INSTITUT POLYTECHNIQUE DE PARIS

NNT : 2020IPPAT033

Thèse de doctorat

TELECOM Paris

IP PARIS

Convolutional Neural Networks for Change Analysis in Earth Observation Images with Noisy Labels and Domain Shifts

Thèse de doctorat de l'Institut Polytechnique de Paris
préparée à Télécom Paris

École doctorale n°626 Institut Polytechnique de Paris (ED IP Paris)
Spécialité de doctorat : Informatique, données, intelligence artificielle

Thèse présentée et soutenue à Palaiseau, le 06/11/2020, par

RODRIGO CAYE DAUDT

Composition du Jury :

Florence Tupin Professor, Télécom Paris (LTCI)	Présidente
Begüm Demir Professor, Technische Universität Berlin (EECS)	Rapportrice
Friedrich Fraundorfer Associate Professor, Technische Universität Graz (ICG)	Rapporteur
Guillaume Charpiat Researcher, INRIA Saclay (TAU)	Examinateur
Maria Vakalopoulou Assistant Professor, CentraleSupélec (CVN)	Examinatrice
Yann Gousseau Professor, Télécom Paris (LTCI)	Directeur de Thèse
Bertrand Le Saux Researcher, European Space Agency (ESRIN)	Encadrant
Alexandre Boulch Researcher, Valeo (Valeo.ai)	Invité

For Dani.

Acknowledgements

I would first like to thank my thesis advisers: Yann Gousseau, Bertrand Le Saux, and Alexandre Boulch. You put your trust in me for this project despite never having worked with me or even met me personally. Choosing to give me this opportunity has changed the course of my life. The feedback, criticism, and help when I got stuck were also essential to the development of my work, which would not be as good without your guidance. And last but not least, thank you for all the discussions we've had about not only our field of work, but also about many other topics. I have learned much from you.

I would also like to thank all members of the jury who took the time to evaluate my work. First, Devis Tuia and Guillaume Charpiat, who first took part in my mid-thesis evaluation during my second year, on which occasion they gave me valuable feedback, and later also agreed to take part in my final thesis defence, although Mr. Tuia was not able to attend the defence due to unforeseen circumstances. Then, I would like to thank Friedrich Fraundorfer and Begüm Demir, who agreed to act as *rapporteurs*, and whose criticism of my work has shown me how to improve. Finally, I would like to thank Maria Vakalopoulou and Florence Tupin, who have also agreed to participate in my thesis defence and evaluate my contributions to our field of study.

It is also important to acknowledge those who have guided me in the past. Thank you not only to past professors who shared their expertise with me, but also to Anesio de Leles Ferreira Filho, Andrew McPherson, Victor Zappi, José Edil Guimarães de Medeiros, Heider Marconi Madureira, Christine Guillemot, Yves Wiaux, and Yoann Altmann, who advised me in various projects and internships in the past. Thank you also to Maite, who first introduced me to the field of remote sensing, even if in a much more casual way.

Thank you to all my colleagues who have welcomed me into the IVA research group and whom I now call friends. Thank you to Marcela, Rodolphe, Pierre, Soufiane, Maxime, Nicolas, Javiera, Guillaume[2], Benjamin, Alexis, Rémy, Gaston, Louis, Simon, Laurane, Anthelme (who I consider a honorary *doctorant*), and all the others who are too many to name. Thank you for sharing with me many climbing injuries, logic puzzles, break room discussions, and often just a cup of tea or coffee while we got ready to go find more bugs in our codes. Thank you also for your patience with me while I slowly learned your language, and my numerous questions while I did so. Despite my frequent criticisms of some of its *n'importe quoi* rules, I appreciate it more than you think.

I am also thankful to ONERA, who has funded this thesis and all the conferences which I have attended, as well

as the staff that helped me during these years. Thank you also to Télécom Paris, especially the IMAGES group. Although I did not spend much time there, I have always felt welcome when I did.

I can also say with certainty that I would not have accomplished what I have so far without the support of my family, mainly from my parents Geraldo and Liliana, but also from my brother Gilberto and my grandparents. You may not have taught me computer vision or machine learning or remote sensing, but you have taught me discipline and hard work, and you have been by my metaphorical side, even if physically distant during most of this journey.

Finally, thank you, Dani, for going through these years with me. It was not always easy, especially at the end, but having you with me made everything better. Hardships are not as bad when you are there to help me, and what would even be the point of my successes if I could not share them, especially with you.

Contents

List of Figures

List of Tables

Résumé

La télédétection est un moyen puissant d'observer les grandes zones avec une grande efficacité. En utilisant des images prises de très haut, comme celles des avions ou des satellites, il est possible de mesurer avec précision plusieurs caractéristiques de régions entières à distance. Les techniques d'imagerie peuvent varier considérablement en fonction des applications souhaitées. Les types d'imagerie passive comprennent les images panchromatiques, RVB (rouge, vert et bleu), multispectrales et hyperspectrales. Ces méthodes d'acquisition consistent à mesurer l'intensité des ondes électromagnétiques émises par l'objet ou la zone observée. Les méthodes d'imagerie actives, telles que le SAR (*synthetic aperture radar*) et le LIDAR (*light detection and ranging*), émettent des ondes électromagnétiques dans une région cible et observent la réponse réfléchie.

Récemment, les réseaux de neurones se sont révélés extrêmement puissants pour résoudre plusieurs problèmes, notamment dans le domaine de l'analyse d'images. La télédétection ne fait pas exception à la règle, et l'application de méthodes basées sur les réseaux de neurones pour extraire des informations significatives des images d'observation de la Terre est actuellement un sujet très recherché.

Les travaux présentés dans cette thèse tournent autour d'un point central: la détéction de changements à partir de paires d'images de télédétection en utilisant des réseaux de neurones convolutifs. Cette question est abordée principalement sous quatre points de vue:

1. **Modalités des données**: y a-t-il des avantages à utiliser des images multispectrales plutôt que des images RVB pour la détection de changements?

2. **Sémantique des données**: comment pouvons-nous extraire des informations sémantiques du paire d'images pour mieux comprendre l'évolution de la zone imagée?

3. **Disponibilité des données**: comment réduire l'effet du bruit des étiquettes et surmonter la rareté des données lors de l'entraînement des réseaux neuronaux convolutifs?

4. **Hétérogénéité des données**: comment pouvons-nous créer des systèmes de détection de changements qui soient robustes dans leur généralisation à des événements jusqu'alors inconnus?

L'intérêt pour la conception de systèmes de détection automatique de changements provient de deux cas prin-

cipaux: les applications à grande échelle et la réponse rapide aux événements. Dans ces deux cas, la vitesse d'analyse manuelle des images devient un problème. Par exemple, lors des incendies de forêt en Australie en 2019, illustrés dans la Fig. 1.3, des images de télédétection ont été utilisées pour obtenir des informations actualisées sur la situation actuelle [Law20]. L'imagerie satellitaire offre souvent un moyen rapide et précis d'obtenir des informations précises sur de vastes zones. C'est particulièrement utile pour l'observation des zones d'accès difficile, comme les hautes montagnes, les forêts profondes et les régions polaires.

L'application de la vision par ordinateur et des méthodes d'apprentissage automatique à l'analyse des images de télédétection multitemporelle a une longue histoire de recherche. Le chapitre 2 contient un résumé des travaux qui sont directement liées aux travaux présentés dans cette thèse, ainsi qu'une base théorique nécessaire à leur compréhension. Ce récapitulatif théorique devrait fournir au lecteur les connaissances minimales nécessaires pour comprendre les idées présentées dans les chapitres suivants.

Les travaux originaux qui ont été menées au cours de cette thèse sont présentées dans les chapitres 3, 4, 5 et 6. Le chapitre 3 décrit comment les réseaux de neurones convolutifs (CNN) et les réseaux neuronaux entièrement convolutifs (FCN ou FCNN) peuvent être utilisés pour effectuer la détection de changements comme un problème de segmentation sémantique de manière entièrement supervisée. Les architectures de réseaux de neurones convolutifs et de réseaux de neurones entièrement convolutifs sont entraînés "*from scratch*" en utilisant un nouvel ensemble de données de détection de changements. L'impact des variations d'architecture est étudié, ainsi que la différence de performance entre les CNN et les FCN. Le chapitre 3 contient également des comparaisons entre l'utilisation des réseaux proposés avec différentes combinaisons de bandes d'images multispectrales, et les résultats obtenus montrent qu'il est intéressant d'utiliser des bandes en dehors du spectre visible pour la détection des changements lorsqu'elles sont disponibles.

Le chapitre 4 explore l'ajout d'informations sémantiques dans le cadre de détection de changements présenté précédemment, et comment l'apprentissage multitâche affecte les performances des réseaux. Un nouvel ensemble de données sémantiques à haute résolution et à grande échelle est créé pour réaliser ces expériences. Différentes approches de ce problème sont analysées, allant de la comparaison des cartes de la couverture terrestre prédite pour la détection de changements jusqu'à un FCN entièrement intégré pour effectuer simultanément la détection de changements et la cartographie de la couverture terrestre des images d'entrée. Un schéma d'entraînement par étapes est proposé pour le réseau multitâche intégré qui réduit le nombre d'hyperparamètres à choisir et qui améliore les performances du réseau.

Le chapitre 5 étudie l'impact du bruit des étiquettes sur le processus d'apprentissage pour la détection de changements, et emprunte des idées à l'apprentissage faiblement supervisé pour améliorer la précision des résultats des réseaux. Le biais des étiquettes dans l'ensemble de données présenté précédemment et généré automatiquement a un effet de biais sur les cartes de changement prévues par les réseaux. Un processus itératif d'entraînement et de nettoyage des étiquettes est utilisé pour augmenter la robustesse du réseau au bruit des étiquettes, et il est

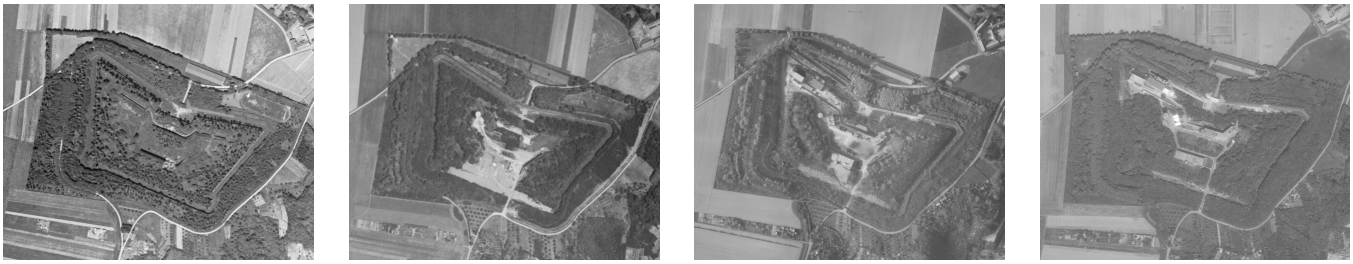
couplé à un nouvel algorithme de filtrage anisotrope guidé qui permet de mieux adapter la prédiction aux bords des images d'entrée. Le chapitre 5 explore également comment la détection des changements au niveau des pixels peut être effectuée en utilisant uniquement des étiquettes au niveau de l'image comme supervision de façon faiblement supervisé. Une couche d'attention spatiale fixe est apprise pendant l'entraînement pour un réseau de classification, et la synergie entre cette couche d'attention et le classifieur du réseau améliore la capacité du réseau à localiser les changements. Ces résultats sont également améliorés par l'utilisation de l'algorithme de diffusion anisotrope guidée.

Le chapitre 6 décrit une approche d'apprentissage antagoniste pour réduire le différences entre les images de différents types de changements dans un espace latent, en utilisant divers types de catastrophes naturelles comme études de cas. Les différentes catastrophes naturelles peuvent avoir des apparences visuelles très différentes dans les images de télédétection, de sorte que les réseaux neuronaux entraînés pour détecter les changements causés par un type d'événement ne se généralisent pas bien aux autres catastrophes. L'apprentissage antagoniste est utilisé pour projeter les images dans un espace latent commun où la supervision de la détection de changements provient d'un domaine source, et peut être appliqué aux images du domaine cible. La méthode proposée est également testée sur d'autres tâches d'adaptation de domaine en vision par ordinateur pour montrer sa polyvalence.

Le chapitre 7 conclut ce manuscrit avec les principales conclusions tirées des travaux présentées. Il contient également des perspectives d'études futures qui pourraient être menées pour poursuivre cette ligne de recherche.

Chapter 1

Introduction



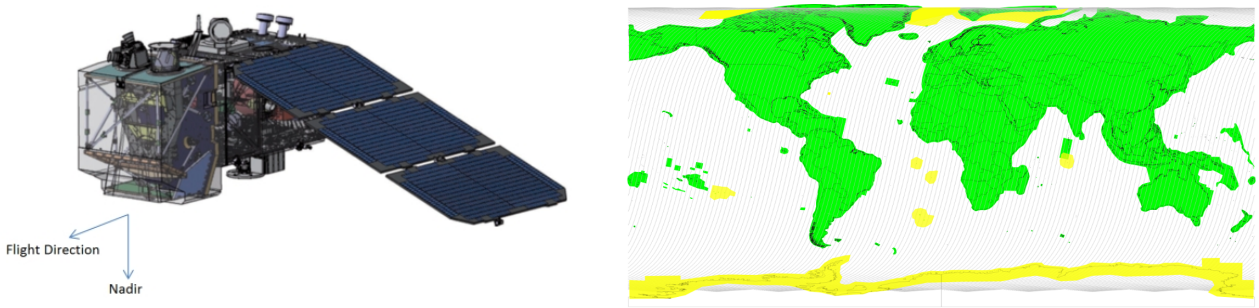
(a) 1937 (b) 1949 (c) 1958 (d) 1961

(e) 1972 (f) 1987 (g) 1999 (h) 2011

Figure 1.1: Aerial images throughout the years of the Fort de Palaiseau, which became a research center for ONERA in 1947. Images courtesy of IGN's BD Historique.

1.1 Context

Eratosthenes, a Greek polymath, was the first person known to have accurately calculated the circumference of the Earth over two thousand years ago [R+13]. By measuring the angle of the shadows cast by the Sun at noon on the summer solstice in Alexandria and Syene, the distance between these two cities could be extrapolated to obtain a surprisingly accurate estimate of Earth's size. Much effort has been spent before and since to understand and catalog our planet. Such understanding of our environments, both natural and human-made, is necessary for



(a) Sentinel-2 satellite (b) Coverage and revisit rate

Figure 1.2: The twin Sentinel-2 satellites regularly image all of Earth's main landmasses. In (b), regions marked in green are imaged every 5 days, and regions marked in yellow are imaged every 10 days (as of October 2019). Images courtesy of `sentinel.esa.int`.

well-informed planning and decision-making in various situations. Urban planning, warfare, forestry, agriculture, and many other sciences benefit from having information about the current state of the world, its past state, and how it changes.

Remote sensing is a powerful way of observing the large areas with high efficiency. Using images taken from high above, such as from airplanes or satellites, it is possible to accurately measure several characteristics from entire regions from a distance. Imaging techniques can vary significantly, depending on the desired applications. Passive imaging types include panchromatic, RGB (red, green and blue), multispectral, and hyperspectral. These acquisition methods consist of measuring the intensity of electromagnetic waves emitted by the observed object or area. Active imaging methods, such as SAR (synthetic aperture radar) and LIDAR (light detection and ranging), emit electromagnetic waves at a target region and observe the reflected response.

Several improvements in imaging technology over the years led to an increase in the quality of the captured images. Figure 1.1 shows images of the Palaiseau Fort acquired during aerial surveys over many decades. Such images were initially obtained by film cameras and in greyscale. Nowadays it is possible to acquire digital photos with color information and at higher spatial resolutions. Other technological improvements, such as better atmospheric correction and registration techniques, further increased our capacity to capture ground-level information from a distance.

The launch of Sputnik 1, the first artificial Earth satellite, marked the beginning of the Space Race in 1957, which has completely changed the field of remote sensing. Before the 1960s, remote sensing of the environment was done predominantly by aerial photography. Nowadays, satellite imaging is widely used to observe the earth. These can monitor atmospheric conditions (e.g. for weather forecasting), as well as ground-level objects. Programs such as Landsat, SPOT, and Sentinel are able to automatically image the Earth at regular intervals. The pair of Sentinel-2 satellites, for example, are able to image all of Earth's main landmasses at intervals of 5-10 days (see Fig. 1.2).

These Earth observation satellites produce extremely large amounts of data. Each of the Sentinel-2 satellites, for example, produces 1.6 TB of multispectral imaging data per orbit, with an orbital period of 100.6 minutes [ESA20a],

resulting in approximately 45.8 TB of data per day. Such large amounts of data are the main motivators for research into automatic Earth observation image analysis. These images contain large amounts of useful information about the Earth and its evolution through time, but the extraction of useful information from remote sensing imagery is a non-trivial problem.

The interpretation of remote sensing images is useful for many applications [Aud18]. Accurate interpretation of such images is extremely important in several impactful areas of study: weather forecasting, climate change, deforestation, monitoring of glaciers, natural disasters, urbanism, agriculture, etc. But the high level or heterogeneity of Earth observation data poses many challenges for its automatic interpretation. Dealing with such large scale and heterogeneous data is a central point of this thesis.

Recently, neural networks have been proven to be extremely powerful in solving several problems, especially in the field of image analysis. Remote sensing is no exception, and the application of neural network based methods to extract meaningful information from Earth observation imagery is currently an extensively researched topic.

1.2 Domain

This thesis is located at the intersection of three scientific domains, with different associated practices and traditions:

1. **Mathematical sciences**: rigorous mathematics is an integral part of image analysis, computer vision, and machine learning. It is the language in which operations, properties, analyses, and results are most often discussed.

2. **Engineering**: engineering has been defined to me in the past as the study of solving problems given economical constraints. In this thesis, the problem at hand is the extraction of information from bitemporal remote sensing image pairs.

3. **Natural sciences**: this is the study of objects, structures, behaviours and phenomena through observation and experimentation. The behavior of neural networks is somewhat unpredictable, and their study is often conducted through probing experiments, similarly to what is done in the natural sciences, rather than by mathematical proofs.

These scientific domains often intersect, but studies in each one tend to be approached in different ways. In mathematical sciences, ideas tend to come from first principles (axioms and previous theorems), and derivations follow rigorous rules. In engineering, the main concern is the problem at hand, and the performance of the proposed solution, often with no formal proofs. In natural sciences, experiments are done to better understand a given subject of interest.

Most of the work in this thesis stems from an original problem, the analysis of multitemporal remote sensing images. The methods that are presented in later chapters and in related research are often the result of a loop

that iterates between proposing a solution to the problem at hand, and performing experiments to quantify the performance and better understand the workings of the proposed solution. It is important to understand that even a system whose operations are known and deterministically defined may give rise to behaviours that are hard to explain [May76], and may have properties that can't be formally proven using formal axiomatic systems [Göd31]. Formal proofs regarding neural networks often lag behind the ideas that are proposed and tested experimentally. The universal approximation theorem, whose origins can be found in the works of Cybenko [Cyb89] and Hornik [Hor91], is one such case.

More precisely, this thesis touches three main areas of research: computer vision, machine learning, and remote sensing of the Earth. The motivations and applications start in the domain of remote sensing, where there is currently an abundance in data being acquired from several sources, and fast and accurate interpretation of such data has several impactful applications. Such interpretation is often done using computer vision and image analysis methods, thus many ideas from these fields are frequently applied in remote sensing research. Computer vision, in turn, often brings machine learning into the context of image analysis, especially in the past decade. These domains will be explored further in Chapter 2.

1.3 Objectives

The work presented in this thesis orbit around a central point: extraction of change information from bitemporal remote sensing image pairs using convolutional neural networks. This is approached from the following main points of view:

1. **Data modalities**: are there advantages to using multispectral images over RGB images for change detection?

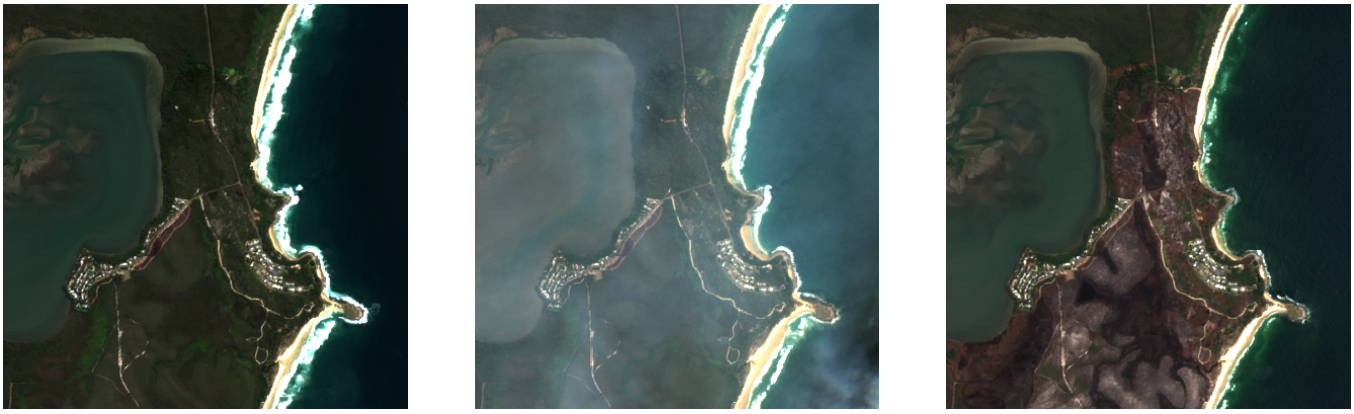
2. **Data semantics**: how can we extract semantic information from the image pair to better understand the evolution of the imaged area?

3. **Data availability**: how can we mitigate the effect of label noise and cope with data scarcity when training convolutional neural networks?

4. **Data heterogeneity**: how can we create change detection systems that are robust in their generalization to previously unseen events?

The interest in designing automatic change detection systems comes from two main cases: large scale applications and rapid response to events. In both of these cases, the speed of manual image analysis becomes a problem. For example, during the Australian wildfires in 2019, illustrated in Fig. 1.3, remote sensing images were used to obtain up-to-date information about the current situation [Law20]. Satellite imagery often provides a fast

(a) 03/09/2019 (b) 09/09/2019 (c) 03/10/2019



(d) 03/09/2019 (e) 09/09/2019 (f) 03/10/2019

Figure 1.3: Images following the evolution of wildfires near Wooloweyah, New South Wales, Australia. Sentinel-2 images courtesy of ESA's Copernicus program.

and accurate way of obtaining precise information over large areas. This is especially valuable for observing areas of difficult access, such as up high mountains, deep into forests, and polar regions.

The application of computer vision and machine learning methods to the analysis of multitemporal remote sensing imagery has a long research history. Chapter 2 contains a summary of the research that is directly related to the work presented in this thesis, as well as a theoretical basis which is necessary for their understanding. This theoretical recapitulation should provide the reader with the minimum knowledge necessary to understand the ideas presented in the subsequent chapters.

The original research that was conducted during this thesis is presented in Chapters 3, 4, 5, and 6. Chapter 3 describes how convolutional neural networks (CNNs) and fully convolutional neural networks (FCNs or FCNNs) can be used to perform change detection as a semantic segmentation problem in a fully supervised manner. Convolutional neural networks and fully convolutional neural networks architectures are trained from scratch using a novel change detection dataset. The impact of architecture variations is studied, as well as the difference in performance between CNNs and FCNs. Chapter 3 also contain comparisons between using the proposed networks with different combinations of bands of multispectral images, and the obtained results show that it is valuable to use bands outside

the visible spectrum for change detection when they are available.

Chapter 4 explores the addition of semantic information into the previously presented change detection framework, and how multitask learning affects the networks' performances. A novel large scale high resolution semantic dataset is created to perform these experiments. Different approaches to this problem are analysed, from comparing predicted land cover maps for detecting changes up to a fully integrated FCN for performing simultaneous change detection and land cover mapping of the input images. A staged training scheme is proposed for the integrated multitask network that reduces the number of hyperparameters to be chosen and that improves the performance of the trained network.

Chapter 5 studies the impact of label noise on the learning process for change detection, and borrows ideas from weakly supervised learning to improve the accuracy of the networks' results. Label bias in the previously presented dataset that was automatically generated have a biasing effect on the predicted change maps by the networks. An iterative training and label cleaning process is employed to increase the network's robustness to label noise, and it is coupled with a novel guided anisotropic filtering algorithm that better fits prediction to boundaries in the input images. Chapter 5 also explores how pixel-level change detection can be performed using only image-level labels as supervision in a weakly supervised setting. A fixed spatial attention layer is learned during the training for a classification network, and the synergy between this attention layer and the network's classifier improve the network's ability to localize changes. These results are also improved by using the guided anisotropic diffusion algorithm.

Chapter 6 describes an adversarial learning approach to bridge the domain gap between images of different types of changes, using various types of natural catastrophes as case studies. Different natural catastrophes can have very different visual appearances in remote sensing images, so neural networks trained to detect changes caused by one type of event don't generalise well to other catastrophes. Adversarial learning is used to project the images into a common latent space where change detection supervision comes from a source domain, and can be applied to images from the target domain. The proposed method is also tested on other domain adaptation tasks in computer vision to show its versatility.

Chapter 7 concludes this manuscript with the main conclusions drawn from the presented research. It also contains perspectives for future studies that could be conducted to continue this line of research.

1.4 Publications

Many of the contributions presented in this thesis have been published in peer reviewed venues. Below is a list of the publications and submissions relative to these works at the time of writing.

1.4.1 Journal Articles

- **R. C. Daudt**, B. Le Saux, A. Boulch and Y. Gousseau, "Multitask Learning for Large-scale Semantic Change Detection", Computer Vision and Image Understanding, 187, p. 102783 (2019).

1.4.2 Conference Articles

- **R. C. Daudt**, B. Le Saux, A. Boulch and Y. Gousseau, "Urban Change Detection for Multispectral Earth Observation Using Convolutional Neural Networks", In IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium, pp. 2115–2118 (July 2018).
- **R. C. Daudt**, B. Le Saux and A. Boulch, "Fully Convolutional Siamese Networks for Change Detection", In 2018 25th IEEE International Conference on Image Processing, pp. 4063–4067 (October 2018).
- **R. C. Daudt**, B. Le Saux, A. Boulch and Y. Gousseau, "Détection Dense de Changements par Réseaux de Neurones Siamois", In Reconnaissance des Formes, Image, Apprentissage et Perception (RFIAP), Marne-la-Vallée, France (June 2018).
- **R. C. Daudt**, A. Chan-Hon-Tong, B. Le Saux and A. Boulch, "Learning to Understand Earth Observation Images with Weak and Unreliable Ground Truth", In IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium, pp. 5602-5605 (July 2019).
- **R. C. Daudt**, B. Le Saux, A. Boulch and Y. Gousseau, "Guided Anisotropic Diffusion and Iterative Learning for Weakly Supervised Change Detection", In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops (June 2019).

Chapter 2

Related Work

Isaac Newton famously said that if he saw further, it was by standing on the shoulders of giants [New19], and this thesis, as all others, is no different. It is, first and foremost, a work of computer vision and image analysis, with an application in the field of remote sensing and using methods rooted in machine learning. This chapter aims to provide an introduction to the concepts most necessary to understand the contributions that will be presented in the following chapters, as well as a review of recent works that have inspired this work or aim to solve similar problems.

It is often impossible to draw sharp lines between the different scientific domains that are the main pillars of this work. The following sections aim to separate the theoretical foundations into meaningful parts, but the interplay between these research fields makes the information in these sections somewhat interdependent.

The sections below attempt to present the theoretical basis for this work coming from a problem solving perspective. First, a basic summary of the fields of computer vision and image analysis is presented. Then, the foundations of machine learning methods are presented, with special attention to image oriented techniques, along with various alternative data paradigms, as this will be a recurring theme throughout the following chapters. A summary of change detection techniques for remote sensing is then presented to contextualize this work among those that have already studied similar problems. Finally, a section containing definitions of the metrics used for quantitative evaluation of the proposed methods concludes this chapter.

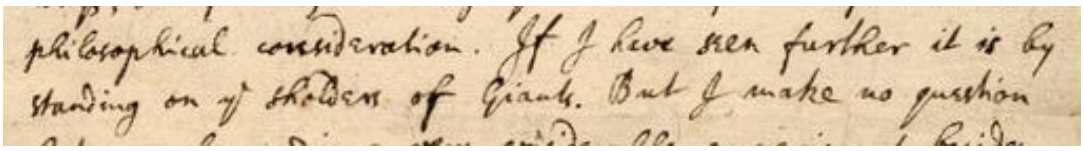
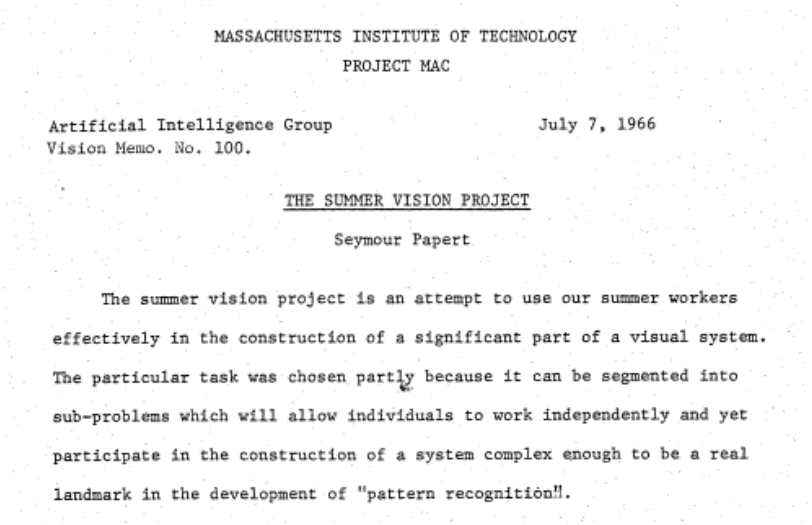


Figure 2.1: Excerpt from letter written by Isaac Newton to Robert Hooke in 1675. Image courtesy of `https://discover.hsp.org/`.

2.1 Computer Vision and Image Analysis

It is widely accepted that the field of computer vision had its beginnings at MIT in the 1960s. In his PhD thesis presented in 1963, Lawrence G. Roberts attempted ”to make it possible for a computer to construct and display a three-dimensional array of solid objects from a single two-dimensional photograph” [Rob63]. In 1966, Seymour Papert conducted ”The Summer Vision Project”, in which basic vision tasks (e.g. segmentation into ’likely objects’, ’likely background areas’, and ’chaos’) were to be performed by a computer [Pap66]. During this project, Marvin Minsky is reported to have told his undergrad student Gerald J. Sussman to ”spend the summer linking a camera to a computer and getting the computer to describe what it saw” [Sze10]. It is safe to say Sussman did not completely solve this problem that summer.



MASSACHUSETTS INSTITUTE OF TECHNOLOGY
PROJECT MAC

Artificial Intelligence Group — July 7, 1966
Vision Memo. No. 100.

THE SUMMER VISION PROJECT

Seymour Papert

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

Figure 2.2: Vision Memo No. 100 from the Artificial Intelligence Group at MIT contained the goals for the Summer Vision Project, including segmentation between objects and background, and scene analysis containing objects such as balls, bricks, cylinders, and other ”objects of known sort”.

Much of the expertise applied to computer vision came from the field of digital signal processing, specifically digital image processing. One way to separate these two fields of research is to focus on their outputs. Digital image processing tasks usually have their outputs in image form, e.g. image filtering, denoising, contrast enhancement, restoration, inpainting, etc. Computer vision studies the tasks where information is to be extracted from images, usually in a similar fashion to the human visual system. Tasks such as classification, segmentation, stereo depth perception, tracking, and many others are often easy for humans, but defining algorithms that perform these tasks from luminosity levels sampled on a grid is far from trivial.

But it is easy to see the motivation for this pursuit. Our visual system allows for us to sense our environment in an astonishingly precise way. In many tasks and benchmarks, human-level performance is implicitly considered as ”the truth”. In many others, human-level performance is a baseline performance that is rarely surpassed, except in speed. Even in very specific and well defined contexts, surpassing the human-level performance baseline is

something that attracts attention [EKN+17, FSR+20].

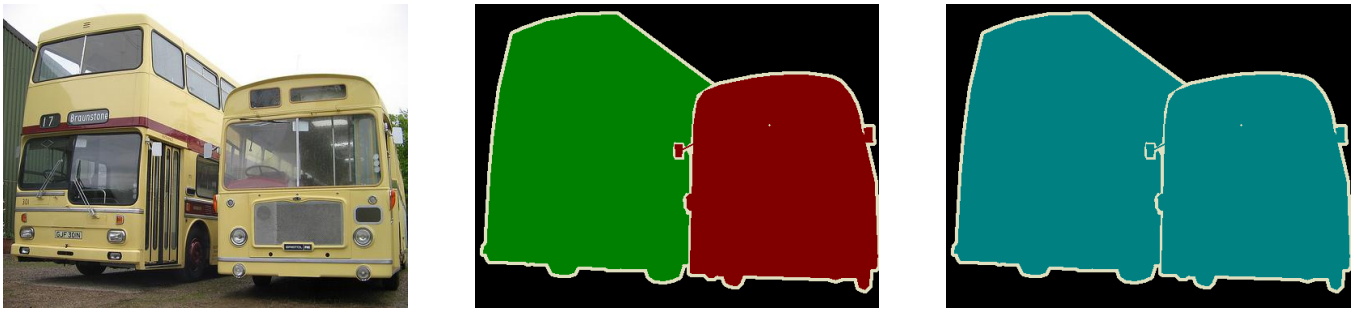
Richard Szeliski provides a good description of the history of computer vision up to about 2010 in [Sze10]. He identifies the origin of the difficulty in computer vision stems from it being a study of inverse problems "in which we seek to recover some unknowns given insufficient information to fully specify the solution". Creating a render from a 3D model, for example, can be done by following deterministic, well defined algorithms. Extracting a 3D model from a rendered image, on the other hand, is a much more poorly defined problem. Szeliski then claims physics-based and probabilistic models are then used to disambiguate between solutions.

Early computer vision research focused on mapping the structure of the 3D world using only images. Approaches such as the "blocks world" one proposed by Roberts used topological analysis of 2D lines identified in the images to model 3D objects [Rob63]. Such approaches motivated research in line extraction techniques, which culminated in algorithms such as Canny's edge detector [Can86] and active contour methods [KWT88]. Computer vision quickly branched out to several other application based on real world needs:

- Optical character recognition (OCR) for automated reading of books, letters, envelopes, etc.;
- Face recognition, fingerprint recognition, biometrics;
- Medical image analysis and computer aided diagnostics;
- Machine inspection for improving manufacturing quality control efficiency;
- Video surveillance for widespread security systems;
- Stereo vision for bioinspired 3D vision systems;
- Optical flow estimation, in which the movement of each pixel is estimated between a pair of images;
- Shape from X, which uses physics-based heuristics for shape estimation;
- Image segmentation, which aims to group pixels into meaningful regions coming from the same object;
- Image classification, where the class of an image is inferred among a group of known options;
- Object detection, where the position, number, and usually the class of objects in an image are extracted;
- Semantic segmentation, where each pixel in an image is classified between a set of known possible class options.

This list in no way covers all the problems studied in computer vision, but it illustrates the variety of inverse problems that our visual cortex performs almost effortlessly and that are very hard to solve using computers.

This work focuses mostly on semantic segmentation, while image classification is also a recurrent topic. Modern techniques have their origins in the older problem of image segmentation and pixel clustering. The main difference between image segmentation and semantic segmentation is, conveniently enough, semantics, as depicted in



(a) Input image (b) Segmentation (c) Semantic segmentation

Figure 2.3: Comparison between image segmentation and semantic segmentation from the PASCAL VOC 2012 dataset [EVGW+12]. In (b), the buses' pixels are separated into two different groups with no semantic information since they come from different objects. In (c), the pixels relative to the regions of both buses are classified as belonging to the semantic class "Bus".

Fig. 2.3. In image segmentation, pixels are simply grouped in meaningful sets that belong to the same object or region. The definitions of what consists of an object or region can be subjective. Picture an image of a cube that is to be segmented: should the whole cube be a single region, or should each face be separated into a different region? In semantic segmentation, on the other hand, a class label is attributed to each pixel based on a known set of possible classes. Pixels in separate regions can belong to the same class. This problem, given a set of classes, is often more accurately defined than that of image segmentation. Semantic segmentation algorithms often depend on learning from examples from each class. These characteristics will be better discussed later on.

Several methods have marked the progress of research into image segmentation algorithms. Mumford and Shah tackled this problem through the minimization of an energy functional containing terms for sub-region smoothness, boundary length, and other desirable properties. Vincent and Soille proposed the watershed method, inspired by the catchment basins of rain water [VS91]. Shi and Malik modelled image segmentation as a graph partitioning problem in a method known as normalized cuts [JM00]. The mean shift method was applied for image segmentation by Comaniciu and Meer, although the method had already been proposed in different contexts [CM02].

The Eigenfaces work by Turk and Pentland was one of the first to introduce machine learning to [TP91]. Principal component analysis was used on training images to find a set of bases able to represent any face. Distances in this new representation space was used to discern between faces of different people and perform facial recognition. Another early application of machine learning that had a strong impact in both academia and industry was the object detection framework proposed by Viola and Jones [VJ04]. It used cascading classifiers based on Haar features that could be efficiently computed, boosted using the AdaBoost algorithm [FSA99]. The efficiency and robustness of this method made it the standard for face detection for many years.

The development of SIFT descriptors in 1999 by Lowe sparked the interest of researchers for local feature descriptors due to its excellent performance in applications such as stereo matching, object recognition, tracking, stitching, and others [Low99, Low04]. It was technically preceded by HOG descriptors in 1986 [McC86], but this

method only really gained traction after it was used by Dalal and Triggs for human detection in 2005 [DT05]. Such works inspired other feature descriptors such as SURF [BTVG06] and ORB [RRKB11]. Such descriptors were often coupled with machine learning methods such as the bag-of-words approach to perform object detection and image classification [SZ09].

Neural networks, especially convolutional neural networks (CNNs) have come dominate many of the research topics in computer vision. Their reliance on data rather than handcrafted methods comes with positive and negative sides, as will be discussed in Section 2.2. Modern CNN-based approaches often couple the standard data-based approach with problem-specific heuristics to improve performance at specific problems.

2.2 Machine Learning

The field of machine learning studies algorithms that are able to improve its performance at a given task automatically using examples, experience, or data, which is usually referred to as ”learning”. Machine learning is often conflated with artificial intelligence, although modern machine learning research rarely discusses the philosophical ideas of what intelligence actually is or how to reproduce it. Research in the field of machine learning usually proposes solutions to a given problem, or attempts to better understand previously proposed algorithms.

The value of machine learning algorithms comes firstly from their versatility. In a way, these algorithms solve a meta-problem, and the solution to the specific problem at hand comes from their learning process. Let's take the problem of image classification as a toy example. Imagine an algorithm which was handcrafted to tell apart images of cats and dogs. Such algorithm is unlikely to be useful if we try to use it to separate images of rabbits and foxes, or cars and buses, or benign and malign skin growths. In this problem of image classification, a machine learning algorithm would solve the meta-problem of separating images into two groups, and the exact operations to do so would be extracted from examples. This way, the same algorithm can be used for solving several problems of a certain category.

This means that problems solved with machine learning methods are usually analysed from a farther point of view. These solutions don't require that the machine learning algorithm, and by extension the machine learning engineer, encompass the exact nature of the problem at hand as long as the general structure of the problem is understood. The need for expert knowledge becomes a need for data. Looking at the ears may be the key to differentiating between cats and dogs, but a machine learning engineer need never know that as long as the machine learning algorithm is able to learn that from data.

Most machine learning algorithms are defined by a set of deterministic operations that are performed to the input to attempt to perform the desired task[1]. These operations, which are determined by the machine learning

[1]There are exceptions to this claim, such as evolutionary algorithms, but these are outside the scope of this work.

engineer[2], use operation parameters that are learned automatically during the learning process. The process of solving problems with machine learning is then often seen as containing two main parts:

1. Construct an algorithm that, given the appropriate operation parameters, is in theory able to perform the task at hand.

2. Find the operation parameters that produce the desired outcome as well as possible.

There are several sources that provide detailed information on various machine learning systems and their history [Mit97, Bis06, Mur12, Sch15, GBC16]. For the sake of simplicity, the review in this section focuses on the ideas and methods that support the work presented in the following chapters.

The following subsections will define from the ground up the basic forms or feedforward neural networks, convolutional neural networks, segmentation networks, and finally some alternative learning paradigms. Virtually all definitions that will be made could be appended with "but someone did it differently". For the sake of simplicity, this will only be acknowledged when found to be relevant.

