., Charpiat, G., and Alliez, P. (2017). Convolutional neural networks for large-scale remote-sensing image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 55(2):645–657.

[Markham and Helder, 2012] Markham, B. L. and Helder, D. L. (2012). Forty-year calibrated record of earth-reflected radiance from landsat: A review. *Remote Sensing of Environment*, 122:30–40.

[Marmanis et al., 2016] Marmanis, D., Datcu, M., Esch, T., and Stilla, U. (2016). Deep learning earth observation classification using imagenet pretrained networks. *IEEE Geoscience and Remote Sensing Letters*, 13(1):105–109.

[Marmanis et al., 2018] Marmanis, D., Schindler, K., Wegner, J. D., Galliani, S., Datcu, M., and Stilla, U. (2018). Classification with an edge: Improving semantic image segmentation with boundary detection. *ISPRS Journal of Photogrammetry and Remote Sensing*, 135:158–172.

[Maugeais et al., 2011] Maugeais, E., Lecordix, F., Halbecq, X., and Braun, A. (2011). Dérivation cartographique multi échelles de la bdtopo de l'ign france: mise en œuvre du processus de production de la nouvelle carte de base. In *Proceedings of the 25th international cartographic conference, Paris*, pages 3–8.

[McCulloch and Pitts, 1943] McCulloch, W. S. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133.

[Melgani and Serpico, 2003] Melgani, F. and Serpico, S. B. (2003). A markov random field approach to spatio-temporal contextual image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 41(11):2478–2487.

[Michel et al., 2012] Michel, J., Grizonnet, M., Jaen, A., Harasse, S., Hermitte, L., Guinet, J., Malik, J., and Savinaud, M. (2012). Open tools and methods for large scale segmentation of very high resolution satellite images. *Proc. OGRS*, pages 179–184.

[Michel et al., 2015] Michel, J., Youssefi, D., and Grizonnet, M. (2015). Stable mean-shift algorithm and its application to the segmentation of arbitrarily large remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 53(2):952–964.

[Möller et al., 2014] Möller, M., Birger, J., and Cornelia, G. (2014). Geometric accuracy assessment of classified land use/land cover changes. *Photogrammetrie - Fernerkundung - Geoinformation*, 2014/2:91–99.

[Mondini, 2017] Mondini, A. (2017). Measures of spatial autocorrelation changes in multitemporal sar images for event landslides detection. *Remote Sensing*, 9(6):554.

[Montero et al., 2014] Montero, E., Van Wolvelaer, J., and Garzón, A. (2014). The european urban atlas. In *Land use and land cover mapping in Europe*, pages 115–124. Springer.

[Moore et al., 2008] Moore, A. P., Prince, S. J., Warrell, J., Mohammed, U., and Jones, G. (2008). Superpixel lattices. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE.

[Moser and Serpico, 2013] Moser, G. and Serpico, S. B. (2013). Combining support vector machines and markov random fields in an integrated framework for contextual image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 51(5):2734–2752.

[Munoz et al., 2010] Munoz, D., Bagnell, J. A., and Hebert, M. (2010). Stacked hierarchical labeling. In *European Conference on Computer Vision*, pages 57–70. Springer.

[Nethercote and Seward, 2007] Nethercote, N. and Seward, J. (2007). Valgrind: a framework for heavyweight dynamic binary instrumentation. In *ACM Sigplan notices*, volume 42, pages 89–100. ACM.

[Nicolai-Shaw et al., 2017] Nicolai-Shaw, N., Zscheischler, J., Hirschi, M., Gudmundsson, L., and Seneviratne, S. I. (2017). A drought event composite analysis using satellite remote-sensing based soil moisture. *Remote Sensing of Environment*, 203:216–225.

[Pelletier et al., 2017] Pelletier, C., Valero, S., Inglada, J., Champion, N., Marais Sicre, C., and Dedieu, G. (2017). Effect of training class label noise on classification performances for land cover mapping with satellite image time series. *Remote Sensing*, 9(2):173.

[Penatti et al., 2015] Penatti, O. A., Nogueira, K., and dos Santos, J. A. (2015). 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*, pages 44–51.

[Pfeffer et al., 2014] Pfeffer, W. T., Arendt, A. A., Bliss, A., Bolch, T., Cogley, J. G., Gardner, A. S., Hagen, J.-O., Hock, R., Kaser, G., Kienholz, C., et al. (2014). The randolph glacier inventory: a globally complete inventory of glaciers. *Journal of glaciology.*, 60(221):537–552.

[Pham et al., 2016] Pham, M.-T., Mercier, G., and Michel, J. (2016). Pw-cog: An effective texture descriptor for vhr satellite imagery using a pointwise approach on covariance matrix of oriented gradients. *IEEE Transactions on Geoscience and Remote Sensing*, 54(6):3345–3359.

[Pieczynski, 1989] Pieczynski, W. (1989). Estimation of context in random fields. *Journal of Applied Statistics*, 16(2):283–290.

[Postadjian et al., 2017] Postadjian, T., Le Bris, A., Sahbi, H., and Mallet, C. (2017). Investigating the potential of deep neural networks for large-scale classification of very high resolution satellite images. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, pages 183–190.

[Ragheb et al., 2016] Ragheb, G., El-Shimy, H., and Ragheb, A. (2016). Land for poor: towards sustainable master plan for sensitive redevelopment of slums. *Procedia-Social and Behavioral Sciences*, 216:417–427.

[Revermann et al., 2016] Revermann, R., Finckh, M., Stellmes, M., Strohbach, B., Frantz, D., and Oldeland, J. (2016). Linking land surface phenology and vegetation-plot databases to model terrestrial plant α-diversity of the okavango basin. *Remote Sensing*, 8(5):370.

[Richmond et al., 2015] Richmond, D. L., Kainmueller, D., Yang, M. Y., Myers, E. W., and Rother, C. (2015). Mapping auto-context decision forests to deep convnets for semantic segmentation. *arXiv preprint arXiv:1507.07583*.

[Rokach, 2009] Rokach, L. (2009). Taxonomy for characterizing ensemble methods in classification tasks: A review and annotated bibliography. *Computational Statistics & Data Analysis*, 53(12):4046–4072.

[Ronneberger et al., 2015] Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer.

[Rosenblatt, 1958] Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386.

[Russell et al., 2006] Russell, B. C., Freeman, W. T., Efros, A. A., Sivic, J., and Zisserman, A. (2006). Using multiple segmentations to discover objects and their extent in image collections. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 1605–1614. IEEE.

[Schindler, 2012] Schindler, K. (2012). An overview and comparison of smooth labeling methods for land-cover classification. *IEEE transactions on geoscience and remote sensing*, 50(11):4534–4545.

[Schwalm et al., 2017] Schwalm, C. R., Anderegg, W. R., Michalak, A. M., Fisher, J. B., Biondi, F., Koch, G., Litvak, M., Ogle, K., Shaw, J. D., Wolf, A., et al. (2017). Global patterns of drought recovery. *Nature*, 548(7666):202.

[Shapovalov et al., 2013] Shapovalov, R., Vetrov, D., and Kohli, P. (2013). Spatial inference machines. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2985–2992.

[Shotton et al., 2008] Shotton, J., Johnson, M., and Cipolla, R. (2008). Semantic texton forests for image categorization and segmentation. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE.

[Simard et al., 2003] Simard, P. Y., Steinkraus, D., and Platt, J. C. (2003). Best practices for convolutional neural networks applied to visual document analysis. In *null*, page 958. IEEE.

[Stoian et al., 2019] Stoian, A., Poulain, V., Inglada, J., Poughon, V., and Derksen, D. (2019). Land cover maps production with high resolution satellite image time series and convolutional neural networks: Adaptations and limits for operational systems. *MDPI Remote Sensing*.

[Tardy et al., 2017] Tardy, B., Inglada, J., and Michel, J. (2017). Fusion approaches for land cover map production using high resolution image time series without reference data of the corresponding period. *Remote Sensing*, 9(11):1151.

[Tatem et al., 2008] Tatem, A. J., Goetz, S. J., and Hay, S. I. (2008). Fifty years of earth observation satellites: Views from above have lead to countless advances on the ground in both scientific knowledge and daily life. *American Scientist*, 96(5):390.

[Teke et al., 2013] Teke, M., Deveci, H. S., Haliloğlu, O., Gürbüz, S. Z., and Sakarya, U. (2013). A short survey of hyperspectral remote sensing applications in agriculture. In *2013 6th International Conference on Recent Advances in Space Technologies (RAST)*, pages 171–176. IEEE.

[Thanh Noi and Kappas, 2018] Thanh Noi, P. and Kappas, M. (2018). Comparison of random forest, k-nearest neighbor, and support vector machine classifiers for land cover classification using sentinel-2 imagery. *Sensors*, 18(1):18.

[Toureiro et al., 2017] Toureiro, C., Serralheiro, R., Shahidian, S., and Sousa, A. (2017). Irrigation management with remote sensing: Evaluating irrigation requirement for maize under mediterranean climate condition. *Agricultural Water Management*, 184:211–220.

[Trias Sanz, 2006] Trias Sanz, R. (2006). Semi-automatic rural land cover classification (phd thesis).

[Tu, 2008] Tu, Z. (2008). Auto-context and its application to high-level vision tasks. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE.

[Tu and Bai, 2010] Tu, Z. and Bai, X. (2010). Auto-context and its application to high-level vision tasks and 3d brain image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(10):1744–1757.

[Turner, 1986] Turner, M. R. (1986). Texture discrimination by gabor functions. *Biological cybernetics*, 55(2-3):71–82.