2.2.1 Feedforward Neural Networks

The first model that we will analyse will be the perceptron, or feedforward neural network, first proposed by Frank Rosenblatt in 1957 [Ros58, Ros60]. The understanding of the perceptron begins with the artificial neuron, which was loosely inspired by neuroscience [GBC16]. The neuron is a function that produces an output $y \in \mathbb{R}$ from an input vector $\mathbf{x} \in \mathbb{R}^n$ and parameters Φ. This is typically done through matrix multiplication, output bias, and a non-linear activation function, here denoted by $g(\cdot)$:

$$f(\mathbf{x}; \Phi) = g(\mathbf{w}^\top \mathbf{x} + b), \tag{2.1}$$

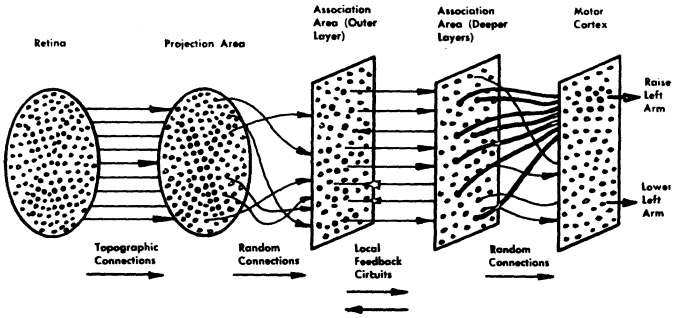
where Φ contains all the operation parameters, in this case $\mathbf{w}$ and b. Most non-linear activation functions don't require any parameters, although that is not always the case. Results discovered by George Cybenko and Kurt Hornik show that functions with this form have powerful approximation capabilities.

A set of m neurons can be used in parallel to compute different activations from the same input using different parameters. This is straightforwardly done by simply adapting the operations and dimensions in Eq. 2.1:

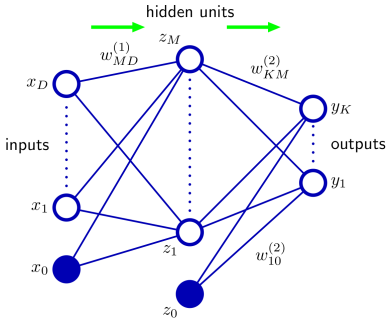
$$f(\mathbf{x}; \Phi) = g(\mathbf{W}^\top \mathbf{x} + \mathbf{b}), \tag{2.2}$$

where $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{b} \in \mathbb{R}^m$, $\mathbf{W} \in \mathbb{R}^{n \times m}$. Such functions are often referred to as single layer perceptrons.

[2]There are also exceptions to this, such as neural architecture search algorithms.



(a) Rosenblatt's inspiration for the perceptron



(b) Modern MLP model

Figure 2.4: (a) While Rosenblatt's inspiration came partially from retinal nerves, the perceptron model is not restricted to vision. (b) Information flows in a single direction from inputs to outputs and through hidden units in a multilayer perceptron. Images reproduced from [Ros60] and [Mur12], respectively.

Single layer perceptrons can be chained together to form multilayer perceptrons (MLPs). Given a set of p appropriately dimensioned single layer perceptrons $f^{(i)}(\mathbf{x}; \Phi^{(i)}), i \in \{1, 2, \ldots, d\}$, a multilayer perceptron can be defined as:

$$f(\mathbf{x}; \Phi) = f^{(d)}(\ldots f^{(1)}(\mathbf{x}; \Phi^{(1)}) \ldots; \Phi^{(d)}), \tag{2.3}$$

where $n^{(i)} = m^{(i-1)}, i \in \{2, \ldots, d\}$, $n^{(i)}$ being the number of inputs and $m^{(i)}$ the number of outputs of the i-th perceptron. The number of layers d is often referred to as the depth of the network, and networks with larger depths are often referred to as "deep", hence the term "deep learning" [Sch15, LBH15, GBC16]. The activations of perceptrons $f^{(i)}, i \in \{1, \ldots, d-1\}$ are called hidden layers, while the activation of perceptron $f^{(d)}$ is referred to as output. These networks are called feedforward networks due to the information flow happening only from the input towards the output. If feedback connections are introduced, the network becomes what is known as a recurrent neural network (RNN) [RHW86, GBC16].

The activation functions $g(\cdot)$ are essential to allow the network to perform complex functions. It is easy to see that without the activation functions, the operations performed by all layers chained together (Eq. 2.3) would be equivalent to a linear algebra matrix operation. Common activation functions include:

- Sigmoid function: $\sigma(z) = \frac{1}{1+\exp(-z)}$.
- Hyperbolic tangent: $\tanh(z) = \frac{\exp(2z)-1}{\exp(2z)+1}$,
- Rectified linear unit: $\text{ReLU}(z) = \max(0, z)$.
- Leaky rectified linear unit: $\text{LeakyReLU}(z) = \max(0, z) - \alpha \cdot \max(0, -z)$, α is fixed.
- Parametric rectified linear unit: $\text{PReLU}(z) = \max(0, z) - \alpha \cdot \max(0, -z)$, α is learnable.

The case in which a dataset $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}, i \in \{1, \ldots, N\}$ containing N pairs of inputs and associated labels is available for training the machine learning algorithm is called supervised learning. In this case, we aim to find the

weights Φ that allow the MLP to approximate the real relationship between $\mathbf{x}$ and $\mathbf{y}$:

$$f(\mathbf{x};\Phi) = \hat{\mathbf{y}} \approx \mathbf{y}. \tag{2.4}$$

This is usually accomplished by searching for Φ that minimises a cost or loss function $\mathcal{L}(\hat{\mathbf{y}}, \mathbf{y})$ that measures the similarity between $\hat{\mathbf{y}}$ and $\mathbf{y}$:

$$\hat{\Phi} = \arg\min_{\Phi} \mathcal{L}(f(\mathbf{x};\Phi), \mathbf{y}). \tag{2.5}$$

Global minima are prohibitively costly to find for even moderately sized networks, so we usually settle for finding local minima during the training process. Assuming f and $\mathcal{L}$ use only differential operations, this is usually done by calculating the gradients using a backpropagation algorithm, an application of the chain rule for differentiation, and performing some form of gradient descent optimization on Φ. Attributing such procedure to a single author is not a trivial task, the reader is referred to Section 5.5 in [Sch15] for a historical summary of backpropagation methods for neural networks.

This form of iterative optimization needs a starting point Φ_0 from which the iterations start. It is necessary to introduce some level of randomness to this initialization of weights, although there are different ways to do so [HZRS15]. The iterative optimization process can then be formalized as

$$\Phi_{i+1} = \Phi_i + \lambda \cdot \nabla\mathcal{L}(f(\mathbf{x};\Phi), \mathbf{y})|_{\Phi_i} = \Phi_i + \lambda \cdot \left.\frac{\partial\mathcal{L}(f(\mathbf{x};\Phi), \mathbf{y})}{\partial\Phi}\right|_{\Phi_i}, \tag{2.6}$$

where λ is a tunable hyperparameter that controls the speed and stability of the optimization. There are several improvements to this basic algorithm that incorporate concepts of momentum and adaptable learning rates. A review of such algorithms can be found in [Rud16].

In many cases, the size of the dataset $\mathcal{D}$ is too large to evaluate $\mathcal{L}$, the gradients and optimization steps for all data points simultaneously. In these cases, subsets of the $(\mathbf{x}, \mathbf{y})$ pairs, called mini-batches, are considered at a time. The resulting problem can then be formalized as a stochastic optimization problem

$$\hat{\Phi} = \arg\min_{\Phi} \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\mathcal{D}}[\mathcal{L}(f(\mathbf{x};\Phi), \mathbf{y})], \tag{2.7}$$

which can be solved by a stochastic gradient descent (SGD) method similarly to Eq. 2.6 by sampling data from $\mathcal{D}$:

$$\Phi_{i+1} = \Phi_i + \lambda \cdot \nabla\mathcal{L}(f(\mathbf{x};\Phi), \mathbf{y})|_{\Phi_i} = \Phi_i + \lambda \cdot \left.\frac{\partial\mathcal{L}(f(\mathbf{x};\Phi), \mathbf{y})}{\partial\Phi}\right|_{\Phi_i}, (\mathbf{x}, \mathbf{y}) \sim \mathcal{D}. \tag{2.8}$$

Taking the case of classification as an example, assume $\mathcal{D}$ contains pairs of input vectors and associated labels $(\mathbf{x}, \mathbf{y})$. The number of considered classes is known to be C and each $\mathbf{x}$ is associated to one and only one class.

Classes are numbered from 1 to C. The ground truth output vectors $\mathbf{y} \in \mathbb{R}^C$ are one-hot encoded, which is to say

$$\mathbf{y}_i = \mathbb{I}_{c_i}(n) = \begin{cases} 1, & \text{if } n = c_i \\ 0, & \text{if } n \neq c_i \end{cases}, \tag{2.9}$$

where c_i is the index of the class associated with $\mathbf{x}_i$.

In this case, it is useful to set the activation function $g^{(d)}$ of the final layer $f^{(d)}$ of the MLP to be the softmax activation function [Bis06, GBC16], which can be defined as

$$\text{softmax}(\mathbf{z})_i = \frac{\exp(z_i)}{\sum_j \exp(z_j)}. \tag{2.10}$$

The dimensionality of the MLP's output must also match the number of classes C. The output $\hat{\mathbf{y}}_i$ of the MLP for input $\mathbf{x}_i$ can then be interpreted as predicted probabilities of it belonging to each of the considered classes.

We can then interpret $\mathbf{y}$ as a probability distribution, and define a loss function as the cross entropy between $\mathbf{y}$ and $\hat{\mathbf{y}}$:

$$\mathcal{L}(\hat{\mathbf{y}}, \mathbf{y}) = -\sum_j y_j \log(\hat{y}_j) = -\log(\hat{y}_c). \tag{2.11}$$

The last part of the equation is a simplification that can only be done due to the one-hot encoding of the labels $\mathbf{y}$ for an element belonging to class c. Minimizing the loss function presented in Eq. 2.11 using the procedure described in Eq. 2.8 allows us to find parameters $\hat{\Phi}$ that enables the MLP to predict the relationship between inputs and outputs in $\mathcal{D}$, i.e. $f(\mathbf{x}, \hat{\Phi}) = \hat{\mathbf{y}} \approx \mathbf{y}$.

Nowadays, automatic differentiation computing libraries such as PyTorch [PGC+17] and TensorFlow [ABC+16] are able to perform all the operations described above including the calculation of gradients and optimizations, as well as many others. Such libraries are essential for the fast development of complex deep learning systems and are used throughout academia and industry.

2.2.2 Convolutional Neural Networks

The neural architecture described in Section 2.2.1 considers the elements of input $\mathbf{x}$ equally and ignores all positional information. While this is appropriate for many applications, image data usually comes in the form $\mathbf{I} \in \mathbb{R}^{M \times N \times C}$, where M and N are the spatial 2D dimensions of the image and C is the number of colour channels ($C = 3$ for RGB images). Information in $\mathbf{x}$ comes not simply from the values attributed to each pixel, but mainly from the relationship between pixels. While images can simply be vectorized as $\text{vec}(\mathbf{I}) \in \mathbb{R}^{M \cdot N \cdot C}$ for it to be used as input for an MLP, this completely ignores the structure of the images and makes the MLP extremely sensitive to small translations, rotations, and other transformations. The number of pixels in common images are also large in

comparison to what is feasibly tractable with MLPs.

The Neocognitron architecture helped tremendously with these two problems. It used operations that were limited to well defined neighbourhoods and with shared parameters [FM82]. The Neocognitron proposed by Fukushima et al. was not trained using the procedures described in Section 2.2.1. This was done by LeCun et al. in 1998 in their work in which convolutional neural networks (CNNs) were proposed [LBBH98]. This was the first time that gradient-based learning was used in conjunction with weight-sharing operations that worked over a sampled 2D grid. This was done through the use of discrete convolutions, which are operations well studied and understood in the field of image processing and digital signal processing, mainly to their application to filtering and interesting properties in Fourier analysis [Opp99].

Convolutions for Neural Networks

Two dimensional convolutions are used in CNNs to increase robustness to translations and to massively decrease the amount of learnable parameters compared to the matrix operations used in MLPs. The basic 2D convolutions used in CNNs can be defined as

$$\mathbf{I_{out}}[m,n] = \mathbf{K} * \mathbf{I}[m,n] = \sum_{i=-p}^{p}\sum_{j=-q}^{q}\sum_{c=1}^{C}\mathbf{I}[m+i,n+j,c]\cdot\mathbf{K}[i,j,c], \tag{2.12}$$

where $\mathbf{K} \in \mathbb{R}^{(2m+1)\times(2n+1)\times C}$ is the convolution kernel that contains the learnable parameters. This convolution operation is not exactly equal to the one typically used in digital signal processing, but it follows the same basic idea of applying a multiply-accumulate operation over a sliding window. Padding of the input (zero padding, mirroring, etc.) is often done to maintain the spatial dimensions of the images after the convolution. Note that in this definition the output has a single ”colour” channel (although the term ”colour” loses its meaning at this point and becomes somewhat interchangeable with ”feature”). Similarly to what was done in Eq. 2.2, a set of these operations can be done in parallel, and their results are considered together

$$\mathbf{I_{out}}[m,n,d] = \mathbf{K} * \mathbf{I}[m,n] = \sum_{i=-p}^{p}\sum_{j=-q}^{q}\sum_{c=1}^{C}\mathbf{I}[m+i,n+j,c]\cdot\mathbf{K}[i,j,c,d], \tag{2.13}$$

where $\mathbf{K} \in \mathbb{R}^{(2m+1)\times(2n+1)\times C\times F}$ can be seen as a collection of F kernels. Combining these convolution operations with bias and activation function operations form what is referred to as convolutional layers.

Pooling operations are also a mainstay of CNNs. These operations aim to increase the robustness of the network to small translations and other spatial transforms, as well as to reduce the size of the calculated features to reduce memory costs and to increase the receptive field of the operations and allow the network to identify larger, more

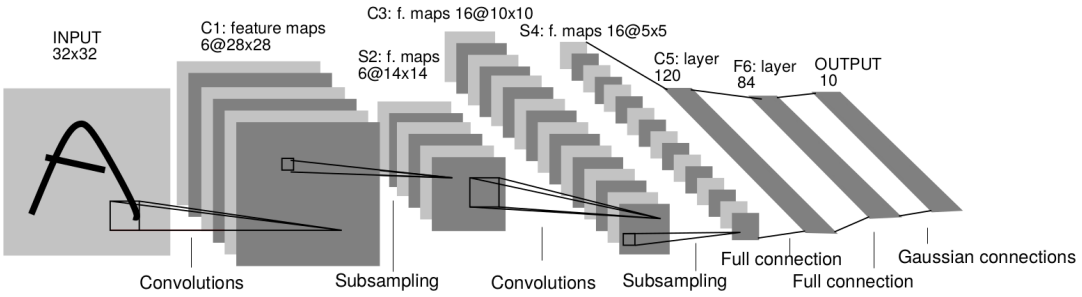


Figure 2.5: LeNet-5 was the first convolutional neural network to combine bioinspired weight sharing operations with gradient-based learning. Image reproduced from [LBBH98].

contextual features. The max-pooling operation can be defined as

$$\mathbf{I_{out}}[m, n, c] = \max(\mathbf{I}[i, j, c]), \text{ s.t. } s \cdot m \leq i < s \cdot m + p \text{ and } s \cdot n \leq i < s \cdot n + q, \tag{2.14}$$

where s defines the stride of the operation and p and q define its window size, assuming an indexing that starts at 0. It is also common to use average-pooling instead of max-pooling, where the maximum operation is replaced by the mean operation, or using convolutions with a stride larger than 1 to reduce the spatial resolution of feature maps. A reduction in spatial resolution is often done along an increase in feature map space (i.e. number of convolutional filters).

Basic Convolutional Neural Network Architectures

Classification CNNs usually begin with several convolutional layers and pooling layers to calculate hierarchical feature maps, as seen in Fig. 2.5. The number of layers, the number of filters per layer (also known as layer depth), and position and properties of the pooling layers are architecture parameters that can be chosen based on factors such as how much computational power is available and how much training data is available, among other factors. The output of convolutional and pooling layers does not match the necessary format for classification. This problem is solved by vectorizing the output feature map and feeding it into a classification MLP. Note that a network with this architecture can only process images of a single size. Modern classification CNNs solve this problem through adaptive pooling operations, whose output is of a fixed size regardless of the input size.

Siamese architectures have been proposed in the past to compare images using CNNs [BGL+94, CHL05a]. In these architectures, the input images are processed separately at first by several convolutional and pooling layers which usually share their weights. The images are then compared in feature space to achieve the desired result, such as face verification [CHL05a], signature recognition [BGL+94], stereo vision [ZK15a], or change detection [ZFY+17].

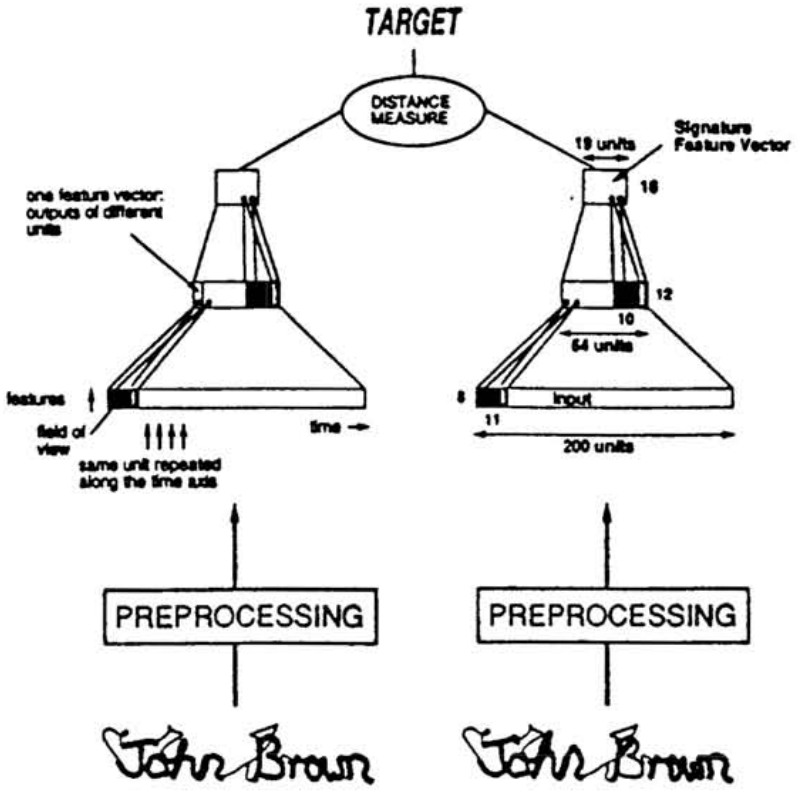


Figure 2.6: The CNN proposed by Bromley et al. was the first one to have a Siamese structure. It was used to compare signatures automatically. Image reproduced from [BGL+94].

Training Deep Neural Networks

It has long been known that training deep neural networks is hard due to the vanishing gradient problem [KK01]. This is mainly a consequence of the multiplication of gradients through the chain rule for activation functions whose derivatives lie between 0 and 1, such as hyperbolic tangent, sigmoid. The result is that operations far from where the loss function is calculated (the output) are hard to optimize using gradient descent algorithms. The opposite problem of exploding gradients can also be harmful to the training of neural networks.

Several techniques have been proposed to help mitigate these gradient problems when training deep neural networks. These include:

- Layer-wise unsupervised pretraining of networks [Sch92].
- Deep supervision (i.e. using additional losses at intermediary points of the network) [SWY+15].
- Normalization operations such as batch normalization [IS15], group normalization [WH18], and instance normalization [UVL16].
- Residual convolutional blocks [HZRS16a].

The main motivation for using deeper architectures is that this allows networks to learn more complex operations,

as long as they are properly trained. This phenomenon has been repeatedly observed, which explains the modern interest in finding ways to handle larger and deeper architectures [SZ15, HZRS16a, HLvdMW17].

In the early 2010s, a breakthrough that in many ways triggered the modern explosion of research into CNNs was the insight of using graphics processing units (GPUs) for inference and training of neural networks. Their ability to perform massively parallelised calculations allow for these computations to happen much faster than on a regular CPU. While this is usually credited to Krizhevsky et al. [KSH12], it is probably more appropriate to credit Cireşan et al. for this idea since they accomplished this one year earlier [CMM+11]. Regardless, using GPUs and other task-specific hardware is now the standard way of handling the computations needed for large neural networks.

2.2.3 Fully Convolutional Neural Networks

Another problem that is efficiently solved by CNN variants is semantic segmentation [MBP+20]. Given an image[3] $\mathbf{I} \in \mathbb{R}^{M \times N \times C}$, a semantic segmentation algorithm predicts a class for each pixel in the input image, i.e. it performs dense classification among a set of known classes. The ground truth labels $\mathbf{Y} \in \mathbb{R}^{M \times N \times N}$ that are used are the one-hot encoded class of each pixel in $\mathbf{I}$. A naive way to accomplish that using CNNs is to train the network to classify the central pixel of patches by creating a dataset with pairs $(\mathbf{I}[i-s \leq m \leq i+s, j-s \leq n \leq j+s], \mathbf{Y}[i,j])$ for training the network. This requires virtually no change to the previous formulation except for rearranging the dimensions, but it is not a very efficient way of performing semantic segmentation.

Long et al. proposed the first architecture that came to be known as a fully convolutional network (FCN) [LSD15b][4]. It aimed to predict class probabilities for all the pixels in the input image simultaneously. Instead of feeding the feature maps calculated by convolutional and pooling layers into an MLP, the feature maps were upsampled back to the scale of the input image, thus retaining spatial information in the activations and performing pixel-wise predictions. It was not only extremely accurate, but it took advantage of redundancies for the calculations of neighbouring pixels which sped up computation speeds compared to competing methods. The decrease in spatial resolution followed by upsampling is why these architectures are also sometimes referred to as encoder-decoder networks.

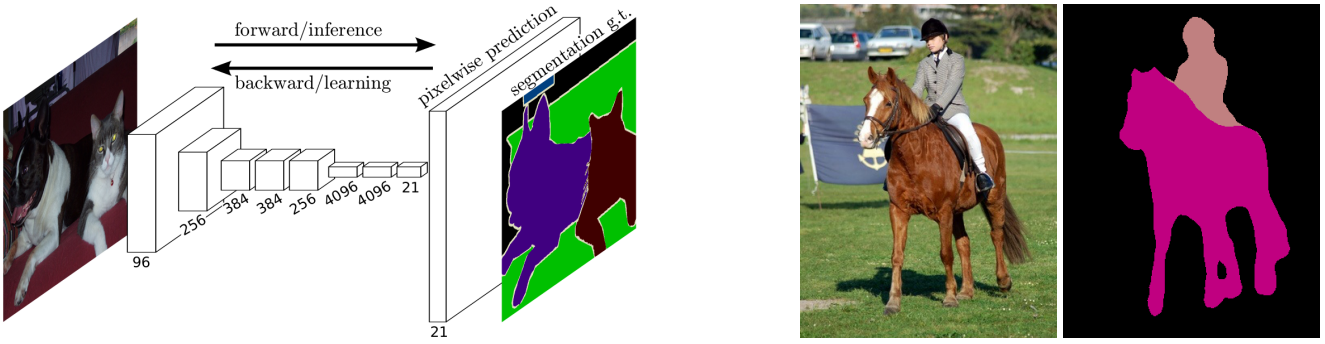
Many improvements have since been proposed to FCNs, and these techniques are now the state-of-the-art in semantic segmentation [MBP+20]. One of the main issues with the original FCN formulation was the accuracy around region boundaries. Given that the features were calculated at a lower spatial resolution, the upsampling operation often resulted in blob-like predictions. Later FCNs performed several gradual upsampling steps mirroring the downsampling operations that are used in the encoders, which allowed a gradual recovery of spatial features from high level features.

Badrinarayanan et al. proposed the SegNet[5] architecture as an improvement of the FCN [BKC17]. In this

[3] Or a set of coregistered images, as will be discussed later.

[4] This architecture is called fully convolutional network, but the group of descending architectures are referred to as fully convolutional networks, which makes the nomenclature a bit confusing at this point.

[5] Around this time it becomes trendy to name your method SomethingNet and title your paper "SomethingNet: A Network That Does Some-



(a) FCN schematic (b) Semantic segmentation by FCN

Figure 2.7: The FCN architecture upsampled feature maps in a single step, which limited the accuracy of predictions around region boundaries. Reproduced from [LSD15a].

symmetrical architecture, the position of the elements that were kept in the encoder's max-pool operations were recorded, and were then used in an unpooling operation in the decoder to upsample the image.

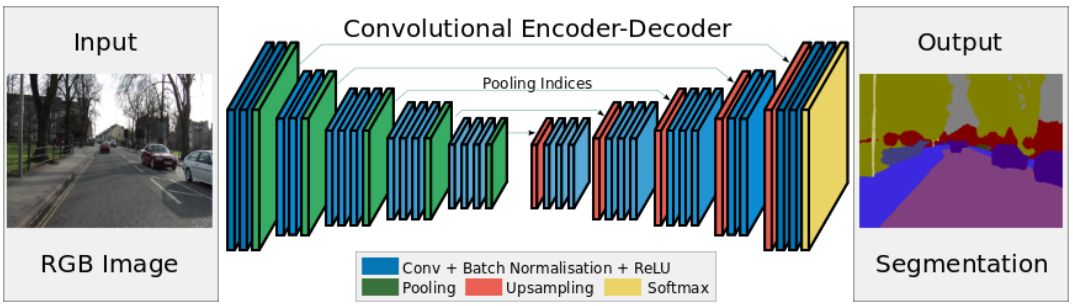


Figure 2.8: SegNet architecture for semantic segmentation, in which the pooling indices were used for the unpooling operations to recover spatial information at each upsampling step. Reproduced from [BKC17].

Ronneberger et al. proposed a different solution to improve the spatial accuracy of FCN semantic segmentations which was named U-Net [RFB15]. First, feature maps were upsampled using a transposed convolution (sometimes also written as a convolution with a fractional stride). These feature maps were then concatenated with ones from an earlier point of the network at the same level of spatial downsampling, as depicted in Fig. 2.9. These concatenation operation were named skip connections, since the feature maps "skipped" part of the network at a lower spatial resolution, thus retaining more accurate boundary position information.

Many of the improvements that are currently used in FCNs were first proposed to improve the performance of classification CNNs. Residual connections (or blocks) were proposed by He et al. [HZRS16b] with the motivation of facilitating the propagation of gradients through backpropagation and under the hypothesis that it is easier to learn identity weights using such operations. Densely connected blocks of convolutions were proposed by Gao et al. to allow the network to reuse previously calculated feature maps and reduce the number of parameters of the networks [HLvdMW17]. These ideas are often used in the design of FCN architectures such as was done in [ZTK+18, ZWZ+19], as can be seen in Fig. 2.10.

thing". How else could readers possibly know it is a deep learning paper?

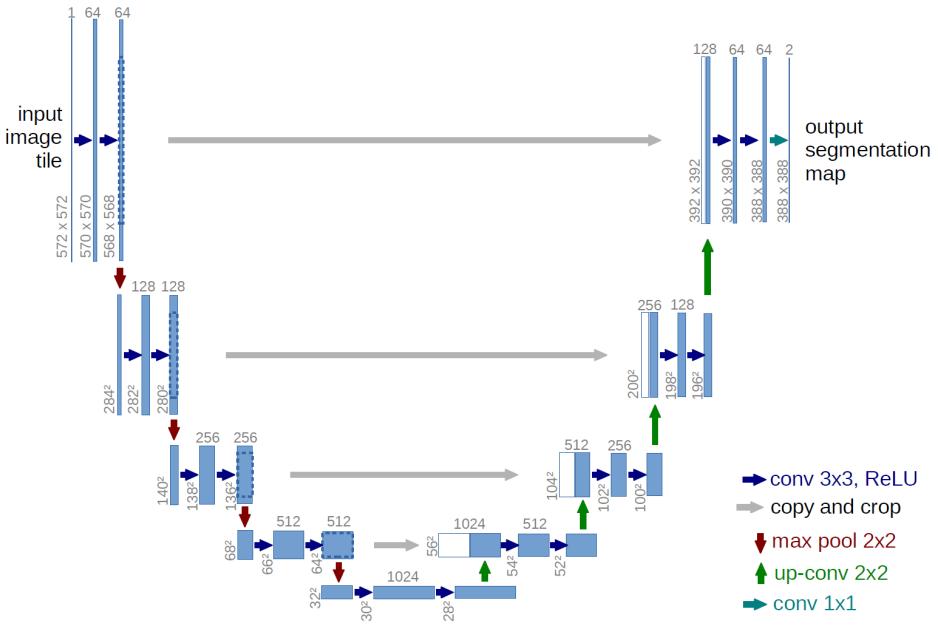
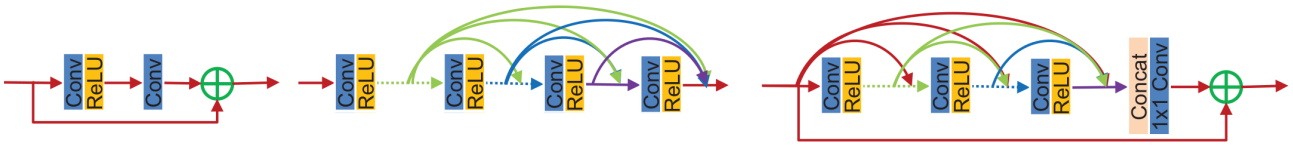


Figure 2.9: U-Net architecture schematic. The output of transposed convolutions are concatenated with feature maps produced by the encoder at several levels to combine high spatial accuracy with high-level features. Reproduced from [RFB15].



(a) Residual block (b) Densely connected block (c) Residual dense block

Figure 2.10: Schematics of (a) residual block, (b) dense block, and (c) residual dense block as described in [ZTK+18].

There are methods that are used to increase the receptive field of the operations in the network, which are used to allow the network to interpret larger scale contexts. Increasing the size of convolutional kernels is not a very attractive way to do so, since the number of parameters scales with the square of the kernel size. Yu et al. proposed the use of dilated or *à trous* convolutions for multi-scale context aggregation [YK16]. Such convolutions theoretically allow an exponential growth of receptive fields with respect to the number of layers, unlike traditional convolutions which make receptive fields grow linearly. These were also used in conjunction with residual connections [YKF17] and separable convolutions [CZP+18]. He et al. proposed the spatial pyramid pooling module which adaptively pooled the image at various spatial resolutions, and used these for context encoding [HZRS14]. Zhuang et al. proposed the usage of recurrent neural network on pixel sequence for context encoding with good results [ZYT+18]. Attention modules which attempt to identify relationships between the pixels in the images such as the one proposed by Li et al. in [LZW+19] are also a powerful tool to accomplish this.

There several variations to the methods described above, as well as various topics that were not discussed (segmentation post-processing, segmentation variations such as instance segmentation, etc.). An in depth review

of semantic segmentation using deep convolutional neural networks by Minaee et al. can be found in [MBP+20].

2.2.4 Learning Paradigms

The previous sections focused on supervised learning methods. While this covers a large family of machine learning methods, several problems don't fit nicely into this framework. Two of the most common reasons to step out of the supervised learning paradigm are data cost and non-deterministic solutions. While the former is fairly straightforward, the latter is slightly more vague. While the answer to the question "is this a picture of a dog or a cat?" is quite clear (given an appropriate image), other problems can be more complex. What is the best next move in the game of Go [SHM+16]? What is the most efficient way to cool data centres [Gao14, EG20]? What did Claude Monet see as he placed his easel by the bank of the Seine near Argenteuil on a lovely spring day in 1873[6] [ZPIE17]? It is possible to tackle such problems with machine learning methods if we step outside of supervised learning.

Unsupervised, Semi-Supervised, and Self-Supervised Learning

In some cases, the dataset at hand contains only raw data $\mathbf{X}$ with no associated outputs $\mathbf{Y}$. Such data can still be used for training machine learning systems in different ways, such as for clustering [Bis06] or representation learning [BCV13], by extracting domain specific information from the data distribution itself. This is referred to as unsupervised learning, since there is no explicit form of supervision. Unsupervised pre-training of neural network was standard procedure for many years for several reasons, but such procedures are no longer seen as essential in cases where enough data is available [Sch15].

At times, only a subset of $\mathbf{X}$ is associated to a supervision signal $\mathbf{Y}$. The field of semi-supervised learning studies how to best leverage unlabelled data to improve the performance of purely supervised systems trained using only the supervised part of the dataset [Zhu05, vEH20]. In many cases, leveraging all of the data, usually through an unsupervised learning auxiliary task, leads to better performing machine learning systems, especially in the case of data-hungry deep neural networks. One traditional way of doing so is to pre-train the networks in an unsupervised manner, then fine-tune the network's weights using the supervised data subset [HOT06, BLPL07]. Another approach is to consider a composite loss function that is composed of supervised and unsupervised components. The former is calculated only for data points with associated supervision signals, while the latter can be calculated using all the data points in the dataset[vEH20].

Weakly Supervised Learning

Data for supervision of machine learning algorithms can come in many different levels of precision. For different levels of spatial classification of images, for example, one possible ladder of complexity is:

[6] Opening line from Zhu et al.'s work [ZPIE17], a highly recommended read.

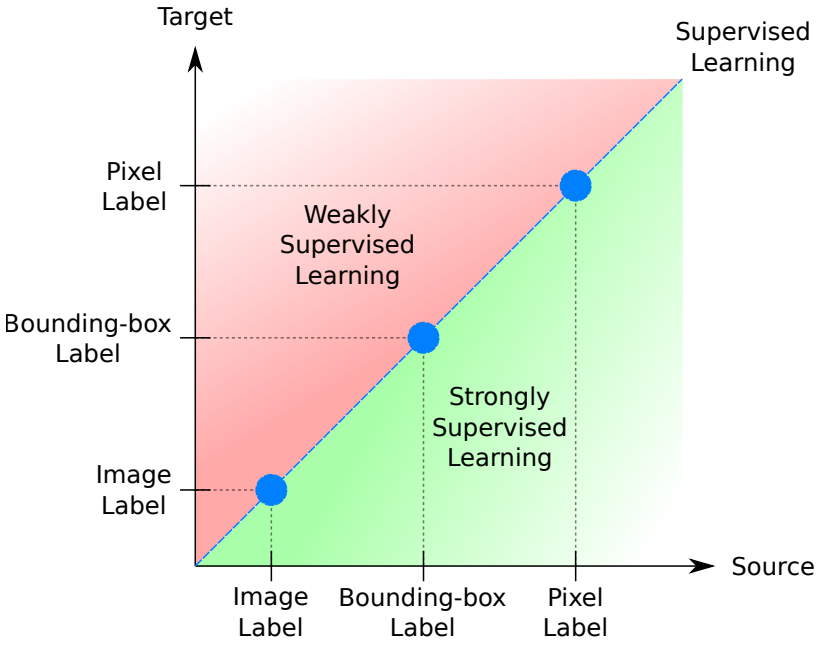


Figure 2.11: Supervision strength paradigms. The challenge of weakly supervised learning is to learn higher precision tasks from lower precision supervision.

1. Image level labels (image classification).
2. Bounding-box level labels (object detection and classification).
3. Pixel level labels (semantic segmentation).
4. Incomplete pixel level and instance labels (instance segmentation).
5. Complete pixel level and instance labels (panoptic segmentation).

Higher level supervision usually enables a higher level of image understanding, but acquiring the necessary ground truth for such supervision comes at a similarly higher cost.

The field of weakly supervised learning studies how higher level tasks can be learned from lower level supervision. This allows for cheaper label generation as well as bringing supervision from already existing data from neighbouring domains. It is easy to see how performing semantic segmentation using only image level labels [AK18], bounding-box annotations [KBO+16, KBH+17, DHS15], scribbles [LDJ+16]. Many weak supervision methods also aim to train networks using user generated tags and labels, which are often free to obtain through crowdsourcing on some online platforms.

Multitask Learning

Machine learning systems can be trained to perform more than one task at once. The term was coined by Richard Caruana in 1993 [Car93], when he noticed evidence that suggested that it is more efficient to learn several tasks at

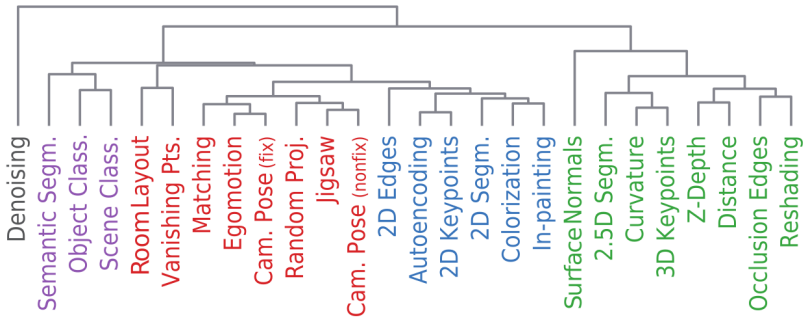


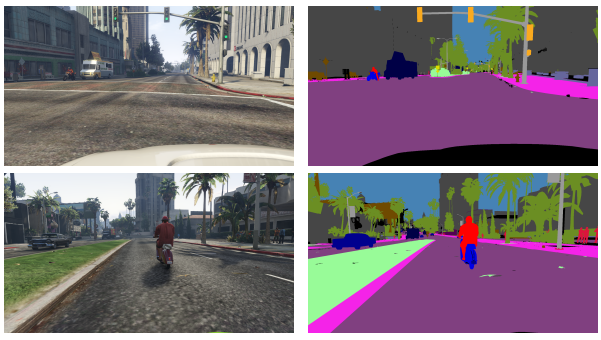
Figure 2.12: Task similarity tree proposed by the Taskonomy project. It was shown that there are several benefits from learning a single latent space representation from which several tasks are performed, such as the need for fewer annotated data. Reproduced from [ZSS+18].

once rather than separately. This topic has been studied in several contexts since [ZY17]. Notably, the Taskonomy project [ZSS+18] showed it was possible for a network to perform 26 2D, 2.5D, and 3D tasks simultaneously, as well as showing multitask learning resulted in a need for smaller amounts of labelled data points.

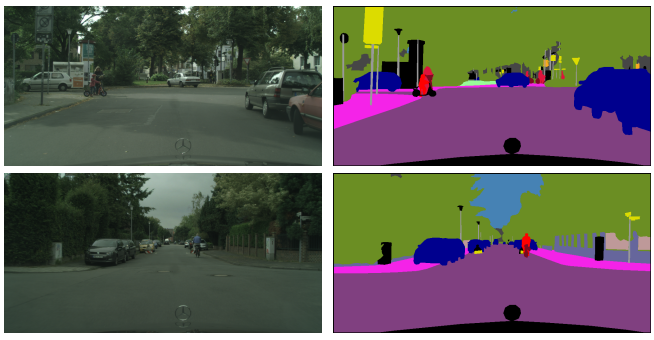
Label Noise

Labels used for supervision of classification and segmentation systems are often noisy to different degrees depending on the considered dataset. Automatically generated labels and user generated labels are especially prone to noise. Neural networks have been shown to be somewhat robust to label noise [RVBS17]. Many methods have also been proposed to identify and mitigate the effects of label noise for supervised learning systems [FV14, FK+14, LWZK19].

Domain Adaptation



(a) GTA 5 images and labels



(b) Cityscapes images and labels

Figure 2.13: One of the main motivation for domain adaptation is to bridge the domain gap between (a) synthetic data and (b) real data to reduce the cost of data acquisition. One such case is the semantic segmentation of GTA 5 images [RVRK16] and real automotive images from the Cityscapes dataset [COR+16].

Domain adaptation is the name given to the study of the generalization of machine learning systems to different, but related, data distribution [RMH+19]. The ability of neural networks to generalise to new cases is

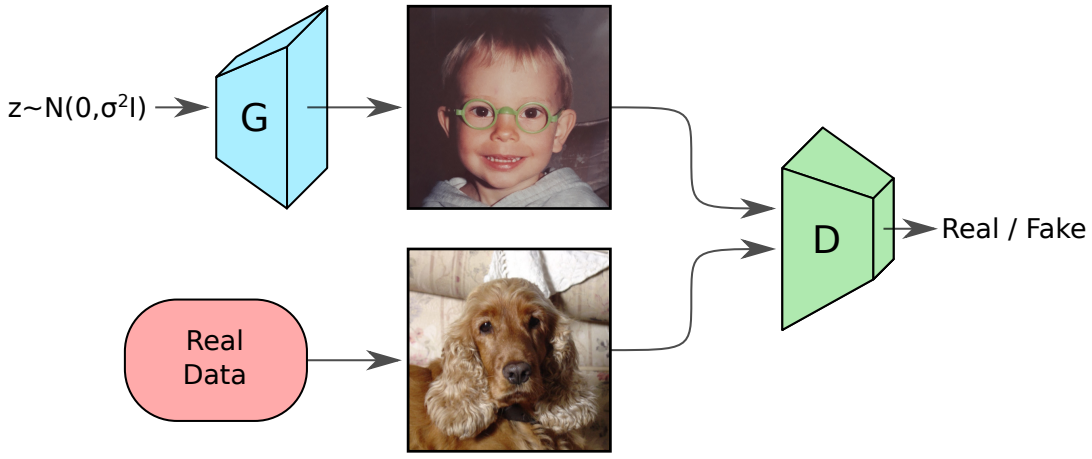


Figure 2.14: General Adversarial Networks are trained by forcing two networks to accomplish contradictory tasks. The generator G attempts to generate realistic images indistinguishable from real samples, while the discriminator D attempts to separate fake from real samples.

usually limited by the variety in the examples used for training. Much work is being done to improve the performance of such network to unseen situations without supervision labels (unsupervised domain adaptation, or UDA) [HWYD16, THSD17, HTP+18, KZYY18, VJB+19, STDE19]. Such methods can help mitigate the effects of seasonal or geographical changes, which can hinder the performance of CNNs. They are also used to bridge the domain gap between synthetic and real data, since it is usually much cheaper to generate large amounts of synthetic data for training deep neural networks, but networks trained only on synthetic data with no UDA techniques often perform badly when tested on real data.

Adversarial Learning

In 2014, Goodfellow et al. proposed the idea of generative adversarial networks (GANs) [GPAM+14], which can be seen as a special case of the ”adversarial artificial curiosity” according to Jürgen Schmidhuber [Sch19]. The initial objective of GANs were to train generative models that were able to translate noise into realistic images. This could not be done through normal supervision, since there is no ”correct” translation from noise to image. Training was done by pitting two networks against each other, as seen in Fig. 2.14, using contradictory or adversarial loss functions. The generator attempts to learn to generate realistic images from input noise in a way that such images trick the discriminator, while the discriminator's task is to correctly identify if an input image is real or was generated by the generator. The training is done my solving a minimax problem by iteratively performing gradient steps on the generator and the discriminator parameters. If properly trained, the generator and the discriminator have a symbiotic relationship where one's good performance helps the other to improve.

There have since been many improvements to this basic idea. Conditional GANs were used to generate images of specific classes [MO14], as well as to perform semantic semantic segmentation [LCCV16]. Adversarial training has since been used for many other applications such as face editing [KLA19, YPN+20], image-to-image

translation [IZZE17, ZPIE17], domain adaptation [HTP+18, VJB+19], and several others.

The training of GANs is notoriously unstable [SGZ+16]. If either the generator or the discriminator becomes much more powerful than the other, the training tends to degenerate and produce useless results through mode collapse. Significant work has been put into improving the training of GANs. LSGAN [MLX+17] and Wasserstein GAN [ACB17] tackle the problem of vanishing gradients in the discriminator. The pix2pix work proposed a dense discriminator that predicts a label for all overlapping 70x70 patches, producing more robust results [IZZE17]. The CycleGAN framework proposed a two-way translation coupled with a reconstruction loss that is very effective against mode collapse. Research into the stability and usage of such adversarial networks is still of high interest.

2.3 Change Detection Using Remote Sensing Images

As was introduced in Chapter 1, remote sensing is a vast field of study with various applications [EVZ06, CW11, Che12]. While remote sensing can be generally defined as "the acquisition of information about an object without being in physical contact with it" [EVZ06], the works in this thesis is done considering the more restricted definition of remote sensing that deals with the acquisition, processing and analysis Earth[7] observation images taken from an overhead perspective.

Such remote sensing imaging techniques enable us to accurately and frequently survey large areas. These images can be combined with geographical information systems (GIS) and global positional systems (GPS) to create what is usually referred to as geospatial data [CW11]. Understanding the geographical area that is imaged by remote sensing systems is essential to interpreting a set of images together, as well as for correlating different images that were taken of the same region at different times of by different sensors.

Remote sensing images can have many types. Panchromatic images are grayscale photos that capture a broad range of wavelengths, typically at a higher resolution than other forms of image acquisition. High resolution panchromatic images are often combined with lower resolution multispectral images (including RGB) to produce high resolution multispectral images through the process of pansharpening [Che12]. The spectrum that is observed may be restricted to the human-visible spectrum (i.e. RGB images) or not (multispectral and hyperspectral images). Active methods such as SAR and LiDAR, where the object's response to an emitted electromagnetic signal is observed, are also commonplace in remote sensing. The studies in this thesis consider only RGB and multispectral images, but there is no reason to believe that the proposed ideas can't be used for other types of images.

Remote sensing image analysis has long used techniques and methods developed for generic image analysis and signal processing [Che12]. Recently, with the rise of deep learning techniques, these have also made their way into the field of remote sensing to solve problems such as road detection [MH10], automatic land classification and cartography [Aud18, WZL+19], and many others [ZTM+17].

[7] Although there is no reason such techniques can be used for other astronomical bodies [MJ89].

Unlike several other problems in computer vision where it can be easy to define a well behaved use case for algorithms, remote sensing image analysis inherits the heterogeneity from the real world. Images can change drastically depending on seasonal variations, weather patterns, acquisition methods, and several other factors. Handling these variations is one of the core challenges in remote sensing image analysis.

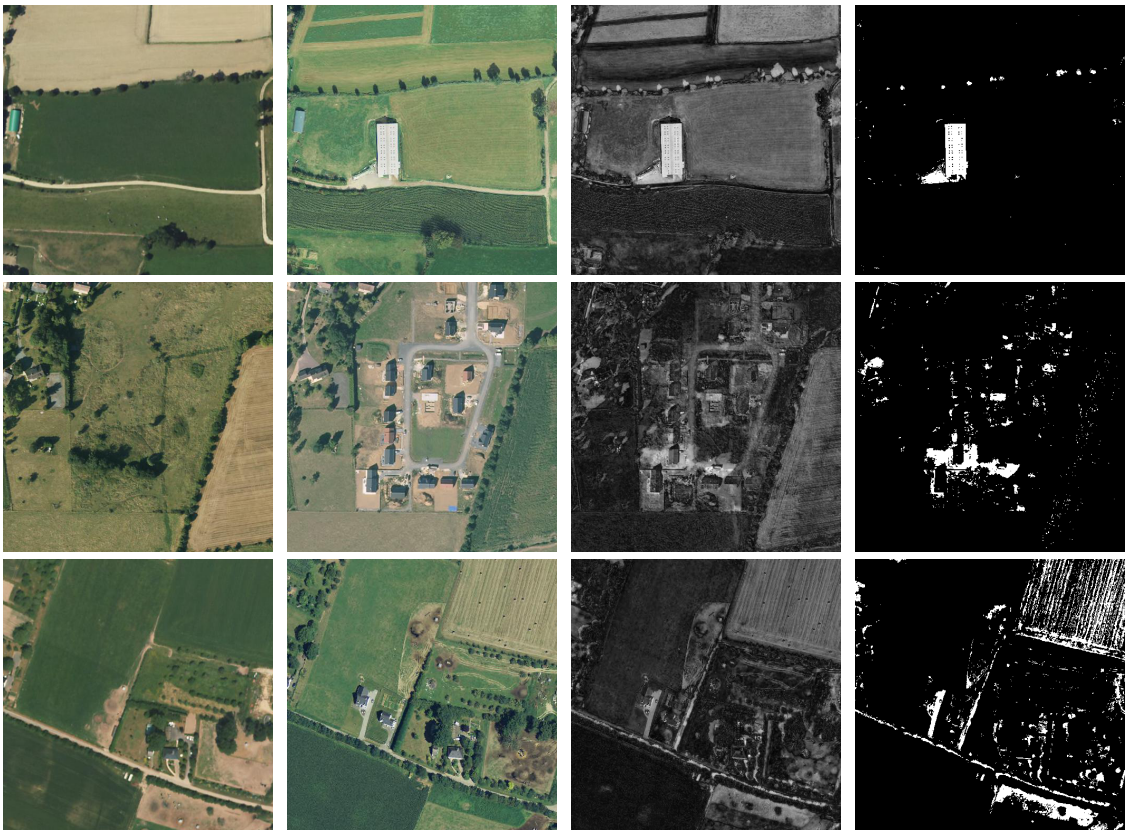


Figure 2.15: Naive change detection methods for producing and thresholding difference images tend to detect either too few or too many changes. Humans are very good at understanding difference based on the context, but the direct pixel colour difference is usually much less discriminative than our intuition suggests. Adaptive thresholding algorithms also assume there are changed and unchanged pixels in the image (often in comparable quantities), which is sometimes a flawed assumption, especially for the automated analysis of large image collections.

2.3.1 Standard Approaches for Change Detection

The ability to correlate information from different images of a given area taken at different times (and possibly by different sensors) allows the study of such images to yield a temporal understanding of the evolution of the imaged area. One such problem that has been researched for many years in this field is change detection, whose study is almost as old as remote sensing itself [Sin89, CJN+04, HCC+13, TCT+15, YCZ20, Alb18]. Change detection is the name given to the task of identifying areas of the Earth's surface that have experienced changes by jointly analysing two or more coregistered images [BB13]. Changes can be of several different types depending on the desired application, e.g. those caused by natural disasters, urban expansion, polar ice melting, deforestation, and

others [XLL+19, GGP+19].

The search for ever more accurate change detection comes from the value of surveying large amounts of land and analysing its evolution over a period of time. Detecting changes manually is a slow and laborious process [Sin89] and the problem of automatic change detection using image pairs or sequences has been studied for many decades. Binary change detection attempts to identify which pixels correspond to areas where changes have occurred, whereas semantic change detection attempts to further identify the type of change that has occurred at each location.

Many change detection algorithms comprise two main steps [Sin89, HCC+13]. First, a difference metric is proposed so that a quantitative measurement of the difference between corresponding pixels can be calculated. The image generated from this step is usually called a difference image. Second, a thresholding method or decision function is proposed to separate the pixels into ”change” and ”no change” based on the difference image. These two steps are usually independent. Post-processing and pre-processing methods are sometimes used to improve results. Many algorithms use out-of-the-box registration algorithms and focus on the other main steps for change detection, or consider that the provided images have already been coregistered [HCC+13]. Most papers on change detection propose either a novel image differencing method [BB05, EALW16, EALW17, ZFY+17] or a novel decision function [BP00, Cel09]. A well established family of change detection methods is change vector analysis (CVA), considering the multispectral difference vector in polar or hyperspherical coordinates and attempting to characterise the changes based on the associated vectors at each pixel [LS94, BB07, HCC+13], which has also been done using CNNs [SBB19a]. The authors of [HCC+13] and [RI03] noted that the performance of such algorithms is scene dependent. *A contrario* modelling has also been used to perform change detection by studying the distribution statistics of changes [LGT19a]. Demir et al. prosed in [DBB13] an active learning method for transferring labels from a source to a target domain. Most methods that propose image differencing techniques followed by thresholding assume that a threshold is chosen based on the difference image. Paris et al. tackled a similar problem by performing domain adaptation to update vector maps using multispectral images [PBF18].

[HCC+13] categorize change detection algorithms into two main groups: pixel based and object based change detection. The former attempt to identify whether or not a change has occurred at each pixel in the image pair, while the latter methods attempt to first group pixels that belong to the same object and use information such as the object’s colour, shape and neighbourhood to help determine if that object has been changed between the acquisitions.

As noted in [HCC+13, BB13], change detection on low resolution images and on very high resolution (VHR) images face different challenges. In low resolution images, pixels frequently contain information about several objects contained within its area. In such cases, a pixel in an image pair may contain both changed and unchanged surfaces simultaneously. VHR images are more susceptible to problems such as parallax, high reflectance variability for objects of the same class, and co-registration problems [BB13]. It follows that algorithms that perform

change detection on very high resolution images must be aware of not only a given pixel's values, but also of information about its neighbourhood. Several change detection algorithms have been developed specifically for VHR images [WLPS18, CWDZ19, FPK19]

Machine learning algorithms, and notably convolutional neural networks (CNNs) in recent years, also have had great impact on change detection research. CNNs were used on the related task of comparing image pairs for different tasks, usually using a Siamese architecture [CHL05b, ZK15b].

2.3.2 Unsupervised Change Detection

Change detection methods with unsupervised parameter estimation have been used in many different ways [Nie07, HCC$^+$13, VKKP15, LGT19b]. In the context of change detection, annotated datasets are extremely scarce and often kept private. Thus, unsupervised methods are extremely useful, since, unlike supervised methods, they do not need labelled data for training. Many of these methods automatically analyse the data in difference images and detect patterns that correspond to changes [BBM05, BP00]. Other methods use unsupervised learning approaches such as iterative training [LGQZ18], autoencoders [ZGLJ14], and principal component analysis with k-means clustering [Cel09] to separate changed pixels from unchanged ones.

Unsupervised change detection methods often utilize some handcrafted heuristic coupled with unsupervised learning to compare descriptors calculated from different images. Several methods use autoencoders to calculate feature vectors for each pixel, which are then compared to performed change detection [SBB20, SSBB19, SBB19b, ZGZ$^+$19, BSBB19]. Other accomplish the same task using transfer learning approaches that use CNNs trained for different tasks [EALW16, EALW17, SSBB20, SBB18, YJL$^+$19]. GANs were also used to obtain pixel-level features for change detection [SBB19b, LKB$^+$20, LHK$^+$20]. Alvarez et al. proposed the S^2-cGAN for self supervised change detection, where an image translation algorithm is learned using unchanged images and the discriminator is used to detect changes by identifying pixels that differ from the expected translated values [ARD20]. Luppino et al. noticed that the changed pixels can have a negative impact in adversarial learning, and worked on mitigating their effects during the training process using a co-attention mechanism [LKB$^+$20, LHK$^+$20], an idea which has also been explored by Jiang et al. [JHL$^+$20]. Adversarial learning was also used by Roy et al. in [RSSD18], which allowed a significant reduction in the amount of data necessary for training the proposed networks.

2.3.3 Supervised Change Detection

Supervised change detection algorithms require labeled training data from which the task of change detection can be learned. Several methods have been proposed for performing change detection using supervised learning algorithms such as support vector machines [HSK$^+$08, VTK$^+$09, VTB$^+$13, LSR13], random forests [SGFT08], and neural networks [GW96, DK99, ZGLJ14]. CNN architectures have also been proposed to perform supervised

change detection [ZFY+17, COA18].

Convolutional neural networks for change detection have been proposed by different authors in the recent years. Many methods avoid the problem of the lack of data by using transfer learning techniques, i.e. using networks that have been pre-trained for a different purpose on a large dataset [EALW16, EALW17]. While transfer learning is a valid solution, it is also limiting. Firstly, end-to-end training tends to achieve the best results for a given problem when possible [GBC16]. Transfer learning also assumes that all images are of the same type. As most large scale datasets contain RGB images, this means that extra bands contained in multispectral images must be ignored. It will however be shown in Chapter 3 that using all available multispectral bands for change detection leads to better results.

Several works have used CNNs to generate the difference image that was described earlier, followed by traditional thresholding methods on those images. [ZFY+17] trained a network to produce a 16-dimensional descriptor for each pixel. Similarly, the work by Mesquita et al. used contrastive loss in FCNs for this purpose using an encoder-decoder architecture, which makes the pixel descriptors be close in a higher dimensional space where no changes have ocurred and far where changes have ocurred [MdM+19]. Descriptors were similar for pixels with no change and dissimilar for pixels that experienced change. [LGQZ18] used deep belief networks to generate pixel descriptors from heterogeneous image pairs, then the Euclidean distance is used to build a difference image. [ZGLJ14] proposed a deep belief network that takes into account the context of a pixel to build its descriptor. [MBZ19] proposed using patch based recurrent CNNs to detect changes in image pairs. CNNs for change detection have also been studied outside the context of remote sensing, such as surface inspection [SGSC15].

FCNs currently achieve state-of-the-art results in semantic segmentation problems, including those in remote sensing [VT17, MTCA17, CWS+18]. Fully convolutional networks trained from scratch to perform change detection were proposed for the first time during the works presented in this thesis, and were published simultaneosly with [COA18] where a similar idea was explored outside the context of remote sensing. A similar architecture was subsequently used by Kolos et al. with the addition of residual blocks [KMAB19]. Both Siamese and early fusion architectures were compared, expanding on the ideas proposed earlier by [CHL05b] and [ZK15b]. The only other time a fully convolutional Siamese network had been proposed previously was by by Bertinetto et al. in [BVH+16] with the purpose of tracking objects in image sequences.

Coupling change detection with a semantic understanding of the detected changes has been explored by Mou et al. using recurrent neural networks [MBZ19]. RNNs were also used by Papadomanolaki et al. to improve the accuracy of change detection networks with respect to bitemporal networks [PVV+19]. Sakurada et al. proposed a way to generate synthetic semantic change data for this purpose using semantically labelled images for weakly supervised semantic change detection [SSW18]. The works by Suzuki et al. and Sakurada et al. focus on the case of street view images, but the techniques are very similar to those used in remote sensing [SSM+16, SSW18].

2.4 Evaluation Metrics

It is important to define quantitative measures that allow us to precisely evaluate the performance of algorithms. In the binary classification case, consider $\mathbf{y}_n \sim \mathbf{Y}$ to be the n-th label in the dataset and $\hat{\mathbf{y}}_n \sim \hat{\mathbf{Y}}$ to be the n-th label predicted by the considered algorithm, $\mathbf{y}_n, \hat{\mathbf{y}}_n \in \{0, 1\}$. Each of the four possible combinations is given, as described in Tab. 2.1. An illustration applied to change detection can be seen in Fig. 2.16. The matrix that is represented in Tab. 2.1 is often referred to as the confusion matrix, and it can be extended to a N-class problem by defining a square matrix $\mathbf{C}_{N\times N}$ where $c_{i,j}$ represent the total number of elements where $\mathbf{y} = i$ and $\hat{\mathbf{y}} = j$.

	$\mathbf{y}_n = 0$	$\mathbf{y}_n = 1$
$\hat{\mathbf{y}}_n = 0$	True negative (TN)	False negative (FN)
$\hat{\mathbf{y}}_n = 1$	False positive (FP)	True positive (TP)

Table 2.1: Two class confusion matrix.

Using the number of elements in each group we can calculate useful metrics that let us evaluate the performance of different methods. The first is the total accuracy of the algorithm, which can be defined as

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}. \tag{2.15}$$

Accuracy (also referred to as overall accuracy, global accuracy, etc.) represents the fraction of correctly classified examples considering the whole dataset. The accuracy can range between 0 and 1, with 1 being the best possible value. Accuracy is a class agnostic metric, and can be applied to cases with any number of classes.

Precision is the measurement of how often are positive predictions correct. It can be defined as

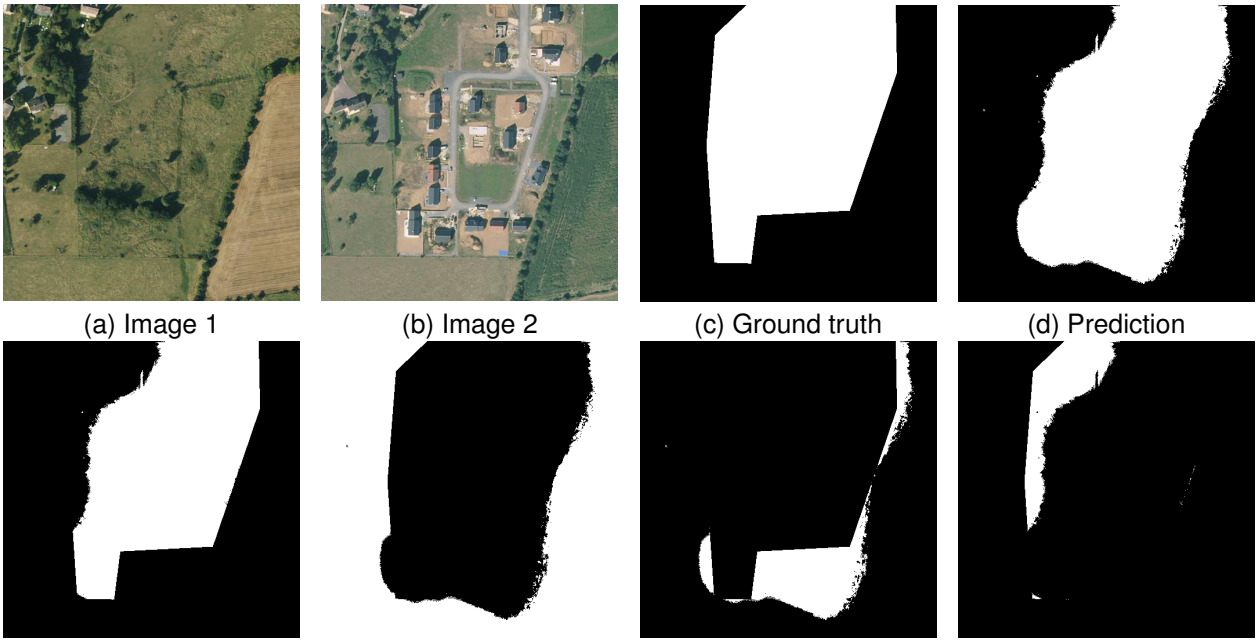
$$Precision = \frac{TP}{TP + FP}. \tag{2.16}$$

High precision scores means we can be highly confident that the ”positive” predictions are correct, but it does not provide any information about how many positives have not been detected by the algorithm. Precision scores can range between 0 and 1, with 1 being the best possible value. Precision is not a class agnostic metric, and in its simplest formulation is restricted to binary classification problems.

Recall is the measurement of how many of the positive examples have been detected by the classification algorithm. It can be defined as

$$Recall = \frac{TP}{TP + FN}. \tag{2.17}$$

High recall scores means we can be highly confident that most of the positive examples are being detected, but it does not provide any information about whether or not the algorithm is also producing many false positives. Recall scores can range between 0 and 1, with 1 being the best possible value. Recall is not a class agnostic metric, and



(a) Image 1 (b) Image 2 (c) Ground truth (d) Prediction

(e) True positives (f) True negatives (g) False positives (h) False negatives

Figure 2.16: Example of a binary classification evaluation at pixel-level applied to change detection. In (e)-(h), the elements of each group are represented in white.

in its simplest formulation is restricted to binary classification problems.

The Dice score (also referred to as F1 score, or Sørensen-Dice coefficient) is a balance between the precision and recall values, and aims to be the a balanced metric of the performance of classification algorithms. It can be defined as the harmonic mean between precision and recall:

$$Dice = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}} = \frac{2 \cdot TP}{2 \cdot TP + FP + FN}. \tag{2.18}$$

A low score in either precision or recall will bring down the Dice score, which makes it a robust metric for quantifying the performance of classification algorithms. It is also better than the accuracy metric for evaluating algorithms in cases where there is a strong class imbalance, since the accuracy metric will be strongly biased towards the class with more examples. Dice scores can range between 0 and 1, with 1 being the best possible value.

The intersection over union (IoU) serves the same purpose as the Dice score, serving as a single metric that is lowered by the presence of either false positives or false negatives, while being more robust to class imbalances than overall accuracy. It can be defined as

$$IoU = \frac{TP}{TP + FP + FN}. \tag{2.19}$$

IoU scores can range between 0 and 1, with 1 being the best possible value. IoU is often used to evaluate multi-class classification and semantic segmentation problems by applying it to each class and averaging the results. This is

referred to as mean IoU (mIoU).

The kappa coefficient (or Cohen's kappa coefficient) is a measurement of how well the true and predicted classes agree when compared to the probability of agreement given the class distributions. It can be defined as

$$\kappa = \frac{p_o - p_e}{1 - p_e}, \tag{2.20}$$

where p_o is the observed agreement probability and p_e is the expected agreement probability given the class distributions. In the two class problem, these can be defined as

$$p_o = Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \tag{2.21}$$

and

$$p_e = \frac{(TP + FP)(TP + FN) + (TN + FP)(TN + FN)}{(TP + TN + FP + FN)^2}. \tag{2.22}$$

The kappa coefficient can also be directly applied in a multiclass setting. Kappa scores can range between -1 and 1, with 1 being the best possible value.

Chapter 3

Supervised Change Detection

Chapter Summary

This chapter presents a complete framework for performing change detection with convolutional neural networks (CNNs) and fully convolutional networks (FCNs) in a supervised manner. Two CNN architectures and three FCN architectures are proposed and compared using a novel dataset that was manually annotated for this purpose. Results show that both Siamese and early fusion architectures are able to learn the necessary operations to perform change detection. The superiority of FCNs over patch-based CNNs in both performance and speed for such pixel-level tasks is clear in the presented results.

The dataset that was manually annotated for these tests contains Sentinel-2 multispectral images, which contains infrared and ultraviolet bands along with colour channels. We explore the usage of these bands for change detection, and observe the effect of using different numbers of spectral bands against the RGB baseline.

At the time this work was done, and to the best of our knowledge, it was the first time CNNs and FCNs were used for change detection in an end-to-end supervised manner. Previous work had utilised such networks simply for the creation of difference images or change vectors, which were then used along with more traditional methods to obtain the final change maps [EALW16, EALW17, ZFY+17]. Such methods tend to assume that there are both changed and unchanged pixels in the image, and will perform poorly in cases where no changes are visible. As will be shown in Chapter 4, this is often the case when attempting to automatically analyse large areas, especially at higher resolutions.

The final contribution presented in this chapter is the multispectral change detection dataset that was created by manually annotating selected Sentinel-2 image pairs of urban areas in several different urban areas around the world. The dataset was openly released to serve as a benchmark for change detection methods and has since been used by many for developing and comparing change detection methods.

3.1 Introduction

Change Detection (CD) is one of the main problems in the area of Earth observation image analysis. Its study has a long history, and it has evolved alongside the areas of image processing and computer vision [Sin89, HCC+13]. Change detection systems aim to assign a binary label per pixel or region based on a pair or sequence of coregistered images of a given region taken at different times. A positive label indicates the area corresponding to that pixel has changed between the acquisitions. While the definition of ”change” may vary between applications, CD is a well defined pixel-level classification problem. Changes may refer, for example, to vegetation changes, urban expansion, polar ice melting, etc. Change detection is a powerful tool in the production of maps depicting the evolution of land use, urban coverage, deforestation, and other types of multi-temporal analysis.

Programs such as Copernicus and Landsat make available large amounts of Earth observation imagery. These can be used in conjunction with advanced supervised machine learning algorithms that have been on the rise for the past decade, especially in the area of image analysis. It is therefore of interest to find efficient ways to make use of the available data. In the context of change detection there is a lack of large annotated datsets, which limits the complexity of the models that can be used. Nevertheless, there are pixelwise annotated change detection datasets available that can be used to train supervised machine learning systems that detect changes in image pairs, such as the Air Change (AC) dataset [BS09].

The history of change detection began not long after aerial images became possible [Sin89, HCC+13]. The proposed techniques have followed the tendencies of computer vision and image analysis: at first, pixels were analyzed directly using manually crafted techniques; later on, descriptors began to be used in conjunction with simple machine learning techniques [LSR13]; recently, more elaborate machine learning techniques (deep learning) are dominating most problems in the image analysis field, and this evolution is beginning to reach the problem of change detection [SGSC15, EALW16, LGQZ16, GZL+16, EALW17, ZFY+17].

Due to the limited amount of available labelled data, most of these methods use various techniques based on transfer learning, taking as a starting point networks that have been trained on larger datasets for different problems. This is limiting in many ways, as it assumes similarities between these datasets and the relevant change detection data. For example, most of the large scale networks have been trained on RGB images, and cannot be directly applied to SAR or multispectral images, which is the case of the dataset presented later in this chapter. These methods also avoid end-to-end training, which tends to have better results for successfully trained systems. For this reason, we focus on algorithms that are able to learn solely from the available change detection data, and can therefore be applied to any datasets with available supervision.

CNNs have been applied in various contexts for the comparison of image pairs [CHL05b, ZK15b, SGSC15]. Recently, FCNs have been proposed for problems that involve dense prediction, i.e. pixel-level prediction [LSD15a, RFB15, BVH+16]. Despite achieving state-of-the-art results in other Earth observation problems [ALSL17a], these

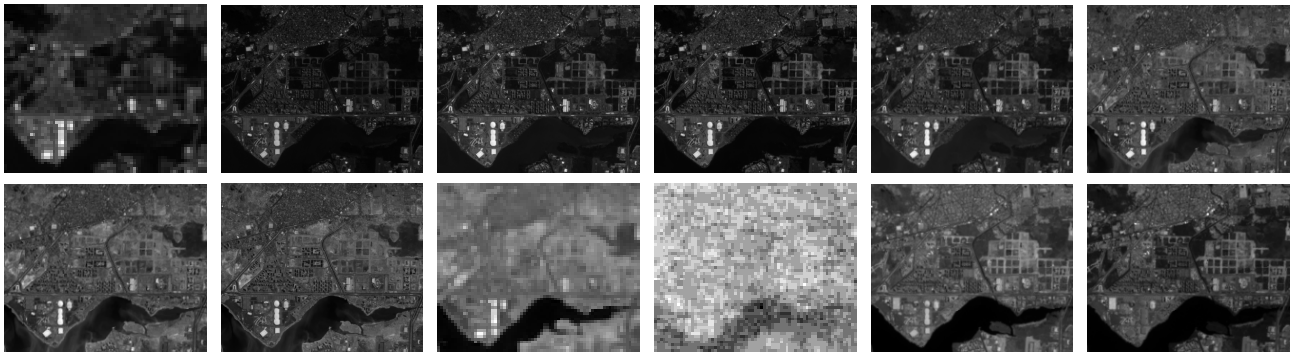


Figure 3.1: Example of Sentinel-2 image bands. Most Sentinel-2 spectral bands are of a lower spatial resolution than the visible bands, but they often contain information that can help with the analysis of the images.

techniques have not yet been applied to CD to the best of our knowledge. The usefulness of such ideas in the context of Earth observation and their superiority over patch based, superpixel based and other approaches has already been studied [ALSL17b]. Siamese architectures have also been proposed in different contexts with the aim of comparing images [BGL+94, ZK15b]. Fully convolutional Siamese architectures have been proposed for performing object tracking in videos [BVH+16] and optical flow estimation [DFI+15], but neither of these works are directly applicable to the task that is studied here.

3.2 ONERA Satellite Change Detection Dataset

With the rise of open access earth observation from programs such as Copernicus and Landsat, large amounts of unlabelled data are available to be used for different applications. The Sentinel-2 satellites generate time series of multispectral images of Earth's landmasses with resolutions varying between 10m and 60m per pixel. Despite the abundance of raw data, there is a lack of open labelled datasets, which are necessary for developing supervised learning methods. Deep learning techniques and CNNs have been on the rise not only due to the exponential growth of the available computing power, but also to the increasingly large amounts of data that is available. The application of these techniques to the problem of change detection is limited while there is a lack of data that can be used for training, testing and comparing these systems.

The lack of common evaluation datasets makes it difficult to quantitatively compare change detection methods. In this section, we present a new change detection dataset that includes manually annotated change maps, aiming to provide a solution to this problem. The objective of this urban change detection dataset is to provide an open and standardized way of comparing the efficacy of different change detection algorithms that are proposed by the scientific community, available to anyone who is interested in tackling the change detection problem[1]. The dataset is focused on urban areas, labelling as "change" only urban growth and urban changes, while ignoring natural changes (e.g. vegetation growth or sea tides).

[1] https://ieee-dataport.org/open-access/oscd-onera-satellite-change-detection

The dataset provides a comparison standard for single band, color or multispectral change detection algorithms that are proposed. Since it contains pixel-wise ground truth change labels for each location on each image pair, the dataset also allows for more elaborate supervised learning methods to be applied to the problem of change detection.

The ONERA Satellite Change Detection (OSCD) dataset was built using images taken by the Sentinel-2 satellites, which belong to the Copernicus program. The satellites capture images at various resolutions between 10m and 60m in 13 bands between ultraviolet and short wavelength infrared. Twenty-four regions of approximately 500x500 pixels at 10m resolution with various levels of urbanization where urban changes were visible were chosen in various countries and continents. The images of all bands were cropped according to the chosen geographical coordinates, resulting in 26 images for each region, i.e. 13 bands for each of the images in the image pair. These images were downloaded and cropped using the Medusa toolbox[2].

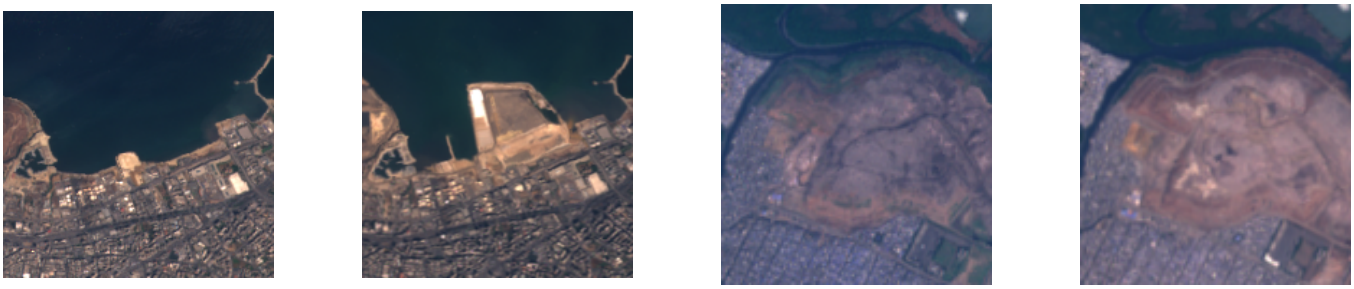
The high variability of the raw data that is available from the Sentinel-2 does not allow a completely scripted generation of image patches. The downloaded images frequently contain large sections of completely black pixels, and the correct images must be selected manually. Furthermore, for the generation of this dataset, it was desired to obtain images with no or very few clouds present in the image. While the *sentinelsat* API allows some control over the amount of clouds present in the images, this also requires manual verification of each of the downloaded images to ensure the presence of clouds in the downloaded image is not too large. The pixel-wise ground truth labels were manually generated by comparing the true color images for each pair.

Since the creation of this dataset, Papadomanolaki et al. have created an extended version of this dataset by adding appropriately cropped Sentinel-2 images that were acquired between the two dates for each of the regions [PVV+19]. This addition extends the reach of the dataset beyond bitemporal image analysis, allowing a multitemporal analysis of image sequences to be performed.

3.2.1 Challenges and Limitations

While the dataset is a very valuable tool for methodically comparing different change detection algorithms and for applying supervised learning techniques to this problem, it is important to understand the limitations of this dataset. First and foremost, the images generated by the Sentinel-2 satellites are not of very high resolution (10–60 m/px depending on the spectral band). This resolution allows the detection of the appearance of large buildings between the images in the image pair. Smaller changes such as the appearance of small buildings, the extension of existing buildings or the addition of lanes to an existing road, for example, may not be obvious in the images as they may occupy a single pixel. For this reason, even the change maps that are generated manually often differ according to its maker.

[2] `https://github.com/aboulch/medusa_tb`



(a) Beirut 1 (b) Beirut 2 (c) Mumbai 1 (d) Mumbai 2

Figure 3.2: Sometimes, such as between images (a) and (b), it is clear to say whether or not changes have occurred and where they are located. In other cases, such as between images (c) and (d), it may not be clear where the changes are or even if changes have occurred at all.

(a) Image 1 (b) Image 2 (c) Change maps

Figure 3.3: Example of change maps between images (a) and (b) manually annotated by three different people, represented by the three colour channels in (c). Even human analysts sometimes disagree at what regions have been changed, especially around the boundaries between changed and unchanged regions.

Figures 3.3 and 3.2 show the difficulty in accurately defining and labelling changes in image pairs. In Fig 3.3(c), we can see the change maps done by three different analysts represented in different color channels. While there is much agreement between the images, there is also a significant amount of disagreement. This difference comes from the difficulty in finding a clear definition for change that covers all situations, even when the images are being manually labelled. Some differences also come from slight differences on the exact locations of the boundaries of the changes. This means that it may not be reasonable to expect an algorithm to reach perfect quantitative results.

One approach which was explored for generating change maps in an automated manner was comparing OpenStreetMap data from different dates. OpenStreetMap provides open map data, and by comparing the maps for the dates of the images in the pairs it is, in theory, possible to identify what changes occurred in the area. This approach proved unsuccessful for a few reasons. First, most of the changes in the maps between the two dates were actually due to things being added to the map, but which had not been built in the period between the dates when the images were taken. Second, it is not possible to have much precision when it comes to the dates of the older maps, where in many cases only one map was available for each year before 2017.

Finally, the Sentinel-2 satellite was launched in 2015, and therefore the data that is available is not able to go further in the past than June of 2015. This means that the changes contained in the dataset are of a temporal

(a) Image 1 (b) OSM 1 (c) Image 2 (d) OSM 2 (e) True CM (f) OSM CM

Figure 3.4: Comparing the OpenStreetMap semantic maps (b) and (d) relative to the images (a) and (c) leads to a change map (f) that does not resemble that which is produced by visual analysis (e).

distance of at most two years approximately since the dataset was created in late 2017. In practice, the actual temporal distance between acquisitions is often less than that due to the limited choice in images due to weather effects. This also means that the images contain many times more pixels labelled as no change, and fewer marked as change.

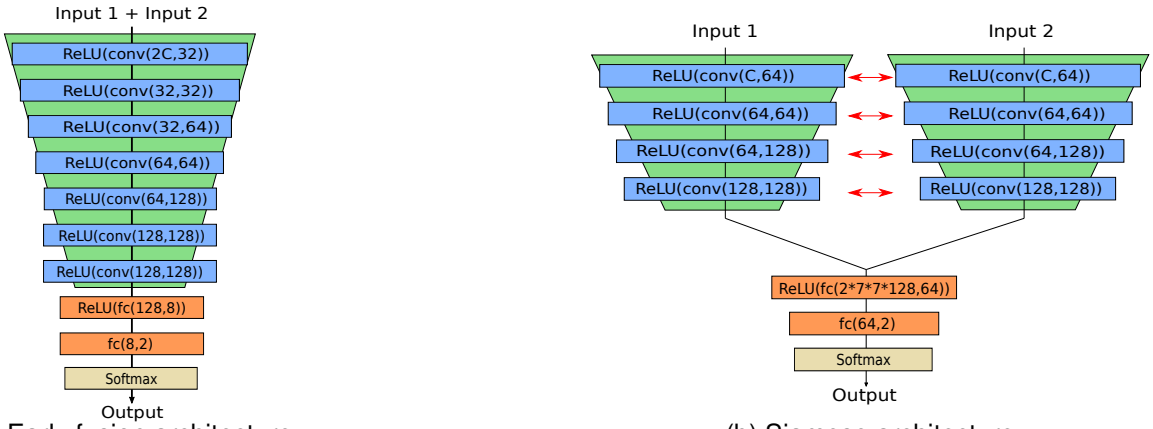
Further information about the OSCD dataset can be found in Appendix A.

3.3 Patch Based Architectures

Unlike previous methods which only use CNNs to build difference images which are later thresholded, our methods are trained end-to-end to attempt to classify a patch between two classes: change and no change. The patches are of size 15x15 pixels, and the networks attempt to classify the label of the central pixel based on its neighborhood's values. The networks should ideally be able to learn to differentiate between artificialization changes and natural changes, given that only artificialization changes are labelled as changes in the OSCD dataset. This goes further than computing a simple difference between the images, as it involves a semantic interpretation of the changes, and is therefore a harder problem.

Two patch based classification CNN architectures are proposed here. These networks take as input two 15x15xC patches, where C is the number of color channels. The output of the networks for each pair of patches is a pair of values which are an estimation of the probability of that patch belonging to each class. By choosing the maximum of these two values we are able to predict if a change has occurred in the central pixel of the patch. Furthermore, we are able to threshold the change probability at values other than 0.5 to further control the results, in case false positives or false negatives are more or less important in a given application, similarly to what is done with difference images.

The first proposed architecture, named Early Fusion (EF), works by concatenating the two image pairs as the first step of the network. The input of the network can then be seen as a single patch of 15x15x2C, which is then processed by a series of seven convolutional layers and two fully connected layers, where the last layer is a softmax layer with two outputs associated with the classes of change and no change. A schematic can be found in Fig. 3.5(a).



(a) Early fusion architecture

(b) Siamese architecture

Figure 3.5: Proposed patch based CNN architectures for change detection. Processing both images since the first layer of the network allows for more total comparison operations, but does not allow for the weight sharing properties of Siamese networks.