[Turner et al., 2003] Turner, W., Spector, S., Gardiner, N., Fladeland, M., Sterling, E., and Steininger, M. (2003). Remote sensing for biodiversity science and conservation. *Trends in ecology & evolution*, 18(6):306–314.

[United Nations Human Settlement Programme, 2012] United Nations Human Settlement Programme, U. N. (2012). State of the world's cities report 2012/2013: prosperity of cities.

[Van der Werff and Van der Meer, 2008] Van der Werff, H. and Van der Meer, F. (2008). Shape-based classification of spectrally identical objects. *ISPRS Journal of Photogrammetry and Remote Sensing*, 63(2):251–258.

[Van Westen, 2013] Van Westen, C. J. (2013). Remote sensing and gis for natural hazards assessment and disaster risk management. *Treatise on geomorphology*, 3:259–298.

[Vincent and Soille, 1991] Vincent, L. and Soille, P. (1991). Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (6):583–598.

[Von Gioi et al., 2012] Von Gioi, R. G., Jakubowicz, J., Morel, J.-M., and Randall, G. (2012). Lsd: a line segment detector. *Image Processing On Line*, 2:35–55.

[Walker and Blaschke, 2008] Walker, J. and Blaschke, T. (2008). Object-based land-cover classification for the phoenix metropolitan area: Optimization vs. transportability. *International Journal of Remote Sensing*, 29(7):2021–2040.

[Wulder and Boots, 1998] Wulder, M. and Boots, B. (1998). Local spatial autocorrelation characteristics of remotely sensed imagery assessed with the getis statistic. *International Journal of Remote Sensing*, 19(11):2223–2231.

[Xie and Tu, 2015] Xie, S. and Tu, Z. (2015). Holistically-nested edge detection. In *Proceedings of the IEEE international conference on computer vision*, pages 1395–1403.

[Yang and Newsam, 2008] Yang, Y. and Newsam, S. (2008). Comparing sift descriptors and gabor texture features for classification of remote sensed imagery. In *2008 15th IEEE international conference on image processing*, pages 1852–1855. IEEE.

[Yang and Newsam, 2010] Yang, Y. and Newsam, S. (2010). 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, pages 270–279, New York, NY, USA. ACM.

[Yang, 2011] Yang, Z. (2011). Tiling and merging framework for segmenting large images. US Patent 8,086,037.

[Yu et al., 2016] Yu, H., Yang, W., Xia, G.-S., and Liu, G. (2016). A color-texture-structure descriptor for high-resolution satellite image classification. *Remote Sensing*, 8(3):259.

[Yu et al., 2018] Yu, L., Su, J., Li, C., Wang, L., Luo, Z., and Yan, B. (2018). Improvement of moderate resolution land use and land cover classification by introducing adjacent region features. *Remote Sensing*, 10(3).

[Zhang et al., 2008] Zhang, H., Fritts, J. E., and Goldman, S. A. (2008). Image segmentation evaluation: A survey of unsupervised methods. *computer vision and image understanding*, 110(2):260–280.

[Zhang et al., 2016] Zhang, X., Sun, Y., Shang, K., Zhang, L., and Wang, S. (2016). Crop classification based on feature band set construction and object-oriented approach using hyperspectral images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 9(9):4117–4128.

[Zhao et al., 2016] Zhao, J., Zhong, Y., Shu, H., and Zhang, L. (2016). High-resolution image classification integrating spectral-spatial-location cues by conditional random fields. *IEEE Trans. Image Processing*, 25(9):4033–4045.

List of Figures

List of Tables

Abstract

This work studies the application of supervised classification for the production of land cover maps using time series of satellite images at high spatial, spectral, and temporal resolutions. On this problem, certain classes such as urban cover, depend more on the context of the pixel than its content. The issue of this Ph.D. work is therefore to take into account the neighborhood of the pixel, to improve the recognition rates of these classes. This research first leads to question the definition of the context, and to imagine different possible shapes for it. Then comes describing the context, that is to say to create a representation or a model that allows the target classes to be recognized. The combinations of these two aspects are evaluated on two experimental data sets, one on Sentinel-2 images, and the other on SPOT-7 images.

Abstract

Ce travail étudie l'application de la classification supervisée pour la production de cartes d'occupation des sols à partir de séries temporelles d'images satellitaires à haute résolution spatiale, spectrale, et temporelle. Sur ce problème, certaines classes, par exemple, les classes urbaines, dépendent plus du contexte des pixels que de leur contenu. L'enjeu de la thèse est la prise en compte du voisinage du pixel, pour améliorer la précision de ces classes. Cette recherche nous mène dans un premier temps à questionner la définition du voisinage, et à imaginer différentes formes. Ensuite, il s'agit de décrire le voisinage, c'est à dire de créer une représentation ou un modèle qui permette de reconnaître les classes ciblées. Les combinaisons de ces deux aspects sont évaluées sur deux jeux de données expérimentales, un sur de l'imagerie Sentinel-2, et un sur une image SPOT-7.