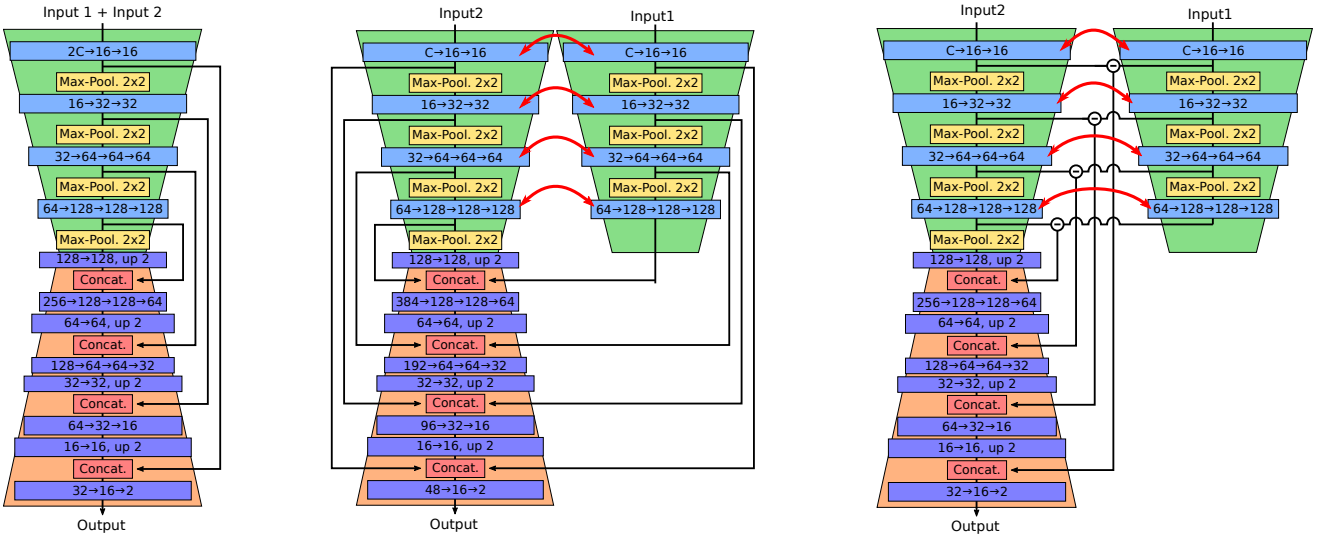
The second approach is a Siamese (Siam) network. The idea is to process each of the patches in parallel by two branches of four convolutional layers with shared weights, concatenating the outputs and using two fully connected layers to obtain two output values as before. This can be seen as projecting the images into a common latent space where a MLP can be used to detect changes. A schematic can be found in Fig. 3.5(b).

3.4 Fully Convolutional Architectures

The proposed fully convolutional architectures are an evolution of the networks presented in Section 3.3. Moving the patch-based architectures to a fully convolutional scheme improves accuracy and speed of inference without affecting significantly the training times. These fully convolutional networks are also able to process inputs of any sizes as long as enough memory is available for the computations, unlike the patch based approaches which require patches with dimensions of exactly 15x15xC.

To extend these ideas we used the concept of skip connections that were used to build the U-Net, which aimed to perform semantic segmentation of images [RFB15]. In summary, skip connections are links between layers at the same subsampling scale before and after the encoding part of an encoder-decoder architecture. The motivation for this is to complement the more abstract and less localized information of the encoded information with the spatial details that are present in the earlier layers of the network to produce accurate class prediction with precise boundaries in the output image.

The first proposed architecture is directly based on the U-Net model and on the patch based EF model, and was named Fully Convolutional Early Fusion (FC-EF). Given the amount of available training data, the original U-Net model is too complex to be directly applied to this problem. The FC-EF (Fig. 3.6(a)) contains therefore only four max pooling and four upsampling layers, instead of the five present in the U-Net model. The layers in FC-EF



(a) FC-EF (b) FC-Siam-conc (c) FC-Siam-diff

Figure 3.6: Schematics for the three FCN architectures for change detection. These architectures are U-Net inspired extensions of the patch based networks presented previously. In (b) all the activations from both streams are used for skip connections, while in (c) the magnitude of their difference is explicitly calculated. Red arrows represent shared weights between Siamese branches.

are also shallower than their U-Net equivalents. As in the patch based EF model, the input of this network is the concatenation the two images in the pair that is to be compared.

The two other proposed architectures are Siamese extensions of the FC-EF model. To do so, the encoding layers of the network are separated into two streams of equal structure with shared weights as in a traditional Siamese network. Each image is given to one of these equal streams. The difference between the two Siamese architectures is only in how the skip connections are done. The first and more intuitive way of doing that is concatenating the two skip connections during the decoding steps, each one coming from one encoding stream. This approach was named Fully Convolutional Siamese - Concatenation (FC-Siam-conc, Fig. 3.6(b)). Since in CD we are trying to detect differences between the two images, this heuristic was used to combine the skip connections in a different way. Instead of concatenating both connections from the encoding streams, we instead concatenate the absolute value of their difference. This approach was named Fully Convolutional Siamese - Difference (FC-Siam-diff, Fig. 3.6(c)).

3.5 Experiments

To evaluate the proposed methods we used two change detection datasets openly available. The first one is the proposed OSCD, and the second is the Air Change Dataset [BS09] (AC). AC contains RGB aerial images, while OSCD contains multispectral satellite images. The networks were also tested using only the RGB layers of the OSCD dataset. The two classes (change and no change) were assigned class weights inversely proportional to the number of pixels in each one. The available data was augmented by using all possible flips and rotations multiple of

90 degrees to the training patches. Dropout with $p = 0.2$ was used after each activation layer to help avoid overfitting during training. All experiments were done using PyTorch framework and with an Nvidia GTX 1070 GPU.

On the OSCD dataset, we split the data in train and test groups as defined in the dataset guidelines, i.e. 14 image pairs for training and 10 for testing. For the AC dataset, we followed the data split that was proposed in [ZFY+17]: the top left 748x448 rectangle of the Szada-1 and Tiszadob-3 images were extracted for testing, and the rest of the data for each location was used for training. This allowed direct comparison to three CD algorithms reported in [ZFY+17]. Each location (Szada and Tiszadob) was treated completely separately as two different datasets, and the images named "Archieve" were ignored, since it contains only one image pair which is not enough data to train the models presented in this chapter.

Table 3.1 contains the quantitative evaluation of the proposed CD architectures, along the same measures of other state-of-the-art methods. For the AC dataset, the methods user for comparison were DSCN [ZFY+17], CXM [BS09], and SCCN [LGQZ16], using the values claimed by Zhan et al. in [ZFY+17]. Benedek et al. proposed the CXM method that uses a conditional mixed Markov model that identify changes using complementary features of global intensity statistics and local correlation [BS09]. Liu et al. proposed the SCCN method, where a symmetrical CNN with coupling layers is used to generate pixel-level descriptors in an unsupervised fashion [LGQZ16]. The DSCN methods was proposed by Zhan et al, in which convolutions are used to project the pixels into a 16-dimensional space using a contrastive loss to lead the network to learn similar representations of unchanged pixels and distant representations of changed ones[ZFY+17]. The table contains the precision, recall and F1 scores from the point of view of the "change" class, as well as the overall accuracy.

The results on the OSCD images show that the fully convolutional methods proposed in this chapter far outperform patch based the ones. While the patch based methods achieve good recall rates due to high numbers of predicted changes, they do poorly on the precision metric, which also reduces the F1 rate. Inference time of the fully convolutional architectures was under 0.1 s per image for all our test cases, while the patch based approach took several minutes to predict a change map for the whole image. On this dataset, the FC-Siam-diff obtained significantly better F1 scores than all other proposed methods, but the other fully convolutional architectures are still clearly superior to the patch based approaches. An illustration of our results on this dataset can be found in Figs. 3.7 and 3.8.Figure 3.9 contains the results for the fully convolutional architectures compared to the annotations of three different manual annotators to illustrate the various possible interpretations.

We then compared the fully convolutional architectures to other state-of-the-art change detection methods using the Air Change dataset. The results obtained on the Air Change dataset also show the superiority of our methods compared to previous ones. For the Szada/1 case, all proposed architectures outperformed the other methods used for comparison in the F1 metric, the FC-Siam-diff being once again the best architecture. For the Tiszadob/3 case, the best F1 score was obtained by our FC-EF architecture, while the other architectures were outperformed by DSCN and SCCN. Once again the inference time of the fully convolutional architectures were below 0.1 s, which

Data	Network	Prec.	Recall	Accuracy	F1
OSCD-3 ch.	Siam.	21.57	79.40	76.76	33.85
	EF	21.56	**82.14**	83.63	34.15
	FC-EF	44.72	53.92	94.23	**48.89**
	FC-Siam-conc	42.89	47.77	94.07	45.20
	FC-Siam-diff	**49.81**	47.94	**94.86**	48.86
OSCD-13 ch.	Siam.	24.16	**85.63**	85.37	37.69
	EF	28.35	84.69	88.15	42.48
	FC-EF	**64.42**	50.97	**96.05**	56.91
	FC-Siam-conc	42.39	65.15	93.68	51.36
	FC-Siam-diff	57.84	57.99	95.68	**57.92**
Szada/1	DSCN [ZFY+17]	41.2	57.4	-	47.9
	CXM [BS09]	36.5	58.4	-	44.9
	SCCN [LGQZ16]	24.4	34.7	-	28.7
	FC-EF	**43.57**	62.65	**93.08**	51.40
	FC-Siam-conc	40.93	65.61	92.46	50.41
	FC-Siam-diff	41.38	**72.38**	92.40	**52.66**
Tiszadob/3	DSCN [ZFY+17]	88.3	85.1	-	86.7
	CXM [BS09]	61.7	93.4	-	74.3
	SCCN [LGQZ16]	**92.7**	79.8	-	85.8
	FC-EF	90.28	96.74	**97.66**	**93.40**
	FC-Siam-conc	72.07	**96.87**	93.04	82.65
	FC-Siam-diff	69.51	88.29	91.37	77.78

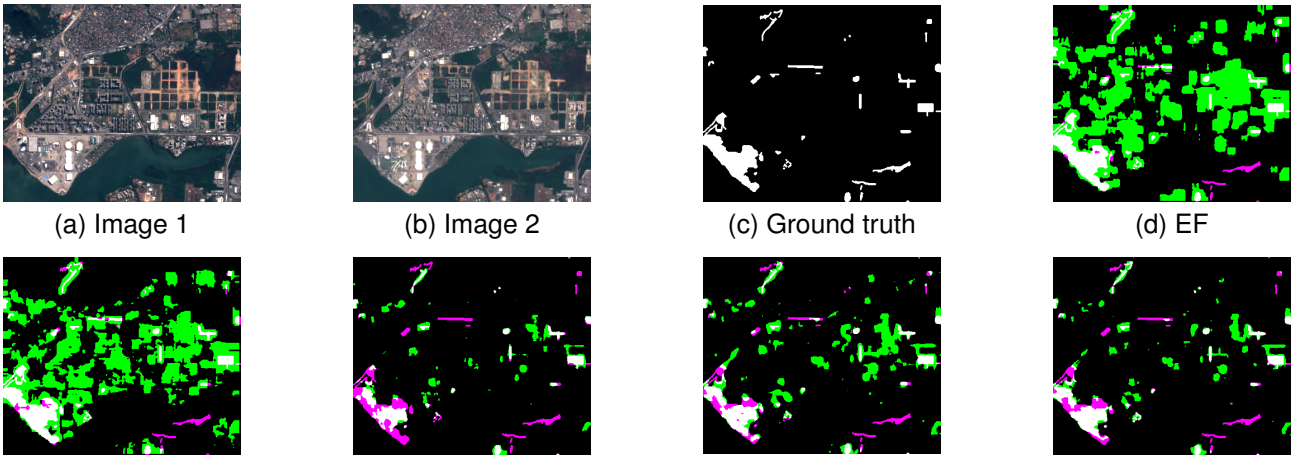
Table 3.1: Quantitative evaluation of the proposed methods on the OSCD and Air Change datasets.

is a speedup of over 500x compared to the processing time of 50 s claimed by Zhan et al. [ZFY+17] for the SCCN method on a similar setup. The results for these cases can be viewed in Figs. 3.10 and 3.11.

In these tests, the FC-Siam-diff architecture seems to be the most suited for change detection. This is, we believe, due to three main factors which make this network especially suited for this problem. First, fully convolutional networks were developed with the express purpose of dealing with dense prediction problems. Second, the Siamese architecture imbues into the system an explicit comparison between two images. Finally, the difference skip connections also explicitly guides the network to compare the differences between the images, in other words, to detect the changes between the two images.

The significant speedup of these fully convolutional networks with no loss in performance compared to patch-based methods is a step towards efficient processing of the massive streams of Earth observation data which are available through programs such as Copernicus and Landsat. These programs monitor very large areas with a high revisit rate. Deployment of systems such as these in conjunction with methods such as the ones proposed in this chapter could enable accurate and fast worldwide monitoring.

These results also validate presented dataset's capability of evaluating change detection algorithms. Despite the low number of image pairs and the small disagreements between experts when labelling some parts of the images, supervised learning was done without the need for transfer learning.



(a) Image 1 (b) Image 2 (c) Ground truth (d) EF

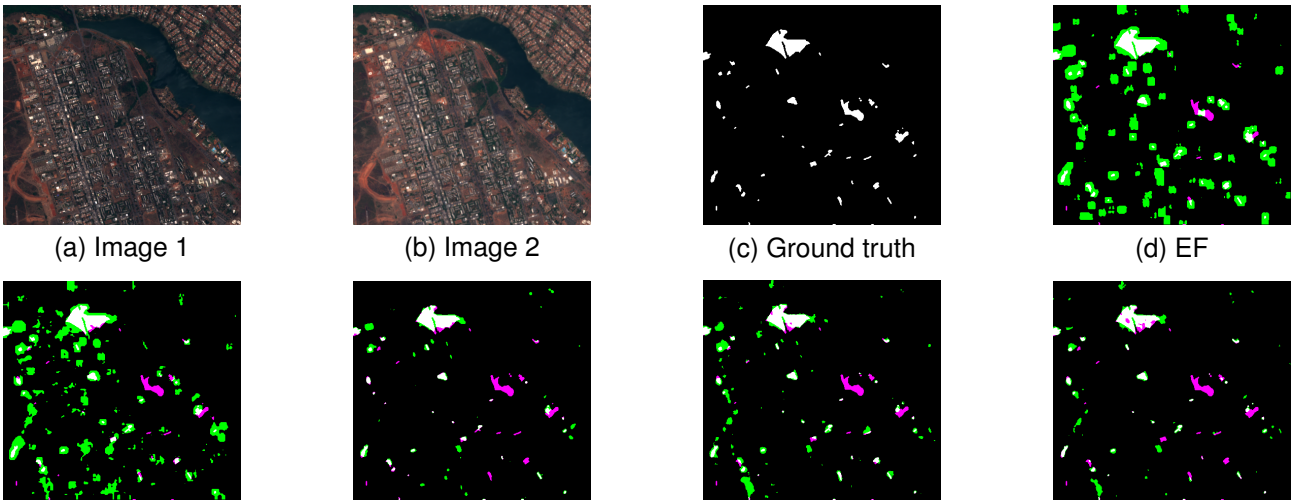
(e) Siam (f) FC-EF (g) FC-Siam-conc (h) FC-Siam-diff

Figure 3.7: Results on the OSCD *rio* image pair using all 13 available spectral channels. White represents true positives, black represents true negatives, green represents false positives, and magenta represents false negatives.

3.6 Conclusion

We presented the first Sentinel-2 pixel-level urban change detection dataset, which has been openly available, along with methods used for its generation, and the main challenges that were faced. We also presented two patch based and three fully convolutional networks trained end-to-end from scratch. The latter surpassed comparable change detection methods at the time it was developed, both in accuracy and in inference speed without the use of post-processing. Most notably, the fully convolutional encoder-decoder paradigm was modified into a Siamese architecture, using skip connections to improve the spatial accuracy of the results.

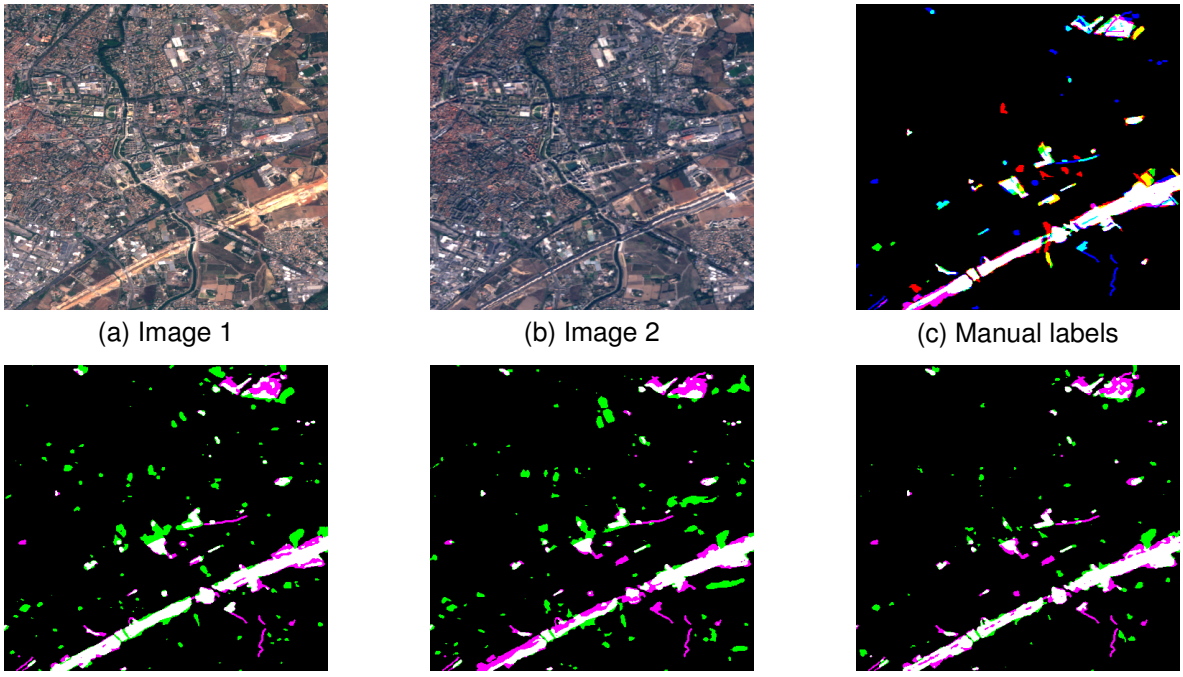
A natural extension of the work presented on this chapter would be to evaluate how these networks perform when attempting to detect semantic changes, as will be seen in the next chapter. It would also be interesting to test them with other image modalities (e.g. SAR images), and to attempt to detect changes in image sequences, as was done in [PVV+19].



(a) Image 1 (b) Image 2 (c) Ground truth (d) EF

(e) Siam (f) FC-EF (g) FC-Siam-conc (h) FC-Siam-diff

Figure 3.8: Results on the OSCD *brasilia* image pair using all 13 available spectral channels. White represents true positives, black represents true negatives, green represents false positives, and magenta represents false negatives.



(a) Image 1 (b) Image 2 (c) Manual labels

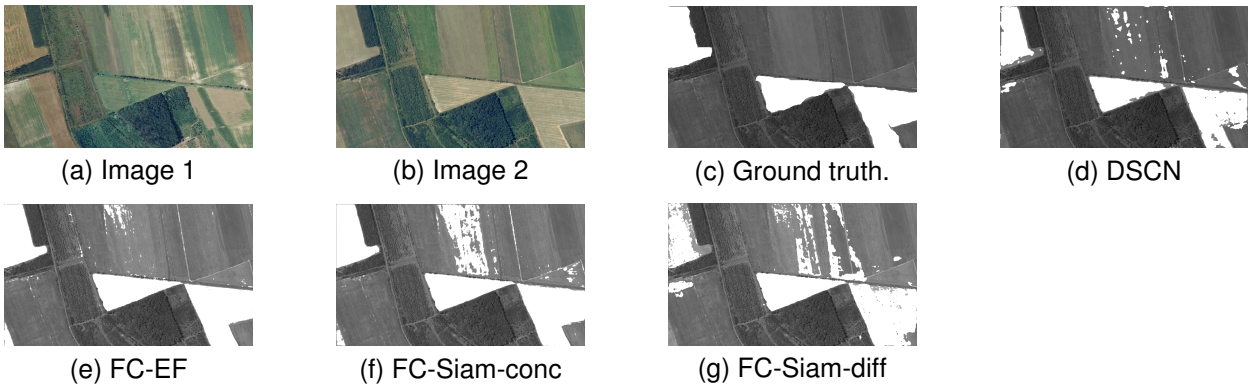
(d) FC-EF (e) FC-Siam-conc (f) FC-Siam-diff

Figure 3.9: Illustrative results on the *montpellier* test case of the OSCD dataset using all 13 color channels. In (d), each colour channel represents the manual annotations by a different user. In images (d), (e), and (f) white means true positive, black means true negative, green is false positive, and magenta is false negative.

(a) Image 1 (b) Image 2 (c) Ground truth (d) DSCN

(e) FC-EF (f) FC-Siam-conc (g) FC-Siam-diff

Figure 3.10: Comparison between (d) the results obtained by the method presented By Zhan et al. in [ZFY+17] and (e-g) the proposed fully convolutional networks on image Szada/1 from the Air Change dataset. Changes are marked in white.



(a) Image 1 (b) Image 2 (c) Ground truth. (d) DSCN

(e) FC-EF (f) FC-Siam-conc (g) FC-Siam-diff

Figure 3.11: Comparison between (d) the results obtained by the method presented By Zhan et al. in [ZFY+17] and (e-g) the proposed fully convolutional networks on image Tiszadob/3 from the Air Change dataset. Changes are marked in white.

Chapter 4

Semantic Change Detection

Chapter Summary

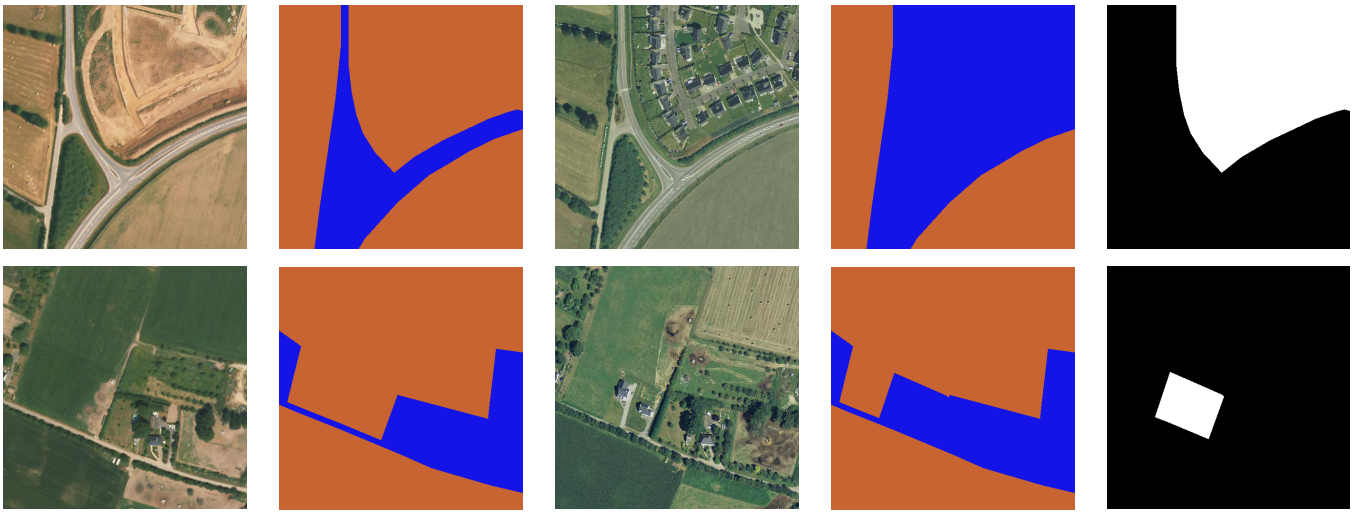
The work presented in this chapter expands the fully convolutional network (FCN) architectures presented previously to allow them to understand as well as detect changes in image pairs. This semantic interpretation of image pairs enable a deeper understanding about the evolution of the imaged terrain. Instead of simply separating changed from unchanged regions, such systems could be used to answer questions such as:

- How much has the urban area expanded into agricultural land?
- How much deforestation has occurred?
- How much water surface loss has happened in this region?
- How much have the ice shelves in these images shrunk?

To enable the learning of such semantic understanding of multitemporal image pairs, a novel very high resolution semantic change detection dataset was created by combining two available databases: one which contained aerial images, and another which contained land cover and change vector annotations. This allowed for the creation of a much larger dataset than those which were previously available, and which contains change maps and land cover maps for all regions in the dataset. This dataset was also released to the scientific community.

Different methods are proposed and tested to solve this semantic change detection problem. These solutions range from comparing separately predicted land cover maps for changes, to proposing a single multi-class semantic segmentation network, and finally a multitask network that produces simultaneously a land cover map for each input image, as well as a binary change map. A staged training scheme is also proposed for training this last proposed network, which not only avoids setting a hyperparameter to balance different loss functions, but also leads to better results.

4.1 High Resolution Semantic Change Detection Dataset



(a) Image 1 (b) LCM 1 (c) Image 2 (d) LCM 2 (e) Change map

Figure 4.1: Examples of image pairs, land cover maps (LCMs) and associated pixel-wise change maps from the HRSCD dataset. In the depicted LCMs, blue represents the ”artificial surfaces” class, and orange represents the ”agricultural areas” class.

Research on the problem of change detection is hindered by a lack of open datasets. Such datasets are essential for a methodical evaluation of different algorithms. [BS09] created a binary change dataset with 13 aerial image pairs split into three regions called the Air Change dataset. The ONERA Satellite Change Detection (OSCD) dataset, presented in the previous chapter, is composed of 24 multispectral image pairs taken by the Sentinel-2 satellites. The Aerial Imagery Change Detection (AICD) dataset contains synthetic aerial images with artificial changes generated with a rendering engine [BDS11]. These datasets allow for CNN-based techniques to be applied to the problem of change detection, but the size and complexity of the models are limited by the size of the datasets. They also do not contain semantic information about the land cover of the images, and contain either lower resolution (OSCD, Air Change) or simulated (AICD) images.

For this reason, we have created the first large scale dataset for semantic change detection, which we present in this section. The High Resolution Semantic Change Detection (HRSCD) dataset has been released to the scientific community to be used as a benchmark for semantic change detection algorithms and to open the doors to the usage of state-of-the-art deep learning algorithms in this context[1]. Examples of image pairs, land cover maps (LCMs) and change maps (CMs) taken from the dataset are depicted in Fig. 4.1.

[1] https://ieee-dataport.org/open-access/hrscd-high-resolution-semantic-change-detection-dataset

4.1.1 Images

The dataset contains a total of 291 RGB image pairs of 10000x10000 pixels. These are mosaics of aerial images taken by the French National Institute of Geographical and Forest Information (IGN). Each image pair contains an earlier image acquired in 2005 or 2006, and a second image acquired in 2012. They come from a database named *BD ORTHO* which contains orthorectified aerial images of several regions of France from different years at a resolution of 50 cm per pixel. The 291 selected image pairs are all the images in this database that satisfy the conditions for the labels, which will be described below. The images cover a range of urban and countryside areas around the French cities of Rennes and Caen. Only the images in the regions mapped in the Urban Atlas project and with a maximum temporal distance of one year from either 2006 or 2012 were kept in the dataset.

The dataset contains more than 3000 times more annotated pixel pairs than either OSCD or Air Change datasets. Also, unlike these datasets, the labels contain information about the types of change that have occurred. This is much more data than was previously available in the context of change detection and it opens the doors for many new ideas to be tested. The amount of labelled pixels and surface area for land cover classification is also about 8 times larger in the proposed HRSCD dataset than in the DeepGlobe Land Cover Classification dataset [DKL+18], both of the datasets containing images of the same spatial resolution (50 cm/px).

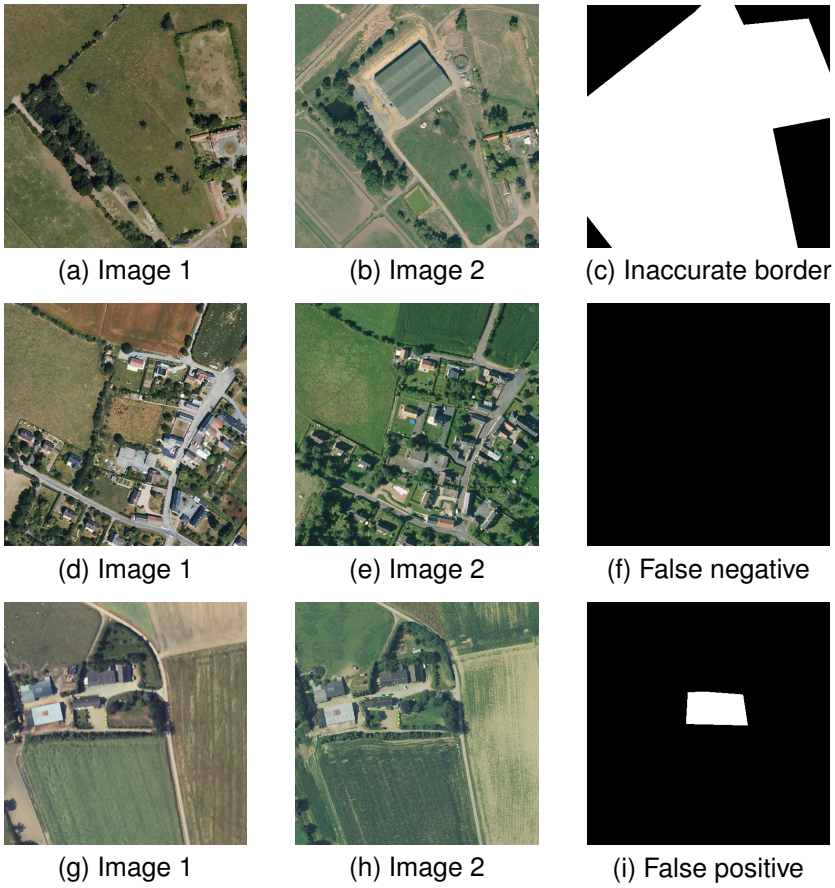
The *BD ORTHO* images provided by IGN are available for free for research purposes, but not all images can be redistributed by the users. That is the case for the images taken in 2005 and 2006. Nevertheless, we have made available all the data for which we have the rights of redistribution and the rasters that we have generated for semantic change detection and land cover mapping. The dataset also contains instructions for downloading the remaining images that are necessary for using the dataset directly from IGN's website.

4.1.2 Labels

The labels in the dataset come from the European Environment Agency's Copernicus Land Monitoring Service - Urban Atlas project. It provides "reliable, inter-comparable, high-resolution land use maps" for functional urban areas in Europe with more than 50000 inhabitants. These maps were generated for the years of 2006 and 2012, and a third map is available containing the changes that took place between those dates.

Code	Class
0	No information
1	Artificial surfaces
2	Agricultural areas
3	Forests
4	Wetlands
5	Water

Table 4.1: Urban Atlas land cover mapping classes at hierarchical level L1, extracted from [Cop20].



(a) Image 1 (b) Image 2 (c) Inaccurate border

(d) Image 1 (e) Image 2 (f) False negative

(g) Image 1 (h) Image 2 (i) False positive

Figure 4.2: Examples of: (a)-(c) overly large change markings, (d)-(f) failure to mark changes, (g)-(i) false positive.

The available land cover maps contain several semantic classes, which are in turn organised in different hierarchical levels. By grouping the labels at different hierarchical levels it is possible to generate maps that are more coarsely or finely divided. For example, grouping the labels with the coarsest hierarchical level yields five classes (plus the ”no information” class) shown in Table 4.1. This hierarchical level will henceforth be referred to as L1.

These maps are openly available in vector form online. We have used these vector maps and the georeferenced *BD ORTHO* images to generate rasters of the vector maps that are aligned with the rasters of the images. These rasters allow us to have ground truth information about each pixel in the dataset.

It is important to note that there are slight differences in the semantic classes present in Urban Atlas 2006 and in Urban Atlas 2012. These differences do not affect the L1 hierarchical grouping and therefore had no consequence in the development of this dataset. It may nevertheless affect future work done with the data. We leave it up to the users how to best interpret and deal with these differences. More information is provided in the dataset files.

4.1.3 Dataset Analysis

Despite its unprecedented size and qualities, we acknowledge in this section the dataset's limitations and challenges. Nevertheless, we will show later that despite these limitations, the dataset allows for the boundaries of the state-of-the-art in semantic change detection through machine learning to be pushed.

One issue is the accuracy of the labels contained in the Urban Atlas vector maps with respect to the *BD ORTHO* images. We do not have access to the images used to build the Urban Atlas vector maps, nor to the exact dates of their acquisitions, nor to the dates of acquisition of the images in *BD ORTHO*. Hence, there are some discrepancies between the information in the vector maps and in the images. Furthermore, the European Environment Agency only guarantees a minimum label accuracy of 80-85% depending on the considered class. Most of the available data is accurate, but it is important to consider that the labels in the dataset are not flawless. Examples of false negatives and false positives can be see in Fig. 4.2 (d)-(f) and Fig. 4.2 (g)-(i), respectively. It is also worth noting that the labels have been created using previously known vector maps, mostly by labelling correctly each of the known regions. This means a single label was given to each region, and this led to inaccurate borders in some cases. This can be clearly seen in Fig. 4.2 (a)-(c).

One of the main challenges involved in using this dataset for supervised learning is the extreme label imbalance. As can be seen in Table 4.2, 99.232% of all pixels are labelled as no change, and the largest class is from agricultural areas to artificial surfaces (i.e. class 2 to class 1), which accounts for 0.653% of all pixels. These two classes together account for 99.885% of all pixels, which means all other change types combined account for only 0.115% of all pixels. Furthermore, many of the possible types of change have no examples at all in any of the images of the dataset. It is of paramount importance when using this dataset for supervised learning and algorithm evaluations to take into account this imbalance. This also means that using the overall accuracy as a performance metric with this dataset is not a good choice, as it virtually only reflects how many pixels of the no change class have been classified correctly. Other metrics, such as Cohen's kappa coefficient or the Sørensen-Dice coefficient, must be used instead. This class imbalance is characteristic of real world large scale data, where changes are much less frequent than unchanged surfaces. Therefore, this dataset provides a realistic evaluation tool for change detection methods, unlike carefully selected image pairs with large changed regions. The problem of supervised learning using noisy labels has already been studied and evidence suggests that supervised learning with noisy labels is possible as long as a dataset of a large enough size is used [RVBS17]. Other works attempt to explicitly deal with the noisy labels present in the dataset and prioritise the correct labels during training [MMMT18].

Finally, we acknowledge how challenging it is to use hierarchical levels finer than L1 due to: 1) a massive increase in the number of possible changes, and 2) the difference between similar classes becomes more abstract and context based. For example, the difference between the "Discontinuous Medium Density Urban Fabric" and the "Discontinuous Low Density Urban Fabric" classes defined in Urban Atlas depends not only in correctly identifying

	1	2	3	4	5
1	0%	0.011%	0%	0.001%	0.001%
2	0.653%	0%	0.001%	0%	0.077%
3	0.014%	0.002%	0%	0%	0%
4	0%	0%	0%	0%	0%
5	0.001%	0.004%	0%	0.004%	0%
No change			99.232%		

Table 4.2: Change class imbalance at hierarchical level L1. Row number represents class in 2006, column number represents class in 2012. Classes were defined in Table 4.1.

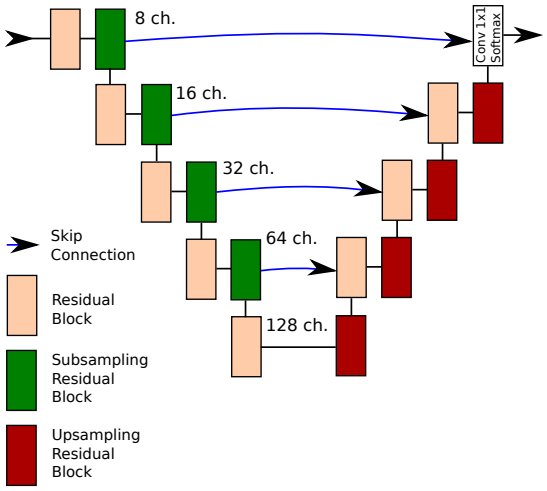


Figure 4.3: FC-EF-Res architecture, used for tests with smaller datasets to avoid overfitting. Using residual blocks improves network performance and facilitates training.

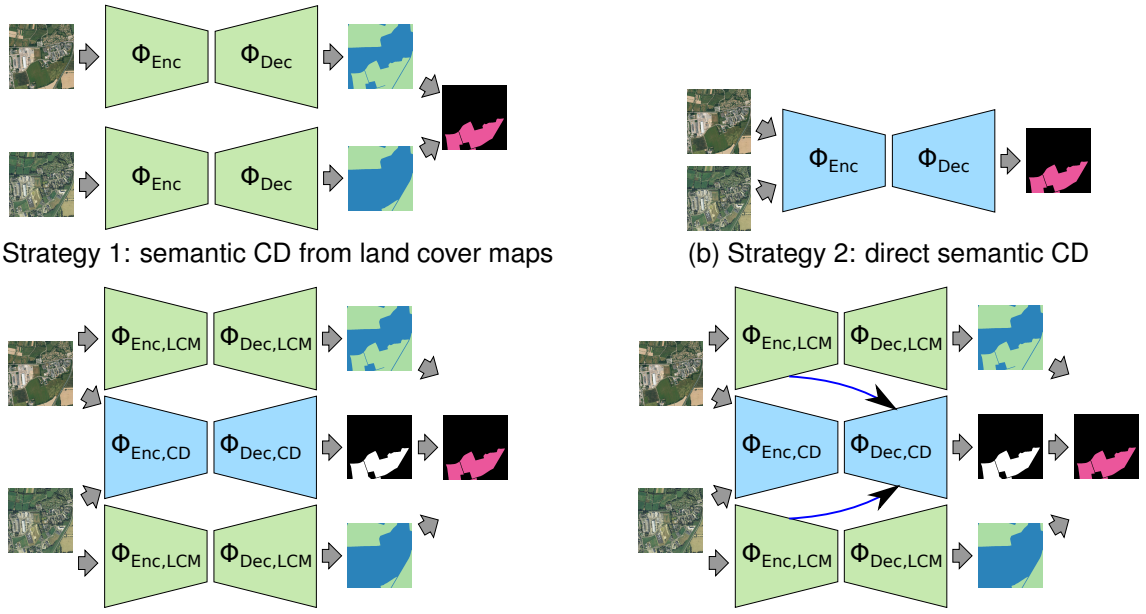
the surface at a given pixel (e.g. building or grass), but also by understanding the surroundings of the pixel and calculating the ratio between these two classes at a given neighbourhood that is not clearly defined.

4.2 Methodology

4.2.1 Binary Change Detection

We have already showed in the previous chapter the efficacy of using three different architectures of fully convolutional neural networks for change detection [DLB18]. [COA18] also proposed a fully convolutional architecture for change detection that is very similar to one of the three initially proposed architectures. In both of these works, FCN architectures performed better than previous methods for supervised change detection.

Building on this previous work, we have modified the FC-EF architecture to use residual blocks, as proposed by [HZRS16b]. The resulting network is later referred to as FC-EF-Res, and is depicted in Fig. 4.3. These residual blocks were used in an encoder-decoder architecture with skip connections to improve the spatial accuracy of the results [RFB15]. Residual blocks are used here to facilitate the training of the network, which is especially important for its deeper variations that will be discussed later.



(a) Strategy 1: semantic CD from land cover maps

(b) Strategy 2: direct semantic CD

(c) Strategy 3: separate CD and LCM

(d) Strategy 4: integrated CD and LCM

Figure 4.4: Schematics for all four proposed strategies for semantic change detection. Φ represents the network branch's learnable parameters, "Enc" means encoder, "Dec" means decoder, "LCM" means land cover mapping, and "CD" means change detection.

For testing on the OSCD dataset, the size of the network has been kept approximately the same that of FC-EF to avoid overfitting. When using the proposed HRSCD dataset (Section 4.3.2), the larger amount of annotated pixels allows us to use deeper and more complex models. In that case, the number of encoding levels and residual blocks per level has been increased, but the idea behind the network is the same as of FC-EF-Res.

4.2.2 Change Semantics

As was mentioned previously, the efficiency of the proposed architecture for binary change detection and the availability of the HRSCD dataset enable us to tackle the problem of semantic change detection. We consider this problem as two separate but related tasks. The first task is binary change detection, i.e. we attempt to determine whether a change has occurred at each pixel in a co-registered multi-temporal image pair. The second task is to differentiate between types of changes. In our case, this consists of predicting the class of the pixel in each of the two given images. The problem of semantic change detection lies in the intersection between change detection and land cover mapping.

Below we will describe four intuitive strategies to perform semantic change detection using fully convolutional networks. Starting from the plain comparison of land cover maps, we then develop more advanced strategies. These strategies vary in complexity and performance, as will be discussed in Section 4.3.

Strategy 1: Direct Comparison of Land Cover Maps

The problem of automatic land cover mapping is a well studied problem. In particular, methods involving CNNs have recently been proposed, yielding good performances [ALL16, Aud18]. When the land cover information is available, as it is the case in the HRSCD dataset, the most intuitive method that can be proposed for semantic change detection would be to train a land cover mapping network and to compare the results for pixels in the image pair (see Fig. 4.4(a)).

The advantage of this method is its simplicity. When using the HRSCD dataset we can assume changes occurred where the predicted class label differs between the two images, and the type of change is given by the predicted labels at each of the two acquisition moments. The weakness of this method is that it heavily depends on the accuracy of the predicted land cover maps. While modern FCNs are able to map areas to a good degree of accuracy, there are still many wrongly predicted labels, especially around the boundaries between regions of different classes. Furthermore, when comparing the results for two acquisitions the prediction errors would accumulate. This means the accuracy of this change detection algorithm would be lower than the land cover mapping network, and would likely predict changes in the borders between classes simply due to the inaccuracy of the network.

Strategy 2: Direct Semantic Change Detection

A second intuitive approach is to treat each possible type of change as a different and independent label, considering semantic change detection as a simple semantic co-segmentation along the lines of what has been done to binary change detection in the previous chapter, but with an increased number of output classes: one for no change, and one for each type of change.

The weakness of this method is that the number of change classes grows proportionately to the square of the number of considered land cover classes. This, combined with the class imbalance problem that was discussed earlier, proves to be a major challenge when training the network.

Strategy 3: Separate Land Cover Mapping and Change Detection

Since it has been proven before that FCNs are able to perform both binary change detection and land cover mapping, a third possible approach is to train two separate networks that together perform semantic change detection (see Fig. 4.4(c)). One network performs binary change detection on the image pair, while the other network performs land cover mapping of each of the input images. The two networks can be trained separately since they are independent.

In this strategy, the two input images produce three outputs: two land cover maps and a change map. At each pixel, the presence of change is predicted by the change map, and the type of change is defined by the classes predicted by the land cover maps at that location. This way the number of predicted classes is reduced relative to the previous strategy (i.e. the number of classes is no longer proportional to the square of land cover classes) without

Strategy number	Description	Training
1	Difference of land cover maps	land cover mapping supervision
2	Direct semantic change detection	Multiclass change detection supervision
3	Separate change detection and LCM	Separate LCM and change detection
4.1	Integrated change detection and LCM	Triple loss function
4.2	Integrated change detection and LCM	Sequential training

Table 4.3: Summary of proposed change detection strategies.

loss of flexibility. This helps with the class imbalance problem. It also avoids the problem of predicting changes at every pixel where the land cover maps differ, since the change detection problem is treated separately from land cover mapping.

We argue that such network may be able to identify changes of types it has not seen during training, as long as it has seen the land cover classes during training. For example, the network could in theory correctly classify a change from agricultural area to wetland even if such changes are not in the training set, as long as it has enough examples of those classes to correctly classify them in the land cover mapping branches. The combination of two separate networks allows us to split the problem into two, and optimise each part to maximise performance.

Strategy 4: Integrated Land Cover Mapping and Change Detection

The last of the proposed approaches is an evolution of the previous strategy of using two FCNs for the tasks of binary change detection and land cover mapping. We propose to integrate the two FCNs into a single multitask network (see Fig. 4.4(d) and Fig. 4.5) so that land cover prediction activations can be used for change detection. The combined network takes as input the two co-registered images and outputs three maps: the binary change map and the two land cover maps.

In the proposed architecture, information from the land cover mapping branches of the network is passed to the change detection branch of the network in the form of difference skip connections, which was previously shown to be the most effective form of skip connections for Siamese FCNs. The weights of the two land cover mapping branches are shared since they perform an identical task, allowing us to significantly reduce the number of learned parameters.

This multitask network gives rise to a new issue during the training phase. Given that the network outputs three different image predictions, it is necessary to balance the loss functions from these results. Since two of the outputs have exactly the same nature (the land cover maps), it follows from the symmetry of these branches that they can be combined into a single loss function by simple addition. The question remains on how to balance the binary change detection loss function and the land cover mapping loss function to maximise performance.

We have proposed and tested two different strategies for training the network. The first and more straightforward approach to this problem is to minimise a loss function that is a weighted combination of the two loss functions. This

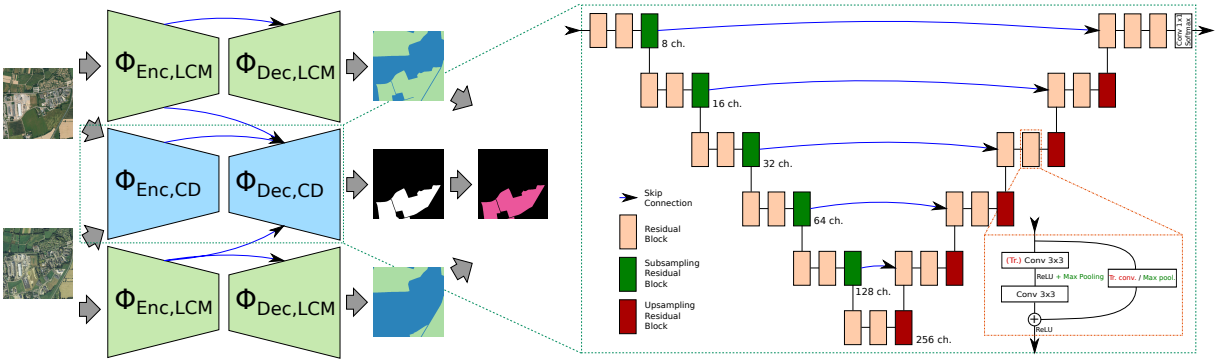


Figure 4.5: Detailed schematics for the integrated change detection and land cover mapping network (Strategy 4). The encoder-decoder architecture is the same that was used for all 4 strategies.

loss function would have the form

$$\mathcal{L}_\lambda(\Phi_{\text{Enc,CD}}, \Phi_{\text{Dec,CD}}, \Phi_{\text{Enc,LCM}}, \Phi_{\text{Dec,LCM}}) = \mathcal{L}(\Phi_{\text{Enc,CD}}, \Phi_{\text{Dec,CD}}) + \lambda \cdot \mathcal{L}(\Phi_{\text{Enc,LCM}}, \Phi_{\text{Dec,LCM}}) \tag{4.1}$$

where Φ represents the various network branch parameters, and $\mathcal{L}$ is a pixel-wise loss function. Here, the pixel-wise cross entropy function was used. The problem then becomes the search for the value of λ that leads to the best balance between the two loss terms. This can be found through a grid search, but the test of each value of λ is done by training the whole network until convergence, which is a slow and costly procedure. This will later be referred to as Strategy 4.1.

To reduce the aforementioned training burden, we propose a second approach to train the network that avoids the need of setting the hyperparameter λ. We train the network in two stages. First, we consider only the land cover mapping loss

$$\mathcal{L}_1(\Phi_{\text{Enc,CD}}, \Phi_{\text{Dec,CD}}, \Phi_{\text{Enc,LCM}}, \Phi_{\text{Dec,LCM}}) = \mathcal{L}(\Phi_{\text{Enc,LCM}}, \Phi_{\text{Dec,LCM}}) \tag{4.2}$$

and train only the land cover mapping branches of the network, i.e. we do not train $\Phi_{\text{Enc,CD}}$ or $\Phi_{\text{Dec,CD}}$ at this stage. Since the change detection branch has no influence on the land cover mapping branches, we can train these branches to achieve the maximum possible land cover mapping performance with the given architecture and data. Next, we use a second loss function based only on the change detection branch:

$$\mathcal{L}_2(\Phi_{\text{Enc,CD}}, \Phi_{\text{Dec,CD}}, \Phi_{\text{Enc,LCM}}, \Phi_{\text{Dec,LCM}}) = \mathcal{L}(\Phi_{\text{Enc,CD}}, \Phi_{\text{Dec,CD}}) \tag{4.3}$$

while keeping the weights for the land cover mapping $\Phi_{\text{Enc,LCM}}$ and $\Phi_{\text{Enc,LCM}}$ fixed. This way, the change detection branch learns to use the predicted land cover information to help to detect changes without affecting land cover mapping performance. This will later be referred to as Strategy 4.2.

Data	Network	Prec.	Recall	Tot. acc.	Dice
RGB	FC-EF	44.72	53.92	94.23	48.89
	FC-Siam-conc	42.89	47.77	94.07	45.20
	FC-Siam-diff	49.81	47.94	94.86	48.86
	FC-EF-Res	**52.27**	**68.24**	**95.34**	**59.20**
MS	FC-EF	**64.42**	50.97	**96.05**	56.91
	FC-Siam-conc	42.39	65.15	93.68	51.36
	FC-Siam-diff	57.84	57.99	95.68	57.92
	FC-EF-Res	54.93	**66.48**	95.64	**60.15**

Table 4.4: Change detection results of several methods on the OSCD dataset, for the RGB and multispectral (MS) cases. Results are in percent.

4.3 Results

4.3.1 Multispectral Change Detection

We first evaluate the performance of the proposed FC-EF-Res network. As explained in Section 4.2.1, this network is an evolution of the fully convolutional architecture FC-EF proposed in Chapter 3, to which residual blocks have been added in place of traditional convolutional layers.

The FC-EF-Res architecture was compared to the previously proposed FCN architectures on the OSCD dataset for binary change detection, which contains lower-resolution Sentinel-2 image pairs with 13 multispectral bands. As expected, the residual extension of the FC-EF architecture outperformed all previously proposed architectures. The difference was noted on both the RGB and the multispectral cases. On the RGB case, the improvement was of such magnitude that the change detection performance on RGB images almost matched the performance on multispectral images. The results can be seen in Table 4.4. This corroborates the claims made by [HZRS16b] that using residual blocks improves the training performance of CNNs. For this reason, all subsequent networks that are tested with the HRSCD dataset use residual modules.

4.3.2 Very High Resolution Semantic Change Detection

To test the methods proposed in Section 4.2.2 we split the HRSCD images into two groups: 146 image pairs for training and 145 image pairs for testing. By splitting the train and test sets this way we can ensure that no pixel in the test set has been seen during training. Class weights were set inversely proportional to the number of training examples to counterbalance the dataset's class imbalance. The results for each of the proposed strategies can be seen in Table 4.5, and illustrative image results can be seen in Fig. 4.6 and the other examples included at the end of this chapter. The networks were trained using minibatches of four 600x600 patches for a total of 500 epochs. The ADAM optimizer was used with an initial learning rate of 10^{-3}, which was reduced by a factor of 10 every 100 epochs. All tests were done using the PyTorch framework.

As is the case for most deep neural networks, the training times for the proposed methods are significantly

larger than the testing times. Once the network has been trained, its fast inference speed allows it to process large amounts of data efficiently. The proposed methods took 3-5 hours of training time using a GeForce GTX 1080 Ti GPU with 11GB of memory. Inference times of the proposed methods were under 0.04 s for 512x512 image pairs using the same hardware.

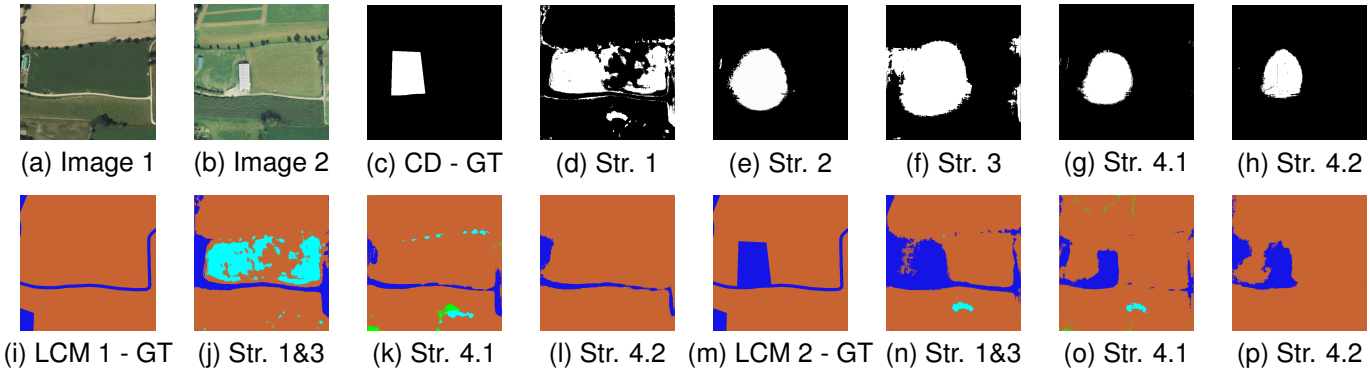
In Strategy 1, which naively attempts to predict change maps from land cover maps, we can see that the network succeeds in accurately classifying the imaged terrains, but this is not enough to predict accurate change maps. The change detection kappa coefficient for this strategy is very low, which means this method is marginally better than chance for change detection.

The results for Strategy 2 are a fair improvement over those of Strategy 1. The change detection Dice coefficient and the land cover mapping results for this method are not reported due to its nature, since Dice coefficients can only be calculated for binary classification problems, and this strategy bypasses the land cover mapping steps. Despite achieving a higher kappa coefficient, the network learned to always predict the same type of change where changes occurred. This means that despite using appropriately tuned class weights, the learning process did not succeed in overcoming the extreme class imbalance present in the dataset. In other words, the network learned to detect changes but no semantic information was present in the results.

For Strategy 3, the land cover mapping network that was used was the same as that of Strategy 1, which achieved good performance. A binary change detection network was trained to be used for masking the land cover maps. The performance of this network was better than that of Strategy 1 but worse than that of Strategy 2. The results show that this is due to an overestimation of the change class. This shows once again how challenging dealing with the extreme class imbalance is.

The results of Strategy 4 are the best ones overall. The simultaneous training strategy (Str. 4.1) achieves excellent performance in both land cover mapping and change detection, proving the viability of this strategy. The reported results were obtained with $\lambda = 0.05$, which is a value that prioritises the training of the change detection branch of the network. We then see that the same network trained with sequential training (Str. 4.2) obtained even better results in both change detection and land cover mapping without needing to search for an adequate parameter λ. This, according to our results, is the best semantic change detection method. By comparing the results for Strategies 3 and 4 we can see the improvements that result directly from integrating the change detection and land cover mapping branches of the networks. In other words, Strategy 4.2 allows us to maximise the change detection performance without reducing the land cover mapping accuracy.

The best performing land cover mapping method was the single purpose network that was trained and used for Strategies 1 and 3. The fact that it achieves a better kappa coefficient than Strategy 4.2 is merely due to the randomness of the initialisation and training of the network, as the land cover mapping branches of Strategy 4.2 are identical to those used in Strategies 1 and 3. This also explains why their results are so similar. By comparing these results to those of Strategy 4.1 it emphasises once again the fact that attempting to train the network shown



(a) Image 1 (b) Image 2 (c) CD - GT (d) Str. 1 (e) Str. 2 (f) Str. 3 (g) Str. 4.1 (h) Str. 4.2

(i) LCM 1 - GT (j) Str. 1&3 (k) Str. 4.1 (l) Str. 4.2 (m) LCM 2 - GT (n) Str. 1&3 (o) Str. 4.1 (p) Str. 4.2

Figure 4.6: Illustrative images of the obtained results: (a)-(b) multitemporal image pair; (c) ground truth change detection map; (d)-(h) predicted change maps; (i)-(l) ground truth and predicted land cover maps for image 1; (m)-(p) ground truth and predicted land cover maps for image 2.

	CD			LCM	
	Kappa	Dice	Tot. acc.	Kappa	Tot. acc.
Strategy 1	3.99	5.56	86.07	**71.92**	87.22
Strategy 2	21.54	-	**98.30**	-	-
Strategy 3	12.48	13.79	94.72	**71.92**	87.22
Strategy 4.1	19.13	20.23	96.87	67.25	85.74
Strategy 4.2	**25.49**	**26.33**	98.19	71.81	**89.01**
CNNF-O	0.74	2.43	64.54	-	-
CNNF-F	3.28	4.84	88.66	-	-
PCA+KM	0.67	2.31	83.95	-	-

Table 4.5: Change detection (CD) and land cover mapping (LCM) results of all four of the proposed strategies on the HRSCD dataset. Comparison with the methods proposed by [EALW16] (Otsu [CNNF-O] and fixed [CNNF-F] thresholding) and by [Cel09] ([PCA+KM]) are included. Results are in percent.

in Fig. 4.5 all at once damages performance in both change detection and land cover mapping.

In Fig. 4.6 we can see the results of the proposed networks on a pair of images from the dataset. Note the amount of false detections by Strategy 1 due to the lack of accuracy of prediction of the land cover maps on region boundaries. The second row shows the predicted classes at each pixel for each image. The semantic information about the changes comes from comparing these two predictions. For example, comparing the images in Fig. 4.6 (k) and (o) we can say that the changes predicted in (g) were from the ”Agricultural areas” class to the ”Artificial surfaces” class.

In our tests we observed that the trained networks had the tendency to overestimate the size of the detected changes. It is likely that this happens simply due to the nature of the data that was used for training. The labels in the HRSCD dataset, which come from Urban Atlas, mark as a change the whole terrain where a change of class happened. This means that not only the pixels associated with a given change are marked as change, but the neighbouring pixels that are in the same parcel are also marked as change. This leads to the networks learning to overestimate the boundary of the detected changes in an attempt to also correctly classify the pixels surrounding the detected change. This once again illustrates the challenges of the HRSCD dataset.

		ReCNN-LSTM	EF
Binary CD	Tot. acc.	98.67	**99.35**
	Kappa	97.28	**98.67**
	No change	98.83	**99.47**
	Change	98.46	**99.19**
Semantic CD	Tot. acc.	**98.70**	98.48
	Kappa	**97.52**	97.10
	No change	**98.49**	97.73
	City exp.	84.72	**100**
	Soil change	**100**	86.07
	Water change	99.25	**99.93**

Table 4.6: Change detection results on Eppalock lake test images. Results are in percent.

The performance of two other change detection methods are also shown in Table 4.5. The first method, proposed in [EALW16], is based on transfer learning and uses features from a pretrained VGG-19 model [SZ15] to create pixel descriptors, whose Euclidean distance is used to build a difference image. The original method uses Otsu thresholding to produce the final change maps (CNNF-O), but we have found that such approach leads to overestimating changes. We therefore tuned a fixed threshold (T = 2300) using a few example images and used that value to test the algorithm on all test data (CNNF-F), which significantly increased its performance by reducing false positives. Also included are the results by the method proposed in [Cel09], which performs principal component analysis (PCA) and k-means clustering on the pixels to detect changes in an unsupervised manner. Both algorithms perform worse than the proposed method on the HRSCD dataset.

To evaluate the size of the dataset, we have also tested Strategy 4.2 using reduced amounts of data for training the network. The kappa coefficient, in percent, obtained by using the whole training dataset is 25.49. This value is reduced to 23.34 by using half the training data, and is further reduced to 22.18 by using a quarter of the data. This shows that, as expected, using more data for training the network leads to better results. Nonetheless, it also shows that the dataset is large enough to allow for even more complex and data hungry methods to be trained using the HRSCD dataset in the future.

Finally, it is important to note that the label imperfections in the HRSCD dataset occur not only in the training images, but also in the test images. This means that the performance of the proposed methods may be even higher than the numbers suggest, since some of the disagreements between prediction and ground truth data are actually due to errors in the ground truth data.

4.3.3 Eppalock Lake Images

We compare our method in this section to the one proposed by [MBZ19], which used recurrent convolutional neural networks for change detection. In that work, pixels were randomly split into train and test sets. We believe that this split leads to overfitting since neighbouring pixels contain redundant information. This is especially true when

using CNNs, which take as inputs patches centred on the considered pixels, meaning the network sees the same information for training and testing. It is likely that overfitting takes place, since an accuracy of over 98% is achieved by using only 1000 labelled pixels to train a network with 67500 parameters (for their long short-term memory (LSTM) architecture, which performed the best). The data consists of a single image pair of 631x602 pixels only partially annotated, with a total of 8895 annotated pixels which is much less data than what is required for deep learning methods. The HRSCD dataset presented in Section 4.1 contains over 3 million times more labelled pixels than the Eppalock lake image pair. Despite the flaws of this testing scheme, we have followed it to achieve a fair comparison between the methods.

Using the CNN architecture labelled EF presented in the previous chapter, we have achieved excellent numeric results thanks to this overfitting phenomenon, which discouraged the usage of more complex methods which would lead to even more extreme overfitting. The results achieved by the EF network were better for binary change detection and equivalent for semantic change detection compared to ReCNN-LSTM. The results can be seen in Table 4.6.

4.4 Conclusion

This chapter presented the first large scale very high resolution semantic change detection dataset, which has been released to the scientific community. This dataset contains 291 pairs of aerial images, together with aligned rasters for change maps and land cover maps. This dataset allows for the first time for much more complex deep learning methods to be used in this context in a supervised manner than what was possible using the previously available smaller datasets. We have then proposed different methods for using deep FCNs for semantic change detection. The best among the proposed methods is an integrated network that performs land cover mapping and change detection simultaneously, using information from the land cover mapping branches to help with change detection. We also proposed a sequential training scheme for this network that avoids the need of tuning a hyperparameter, which circumvents a costly grid search.

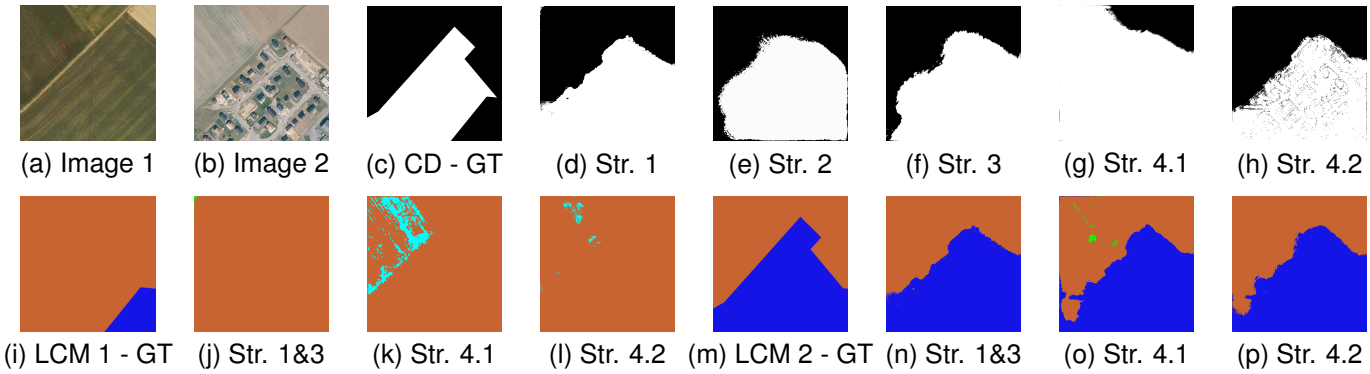


Figure 4.7: Illustrative images of the obtained results: (a)-(b) multitemporal image pair; (c) ground truth change detection map; (d)-(h) predicted change maps; (i)-(l) ground truth and predicted land cover maps for image 1; (m)-(p) ground truth and predicted land cover maps for image 2.

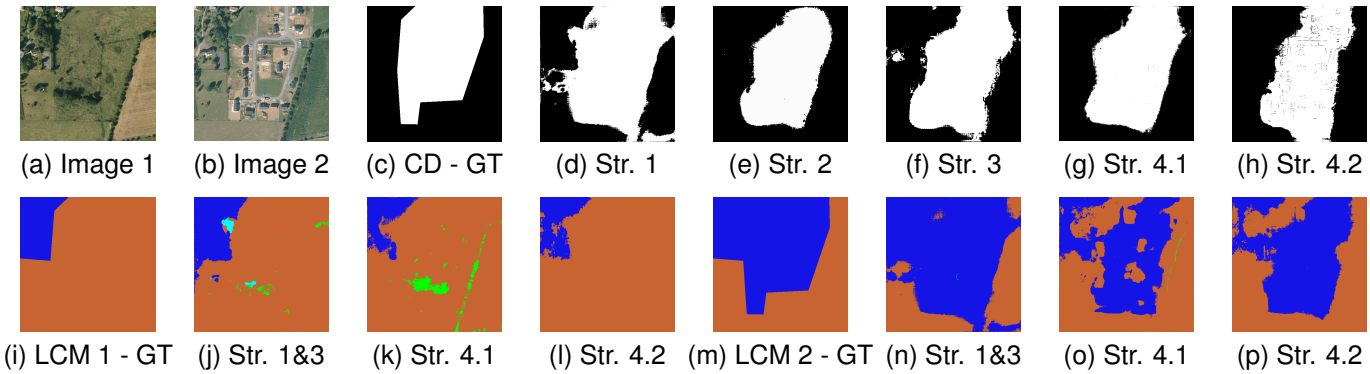


Figure 4.8: Illustrative images of the obtained results: (a)-(b) multitemporal image pair; (c) ground truth change detection map; (d)-(h) predicted change maps; (i)-(l) ground truth and predicted land cover maps for image 1; (m)-(p) ground truth and predicted land cover maps for image 2.

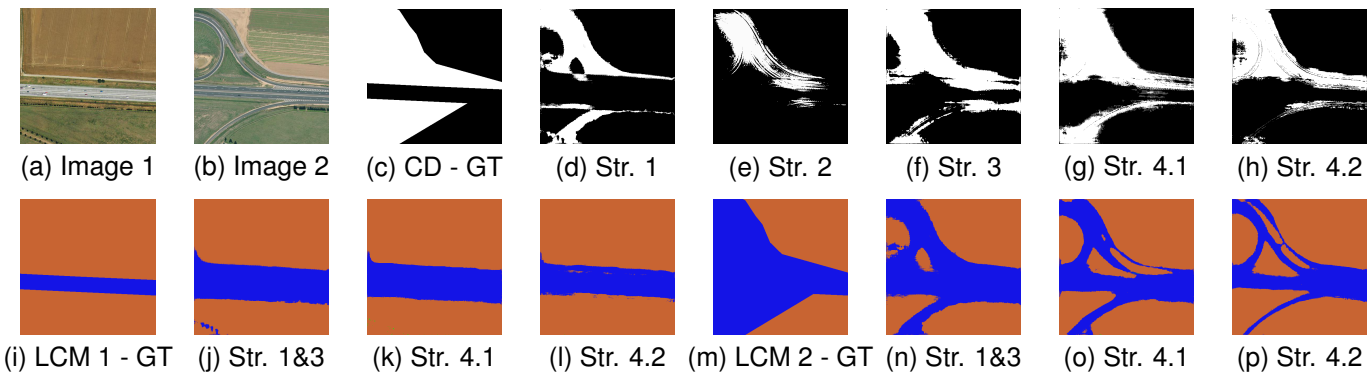


Figure 4.9: Illustrative images of the obtained results: (a)-(b) multitemporal image pair; (c) ground truth change detection map; (d)-(h) predicted change maps; (i)-(l) ground truth and predicted land cover maps for image 1; (m)-(p) ground truth and predicted land cover maps for image 2.

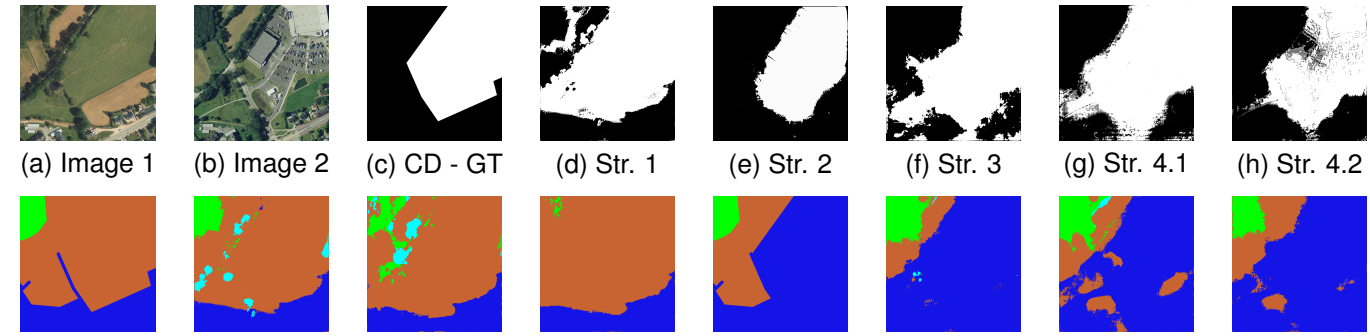


Figure 4.10: Illustrative images of the obtained results: (a)-(b) multitemporal image pair; (c) ground truth change detection map; (d)-(h) predicted change maps; (i)-(l) ground truth and predicted land cover maps for image 1; (m)-(p) ground truth and predicted land cover maps for image 2.

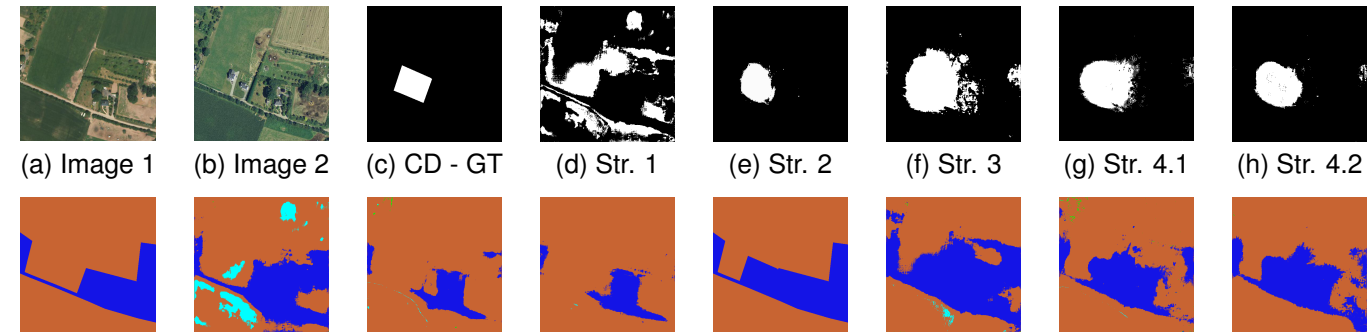


Figure 4.11: Illustrative images of the obtained results: (a)-(b) multitemporal image pair; (c) ground truth change detection map; (d)-(h) predicted change maps; (i)-(l) ground truth and predicted land cover maps for image 1; (m)-(p) ground truth and predicted land cover maps for image 2.

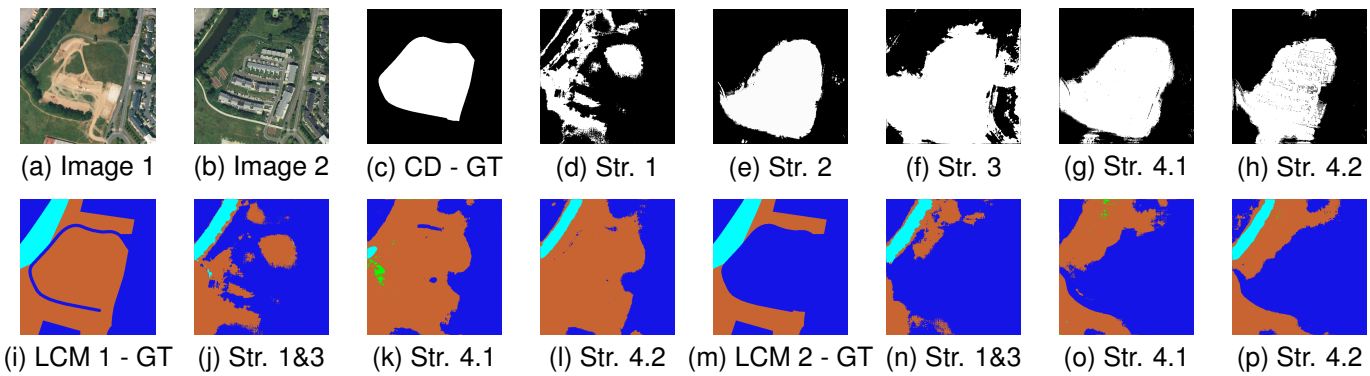
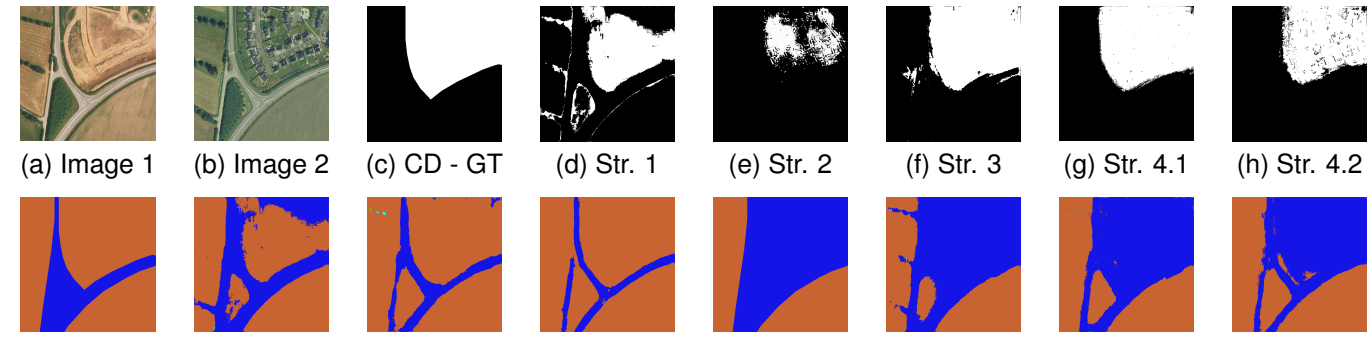


Figure 4.12: Illustrative images of the obtained results: (a)-(b) multitemporal image pair; (c) ground truth change detection map; (d)-(h) predicted change maps; (i)-(l) ground truth and predicted land cover maps for image 1; (m)-(p) ground truth and predicted land cover maps for image 2.



(a) Image 1 (b) Image 2 (c) CD - GT (d) Str. 1 (e) Str. 2 (f) Str. 3 (g) Str. 4.1 (h) Str. 4.2

(i) LCM 1 - GT (j) Str. 1&3 (k) Str. 4.1 (l) Str. 4.2 (m) LCM 2 - GT (n) Str. 1&3 (o) Str. 4.1 (p) Str. 4.2

Figure 4.13: Illustrative images of the obtained results: (a)-(b) multitemporal image pair; (c) ground truth change detection map; (d)-(h) predicted change maps; (i)-(l) ground truth and predicted land cover maps for image 1; (m)-(p) ground truth and predicted land cover maps for image 2.

Chapter 5

Weakly Supervised Change Detection

Chapter Summary

In the previous chapter, the HRSCD dataset was presented. Due to the automatic way it was created, i.e. combining two unrelated databases for images and labels, the generated change maps in the dataset were somewhat noisy and biased. Not only can false positives and false negatives be found, but change labels were often larger than the changed object, since the annotations came from parcel-level vector data.

In this chapter, concepts are borrowed from the field of weakly supervised learning, drawing a parallel between these imprecise annotations and bounding box annotations, which are also imprecise when considered at pixel level. The aim here is to extract the correct information from the annotations, while ignoring the incorrect labels. This is also connected to the problem of supervision using noisy data.

We propose an iterative training method that uses the network predictions, once training has converged, to try to isolate bad labels from good ones or to correct badly labelled samples. Training is then resumed using the cleaner version of the dataset until convergence, and this process is repeated. This training method is coupled with a novel post-processing method named *guided anisotropic diffusion*, which applies an anisotropic diffusion algorithm to the softmax activations with diffusion coefficients calculated using the input images. This serves as a way to better fit predictions to boundaries present in the input images, which helps reduce the effects of the biased overlarge predictions of the network trained on the original reference data.

Finally, true weakly supervised semantic co-segmentation from image-level labels is performed for the first time, to the best of our knowledge. Using building-centred image patch pairs with binary classification labels, a Siamese network with a novel spatial attention layer is used for this purpose. The attention layer allows the network to learn more localised features, which in turn helps the attention layer to further narrow down the area of interest in a positive feedback loop during the training procedure.

5.1 Change Detection with Unreliable Data

The HRSCD that was presented in the previous sections is the first large scale change detection dataset of its kind ever released. It was generated by combining an aerial image database with open change and land cover data. Change maps and land cover maps were generated for almost 30 billion pixels, over 3000 times larger than previous change detection datasets. This dataset, however, contains unreliable labels due to having been generated automatically. The effects of naively using these data for supervised learning with change detection networks are shown in Fig. 5.1. Inaccuracies in the reference data stem primarily from two causes: imperfections in the vector data at different semantic levels, and temporal misalignment between the annotations and the images. Naive supervision using such data leads to overestimation of the detected changes, as can be seen in Fig. 5.1(e). Nevertheless, there is much useful information in the available annotations that, if used adequately, can lead to better CD systems.

Due to the way the ground truth was generated, the labels in the dataset mark changes at a land parcel level with imprecise boundaries. While useful for global monitoring of changes in land cover, it cannot delineate precise object-level changes. In order to achieve a precise pixel-wise change detection, we propose a weakly supervised learning approach to change detection. We consider the parcel-wise reference data as approximations, similar to bounding-boxes, of an ideal unknown ground truth corresponding to changes at pixel level. For each parcel with detected changes, the reference data in HRSCD contained both good and bad labels. For this reason, the noise in the labels is not randomly distributed, but it is conditioned on the pixels' neighborhoods and highly structured.

Other change detection datasets rely on cross referencing data obtained by on-site surveys with available remote sensing imagery. Such is the case of the ABCD dataset [FSI+17], which contains image pairs centered on buildings in a region that has been affected by a tsunami. Images before and after the event were taken with different sensors, and were registered and cropped around each known building in the area. Binary change labels for each image pair are available, but segmentation labels are not. This dataset contains over 8000 labelled image pairs, and is available in two versions: *fixed scale*, where the spatial resolution of the images is kept constant, and *resized*, where images are resized so that the length of the imaged building takes up roughly a third of the patch size.

Noisy labels for supervised learning is a topic that has already been widely explored [FK+14, FV14]. In many

(a) Image 1 (b) Image 2 (c) Reference data (d) Manual annotations (e) Naive method (f) Proposed method

Figure 5.1: (a)-(b) image pair, (c) change labels from the HRSCD dataset, (d) ground truth created by manually annotating changes, (e) result obtained by naive supervised training, (f) result obtained by our proposed method.

cases, label noise is completely random and independent from the data, and is modelled mathematically as such [NDRT13, XXY+15, RVBS17]. Rolnick *et al.* showed that supervised learning algorithms are robust to random label noise, and proposed strategies to further minimize the effect label noise has on training, such as increasing the training batch sizes [RVBS17]. In the case presented in this chapter, the assumption that the label noise is random does not hold. Incorrect change detection labels are usually around edges between regions or grouped together, which leads the network to learn to overestimate detected changes as seen in Fig. 5.1(e). Ignoring part of the training dataset, known as data cleansing (or cleaning), has already been proposed in different contexts [MGB+92, Joh95, GMV+96, JWF10].

Weakly supervised learning is the name given to the group of machine learning algorithms that aim to perform different or more complex tasks than normally allowed by the training data at hand. Weakly supervised algorithms have recently gained popularity because they provide an alternative when data acquisition is too expensive. The problem of learning to perform semantic segmentation using only bounding box data or image level labels is closely related to the task discussed in this chapter, since most methods propose the creation of an approximate semantic segmentation ground truth for training and dealing with its imperfections accordingly. Dai *et al.* proposed the BoxSup algorithm [DHS15] where region proposal algorithms are used to generate region candidates in each bounding box, then a semantic segmentation network is trained using these annotations, and finally it is used to select better region proposal candidates iteratively. Khoreva *et al.* proposed improvements to the BoxSup algorithm that includes using *ad hoc* heuristics and an ignore class during training [KBH+17]. They obtained best results using region proposal algorithms to create semantic segmentation training data directly from bounding boxes. Lu *et al.* modelled this problem as a simultaneous learning and denoising task through a convex optimization problem [LFX+17]. Ahn and Kwak proposed combining class activation maps, random walk and a learned network that predicts if pixels belong to the same region to perform semantic segmentation from image level labels [AK18]. Zhou *et al.* proposed the class activation mapping technique [ZKL+16], which allows the networks to localize what regions in the image contribute to the prediction of each class, which can be harnessed for generating pixel-level predictions from image-level labels.

Results generated by such methods often lack in spatial accuracy and benefit from post-processing methods to increase the precision of the predictions, especially around region boudaries. Post-processing methods that use information from guide images to filter other images, such as semantic segmentation results, have also been proposed [PSA+04, KCLU07, FRR+13]. A notable example is the Dense CRF algorithm proposed by Krähenbühl and Koltun, in which an efficient solver is proposed for fully connected conditional random fields with Gaussian edge potentials [KK11]. The idea of using a guide image for processing another is also the base of the Guided Image Filtering algorithm proposed by He *et al.* [HST13], where a linear model that transforms a guide image into the best approximation of the filtered image is calculated, thus transferring details from the guide image to the filtered image. The use of joint filtering is popular in the field of computational photography, and has been used for several applications [PSA+04, KCLU07, FRR+13]. One of the building blocks of the filtering method we propose in this

chapter is the anisotropic diffusion, proposed by Perona and Malik [PM90], an edge preserving filtering algorithm in which the filtering of an image is modelled as a heat equation with a different diffusion coefficient at each edge between neighbouring pixels depending on the local geometry and contrast. However, to the best of our knowledge, this algorithm has not been used for guided filtering previously to this work.

5.2 Method

The main contributions of this chapter are: 1) the guided anisotropic diffusion algorithm, which uses information from the input images to filter and improve semantic segmentation results, 2) an iterative training scheme that aims to efficiently learn from inaccurate and unreliable ground truth semantic segmentation data, and 3) a learned spatial attention layer that improves classification and weakly supervised semantic segmentation with class activation maps for datasets with object aligned crops. These contributions are described in detail below. While these ideas are presented in this chapter in the context of change detection, the proposed methods' scope is broader and they could be used for other semantic segmentation problems, together or separately.

5.2.1 Guided Anisotropic Diffusion

In their seminal paper, Perona and Malik proposed an anisotropic diffusion algorithm with the aim of performing scale space image analysis and edge preserving filtering [PM90]. Their diffusion scheme has the ability to blur the inside of regions with homogeneous colours while preserving or even enhancing edges. This is done by modelling the filtering as a diffusion equation with spatially variable coefficients, and as such is an extension of the linear heat equation, whose solution is mathematically equivalent to Gaussian filtering when diffusion coefficients are constant [Koe84]. Diffusion coefficients are set to be higher where the local contrast of the image is lower.

More precisely, we consider the anisotropic diffusion equation

$$\frac{\partial I}{\partial t} = div(c(x,y,t)\nabla I) = c(x,y,t)\Delta I + \nabla c \cdot \nabla I \tag{5.1}$$

where I is the input image, $c(x,y,t)$ is the coefficient diffusion at position (x,y) and time t, div represents the divergence, ∇ represents the gradient, and Δ represents the Laplacian. In its original formulation, $c(x,y,t)$ is a function of the input image I. To perform edge preserving filtering, one approach is using the coefficient

$$c(x,y,t) = \frac{1}{1 + \left(\frac{||\nabla I(x,y,t)||}{K}\right)^2}, \tag{5.2}$$

which approaches 1 (strong diffusion) where the gradient is small, and approaches 0 (weak diffusion) for large gradient values. Other functions with these properties and bound in $[0,1]$ may also be used. The parameter K

Algorithm 1 Guided Anisotropic Diffusion pseudocode.

Input: I_1, I_2, I_in, N, K, λ
Output: I_f
$I_f \leftarrow I_in$
for $(i \leftarrow 1; i \leq N; i++)$ **do**
  for $(I_j = \{I_1, I_2\})$ **do**
    $\nabla I_j \leftarrow$ Calculate gradient of I_j
    $c_{I_j} \leftarrow$ Calculate diffusion coefficients using Eq. 5.3
    $I_j \leftarrow I_j + \lambda \cdot \nabla I_j \cdot c_{I_j}$
  end for
  $\nabla I_f \leftarrow$ Calculate gradient of I_f
  $c_f \leftarrow$ Calculate diffusion coefficients using Eq. 5.4
  $I_f \leftarrow I_f + \lambda \cdot \nabla I_f \cdot c_f$
end for

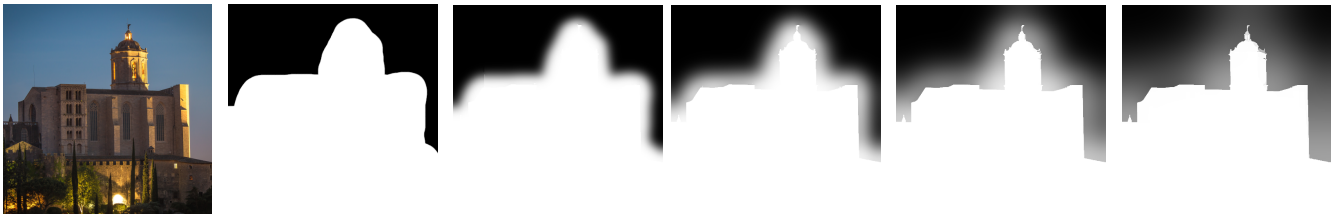
controls the sensitivity to contrast in the image.

In the *guided* anisotropic diffusion (GAD) algorithm, the aim is to perform edge preserving filtering on an input image, but instead of preserving the edges in the filtered image we preserve edges coming from a separate guide image (or images). Doing so allows us to transfer properties from the guide image I_g into the filtered image I_f. An illustrative example is shown in Fig. 5.2, where the image of a cathedral (a) is used as a guide to filter the image of a rough segmentation (b). The edges from the guide image I_g are used to calculate $c(x, y, t)$, which in practice creates barriers in the diffusion of the filtered image I_f, effectively transferring details from I_g to I_f. These edges effectively separate the image in two regions, inside and outside the region of interest, and the pixel values in each of these regions experience diffusion, but there is virtually no diffusion happening between them.

Our primary aim is to use this GAD algorithm to improve semantic segmentation results based on the input images. Weakly supervised learning methods are often used when there is an overestimation of the target area: either the whole image is the starting point in classification to segmentation tasks, or the reference region is too large in parcel to region segmentation tasks. GAD provides a way to improve these semantic segmentation results by making them more precisely fit the edges present in the input images. A few design choices were made to extend the anisotropic diffusion from gray level images to RGB image pairs. The extension to RGB image was done by taking the mean of the gradient norm at each location

$$c_I(x, y, t) = \frac{1}{1 + \left(\sum_{C \in \{R,G,B\}} \frac{||\nabla I_C(x,y,t)||}{3 \cdot K} \right)^2}, \tag{5.3}$$

so that edges in any of the color channels would prevent diffusion in the filtered image. To extend this further to be capable of taking multiple guide images simultaneously, which is necessary for the problem of change detection, the minimum diffusion coefficient at each position (x, y, t) was used, once again to ensure that any edge present in



(a) Guide image (b) Input image (c) 300 it. (d) 1000 it. (e) 3000 it. (f) 10000 it.

Figure 5.2: Results of guided anisotropic diffusion. Edges in the guide image (a) are preserved in the filtered image (b). (c)-(f) show results using different numbers of iterations.

any guide image would be transferred to the filtered image:

$$c_{I_1,I_2}(x,y,t) = min_{i\in\{1,2\}} c(I_i)(x,y,t). \tag{5.4}$$

Guided anisotropic diffusion aims to improve semantic segmentation predictions by filtering the class probabilities yielded by a fully convolutional network. It is less adequate to correct for large classification mistakes, as opposed to non-local methods such as Dense CRF, but it leads to smoother predictions with more accurate edges. It can also be easily extended for any number of guide images by increasing the number of images considered in Eq. 5.4. The pseudocode for the GAD algorithm can be found in Algorithm 1. As mentioned in the original anisotropic diffusion paper, the algorithm is unstable for $\lambda > 0.25$ when using 4-neighborhoods for the calculations. For more information the reader can refer to the mathematical derivations presented in [PM90, AK06].

In the following sections, we show two ways to use GAD to improve weakly supervised semantic segmentation and reduce the effect of label noise. First, we address in Section 5.2.2 the inaccurate labelling problem with an iterative data cleansing scheme. Second, in Section 5.2.3 we use GAD to learn to segment changes from classification labels only.

5.2.2 Iterative Training Scheme

The label noise present in parcel-based change detection datasets such as the HRSCD dataset is challenging due to its spatial structure and correlation between neighbors. In the taxonomy presented in [FK$^+$14, FV14], this type of label noise would be classified as ”label noise not at random” (NNAR). NNAR is the most complex among the label noise models in the taxonomy. In the case of HRSCD, most errors can be attributed to one of the following reasons: the available information is insufficient to perform labelling, errors on the part of the annotators, subjectiveness of the labelling task, and temporal misalignment between the databases used to create the HRSCD dataset.

It is important to note that, as discussed by Frénay and Kabán in [FK$^+$14], label noise has an even more powerful damaging impact when a dataset is imbalanced since it alters the perceived, but not the real, class imbalance and therefore the methods used to mitigate class imbalance during training are less effective. In the case of change

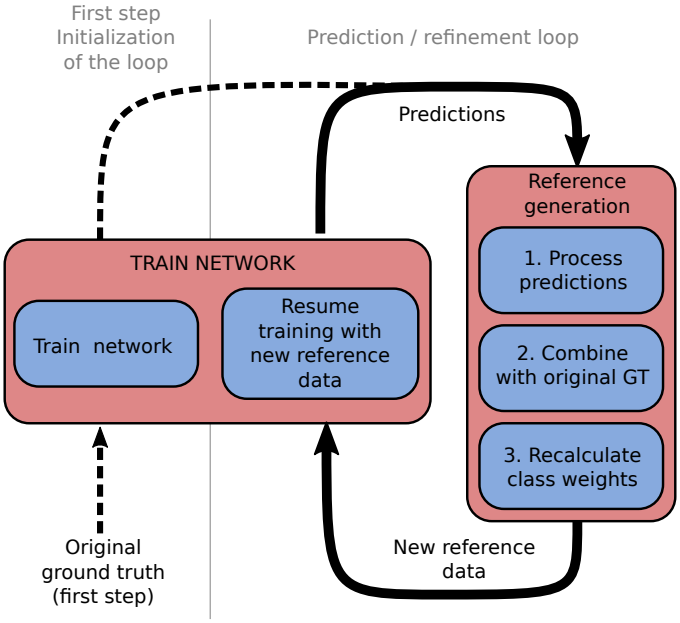


Figure 5.3: Iterative training method: alternating between training and data cleaning allows the network to simultaneously learn the desired task and to remove bad examples from the training dataset.

Algorithm 2 Iterative training pseudocode.

Input: I: Image pairs, GT_o: Original unreliable ground truths, N: Number of hyperepochs, Φ_r: Initial random network weights.
Output: Φ_N: Trained network weights.
$w_0 \leftarrow$ calculate class weights inversely proportional to number of class examples
$\Phi_0 \leftarrow$ Train network with I and GT_0 until convergence or fixed number of epochs
for $(i \leftarrow 1; i \leq N; i++)$ **do**
  $P_i \leftarrow$ generate predictions for training dataset with current network
  $P_{i,pp} \leftarrow$ Post-processing of predictions
  $GT_i \leftarrow$ Combine $P_{i,pp}$ with GT_0 to generate cleaner ground truth data
  $\Phi_i \leftarrow$ Continue training network from Φ_{i-1} using I and GT_i until convergence
end for

detection with the HRSCD dataset, the no change class outnumbers the change class 130 to 1, which means the label noise could significantly alter the calculated class weights used for training.

It has been noticed in this chapter and in the previous one that change detection networks trained directly on the HRSCD dataset had the capacity to detect changes in image pairs but tended to predict blobs around the detected change instances, as is depicted in Fig. 5.7(c), likely in an attempt to minimize the loss for the training images where the surrounding pixels of true changes are also marked as having experienced changes. In many cases, it was observed that the network predictions were correct where the ground truth labels were not. Based on this observation, we propose a method for training the network that alternates between actual minimization of a loss function and using the network predictions to clean the reference data before continuing the training. A schematic that illustrates the main ideas of this method is shown in Fig. 5.3. For the remainder of this chapter, the iteration cycles of training the network and cleaning of training data will be referred to as *hyperepochs*.

Pred. \ Original GT	0	1
0	0	0
1	0	1

(a) Intersection

Pred. \ Original GT	0	1
0	0	2
1	0	1

(b) FN← Ignore

Pred. \ Original GT	0	1
0	0	2
1	2	1

(c) FN∪FP← Ignore

Figure 5.4: Proposed methods for merging original labels and network predictions. Classes: 0 is no change, 1 is change, 2 is ignore. (a) Intersection between original and detected changes. (b) Ignore false negatives from the perspective of original labels. (c) Ignore all pixels with label disagreements.

Alternating between training a semantic segmentation network and using it to make changes to the training data has already been explored [DHS15, KBH+17]. Such iterative methods are named "classification filtering" [FV14]. The main differences between the method proposed in this chapter and previous ones are:

1. **No bounding box information is available**: we work directly with pixel level annotations, which were generated from vector data;

2. **Each annotated region may contain more than one instance**: the annotations often group several change instances together;

3. **Annotations are not flawless**: the HRSCD dataset contains both false positives and false negatives in change annotations.

It has also been shown by Khoreva *et al.* in [KBH+17] that simply using the outputs of the network as training data leads to degradation of the results, and that it is necessary to use priors and heuristics specific to the problem at hand to prevent a degradation in performance. Here we use two methods to avoid degradation of the results with iterative training. The first is using post-processing techniques that bring information from the input images into the predicted semantic segmentations, improving the results and providing a stronger correlation between inputs and predictions. The GAD algorithm presented in Section 5.2.1 serves this purpose, but other algorithms such as Dense CRF [KK11] may also be used. The second way the degradation of results is avoided is by combining network predictions with the original reference data at each iteration, instead of simply using predictions as reference data.

We propose three ways of merging the original labels with network predictions. When merging, each pixel will have a binary label from the original ground truth and a binary label from the network prediction. If these labels agree, there is no reason to believe the label for that pixel is wrong, and it is therefore kept unchanged. In case the labels disagree, the following options to decide the pixel's label are proposed:

1. **The intersection of predicted and reference change labels is kept as change**: this strategy assumes all changes are marked in both the reference data and in the prediction. It also puts pixels with uncertain labels in the no change class, where they are more easily diluted during training due to the class imbalance.

2. **Ignore false negatives**: using an ignore class for false negatives attempts to keep only good examples in

(a) Image 1 (b) Image 2 (c) GT and prediction

(d) Intersection (e) FN← Ignore (f) FN∪FP← Ign.

Figure 5.5: Example case of the three proposed merge strategies. In (c), black is true negative, white is true positive, magenta is false negative, and green is false positive. In (d)-(f) blue represents the ignore class.

the change class, improving the quality of the training data. It assumes all changes are marked in the original labels provided.

3. **Ignore all disagreements**: marking all label disagreements to be ignored during training attempts to keep only clean labels for training at the cost of reducing the number of training examples. This approach is the only one that is class agnostic.

In practice, the ignored pixels are marked as a different class that is given a class weight of 0 during the training. Tables for the three proposed methods can be found in Fig. 5.4, and an example can be found in Fig. 5.5.

5.2.3 Scene-Invariant Spatial Attention Layer

Many datasets in remote sensing contain georeferenced data. In many cases, patches are cropped from large images using the coordinates of known objects for which a label is known. In the case of the ABCD dataset [FSI+17], for example, ground surveys were used to identify which buildings had been damaged by a tsunami, and a dataset was created using crops centred on each of the buildings that were surveyed.

In such cases, objects to which the labels refer are located in the center of the images, while the characteristics of their surroundings are not directly related to the available labels. Pooling techniques such as max pooling and average pooling that are very often used in CNNs are invariant with respect to the image position. These operations

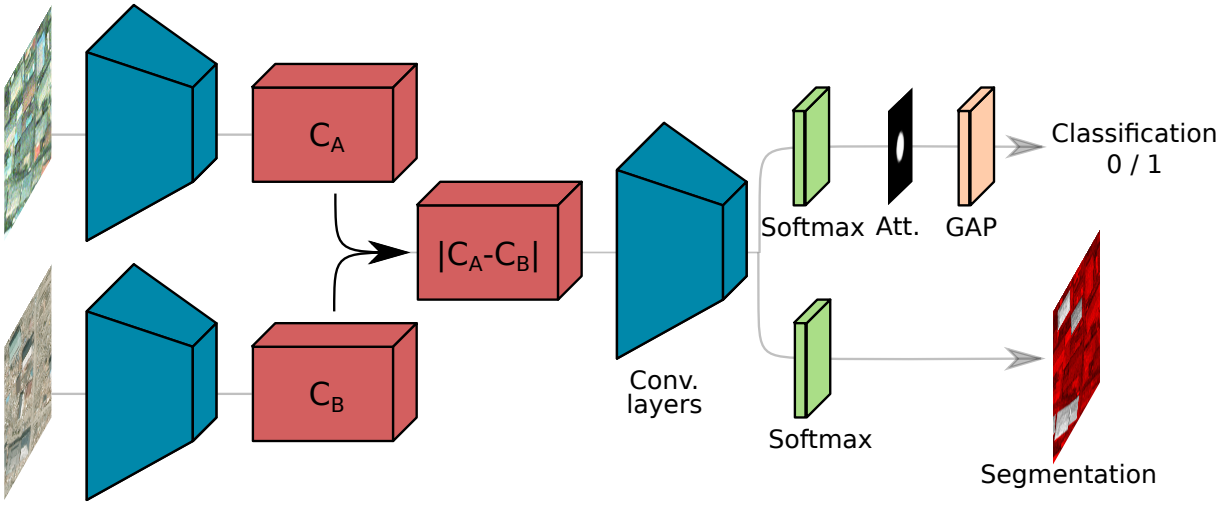


Figure 5.6: Basic schematic of the network used for weakly supervised change detection. Two paths can be taken: the classification path uses the proposed attention layer and global average pooling to produce a classification of the image, while the segmentation path avoids these steps to output pixel-level predictions. Supervision is only available on the classification path.

fail to make use of the heuristic described above, and do not learn to prioritize some areas of the image over others when making classification predictions.

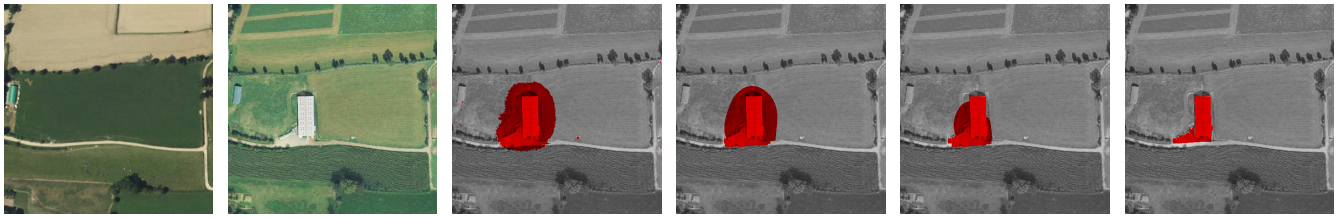
Attention mechanisms have been successful in applications such as machine translation [VSP+17] and class-attention for classification of aerial images [HMZ19]. We propose a learned spatial-attention layer that can be used to allow the network to learn which positions of the images are more discriminative and should be prioritized over others when making inferences. We also propose to use the GAD algorithm to further focus the attention of the network on the most relevant features. Let's assume that a feature map x of size $C \times M \times N$ is obtained after any number of convolution, pooling and other operations from an input image (or images in the case of change detection), where C is the number of channels and M and N are spatial dimensions. We define a matrix A of size $M \times N$ which will be learned by the network, and will serve as attention weights given to spatial locations. The attention operation $f(x)$ can then be defined as

$$f(x)_{c,i,j} = \alpha \cdot x_{c,i,j} \cdot \sigma(a_{i,j}), \tag{5.5}$$

where $a_{i,j}$ denotes the element of A in position (i, j), σ denotes the sigmoid function and α is a normalization term defined as

$$\alpha = \sum_{i=1}^{M} \sum_{j=1}^{N} \sigma(a_{i,j}). \tag{5.6}$$

The sigmoid function is used to ensure the attention weights given to each spatial location is in the range $(0, 1)$. The matrix A is initialized as a null matrix so that all spatial locations have equal attention values of $\sigma(a_{i,j}) = 0.5$. Random initialization of A is neither necessary nor recommended.



(a) Image 1 (b) Image 2 (c) Naive prediction (d) 2000 it. (e) 5000 it. (f) 20000 it.

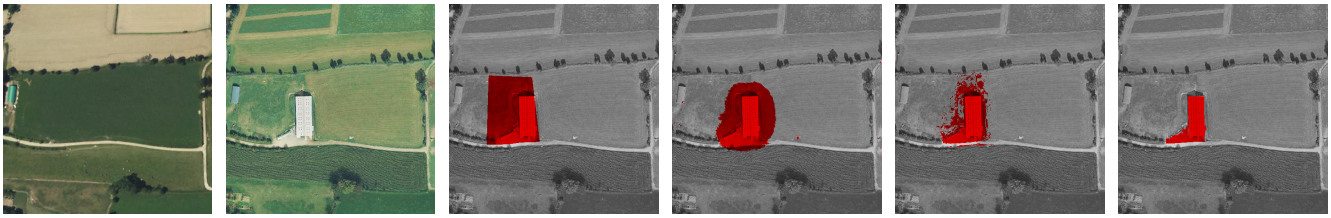
Figure 5.7: Guided anisotropic diffusion for filtering a real example of semantic segmentation. The diffusion allows edges from the guide images to be transferred to the target image, improving the results.

The proposed attention layer is designed to be used after a softmax operation and before a global average pooling (GAP) layer. Doing so will force the network to produce per-pixel classification predictions, which are then put through a weighted average operation whose weights are learnable parameters which depend only on the spatial position of each feature. Global average pooling is preferable to max pooling at the end of a network when we want the network to be able to localize objects, as was discussed in [ZKL+16]. Note that the number of learnable parameters introduced by this attention layer is only $M \cdot N$, which is extremely small in the context of deep neural networks. At inference time, the learned attention weights can be further adapted to the input images by using the GAD algorithm proposed in Section 5.2.1. This helps the network to focus its attention at the building at the center of the image pair, further increasing classification performance.

In this work, we incorporate this attention layer into the classification branch of the architecture depicted in Fig. 5.6. The images are processed by two convolutions with stride $\frac{1}{2}$ and 5 residual blocks before their features are merged, and 4 residual blocks after. This architecture allows us to perform either classification or segmentation by choosing either of the paths at the end. This architecture is a straightforward Siamese extension of the ideas presented in [ZKL+16] with residual blocks. Supervision is only available for the classification branch, but the structure of the network allows us to apply equivalent classification operations at each spatial locations by avoiding the attention and global average pooling layers, effectively performing semantic segmentation.

5.3 Experiments

The experiments presented in this chapter have been divided into two sections. The first ones explore how the proposed methods allow us to refine approximate labels to obtain pixel-level change detection more accurately than through direct supervision. Then, we present experiments that show the effectiveness of the spatial attention layer in performing weakly supervised change detection, using only image-level labels to perform pixel-level predictions. The GAD algorithm is used in both cases to improve the obtained results.



(a) Image 1 (b) Image 2 (c) Reference data (d) Naive prediction (e) Dense CRF (f) GAD

Figure 5.8: Comparison between (c) original dataset ground truth, (e) prediction filtered by Dense CRF, and (f) prediction filtered with guided anisotropic diffusion for 20000 iterations.

5.3.1 Label Refinement Through Iterative Learning

To validate the iterative training scheme proposed in Section 5.2 we adopted the hybrid change detection and land cover mapping fully convolutional network presented in Chapter 4, since it was already proven to work with the HRSCD dataset. We adopted *strategy 4.2*, in which the land cover mapping branches of the network are trained before the change detection one to avoid setting a balancing hyperparameter. The land cover mapping branches of the network were fixed to have the same parameter weights for all tests presented in this section, and evaluating those results is not done here as the scope of these experiments is restricted to the problem of change detection refinement.

We applied the GAD algorithm to the predictions from a network trained directly on the reference data from HRSCD to evaluate its performance. In Fig. 5.7 there is an example of the obtained results. As noted before, we can see in (c) that the change is detected but unchanged pixels around it are also classified as changes by the network. In (d)-(f) it can be clearly seen how the GAD algorithm improves the results by diffusing the labels across similar pixels while preserving edges from the input images in the semantic segmentation results. As expected, more iterations of the algorithm lead to a stronger erosion of "out-of-bounds" labels. For these results, GAD was applied with $K = 5$ and $\lambda = 0.24$. In Fig. 5.8 we can see a comparison between GAD and the Dense CRF[1] algorithm [KK11]. While the non-local nature of fully connected CRFs is useful in some cases, we can see the results are less precise and significantly noisier than the ones obtained by using GAD.

To perform quantitative analysis of results, it would be meaningless to use the test data in the HRSCD dataset given that we are attempting to perform a task which is not the one for which ground truth data are available, *i.e.*we are attempting to perform pixel-level precise change detection and not parcel-level change detection. For this reason we have manually annotated the changes as precisely as possible for two 10000x10000 image pairs in the dataset, for a total of $2{\cdot}10^8$ test pixels, or 50 km^2. The image pairs were chosen before any tests were made to avoid biasing the results. Due to the class imbalance, total accuracy, *i.e.*the percentage of correctly classified pixels, provides us with a skewed view of the results biased towards the performance on the class more strongly

[1] `https://github.com/lucasb-eyer/pydensecrf`

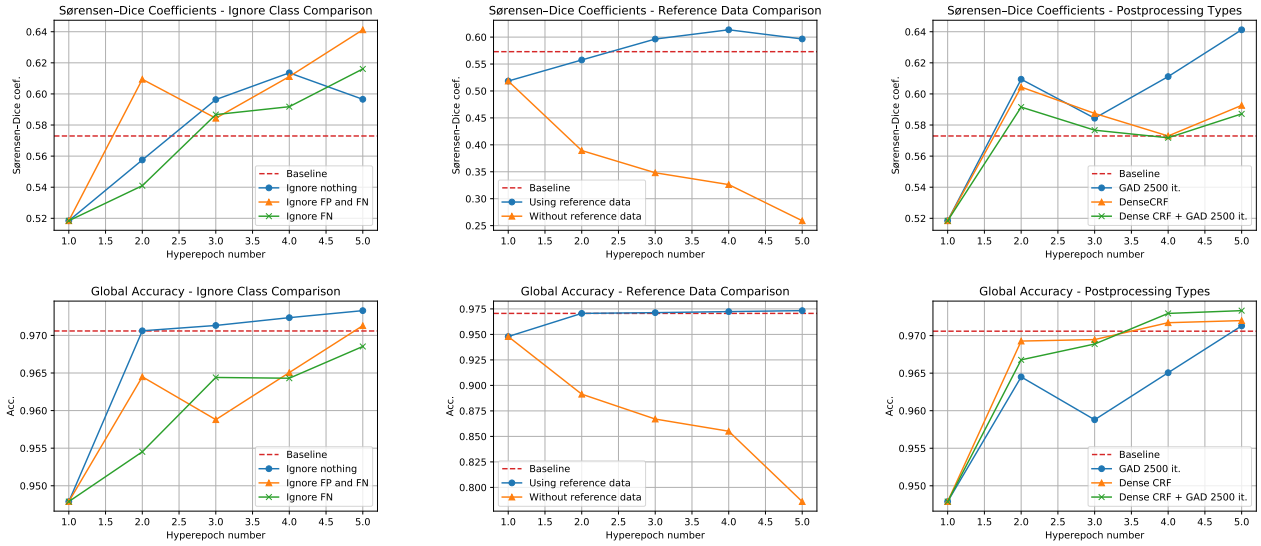


Figure 5.9: Ablation studies. (a) Comparison between strategies for merging network predictions and reference data. (b) Comparison between iterative training with and without the usage of original reference data. (c) Comparison between GAD and Dense CRF. Top row contains Dice scores, bottom row contains global accuracy curves.

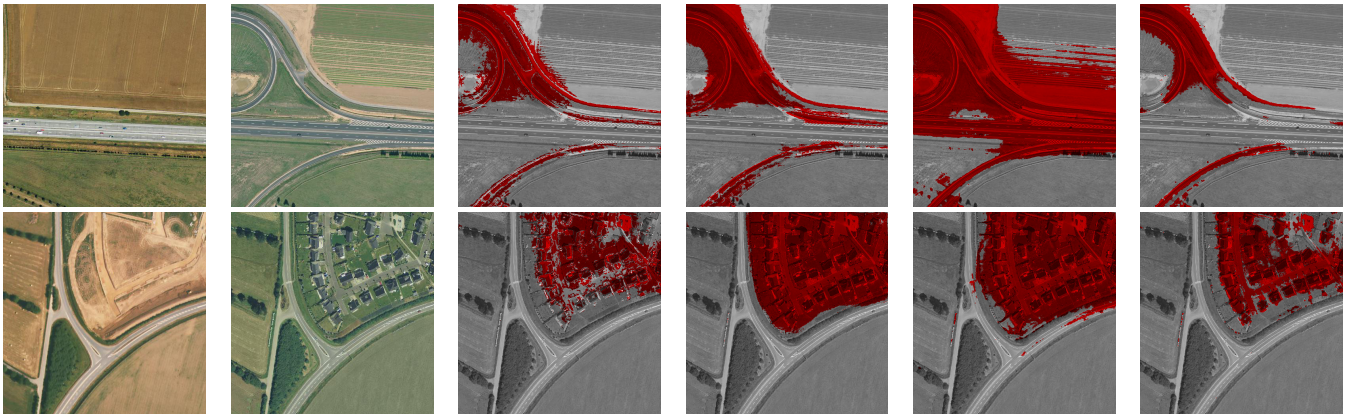
represented. Therefore, the Sørensen-Dice coefficient (equivalent to the F1 score for binary problems) from the point of view of the change class was used [Dic45, Sør48]. The Sørensen-Dice coefficient score is defined as

$$Dice = (2 \cdot TP)/(2 \cdot TP + FP + FN) \tag{5.7}$$

where TP means true positive, FP means false positive, and FN means false negative. It serves as a balanced measurement of performance even for unbalanced data.

All tests presented here were done using PyTorch [PGC+17]. At each hyperepoch, the network was trained for 100 epochs with an ADAM algorithm for stochastic optimization [KB14], with learning rate of 10^{-3} for the first 75 epochs and 10^{-4} for the other 25 epochs. The tests show the performance of networks trained with the proposed method for 5 hyperepochs (iterations of training and cleaning the data), where the first one is done directly on the available data from the HRSCD dataset. For accurate comparison of methods and to minimize the randomness in the comparisons, the obtained network at the end of hyperepoch 1 is used as a starting point for all the methods. This ensures all networks have the same initialization at the point in the algorithm where they diverge. A baseline network was trained for the same amount of epochs and hyperepochs but with no changes done to the training data. This serves as a reference point as to the performance of the fully convolutional network with no weakly supervised training methods.

The first comparison, shown in Fig. 5.9(a), compares the three methods proposed in Section 5.2.2 to combine the network predictions with the original ground truth from the HRSCD dataset. We notice that all three strategies surpass the baseline network using the proposed iterative training method, which validates the ideas presented



(a) Image 1 (b) Image 2 (c) Baseline (d) GAD 2500 it. (e) No reference constraint (f) Dense CRF

Figure 5.10: Change maps obtained by using different methods on two image pairs. Detected changes are marked in red color.

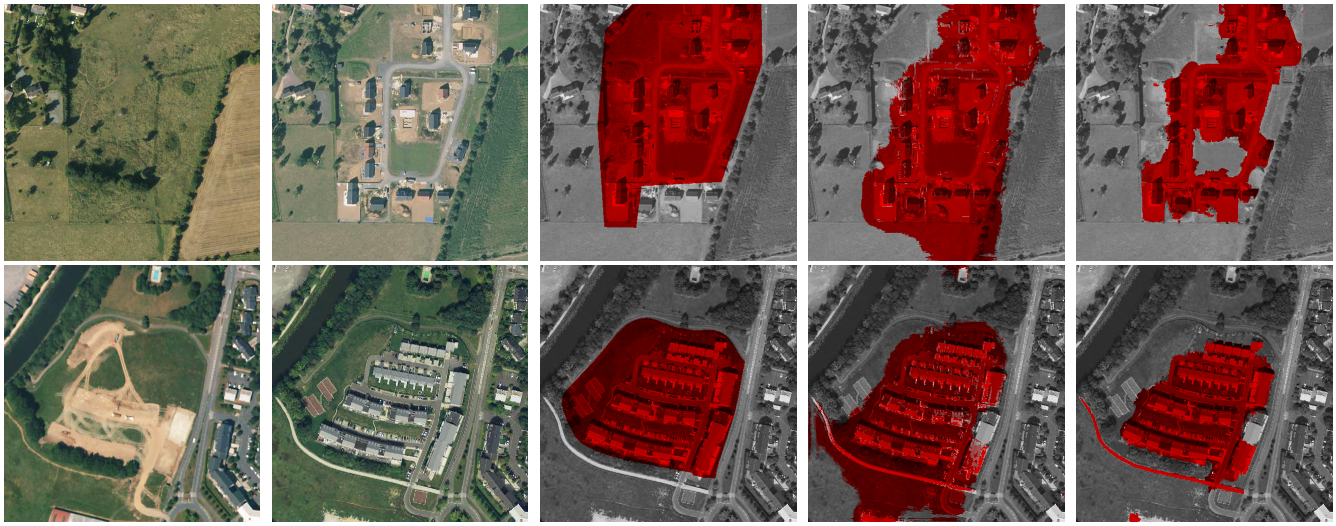
earlier. In Fig. 5.9(b) we see a comparison between a training using the full training scheme proposed in this chapter (without the usage of an ignore class) and the same method but without using the original reference data, *i.e.*using only network predictions processed by GAD to continue training at each hyperepoch. Our results, which corroborate the ones in [KBH+17], show that referring back to the original data at each hyperepoch is essential to avoid a degradation in performance.

In Fig. 5.9(c) we show a comparison between using the proposed GAD algorithm versus the Dense CRF [KK11] algorithm in the iterated training procedure, as well as using both together. We see that using the Dense CRF algorithm to process predictions leads to good performance in early hyperepochs, but is surpassed by GAD later on. This is likely explained by the non local nature of Dense CRF and its ability to deal with larger errors, but its inferior performance relative to GAD for finer prediction errors.

Figure 5.10 shows the predictions by networks trained by different methods on two example images. We see that the best results are obtained by using the full training scheme with GAD in (d), followed by Dense CRF, which also achieves good results shown in (f). The baseline results in (c), obtained by naively training the network in a supervised manner, and the ones without using the reference data as constraint in the iterative training scheme shown in (e) are significantly less accurate than those using GAD or Dense CRF.

5.3.2 Scene-Invariant Spatial Attention Layer

We tested the proposed method using the ABCD dataset proposed by Fujita *et al.* [FSI+17]. This dataset contains pairs of crops of images centered on buildings that have been surveyed to evaluate their destruction after a tsunami. We have followed the 5-fold cross validation that was defined by the dataset's creators. All networks were trained from scratch using only the ABCD dataset, using an initial learning rate of 0.005 for 10 epochs, then with a linearly decaying learning rate for 90 epochs for a total of 100 epochs. The classification results for these tests are presented



(a) Image 1 (b) Image 2 (c) Reference data (d) Naive prediction (e) GAD

Figure 5.11: Results using the complete inference pipeline. GAD is used to improve predictions during the iterative training process as well as for improving the final segmentations.

in Table 5.1. These results show that our network with the attention module performed very similarly to the ones presented in [FSI+17]. It is also clear that the proposed attention module improved the classification accuracy of the networks significantly. The obtained results also show that filtering the attention weights using the GAD algorithm further increases the classification performance of the proposed network, improving the quality of the attention weights by using the input images as guides.

Figure 5.12 show the learned spatial attention weights learned in each of the performed tests. We can clearly see how consistent the network was in identifying that the most discriminative region of the images was located in the center. It is also apparent that the network identified that the scale of this discriminative region is larger in the *resized* version of the dataset compared to the *fixed-scale* version.

Qualitative analysis of segmentation results show that the usage of the proposed spatial attention operation allowed the network to vastly increase its capacity to localize features in the input images, which led to much more accurate segmentation, as depicted in Fig. 5.13. The results also show how using the GAD algorithm for post-processing further increased the spatial accuracy of the segmentation results.

These results suggest that there is a positive feedback loop that happens during the training process between the network's ability to localize discriminative features and the spatial attention operation. Once the network develops the ability to roughly localize discriminative features, this allows the training of the spatial attention layer, which leads the network to learn even more local features, and so on.

Two notable examples can be seen in Fig. 5.13. The first one is the example in the fourth row, which shows that the network is not simply finding buildings in the second image and marking those as unchanged. In this example, a building is present in the second image but it is marked as a change nonetheless since it doesn't match the buildings

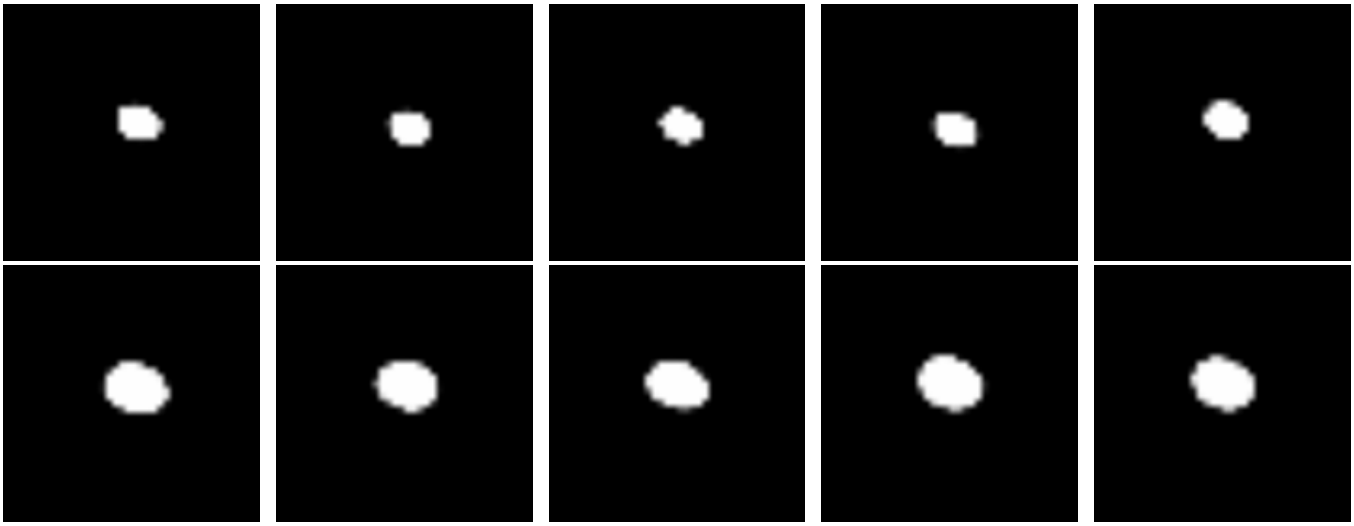


Figure 5.12: Spatial attention weights that were learned in each of the cross-validation tests. Top row contains all 5 tests using fixed scale ABCD dataset, bottom row are the results using the rescaled version of the dataset. Note that the network was incredibly consistent in identifying the center of the images as most discriminative without any explicit knowledge. These attention matrices are of size 40×40.

Method	Fixed scale	Resized
6-ch [FSI$^+$17]	94.5 ± 0.5	94.7 ± 0.3
siam [FSI$^+$17]	94.8 ± 0.3	94.9 ± 0.4
No attention	89.33 ± 0.79	90.96 ± 0.65
Attention	94.36 ± 0.26	94.88 ± 0.18
Attention + GAD	94.58 ± 0.27	94.90 ± 0.22

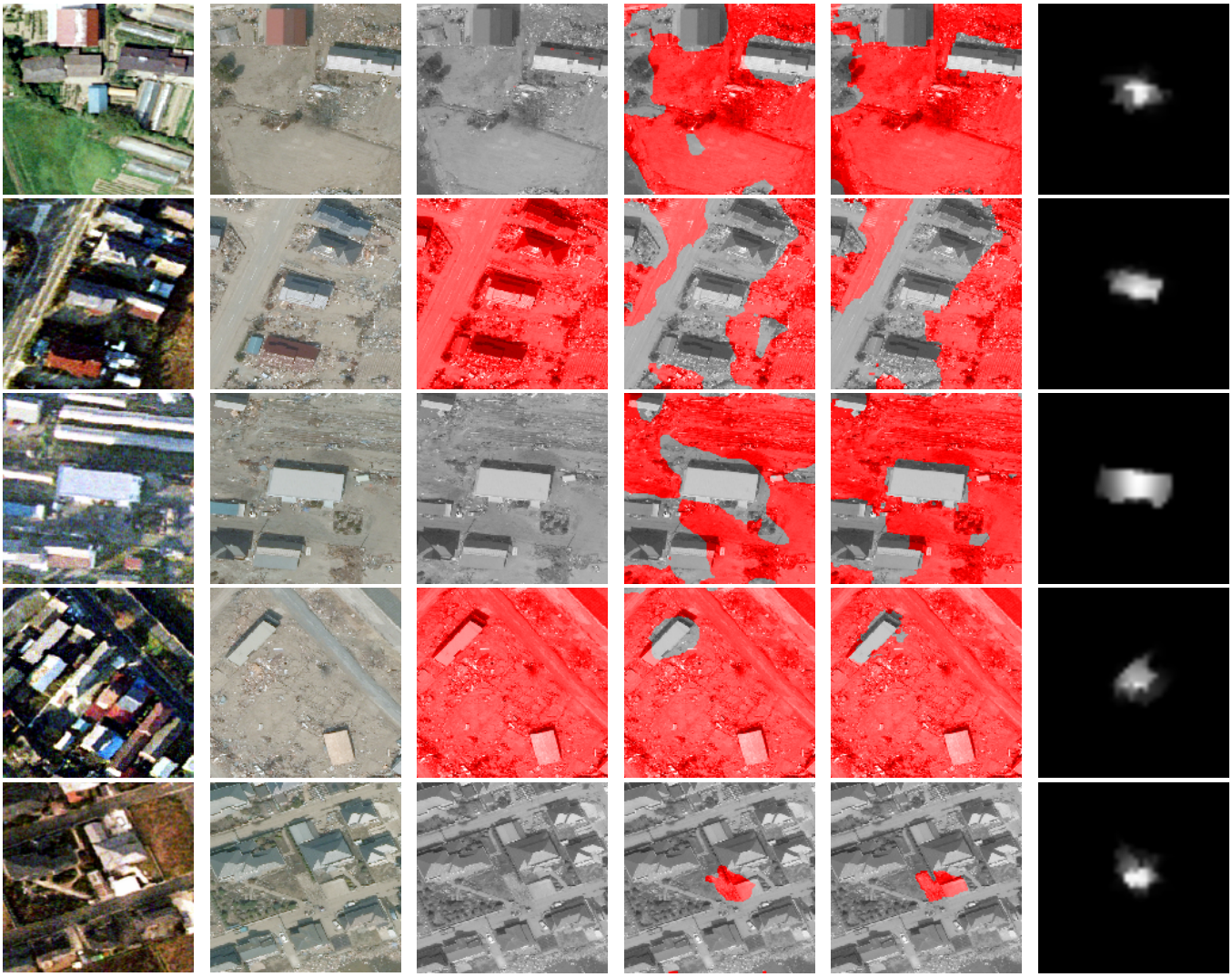
Table 5.1: Accuracy and standard deviation for each test on ABCD dataset using 5-fold cross validation. Fixed scale and resized variations of the ABCD dataset were tested. Results from methods proposed by Fujita *et al.* are included for comparison.

in the first image. The second notable example is the one showed in the last row, where a very small change was detected in the center of the image, surrounded only by unchanged buildings. Since the position of this detected change coincided to the spatial attention position, the network was able to mark this image pair as a change, which is correct according to the ground truth label. The same was not accomplished by the network without the attention layer.

5.4 Analysis

The experiments presented in the previous section showed how GAD was successfully used in two different weakly supervised change detection settings. The results show an increase in performance in object-level segmentation from parcel-level labels through label cleaning, as well as the seldom explored task of weakly supervised image co-segmentation using classification labels.

The iterative training results made clear that it is of paramount importance to refer back to the ground truth data



(a) Image 1 (b) Image 2 (c) No attention (d) Attention (e) Att. + GAD (f) Att. coef.

Figure 5.13: Results obtained by using the proposed method. Note that when the attention layer is not used, the network does not learn to localize the features and tends to predict all pixels into the same class. The attention layer enables the network to localize features much more accurately, and the GAD post-processing further increases the spatial accuracy of such predictions.

every time the training ground truth is being modified. Not doing so leads to a fast degradation in performance, since the network simply attempts to learn to copy itself and stops learning useful operations from the data. The results also showed that separating dubiously labelled pixels leads to a small increase in performance, likely due to the fact that we end up providing a cleaner and more trustworthy dataset at training time.

The guided anisotropic diffusion algorithm was compared against the Dense CRF algorithm for using information from the input images to improve semantic segmentation results. While both algorithms were successful when used in the proposed iterative training scheme, GAD outperformed Dense CRF at later hyperepochs for quantitative metrics. Both algorithms yielded visually pleasing results, each performing better in different test cases.

One possible criticism of the proposed iterative training method is that it would get rid of hard and important examples in the training dataset. It is true that the performance of this weakly supervised training scheme would likely never reach that of one supervised with perfectly clean data, but the results in Section 5.3 show that using the proposed method we can consistently train networks that perform better than those naively trained with noisy data directly.

The proposed spatial attention operation was showed to be useful in improving the classification and weakly supervised segmentation results for datasets which are cropped using object locations as reference points. While this is a particular case, such datasets are often available or can be easily generated for remote sensing applications, where georeferenced data is widely available. The proposed ideas have been only tested in a two-class problem, but there is nothing that indicates that such methods would not work just as well in a multi-class context. Filtering the attention weights with the GAD algorithm further increased the classification performance of the network by increasing the coherence between the attention weights and the region where the building of interest is located in each image.

5.5 Conclusion

In this chapter we have proposed the guided anisotropic diffusion algorithm for improving semantic segmentation results by performing a cross-image edge preserving filtering. We have proposed two GAD-based weakly supervised change detection methods to demonstrate how it can help to recover from inaccurate segmentation labels or go beyond the available classification labels.

We first proposed an iterative training method for training networks with noisy data that alternates between training a fully convolutional network and leveraging its predictions to clean the training dataset from mislabelled examples. We showed that the proposed method outperforms naive supervised training using the provided reference data for change detection. The GAD algorithm was used in conjunction with the iterative training method to obtain the best results in our tests. The GAD algorithm was compared against the Dense CRF algorithm, and was found to be superior in performance.

We then proposed a spatial attention operation that can be easily incorporated into existing classification networks that significantly improve the classification and weakly supervised segmentation performances for datasets with object-aligned crops.

The proposed methods are useful when using data-based approaches in data-scarce domains, as is the case of change detection. We have observed improvement in all of our tests when approaching the problem from a weakly supervised perspective, as opposed to naive supervision. This is a step towards more precise systems for accurately measuring and quantifying the surface areas of changes and the evolution of image terrains. The value of this would be to more efficiently extract information from noisy and lower complexity data and to perform analyses such as urban expansion quantification, deforestation measurement, natural disaster evaluation, etc. It would be interesting to test the efficacy of the proposed ideas outside the context of change detection. The proposed methods could be applied with minor adaptations to other applications to help mitigate the effects of data scarcity and data noise. The proposed attention layer could be used to improve classification results when crops are centred on the object of interest (e.g. a classifier coupled with a region proposal network). GAD and iterative training could be used to improve segmentation results in other cases where overlarge prediction of some classes is a recurring problem.

Chapter 6

Domain Adaptation for Change Detection

Chapter Summary

Different events can lead to changes with different appearances, which can be a problem for neural networks if the test examples differ too much from the ones used for training. Finding ways to bridge this domain gap between different events or even different geographical regions could help networks generalise to unseen cases that are farther from those in the training set. Unlike in Chapter 5, where label quality was the focus, here we deal with issues with the data distribution itself and how to mitigate the effects of domain shifts between training and test data.

This chapter studies a novel domain adaptation method based on adversarial learning, with the final aim of performing change detection across datasets containing images from different types of natural catastrophes. The scope of the proposed method is nevertheless wider than just change detection. Thus, this chapter first steps away from the problem of change detection to present the domain adaptation method in a more general formulation, and the experiments contain not only change detection tests but also image classification and semantic segmentation ones.

The main contribution that is presented in this chapter is a domain adaptation method, named *domain invariant encoding*, that aims to find a latent representation space where images from either domain are indistinguishable. The desired task is then performed using the projections of the images in this latent space. Unlike previous methods, feature space alignment is done indirectly through cycle consistent adversarial learning with the addition of a simple L1 similarity loss.

The proposed method is validated in three main experiments:

- **Image classification**: digit recognition across different datasets, including handwritten and house numbers.
- **Semantic segmentation**: synthetic to real street scenes interpretation.
- **Change detection**: damaged building detection using wildfire, hurricane, and tsunami images.

6.1 Motivation

Deep neural networks excel at learning to perform various visual tasks from large amounts of accurately labeled data. Such networks often fail at generalizing to new datasets with different characteristics, even in cases that humans find trivial. Such failures often occur when the target domain is shifted in any way from the source domain, *e.g.*synthetic to real images, different geographical regions, different weather conditions, or different sensors [PUK+17, SDVG18]. Ideally, these systems would be robust to domain shifts or, alternatively, allow for unsupervised domain adaptation (UDA).

Most UDA methods fall into one of three main groups. The first group of methods attempt to align the domains in a feature space [LCWJ15, SS16, GL15, THSD17]. Methods in the second group aim to translate target data into the source domain, and use this representation to perform the desired task [HTP+18, LT16, GKZ+16]. Finally, a last group of methods attempt to promote the learning of operations that are robust to domain shifts by means of proxy measurements which do not require labels [LCWJ15, VJB+19]. The method presented in this chapter belongs to the first group, although a possible variation that will be presented later belongs to the third group.

The human visual system possesses an incredible ability to extract abstract representations from images in a very robust way regardless of domain shifts. UDA algorithms in the first family described above attempt to emulate this capability, but they rely on proxies for comparing representations computed from different domains such as maximum mean discrepancy [LCWJ15], correlation distance [SS16], or adversarial training [GL15, THSD17]. This is necessary since the source and target data are not paired and therefore representations cannot be compared directly.

Many UDA methods rely on adversarial training to approximate source and target domains. Such methods use a discriminator whose task is to separate data coming from source and target domains, be it at image level [HTP+18], feature level [THSD17, HWYD16], or output level [THS+18, VJB+19]. As noted in [HTP+18], feature-level methods often fail at maintaining semantic consistency when translating domains, and struggle to encode low-level visual appearance variance.

We propose a novel UDA method, named *domain-invariant encoding* (DINE), that uses cycle-consistent ad-

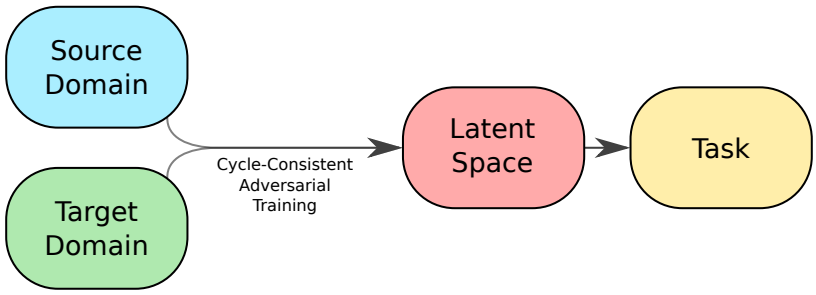


Figure 6.1: DINE aims to extract a code in a latent space where the desired task can be performed regardless of the image's domain of origin.

versarial training [ZPIE17] to obtain a feature-space representation of a scene robust to source-to-target domain shifts. The unsupervised two-way image translation method CycleGAN [ZPIE17] enables the network to find a domain-invariant feature space into which images can be projected from either the source or the target domains through the use of a simple $\ell 1$ norm loss. We show that this method is more effective in enforcing feature coherence between source and target features than direct adversarial training, and can provide qualitative and quantitative measurements of how well the feature spaces are aligned.

We validate our domain adaptation method on three image analysis tasks: classification, semantic segmentation, and semantic co-segmentation. Classification tests are done using various digit recognition datasets, validating our method in a case where data sources differ. Our semantic segmentation tests show how effective our method is at bridging the gap between simulated and real data. Finally, we tackle change detection in remote sensing image pairs [Sin89, HCC+13] as an image co-segmentation problem, showing that our method allows deep neural networks to generalize between different geographical areas and different types of natural disasters. Such tests show how the proposed method can help address harmful accidents that happen worldwide, and may help mitigate the consequences of such events.

6.2 Unsupervised Domain Adaptation

The motivation for unsupervised domain adaptation comes from the need to overcome the lack of supervision in some situations. In some cases, labeled data is expensive to acquire, while labeled synthetic data is much cheaper to generate [RVRK16]. The lack of labels is also an issue when timely results are necessary, such as when analyzing natural disaster images [GGP+19]. UDA methods attempt to bridge the shift between a source domain where labeled data is available and the target domain where a desired task is to be performed. Supervised methods often fail when applied to data significantly different from that which has been seen during training, and UDA methods attempt to bridge that gap [Csu17].

Several successful UDA methods have proposed ways to approximate image representations in feature space. In deep adaptation networks [LCWJ15] hidden representations of layers are embedded in a reproducing kernel Hilbert space where the mean embeddings of different domain distributions can be explicitly matched. In deep CORAL [SS16], second-order statistics of the source and target distributions are aligned with a linear transformation.

Many authors have proposed adversarial methods for UDA based on generative adversarial networks (GANs) [GPAM+14, SGZ+16]. Adversarial methods consist of solving a minimax problem by training two or more networks to perform contrary tasks. Notably, a generator G attempts to generate images that are indistinguishable from real images, while a discriminator D attempts to separate real and generated images. Ganin *et al.* proposed a gradient reversal layer that allows the adversarial training of a discriminator that attempts to separate source and target data [GL15, GUA+16]. CoGAN [LT16] jointly learns a representation space for source and target domains which

is used for image translation. CyCADA [HTP+18] performs image translation from target to source domain using a modified version of CycleGAN [ZPIE17] that attempts to preserve semantic information using a pre-trained network on source domain data, which assumes a network trained on source data should also apply to target data. ADDA combines discriminative modeling, untied weight sharing, and a GAN loss to approximate source and target domain in feature space [THSD17]. Adapt-SegNet uses discriminators on segmentation maps instead of image features with good results [THS+18]. ADVENT uses adversarial training to reduce the entropy of segmentation predictions [VJB+19]. We refer the reader to [Csu17] for a more in-depth review of visual domain adaptation methods.

6.3 Formulation

The unsupervised domain adaptation method we propose in this chapter is based on the CycleGAN framework [ZPIE17], where image-to-image translation is learned from unpaired data using cycle-consistent adversarial training. In this section we will briefly summarize the ideas from the CycleGAN framework that are essential to our UDA algorithm. We will then explain in detail how our method learns to perform domain-invariant encoding from unpaired data.

6.3.1 Cycle-Consistent Unpaired Image-to-Image Translation

Zhu *et al.* remark in [ZPIE17] that attempting to learn image-to-image translation in an adversarial fashion using unpaired data is very prone to mode collapse, *i.e.*the networks output very similar images regardless of the given inputs. Their work solves this problem by introducing an auxiliary cycle-consistency loss, which is based on the observation that an image that is put through a forward translation from domain A to domain B and a backward translation back to domain A should remain mostly unchanged[1]:

$$F_{B\to A}(F_{A\to B}(I)) \approx I \ . \tag{6.1}$$

Thus, image translation networks $F_{A\to B}$ and $F_{B\to A}$ are learned simultaneously. At the same time, discriminators D_A and D_B are trained in an adversarial manner to differentiate between real and translated image in domains A and B, respectively. This is done by solving the following problem:

$$F^*_{A\to B}, F^*_{B\to A} = \underset{F_{A\to B},F_{B\to A}}{\arg\min} \max_{D_A,D_B} \mathcal{L}(F_{A\to B}, F_{B\to A}, D_A, D_B) \ , \tag{6.2}$$

[1]The notation here differs from the one used in [ZPIE17] for consistency with the following section.

where $\mathcal{L}$ is composed of adversarial and cycle consistency terms

$$\mathcal{L}(F_{A\to B}, F_{B\to A}, D_A, D_B) = \mathcal{L}_{cyc}(F_{A\to B}, F_{B\to A}) + \mathcal{L}_{GAN}(F_{A\to B}, D_B) + \mathcal{L}_{GAN}(F_{B\to A}, D_A) \; . \tag{6.3}$$

The adversarial term $\mathcal{L}_{GAN}(F_{A\to B}, D_B)$ is defined as

$$\mathcal{L}_{GAN}(F_{A\to B}, D_B) = \mathbb{E}_{I_A\sim A}[\log(1 - D_B(F_{A\to B}(I_A)))] + \mathbb{E}_{I_B\sim B}[\log(D_B(I_B))] \; . \tag{6.4}$$

The symmetric term $\mathcal{L}_{GAN}(F_{B\to A}, D_A)$ can be obtained by simply flipping the domains in Eq. 6.4. The LSGAN loss function can also be used here instead [MLX+17]. Finally, the cycle consistency loss term $\mathcal{L}_{cyc}(F_{A\to B}, F_{B\to A})$ is defined as

$$\mathcal{L}_{cyc}(F_{A\to B}, F_{B\to A}) = \mathbb{E}_{I_A\sim A}[||F_{B\to A}(F_{A\to B}(I_A)) - I_A||_1] + \mathbb{E}_{I_B\sim B}[||F_{A\to B}(F_{B\to A}(I_B)) - I_B||_1] \; . \tag{6.5}$$

6.3.2 Domain-Invariant Encoding

The main idea behind our method is to condition the translation functions to pass through a latent code that is shared between domains. This code can be obtained from the real or translated images, as well as be decoded into either of them. In the CycleGAN framework, $F_{A\to B}$ and $F_{B\to A}$ are encoder-decoder CNNs. We begin by looking at the encoder (Enc) and decoder (Dec) sections of the networks separately[2]:

$$F_{A\to B}(I) \triangleq Dec_{A\to B}(Enc_{A\to B}(I)) \; . \tag{6.6}$$

This allows us to directly observe the code generated by $Enc_{A\to B}(I)$. We aim to condition the network to find a code that is agnostic to the domain of origin of the image. Thus, the code corresponding to the $A \to B$ translation should be the same as the code corresponding to the $B \to A$ translation:

$$Enc_{A\to B}(I) \approx Enc_{B\to A}(Dec_{A\to B}(Enc_{A\to B}(I))) \; . \tag{6.7}$$

To condition the network to find this domain agnostic encoding, the following code similarity loss functions are defined:

$$\mathcal{L}_{cs,A}(Enc_{A\to B}, Dec_{A\to B}, Enc_{B\to A}) = \mathbb{E}_{I_A\sim A}[||Enc_{B\to A}(Dec_{A\to B}(Enc_{A\to B}(I_A))) - Enc_{A\to B}(I_A)||_1] \tag{6.8}$$

[2]Due to the dual symmetry of this method, some of the derivations are only shown for one of the two cases for the sake of simplicity.

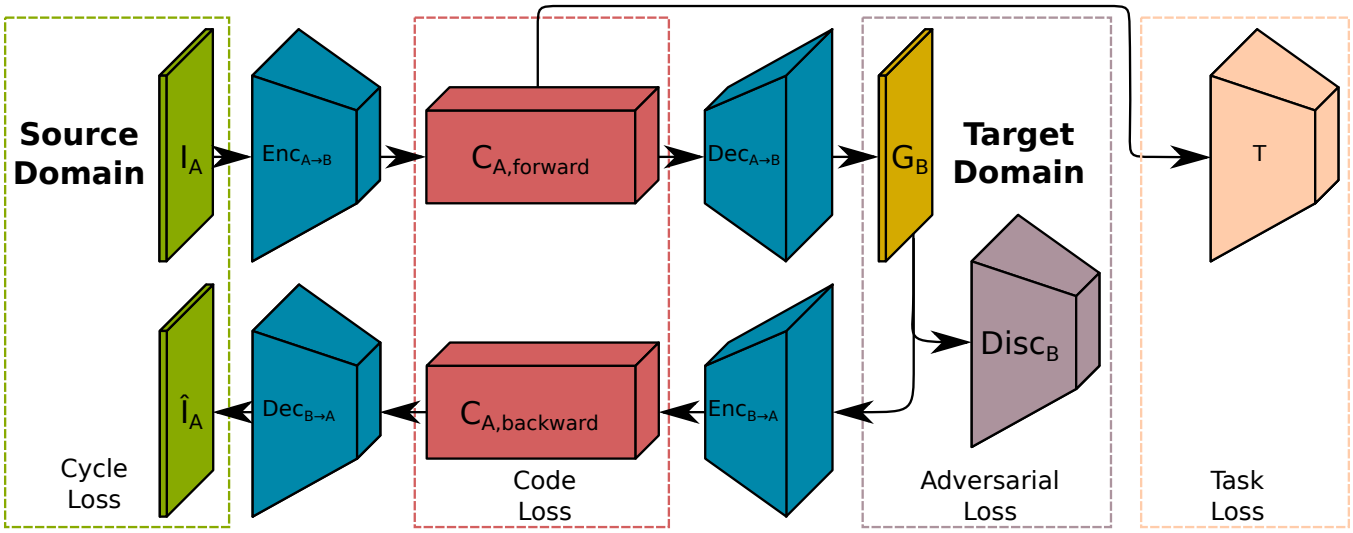


Figure 6.2: Diagram of the DINE algorithm. The two way image translation allows us to enforce code similarity during the forward and backward translations, forcing the network to find a domain agnostic latent representation space.

$$\mathcal{L}_{cs,B}(Enc_{B\to A}, Dec_{B\to A}, Enc_{A\to B}) = \\ \mathbb{E}_{I_B\sim B}[||Enc_{A\to B}(Dec_{B\to A}(Enc_{B\to A}(I_B))) - Enc_{B\to A}(I_B)||_1] \,. \tag{6.9}$$

Note that this formulation does not increase the number of trainable parameters in the system as a feature discriminator would, nor does it increase the training complexity in any way. It is also important that the code generated by the encoders fully encode the input images, therefore techniques frequently used in encoder-decoder architectures such as skip connections [RPB15] should not be used.

We now assume that we want to train a network T to perform a task for which supervision labels Y are only available for the data in domain A. The network F takes as input the features obtained previously, *i.e.*$T(Enc_{A\to B}(I))$. In the later examples, this task will be image classification, semantic segmentation, and image pair co-segmentation. The applications of our method are not restricted to these cases, and it could in principle be used for other pattern recognition tasks (*e.g.*optical flow estimation). The CycleGAN networks are then trained simultaneously with T, which helps $Enc_{A\to B}$ and $Enc_{B\to A}$ learn task-specific features. For the classification case, the cross-entropy loss function is used:

$$\mathcal{L}_{task}(Enc_{A\to B}, T) = -\mathbb{E}_{(I_A,y)\sim(A,Y)} \sum_{k=1}^{K} \mathbb{1}_{k=y} \log(\sigma(T^k(Enc_{A\to B}(I_A)))) \tag{6.10}$$

where σ denotes the softmax function, $\mathbb{1}$ is an indicator function, and K is the number of considered classes. The pixel-wise cross entropy used for semantic segmentation and co-segmentation is nearly identical, except that it is applied at each pixel instead of once for the whole image. Note that this supervised loss is a function of $Enc_{A\to B}$, and therefore has an effect on the CycleGAN networks, helping the encoder learn task specific features.

The total loss function for this problem can then be defined as

$$\mathcal{L}_{DINE} = \mathcal{L}_{cyc,A} + \mathcal{L}_{cyc,B} + \mathcal{L}_{GAN,A} + \mathcal{L}_{GAN,B} + \mathcal{L}_{cs,A} + \mathcal{L}_{cs,B} + \mathcal{L}_{task} \ , \tag{6.11}$$

where the function arguments have been omitted for simplicity. In practice, different loss function terms are weighted by hyperparameters that are used to tune the method's performance. We aim to train the networks in an adversarial manner by solving

$$\Phi^* = \arg\min_{\Phi} \max_{\Psi} \mathcal{L}_{DINE} \ , \tag{6.12}$$

where $\Phi = \{Enc_{A\to B}, Dec_{A\to B}, Enc_{B\to A}, Dec_{B\to A}, T\}$ and $\Psi = \{D_A, D_B\}$.

Finally, one possible variant of the proposed method is to share the encoder weights between $Enc_{A\to B}$ and $Enc_{B\to A}$. In this case, the network attempts to learn domain-invariant operations that extract similar features from each domain, instead of learning domain specific operations that project the images into the same latent space.

6.3.3 Shortcut Decoding

The proposed method opens the door to a validation method that we will refer to as shortcut decoding. Given that the forward and backward translations are supposed to pass through the same domain agnostic code, we can verify feature-space alignment by cross-decoding the encoded image:

$$I_{A,shortcut} = Dec_{B\to A}(Enc_{A\to B}(I_A)) \ . \tag{6.13}$$

The obtained result can then be visually analyzed to qualitatively assess feature-space alignment, or directly compared to I_A to obtain quantitative measurements.

Note that an auxiliary loss function comparing I_A and $I_{A,shortcut}$ could be used but is not necessary, since feature space alignment implies accurate shortcut decoding, but the opposite is not true. Using a code similarity is therefore more adequate.

6.4 Implementation

We adopt for the CycleGAN networks the same architectures as in [ZPIE17]. The main difference is that we consider encoder and decoder separately. The encoders $Enc_{A\to B}$ and $Enc_{B\to A}$ are composed of two convolutional layers with stride 2, followed by 5 residual blocks [HZRS16a]. Instance normalization [UVL17] is used after each convolution. The decoders $Dec_{A\to B}$ and $Dec_{B\to A}$ are composed of 4 residual blocks, followed by two convolutional layers with stride $\frac{1}{2}$. PatchGAN discriminators [IZZE17] are used for D_A and D_B, where 70×70 overlapping patches

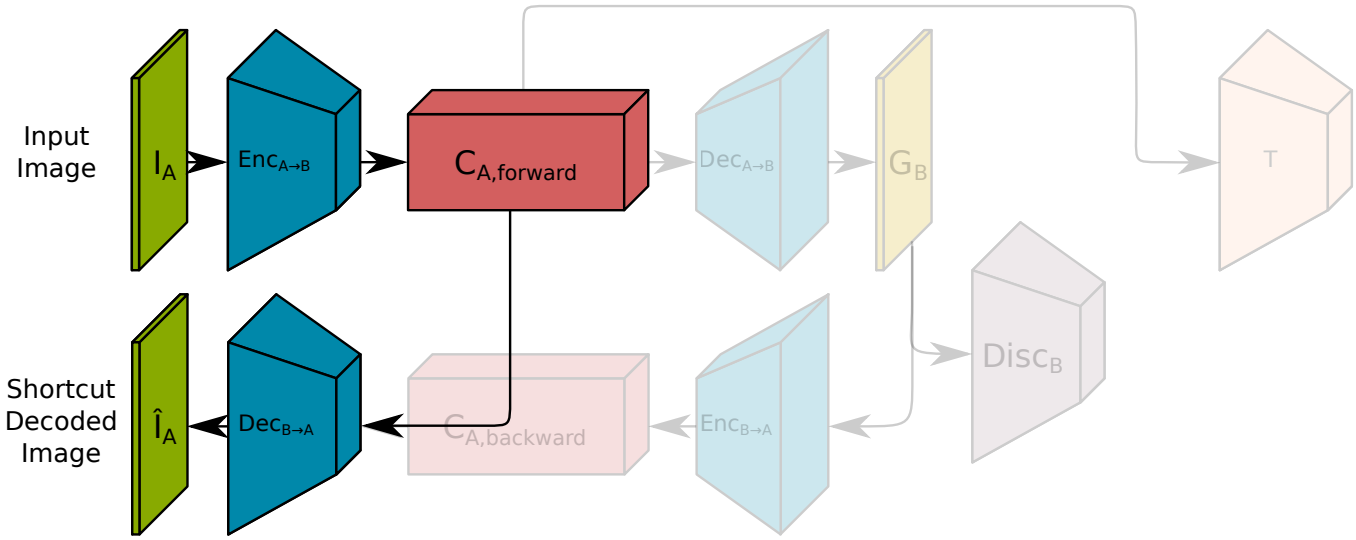
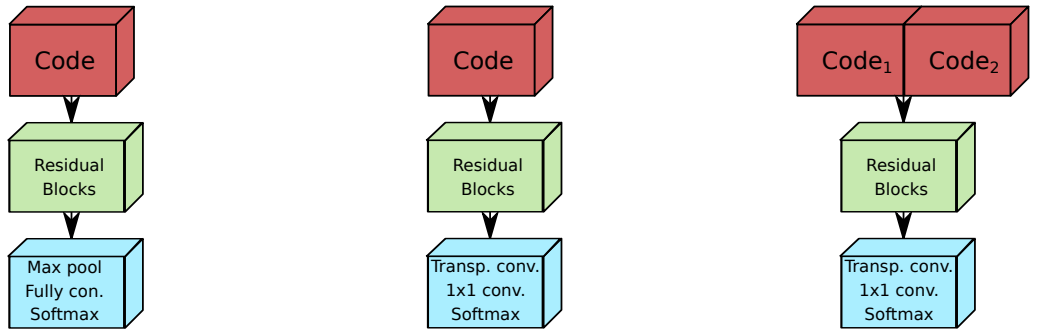


Figure 6.3: Shortcut decoding path, used for validating alignment of feature spaces during forward and backward translations.



(a) Classification T (b) Segmentation T (c) Co-segmentation T

Figure 6.4: Basic schematic for the architectures of T network for each of the considered tasks.

are analyzed in a fully convolutional manner. This discriminator architecture is able to work with images of any size, and has fewer parameters than a full-image discriminator [IZZE17, ZPIE17].

Different architectures for T were explored for each task, as the architecture of T is task specific. The chosen architectures are based on the decoder architecture described above. For the classification task, the final two upsampling convolutional layers are replaced by a global max pooling operation, followed by a fully connected layer. To obtain the segmentation network, the decoder architecture's last layer is replaced by a 1×1 convolution followed by a softmax layer. For the co-segmentation task, each image is encoded separately, and the codes are concatenated before being decoded by an architecture identical to the segmentation one, except for the number of input and output channels. This effectively creates a Siamese network [CHL05a, ZK15a] with shared encoder weights, which has been shown to be very effective to this task [DLB18, COA18, DLBG19].

All tests were done using PyTorch [PGC+17], and trained using the ADAM optimization method [KB14]. All tests have been performed using Nvidia GeForce GTX 1080 Ti 11GB GPUs. While this setup is not able to support some of the larger state-of-the-art architectures, our tests were organized in a way that compares our UDA method to

Model	MNIST → USPS	USPS → MNIST	SVHN → MNIST
Source only (from [HTP+18])	0.822	0.696	0.671
Source only (our tests)	0.957	0.779	0.723
DANN [GUA+16]	-	-	0.736
DTN [TPW17]	-	-	0.844
CoGAN [LT16]	0.912	0.891	-
ADDA [THSD17]	0.894	0.901	0.760
CyCADA [HTP+18]	0.956	0.965	**0.904**
DINE‡	0.980	**0.985**	0.835*
DINE†	**0.982**	0.973	0.713*
DINE (ours)	**0.982**	0.981	0.803
Target only (from [HTP+18])	0.963	0.992	0.992
Target only (our tests)	0.973	0.995	0.995

Table 6.1: Classification accuracy. Models marked with † share encoder parameters. Models marked with ‡ share encoder parameters and are trained without the code similarity loss. Results marked with $*$ convert SVHN images to grayscale to enable symmetric encoders.

other methods in a fair way regardless of the backbone architecture.

6.5 Results

We use DINE to perform unsupervised domain adaptation for three applications. The first one is classification of digits, both handwritten (MNIST [LBBH98] and USPS [Hul94]) and building identification numbers (SVHN [NWC+11]). For segmentation, we explore the adaptation from synthetic images (GTA5 [RVRK16]) to real images (Cityscapes [COR+16]). For co-segmentation, the adaptation between different geographical regions and natural catastrophes is studied using the xBD dataset [GGP+19] that was released in late 2019.

6.5.1 Classification

Our classification tests follow the ones presented in [HTP+18]. UDA is performed between MNIST and USPS datasets of handwritten digits. UDA is also performed from the SVHN dataset [NWC+11] to MNIST [LBBH98]. This is a much more challenging case, since SVHN images do not contain handwritten digits, and images are RGB instead of grayscale. Our proposed method is tested both without and with shared encoder weights. The results can be found in Table 6.1.

Our method yields state-of-the-art performances for UDA between the MNIST and USPS datasets. Notably, the performance of DINE when adapting from MNIST to USPS surpasses target only supervision. This is likely a result of two main factors. Firstly, the number of training images is larger in the MNIST dataset. Second, the image translation in DINE can be seen as a self-supervised auxiliary loss that helps the network learn discriminative features. On the difficult task of adapting from street view house numbers to handwritten digits, DINE obtains the third best performance behind [HTP+18] and [TPW17]. Here we observed that DINE struggled to stabilize the

Model	Backbone	Parameters	mIoU
Source supervision	ResNet-9 [ZPIE17]	11.4 M	0.117
FCNs in the Wild [HWYD16]	VGG-16 [YK16]	50.5 M	0.271
Adapt-SegNet [THS+18]	Deeplab-v2 VGG16 [CPK+16]	29.6 M	0.350
Adapt-SegNet [THS+18]	Deeplab-v2 ResNet-101 [CPK+16]	44.5 M	0.424
CyCADA [HTP+18]	VGG16-FCN8s [LSD15b]	134.4 M	0.354
CyCADA [HTP+18]	DRN-26 [YKF17]	20.6 M	0.395
AdvEnt [VJB+19]	ResNet-101 [HZRS16a]	44.5M	**0.438**
AdvEnt [VJB+19]	ResNet-9 [ZPIE17]	11.4 M	0.108
CyCADA [HTP+18]	ResNet-9 [ZPIE17]	11.4 M	0.117
Adapt-SegNet [THS+18]	ResNet-9 [ZPIE17]	11.4 M	0.125
DINE‡	ResNet-9 [ZPIE17]	11.4 M	0.126
DINE†	ResNet-9 [ZPIE17]	11.4 M	0.137
DINE	ResNet-9 [ZPIE17]	11.4 M	**0.201**

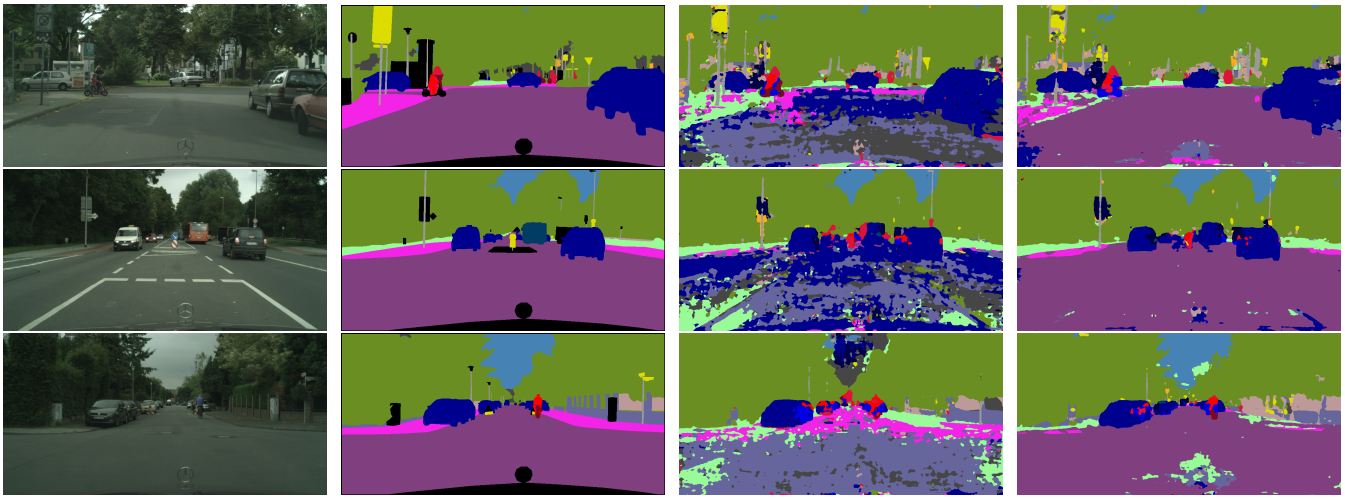
Table 6.2: GTA5 to Cityscapes segmentation performance. Models marked with † share encoder parameters. Models marked with ‡ share encoder parameters and are trained without the code similarity loss.

adversarial training between grayscale and RGB images.

6.5.2 Segmentation

The semantic segmentation tests focus on the adaptation from synthetic GTA5 images to real world Cityscapes images, using the 19 common classes between the two datasets as is done in [HWYD16, THS+18, HTP+18, VJB+19]. Training is done using the training images of each dataset, and testing is done using the validation images from the Cityscapes dataset. While running comparison tests we have realised that the tests presented in most UDA papers vary not only in method, but also in the segmentation architectures that are used. The different backbones have a strong impact in the final results, and these results make it hard to identify if the performance gain comes from the proposed method of from using a better backbone segmentation network. For this reason, we have split our results in two groups. First, results are compared using the reported results for several UDA results and their original segmentation architectures. Second, we used the official codes found online for several UDA methods and tested using a standardized simple segmentation backbone, which should isolate the effect of the considered UDA methods from the chosen architectures.

Table 6.2 contains the obtained semantic segmentation results, as well as a comparison of the used segmentation backbones and number of trainable parameters. The first part of the table contains the results reported in the original paper of each method. The second part contains the results of our tests where several methods were tested using the ResNet-9 segmentation backbone. DINE outperformed all other methods when tested with comparable segmentation architectures. We believe DINE's performance would be more comparable to those reported in other papers if a more advanced segmentation network is used, but our computational setup did not allow for these tests to be made. These results also show that sharing the encoder weights between source and target encoders leads to a performance reduction. This suggests that learning domain-invariant representations is more powerful than



(a) Target image (b) GT (c) No DA (d) DINE

Figure 6.5: Segmentation results on the Cityscapes dataset using the ResNet-9 backbone for segmentation using supervision from the GTA5 dataset.

learning domain-invariant operations.

6.5.3 Co-segmentation

Our final tests for the co-segmentation tasks focus on the problem of change detection on remote sensing image pairs [Sin89, HCC+13]. We use the xBD dataset, which contains images before and after various natural disasters. We have chosen three types of natural disaster present in the dataset for our tests: wildfires, hurricanes and tsunamis. From each domain, we have selected the 50 images with the most changed pixels. Change labels were generated using the ”destroyed building” annotations present in the dataset. The task here is to attribute dense per-pixel change labels based on an input of two co-registered images.

This task is challenging in several ways. For one, several images are affected by atmospheric conditions. This reflects the use case for such images in the real world, in which timely results are expected and the image acquisition conditions are seldom ideal. Furthermore, class distributions are significantly different between some domains. The fire and hurricane domains contain 7-8% of pixels belonging to the change class, while tsunami images contain about 4% of such pixels. This large class imbalance also means that small differences in total accurately classified pixels may have a strong impact on the Dice score.

The difference between the considered domains is clear when looking at the images. The images of destroyed buildings are extremely different in each of the three cases. For wildfires, most of the buildings is gone but the structure and footprint of the buildings usually remains visible in the same location. For flood images, changes are often inferred from contextual clues, i.e. if water surrounds the building in the second image in the pair, that building has likely suffered heavy damage. In the case of the tsunami images, the affected buildings are almost unrecognisable afterwards. Often, it is impossible to identify any signs of destroyed buildings, as most of it has been

Source domain:	Fire						
Target domain	Hurricane			Tsunami			None
Test data ↓	DINE†	DINE‡	DINE	DINE†	DINE‡	DINE	
Hurricane	0.656	0.677	0.651	0.668	0.681	0.120	**0.693**
Tsunami	0.176	0.174	0.110	**0.258**	0.178	0.237	0.156

Source domain:	Hurricane						
Target domain	Fire			Tsunami			None
Test data ↓	DINE†	DINE‡	DINE	DINE†	DINE‡	DINE	
Fire	0.668	0.672	0.686	0.619	0.604	0.333	**0.743**
Tsunami	0.089	0.105	0.086	**0.112**	0.077	0.101	0.042

Source domain:	Tsunami						
Target domain	Fire			Hurricane			None
Test data ↓	DINE†	DINE‡	DINE	DINE†	DINE‡	DINE	
Fire	0.386	0.603	0.384	0.332	**0.646**	0.174	0.505
Hurricane	0.261	**0.550**	0.043	0.280	0.548	0.316	0.493

Table 6.3: Dice score in change detection tests. Models marked with † share encoder parameters. Models marked with ‡ share encoder parameters and are trained without the code similarity loss. Column marked as ”none” refers to source domain supervision. Results highlighted in yellow are the ones where target domain and test data are the same.

washed away or displaced in the image.

Tables 6.3 and 6.4 contain the Dice scores and accuracies obtained in each test, respectively. Domain adaptation is performed in both directions between each pair of domains. Tests of the obtained networks were done using the target domain images as well as the ignored domain images to evaluate how well the networks generalize to unseen cases. Tests on target domain have been highlighted with a yellow background. We observe that the domains of fire and hurricane are not too distant, and that using domain adaptation methods does not improve results over source domain supervision. We do notice that using DINE improves the performances when transferring between either fire↔tsunami and hurricane↔tsunami cases, validating our method when domains are sufficiently distant.

6.6 Limitations and Discussion

The proposed method achieved good results in many of the test cases. As is generally the case with adversarial methods, training stability can be an issue. In our tests this was only a problem when bridging the domain shift between the MNIST and SVHN datasets, probably due to the difference in imaged objects (handwritten digits and house numbers), as well as the different number of color channels. DINE obtained excellent results in the MNIST to USPS tests, surpassing even target only supervision.

In the segmentation tests, our method outperformed state-of-the-art UDA methods in tests with a standardized segmentation network. It did not outperform some other methods on the results reported in their original papers. It is possible that with higher computational power and a more advanced segmentation backbone DINE will achieve more comparable results, but that remains to be tested.

Source domain:	Fire						
Target domain	Hurricane			Tsunami			None
Test data ↓	DINE†	DINE‡	DINE	DINE†	DINE‡	DINE	
Hurricane	0.945	0.946	0.943	0.942	0.948	0.925	**0.950**
Tsunami	0.956	**0.959**	0.952	0.956	**0.959**	0.954	**0.959**

Source domain:	Hurricane						
Target domain	Fire			Tsunami			None
Test data ↓	DINE†	DINE‡	DINE	DINE†	DINE‡	DINE	
Fire	0.958	0.960	0.960	0.955	0.951	0.935	**0.966**
Tsunami	**0.958**	**0.958**	**0.958**	0.957	0.957	**0.958**	0.957

Source domain:	Tsunami						
Target domain	Fire			Hurricane			None
Test data ↓	DINE†	DINE‡	DINE	DINE†	DINE‡	DINE	
Fire	0.935	0.949	0.935	0.935	**0.952**	0.930	0.945
Hurricane	0.925	0.934	0.919	0.925	0.937	0.926	**0.939**

Table 6.4: Accuracy in change detection tests. Models marked with † share encoder parameters. Models marked with ‡ share encoder parameters and are trained without the code similarity loss. Column marked as ”none” refers to source domain supervision. Results highlighted in yellow are the ones where target domain and test data are the same.

(a) Image 1 (b) Image 2 (c) GT (d) DINE (e) No UDA

Figure 6.6: Change detection results between images (a) before and (b) after a natural disaster. (d) DINE results show a significant improvement in adapting to new natural disaster with respect to (e) source-only supervision when compared to the (c) ground truth. Changes are marked in red.

(a) (b) (c) (d) (e)

Figure 6.7: Comparison of shortcut decoded images. (a) Input images. (b) Translated images. (c) Full cycle decoding. (d) Shortcut decoding (DINE). (e) Shortcut decoding (discriminative feature loss). This clearly shows that DINE successfully aligned feature spaces, while a discriminative code loss did not.

The co-segmentation tests showed how challenging it is to apply pattern recognition methods to some real world applications. DINE showed a clear improvement in results in the fire↔tsunami and hurricane↔tsunami tests. Tests like these are still uncommon in the computer vision community, and we hope that the results presented here motivates more applications such as this, as we believe this would help a concrete positive impact in the world.

Our method is lighter than CyCADA [HTP+18], a well known UDA method that is also based on CycleGAN. CyCADA requires pre-training a network on source domain data, and using it to preserve semantic information in the images during translation. It is also based on the assumption that a network trained on source domain data should apply directly to target domain data. We make no such assumption, as we only use the domain agnostic code for the desired task. DINE is also lighter at inference time. At test time, CyCADA first translates the image from target to source domain to then apply the desired task. In DINE, image translation is not done at test time since we have trained encoders for each of the considered domains.

Shortcut decoding images also show that DINE is much more effective at aligning the latent spaces of source and target domain representations than adversarial training at feature level. Results that show this phenomenon can be seen in Fig. 6.7.

6.7 Unpaired Translation of Change Detection Images

Around the same time this work was being developed, other authors, including Luppino et al. [LKB+20] and Saha et al. [SBB18], have explored unpaired image translation using the CycleGAN framework for unsupervised change detection, based on the principle that translated images should more closely match where no changes have occurred,

while not being able to predict changes.

It is easy to see why such translation would be of interest in the case of cross-sensor change detection, where pixel values can't be compared directly. In an ideal scenario, using such image translation techniques would abstract away the different sensor characteristics or even seasonal and other irrelevant variations, allowing the change detection algorithms to compare the images directly.

During the development of the DINE algorithm that was presented above, some tests were conducted using the CycleGAN algorithm to translate images from the ABCD dataset, which contained image pairs taken before and after a tsunami with patch-level change labels. Since the images from before and after were obtained using sensors with different properties, the aim was to translate images taken by the lower quality sensor into higher quality images with similar sensor properties as the other images in the dataset while preserving the content.

An interesting phenomenon was observed when this was attempted, which can be seen in Fig. 6.8. In this case, adversarial training led the CycleGAN translation operations to hallucinate changes to match the statistics in the images taken after the tsunami. This contradicts the original purpose of this operation, since the translation affects not only the sensor statistics (i.e. image style) but also the imaged objects (i.e. image content). This effect was also observed by Luppino et al. in [LKB+20], where pixel relationships are used in attempt to separate changed pixels from unchanged ones during the adversarial training process to reduce these effects.

In the ABCD dataset, where change labels for the central building in each patch are available, this effect can be mitigated by using only image pairs marked as unchanged for training the image translation algorithm. Figure 6.8 shows the different effects of training the translation algorithm using the full dataset (*all*) versus that of training using only unchanged patches (*clean*). Interestingly, the translation algorithm often hallucinates destructed buildings when translating from 1 to 2, but rarely hallucinates real buildings from destructed ones when translating from 2 to 1. The reason for this difference is unclear from these experiments. Furthermore, using only unchanged image patches is not enough to completely stop the network from hallucinating destructed buildings. This is likely due to the fact that these images still contain some destructed buildings surrounding the central building in each patch (which is the one the label refers to).

This highlights the perils of using such adversarial algorithms in the context of image analysis. The fact that a generator network is successful in confounding a discriminator does not necessarily mean that the produced images are "correct" or even realistic. The application of such algorithms in sensitive contexts, such as autonomous driving, must be used very carefully.

6.8 Conclusion

The proposed DINE algorithm for unsupervised domain adaptation has been validated for several cases of domain shift, including different data sources, synthetic to real, and different geographical regions. It was also validated for

(a) Image 1 (b) 1 → 2 (clean) (c) 1 → 2 (all) (d) Image 2 (e) 2 → 1 (clean) (f) 2 → 1 (all)

Figure 6.8: Including changed image pairs when training the CycleGAN networks affects the results, leading to more frequent hallucination of destructed buildings as seen in (b) versus (c). Interestingly, the hallucination of buildings where there are none is much less frequent, as seen in (e) versus (f).

the tasks of image classification, semantic segmentation, and co-segmentation. The results were mostly positive, although its performance did not match that of other method in some tests. In one of the classification tests, DINE's performance was superior to target only supervision due to larger training size and the self-supervision of CycleGAN. Its performance in segmentation was superior to other methods when methods were compared with an identical segmentation network. The co-segmentation tests showed how such methods can help interpret aerial images in the case of natural disasters, even when no disaster-specific labels are available for training neural networks.

Chapter 7

Conclusion

This thesis began with the aim of bridging the gap between the recent advances in computer vision and machine learning and the application of change detection using pairs of remote sensing images. The breakthroughs that convolutional neural networks had for accomplishing other computer vision tasks had not yet happened in change detection. It quickly became clear that there was one main problem holding back these advances: data. Along with other properties that separate the problem of change detection from other computer vision tasks, data availability and quality for this task is a recurrent issue, especially in the amounts that are usually required to apply deep learning methods. This explains the data-centred point of view from where this work is written. Data modalities, amounts, variability, quality and other aspects were central topics in the analyses presented in the previous chapters.

Unlike many other works in this field, the methods presented in this thesis focus on supervised change detection. Unsupervised methods are valuable, especially when labelled data is not available. But supervised methods are often superior in many ways when successfully applied as they benefit from prior information on the given task. Such methods are able to learn from examples which changes are important and which are not. In a way, everything could be defined as having been changed between two images taken at different times since nothing is exactly the same, but that would hardly be useful. Unsupervised methods decide by themselves what should be considered as a change, simply based on the appearance variance. But what is a change? While the answer to that question in isolation is best left to the philosophers, real applications of change detection methods usually have a concrete applications in mind. Should the algorithm look for new buildings that have been built? Or ones that have been destroyed? Or loss of ice surfaces due to polar melting? Or loss of forest areas due to fires or deforestation? Each of these applications can be modelled as change detection problems, and each of these events would lead to modified appearances in the imaged areas. This is the value of supervised methods, they can be taught to perform the task that the current application demands.

Two datasets were created during the development of this thesis. The first, the ONERA Satellite Change Detection dataset (OSCD), contains pairs of multispectral satellite images with manually annotated pixel-level change

maps. The second, the High Resolution Semantic Change Detection dataset (HRSCD), combined aerial VHR image pairs with open vector annotations of changes and land cover to obtain the largest openly available change detection dataset at the time of development. Both of these datasets opened the doors to new developments in change detection methods using CNNs. The datasets have been openly released to the scientific community to serve as a benchmark for various change detection algorithms.

Pixel level was initially studied as a patch classification problem, as well as a two class semantic segmentation problem in Chapter 3. This was, to the best of our knowledge, the first time CNNs were applied in an end-to-end fashion to the problem of change detection in such a framework. Previous CNN-based algorithms mostly focused on generating high dimensional pixel-level descriptors, whose difference could then be analysed using various techniques. Using the OSCD and an earlier dataset for our tests, the superiority of end-to-end methods over other supervised methods became clear. The experiments that were conducted also showed that the information that is present in the bands outside the visible spectrum can also help supervised change detection methods to achieve better performances.

The development of the HRSCD dataset allowed two main improvements to the methods presented in Chapter 3. First, the unprecedented scale of this change detection dataset allowed for deeper architectures to be trained with a smaller risk of overfitting. Second, the availability of land cover maps allowed the proposed change detection methods to not only detect, but also understand the changes in image pairs. In Chapter 4, several different approaches to harness this semantic information to perform semantic change detection were studied, and it became clear that the integrated multitask approach was superior to the other ones that were considered.

At this point, it became very clear that the considered data were far from ideal, and that simply proposing more advanced methods for change detection without considering such issues would not lead to concrete advancements that would fare well in realistic application cases. Data problems such as label noise, bias, imbalance and other issues had to be considered to produce robust CNN-based change detection algorithms.

To handle the problem of label noise and bias, the work presented in Chapter 5 presented an iterative training algorithm for the multitask network that had been presented in the previous chapter. This training procedure, when coupled with the novel guided anisotropic diffusion algorithm, allowed us to use the network predictions themselves and edge information from the input images to filter out noisy labels from the training dataset, allowing the network to produce much more precise predictions of changes. The guided anisotropic diffusion was also coupled with a class activation map technique and a novel spatial attention layer to perform weakly supervised semantic cosegmentation using only image-level change labels as supervision. This was, to the best of our knowledge, the first time image-level to pixel-level weakly supervised change detection was performed.

Chapter 6 presents an exploration of a novel domain adaptation method based on adversarial training for a domain invariant encoding of input images into a common latent space where a given task could be performed. This method was initially proposed with the aim of performing change detection, but the generality of the proposed

method was shown by applying it to classification and semantic segmentation tasks. In the context of change detection, this method allowed the change detection network to learn to detect destroyed buildings using images from one type of natural disaster, and to better generalise to other types of disasters. The range of natural disasters that were considered (wildfires, hurricanes, and tsunamis) illustrated the flexibility of the proposed method.

The exploration of CNN-based techniques for supervised changed detection presented in this thesis encountered several challenges, most of which centred around the properties of the considered data. While many of the recently proposed CNN-based computer vision methods can be somewhat straightforwardly extended to change detection, the peculiar data characteristics often make it quite challenging to actually achieve good results using realistic data. Nevertheless, end-to-end FCNs were successfully used to perform various change detection tasks.

Future Work

One of the main possible continuations for this work would be to extend the presented methods to deal with multi-temporal inputs, i.e. time series of coregistered images. While some work has already been done along those lines, notably in [PVV+19], the number of time-steps considered remains very low compared to what recurrent neural networks are capable of handling. Such networks are designed to identify long-range relations, which could be very useful when applied to detecting changes using sources with a high revisit rates such as the Sentinel-2 or Landsat satellites. One example would be to input time series of satellite images that have not been selected for good weather, and letting the network identify and interpret images automatically. This would be extremely useful for real time large scale monitoring of landmasses using data streams from Sentinel-2 satellites, for example. Currently, images are still manually selected to avoid images where atmospheric conditions interfere with image interpretation, but such manual selection does not lend itself well to large scale monitoring.

It would also be interesting to extend the methods presented in this thesis to handle multimodal data. Since one can not always have full control over the data that is available to study specific events, it is at times necessary to handle several types of data simultaneously. Data multimodality can be as simple as sensors with different ground sample distance, or as complex as comparing multispectral images to SAR images. Handling such multimodal cases could significantly increase the potential applications of such algorithms. The domain adaptation method presented in Chapter 6 is a good starting point for such approaches. The idea of representing images from different sources in a common latent space could be used to compare Earth observation images. To the best of my knowledge, this has not yet been done using adversarial training for multitemporal remote sensing image analysis.

Finally, the proposed change detection methods could be integrated into a large scale monitoring system to perform online monitoring of large areas. Such systems could be used for automatically updating maps, monitoring illegal deforestation, and many other applications. The integration of change detection into such a large scale data processing pipeline is a necessary step for a global automatic monitoring system, which would be made possible

by harnessing the full potential of Earth observation satellites. This would consist of bridging the gap between purely academic works and real world applications. Automatic monitoring of large areas is a task of interest in many different industries, but it is a challenging task that is hard to fully automate, which subsequently hinders the speed of the developed systems.

Appendix A

ONERA Satellite Change Detection Dataset

Images

The ONERA Satellite Change Detection (OSCD) dataset contains image pairs that were captured by the Sentinel-2 satellites[1]. The images were obtained by following two basic steps:

1. Create a *geojson* file that precisely defines the coordinates of the region of interest using a tool such as `geojson.io`[2]. Urban areas were chosen in several different continents with the aim of obtaining representative examples of different geographical areas.

2. Images were downloaded using the *sentinelsat* API[3] with help of the Medusa toolbox[4]. The first and last available images with high visibility (i.e. very small number of clouds) were downloaded and cropped using the coordinates from the *geojson* files.

Given that the dates of the development of this dataset and the launch of the Sentinel-2 satellites, the image pairs cover a temporal distance of 1–2 years. More information about the chosen regions can be found in Tabs. A.1 and A.2.

Sentinel-2 images contains different different bands inside and outside the visible spectrum [ESA20b]. Spectral bands come at three different resolution, ranging between 10 and 60 meters per pixel as described in Tab. A.3.

The different image resolutions across spectral bands require an additional step for considering the images simultaneously. Each band was cropped according to the chosen geographical coordinates contained in the *geojson* files, and then were resampled using the *zoom*[5] function of the *scipy* python library[6]. Both the original and resampled

[1] `https://sentinel.esa.int/web/sentinel/missions/sentinel-2`
[2] `https://geojson.io/`
[3] `https://sentinelsat.readthedocs.io/en/stable/api.html`
[4] `https://github.com/aboulch/medusa_tb`
[5] `https://docs.scipy.org/doc/scipy/reference/generated/scipy.ndimage.zoom.html`
[6] `https://docs.scipy.org/doc/scipy/reference/index.html`

Split	Name	Northwest corner	Southeast corner	Size at 10 m/px	Num. of pixels
Train	abudhabi	(24.3664,54.5402)	(24.2914,54.6140)	785x799	627215
	aguasclaras	(-15.8163,-48.0498)	(-15.8596,-48.0015)	525x471	247275
	beihai	(21.6159,109.4839)	(21.5316,109.5560)	772x902	696344
	beirut	(33.9181,35.4618)	(33.8014,35.5677)	1070x1180	1262600
	bercy	(48.8666,2.3593)	(48.8206,2.4011)	360x395	142200
	bordeaux	(44.8668,-0.6009)	(44.8105,-0.5508)	461x517	238337
	cupertino	(37.4012,-122.0537)	(37.2976,-121.9733)	788x1015	799820
	hongkong	(22.3413,114.2281)	(22.2763,114.2785)	540x695	375300
	mumbai	(19.1052,72.9016)	(19.0256,72.9533)	557x858	477906
	nantes	(47.2341,-1.5773)	(47.1755,-1.5121)	582x522	303804
	paris	(48.8450,2.3101)	(48.7975,2.3555)	390x408	159120
	pisa	(43.7583,10.3630)	(43.6748,10.4402)	718x776	557168
	rennes	(48.1325,-1.6851)	(48.0937,-1.6209)	563x339	190857
	saclay_e	(48.7402,2.2174)	(48.6668,2.2964)	688x639	439632
Test	brasilia	(-15.7267,-47.9076)	(-15.7664,-47.8645)	469x433	203077
	chongqing	(29.4497,106.2697)	(29.3791,106.3222)	544x730	397120
	dubai	(25.0737,55.1937)	(25.0003,55.2538)	634x774	490716
	lasvegas	(36.0654,-115.2723)	(35.9827,-115.2004)	716x824	589984
	milano	(45.5330,9.1449)	(45.4728,9.2065)	558x545	304110
	montpellier	(43.6238,3.879)	(43.5777,3.9277)	451x426	192126
	norcia	(42.8068,13.0720)	(42.7807,13.1137)	385x241	92785
	rio	(-22.9514,-43.4033)	(-22.9846,-43.3632)	426x353	150378
	saclay_w	(48.7402,2.1383)	(48.6668,2.2174)	688x639	439632
	valencia	(39.5287,-0.4460)	(39.4816,-0.3971)	476x458	218008
	Total	—	—	—	9595514

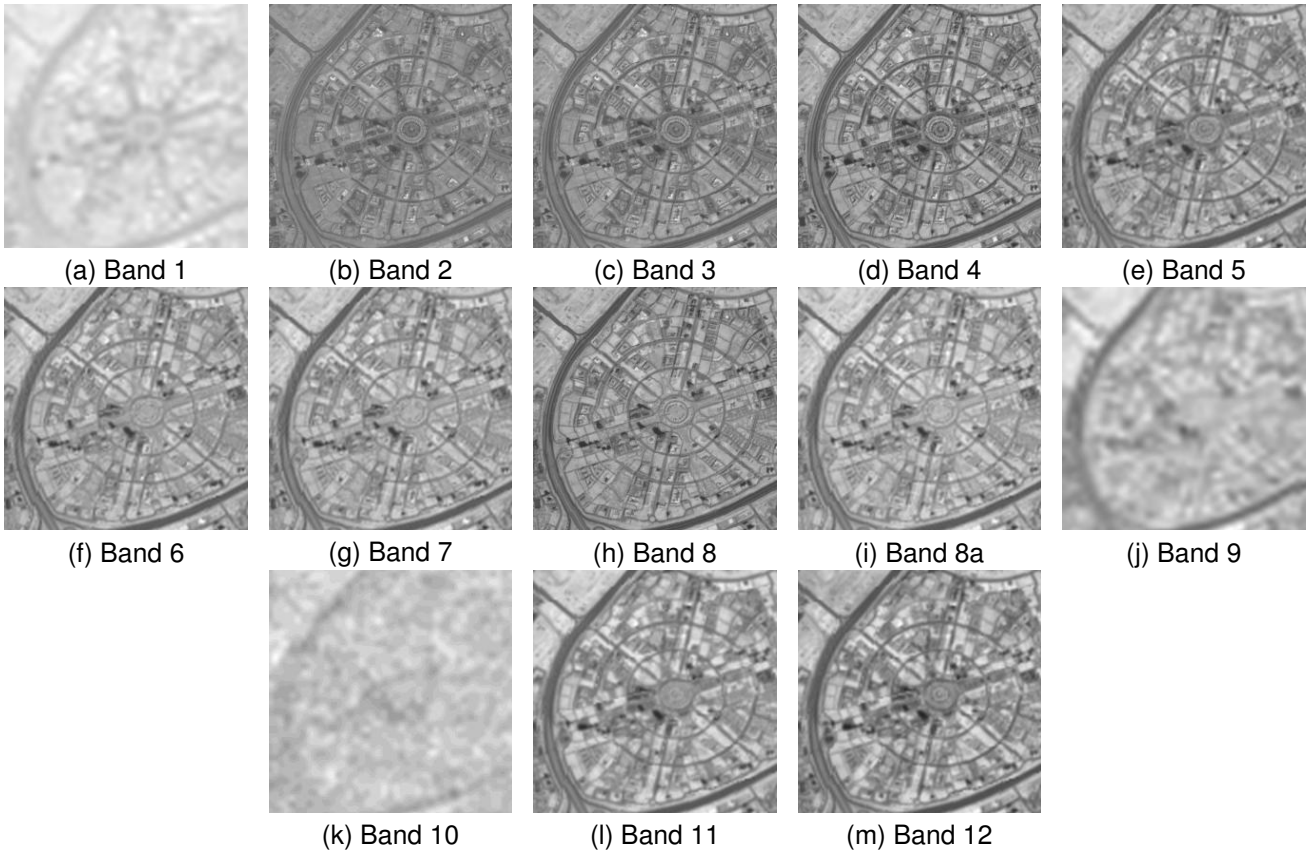
Table A.1: Locations and sizes of the images in the OSCD dataset.

Split	Name	Date - Image 1	Date - Image 2
Train	abudhabi	20/01/2016	28/03/2018
	aguasclaras	16/09/2015	15/10/2017
	beihai	09/12/2016	09/03/2018
	beirut	20/08/2015	03/10/2017
	bercy	30/11/2016	29/08/2017
	bordeaux	04/05/2016	26/10/2017
	cupertino	18/09/2015	26/03/2018
	hongkong	27/09/2016	23/03/2018
	mumbai	30/11/2015	19/03/2018
	nantes	21/08/2015	14/10/2017
	paris	30/11/2016	07/11/2017
	pisa	04/07/2015	11/02/2018
	rennes	21/08/2015	21/06/2017
	saclay_e	15/03/2016	29/08/2017
Test	brasilia	16/09/2015	17/10/2017
	chongqing	14/04/2017	02/04/2018
	dubai	11/12/2015	30/03/2018
	lasvegas	20/08/2015	05/02/2018
	milano	28/12/2016	22/01/2018
	montpellier	12/08/2015	30/10/2017
	norcia	11/07/2015	18/10/2017
	rio	24/04/2016	11/10/2017
	saclay_w	15/03/2016	29/08/2017
	valencia	30/07/2016	07/11/2017

Table A.2: Dates of acquisition of the images in the OSCD dataset.

Band	Central wavelength (nm)	Resolution (m/px)
1	443	60
2	490 (blue)	10
3	560 (green)	10
4	665 (red)	10
5	705	20
6	740	20
7	783	20
8	842	10
8b	865	20
9	945	60
10	1375	60
11	1610	20
12	2190	20

Table A.3: Sentinel-2 bands and resolutions.



(a) Band 1 (b) Band 2 (c) Band 3 (d) Band 4 (e) Band 5

(f) Band 6 (g) Band 7 (h) Band 8 (i) Band 8a (j) Band 9

(k) Band 10 (l) Band 11 (m) Band 12

Figure A.1: Example of Sentinel-2 image bands. Differences in spatial resolutions can be perceived as a variance in sharpness in the images above.

versions of each image are available in the dataset in case the users decide to handle the differences in resolution in other ways.

Labels

The labels in the OSCD were created using GNU Image Manipulation Program (GIMP)[7]. Binary change maps were created by comparing each pair of coregistered true colour images, and marking the observed changes manually as closely as possible to the boundaries in the original images.

It is not always clear to clearly separate changed from unchanged reasons, especially at a resolution of 10 m/px, since reality is not binary. The effects of this ambiguity are illustrated in Fig A.2, where even human analysts disagree about some of the marked changes. At the edge cases, where it is not clear whether a region should or not be marked as changed, a criterion was used to better standardize the creation of such maps. Such regions were marked as changed if it looked like human activities had caused the perceived changes. Other natural changes, such as seasonal variation, was largely ignored.

[7] `https://www.gimp.org/`

(a) Image 1 (b) Image 2 (c) Change maps

Figure A.2: Pair of coregistered images from the OSCD and manual change maps created by three different people. In (c), each colour channel contains the change map produced by a different person.

Binary pixel-level change maps are available for all training and test images at a resolution of 10 m/px.

Size

The detailed size of each region in the dataset at 10 m/px is described in Tab. A.1. The total size of about 9.6 million annotated pixel is larger than previous open datasets, including the Air Change dataset by Benedek et al. [BS09]. The complete downloadable version of the dataset amounts to 512.9 MB of data. It occupies 756.7 MB of disk space once uncompressed.

Class Distribution

Table A.4 contains the percentage of changes in each change map in the OSCD dataset. The distribution of changes is not homogeneous across images. Approximately 3.21% of the pixels in the dataset are marked as change, which consists of a 1:30 class imbalance. This should be taken into account when using the dataset.

Contents

The dataset files contain:

1. Information about the suggested train/test split.
2. Information about the acquisition dates of each image.
3. *Geojson* files for each considered geographical region.
4. All image bands in TIFF format for all images in the dataset at the original resolution cropped according to the *geojson* files.

Split	Name	Change percentage
Train	abudhabi	3.76%
	aguasclaras	1.64%
	beihai	2.49%
	beirut	2.69%
	bercy	0.74%
	bordeaux	1.00%
	cupertino	2.37%
	hongkong	3.56%
	mumbai	2.56%
	nantes	1.14%
	paris	0.29%
	pisa	1.64%
	rennes	2.58%
	saclay_e	0.99%
Test	brasilia	2.58%
	chongqing	7.21%
	dubai	9.92%
	lasvegas	7.67%
	milano	0.80%
	montpellier	6.79%
	norcia	1.32%
	rio	5.69%
	saclay_w	1.14%
	valencia	0.44%
	Total	3.21%

Table A.4: Percentage of changes in each change map in the OSCD dataset.

5. All image bands in TIFF format for all images in the dataset resampled at 10 m/px for ease of use.

6. All change maps in TIFF and PNG formats.

7. True colour images for all the acquisitions in the dataset, generated from the red, green, and blue bands.

8. A README containing copyright and general information about the dataset.

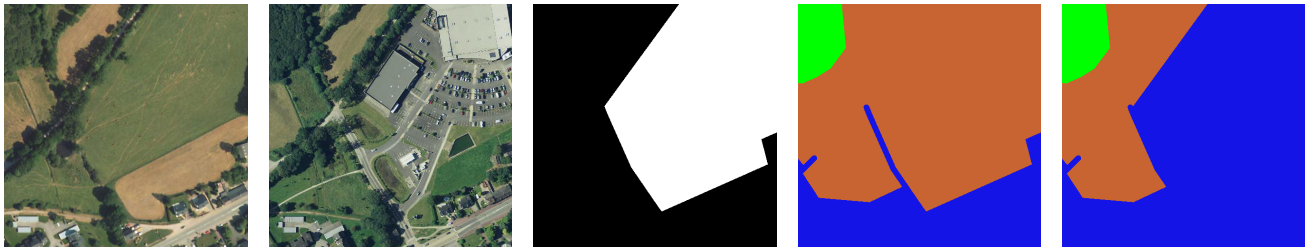
Access and Distribution

The dataset files can be downloaded from the official IEEE DataPort page[8] or from the mirrors found on the dataset page on my personal website[9]. The dataset is released under a Creative-Commons BY-NC-SA licence[10].

[8] https://ieee-dataport.org/open-access/oscd-onera-satellite-change-detection
[9] https://rcdaudt.github.io/oscd
[10] https://creativecommons.org/licenses/by-nc-sa/4.0/legalcode

Appendix B

High Resolution Semantic Change Detection Dataset



(a) Image 1 (b) Image 2 (c) Change map (d) LCM 1 (e) LCM 2

Figure B.1: Example of data that can be found in the HRSCD dataset. Coregistered aerial image pairs, change maps, and land cover maps contain semantic information about how the terrain has evolved over a period of six years.

Images

The images in the High Resolution Semantic Change Detection (HRSCD) dataset come from the *BD ORTHO*[1] database, which is provided by the French Institut National de l'Information Géographique et Forestière (IGN)[2]. This database contains orthorectified RGB mosaics created using aerial imagery of urban areas in France at a resolution of 50 cm/px split into crops of size 10000x10000 pixels. Some of the images in the *BD ORTHO* database are under an open licence (*licence ouverte*), which means they are openly downloadable and redistributable under certain conditions. Others can be accessed (but not redistributed) at no cost for research purposes if a request is made.

For reasons that will become clear in the next section, only regions for which images were available around the

[1] https://www.data.gouv.fr/en/datasets/bd-ortho-r-50-cm/
[2] http://www.ign.fr/

years of 2006 and 2012 were selected. This restricted the selection to only two regions:

1. **14 - Calvados**: region around the city of Caen.
2. **35 - Ille-et-Vilaine**: region around the city of Rennes.

The selection of images was further narrowed down following two steps:

1. Only the regions for which images were available both around 2006 and 2012 were kept (the imaged regions were not identical in the two years).
2. The regions for which no labels were available were removed from the dataset.

The final size of the dataset is described in Tab. B.1. The HRSCD dataset contains over 3000 times the number of pixels as the OSCD dataset that was presented in Appendix A.

Region	Number of images	Total number of pixels	Total surface area (km^2)
D14 - Calvados	101	$1.01 \cdot 10^{10}$	2525
D35 - Ille-et-Vilaine	190	$1.9 \cdot 10^{10}$	4750
Total	291	$2.91 \cdot 10^{10}$	7275

Table B.1: Size of the HRSCD dataset.

Labels and Class Distribution

The labels in the HRSCD dataset were generated by sampling the Urban Atlas Change 2006-2012 database[3]. This database contains vector data for changes that occurred in all major urban areas in Europe between 2006 and 2012. The annotations come in the form of polygon coordinates, coupled with ”from” and ”to” information to define what type of change has occurred in that location. These vector data were rasterized using the *gdal* library[4].

To limit the complexity of data and reduce the number of considered classes, several classes were grouped together according to the hierarchical breakdown presented in [Cop20]. This leaves us with 5 distinct classes (as well as the ”no information” class for pixels not covered by any polygons), as can be seen in Tab. B.2.

The land cover maps in the HRSCD dataset were generated in a similar way to the change labels using the Urban Atlas 2006[5] and Urban Atlas 2012[6] land cover mapping vector data. The class distribution for different sections of the dataset can be found in Tab. B.2.

The information contained in the generated land cover maps includes the semantic information in the Urban Atlas Change 2006-2012 data. Therefore, change maps were generated as binary maps, and semantic information

[3] https://land.copernicus.eu/local/urban-atlas/change-2006-2009
[4] https://gdal.org/
[5] https://land.copernicus.eu/local/urban-atlas/urban-atlas-2006
[6] https://land.copernicus.eu/local/urban-atlas/urban-atlas-2012

Code	Class	Incidence in 2006			Incidence in 2012			Total
		D14	D35	Combined	D14	D35	Combined	
0	No information	13.55%	19.91%	17.70%	13.52%	19.90%	17.69%	17.70%
1	Artificial surfaces	11.74%	10.88%	11.18%	12.02%	11.74%	11.84%	11.51%
2	Agricultural areas	67.79%	59.26%	62.22%	67.25%	58.37%	61.45%	61.83%
3	Forests	6.56%	9.31%	8.35%	6.59%	9.32%	8.38%	8.36%
4	Wetlands	0.00%	0.00%	0.00%	0.10%	0.01%	0.04%	0.02%
5	Water	0.36%	0.64%	0.54%	0.53%	0.66%	0.61%	0.58%

Table B.2: Urban Atlas land cover mapping classes at hierarchical level L1, extracted from [Cop20], and their frequency in the generated land cover maps.

can be extracted from the land cover maps. This approach allows for full flexibility for the users of the dataset with no loss of information.

The class imbalance in the dataset is described in Tab. B.3. The unchanged pixels vastly outnumber the changed pixels. The imbalance is especially strong when we consider all types of transitions between the 5 considered classes.

	1	2	3	4	5
1	—	0.011%	0%	0.001%	0.001%
2	0.653%	—	0.001%	0%	0.077%
3	0.014%	0.002%	—	0%	0%
4	0%	0%	0%	—	0%
5	0.001%	0.004%	0%	0.004%	—
No change			99.232%		

Table B.3: Change class imbalance. Row number represents class in 2006, column number represents class in 2012.

Size

The download size of the HRSCD dataset is 9.2 GB. The dataset files occupy 12.5 GB of disk space when uncompressed. More information about the size of the dataset can be found in Tab. B.1.

Contents

The dataset files contain:

1. 291 RGB aerial image pairs of size 10000x10000 pixels taken in the regions of Calvados and Ille-et-Vilaine in the GeoTIFF format.
2. Binary change maps for all image pairs in the dataset in the GeoTIFF format.
3. Land cover maps for all images in the dataset in the GeoTIFF format.

4. A README containing copyright and general information about the dataset.

Access and Distribution

The dataset files, except for the 2006 images, can be downloaded from the official IEEE DataPort page[7] or from the mirrors found on the dataset page on my personal website[8]. The 2006 images can be downloaded using IGN's data portal[9]. The dataset is released under a Creative-Commons BY-NC-SA licence[10].

[7] https://ieee-dataport.org/open-access/hrscd-high-resolution-semantic-change-detection-dataset
[8] https://rcdaudt.github.io/hrscd
[9] http://professionnels.ign.fr/donnees
[10] https://creativecommons.org/licenses/by-nc-sa/4.0/legalcode

Bibliography

[ABC+16] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th Symposium on Operating Systems Design and Implementation*, pages 265–283, 2016.

[ACB17] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan, 2017.

[AK06] Gilles Aubert and Pierre Kornprobst. *Mathematical problems in image processing: partial differential equations and the calculus of variations*, volume 147. Springer Science & Business Media, 2006.

[AK18] Jiwoon Ahn and Suha Kwak. Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4981–4990, 2018.

[Alb18] Vinicius Ferraris Pignataro Mazzei Albert. *Fusion-based change detection for remote sensing images of different resolutions and modalities*. Theses, Institut National Polytechnique de Toulouse, October 2018.

[ALL16] Nicolas Audebert, Bertrand Le Saux, and Sébastien Lefèvre. Semantic segmentation of earth observation data using multimodal and multi-scale deep networks. In *Asian Conference on Computer Vision*, pages 180–196, 2016.

[ALSL17a] Nicolas Audebert, Bertrand Le Saux, and Sébastien Lefèvre. Beyond rgb: Very high resolution urban remote sensing with multimodal deep networks. *ISPRS Journal of Photogrammetry and Remote Sensing*, 2017.

[ALSL17b] Nicolas Audebert, Bertrand Le Saux, and Sébastien Lefèvre. Segment-before-detect: Vehicle detection and classification through semantic segmentation of aerial images. *Remote Sensing*, 9(4):368, 2017.

[ARD20] Jose Luis Holgado Alvarez, Mahdyar Ravanbakhsh, and Begüm Demir. S2-cgan: Self-supervised adversarial representation learning for binary change detection in multispectral images, 2020.

[Aud18] Nicolas Audebert. *Machine learning for classification of big remote sensing data*. Theses, Université de Bretagne Sud, October 2018.

[BB05] Francesca Bovolo and Lorenzo Bruzzone. A wavelet-based change-detection technique for multitemporal sar images. In *International Workshop on the Analysis of Multi-Temporal Remote Sensing Images*, pages 85–89. IEEE, 2005.

[BB07] Francesca Bovolo and Lorenzo Bruzzone. A theoretical framework for unsupervised change detection based on change vector analysis in the polar domain. *IEEE Transactions on Geoscience and Remote Sensing*, 45(1):218–236, 2007.

[BB13] Lorenzo Bruzzone and Francesca Bovolo. A novel framework for the design of change-detection systems for very-high-resolution remote sensing images. *Proceedings of the IEEE*, 101(3):609–630, 2013.

[BBM05] Yakoub Bazi, Lorenzo Bruzzone, and Farid Melgani. An unsupervised approach based on the generalized gaussian model to automatic change detection in multitemporal sar images. *IEEE Transactions on Geoscience and Remote Sensing*, 43(4):874–887, 2005.

[BCV13] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013.

[BDS11] Nicolas Bourdis, Marraud Denis, and Hichem Sahbi. Constrained optical flow for aerial image change detection. In *International Geoscience and Remote Sensing Symposium*, pages 4176–4179, 2011.

[BGL+94] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Signature verification using a ”siamese” time delay neural network. In *Advances in Neural Information Processing Systems*, pages 737–744, 1994.

[Bis06] Christopher M Bishop. *Pattern recognition and machine learning*. Springer, 2006.

[BKC17] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12):2481–2495, 2017.

[BLPL07] Yoshua Bengio, Pascal Lamblin, Dan Popovici, and Hugo Larochelle. Greedy layer-wise training of deep networks. In *Advances in neural information processing systems*, pages 153–160, 2007.

[BP00] Lorenzo Bruzzone and Diego F Prieto. Automatic analysis of the difference image for unsupervised change detection. *IEEE Transactions on Geoscience and Remote sensing*, 38(3):1171–1182, 2000.

[BS09] Csaba Benedek and Tamás Szirányi. Change detection in optical aerial images by a multilayer conditional mixed markov model. *IEEE Transactions on Geoscience and Remote Sensing*, 47(10):3416–3430, 2009.

[BSBB19] Luca Bergamasco, Sudipan Saha, Francesca Bovolo, and Lorenzo Bruzzone. Unsupervised change-detection based on convolutional-autoencoder feature extraction. In Lorenzo Bruzzone and Francesca Bovolo, editors, *Image and Signal Processing for Remote Sensing XXV*, volume 11155, pages 325 – 332. International Society for Optics and Photonics, SPIE, 2019.

[BTVG06] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In Aleš Leonardis, Horst Bischof, and Axel Pinz, editors, *Computer Vision – ECCV 2006*, pages 404–417, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[BVH+16] Luca Bertinetto, Jack Valmadre, Joao F Henriques, Andrea Vedaldi, and Philip HS Torr. Fully-convolutional siamese networks for object tracking. In *European conference on computer vision*, pages 850–865. Springer, 2016.

[Can86] John Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698, 1986.

[Car93] Richard Caruana. Multitask learning: A knowledge-based source of inductive bias. In *Proceedings of the Tenth International Conference on Machine Learning*, pages 41–48. Morgan Kaufmann, 1993.

[Cel09] Turgay Celik. Unsupervised change detection in satellite images using principal component analysis and k-means clustering. *IEEE Geoscience and Remote Sensing Letters*, 6(4):772–776, 2009.

[Che12] Chi-hau Chen. *Signal and image processing for remote sensing*. CRC press, 2012.

[CHL05a] Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2005.

[CHL05b] Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 539–546. IEEE, 2005.

[CJN+04] Pol Coppin, Inge Jonckheere, Kristiaan Nackaerts, Bart Muys, and Eric Lambin. Digital change detection methods in ecosystem monitoring: A review. *International Journal of Remote Sensing*, 25(9):1565–1596, 2004.

[CM02] D. Comaniciu and P. Meer. Mean shift: a robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619, 2002.

[CMM+11] Dan Claudiu Ciresan, Ueli Meier, Jonathan Masci, Luca Maria Gambardella, and Jürgen Schmidhuber. Flexible, high performance convolutional neural networks for image classification. In *Twenty-Second International Joint Conference on Artificial Intelligence*, 2011.

[COA18] Ying Chen, Xu Ouyang, and Gady Agam. MFCNET: End-to-end approach for change detection in images. In *2018 25th IEEE International Conference on Image Processing*, pages 4008–4012. IEEE, 2018.

[Cop20] Copernicus. Mapping guide for a european urban atlas. `https://land.copernicus.eu/user-corner/technical-library/urban_atlas_2012_2018_mapping_guide_v6-1.pdf`, 2020. Accessed: 2020-08-20.

[COR+16] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[CPK+16] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP, 06 2016.

[Csu17] Gabriela Csurka. Domain adaptation for visual applications: A comprehensive survey. In *Advances in Computer Vision and Pattern Recognition*, pages 1–35. Springer, 2017.

[CW11] James B Campbell and Randolph H Wynne. *Introduction to remote sensing*. Guilford Press, 2011.

[CWDZ19] H. Chen, C. Wu, B. Du, and L. Zhang. Deep siamese multi-scale convolutional network for change detection in multi-temporal vhr images. In *2019 10th International Workshop on the Analysis of Multitemporal Remote Sensing Images (MultiTemp)*, pages 1–4, 2019.

[CWS+18] Kaiqiang Chen, Michael Weinmann, Xian Sun, Menglong Yan, Stefan Hinz, Boris Jutzi, and Martin Weinmann. Semantic segmentation of aerial imagery via multi-scale shuffling convolutional neural networks with deep supervision. *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, 4(1), 2018.

[Cyb89] George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

[CZP+18] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[DBB13] B. Demir, F. Bovolo, and L. Bruzzone. Updating land-cover maps by classification of image time series: A novel change-detection-driven transfer learning approach. *IEEE Transactions on Geoscience and Remote Sensing*, 51(1):300–312, 2013.

[DFI+15] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[DHS15] Jifeng Dai, Kaiming He, and Jian Sun. Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1635–1643, 2015.

[Dic45] Lee R Dice. Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302, 1945.

[DK99] XL Dai and Slamak Khorram. Remotely sensed change detection based on artificial neural networks. *Photogrammetric engineering and remote sensing*, 65:1187–1194, 1999.

[DKL+18] Ilke Demir, Krzysztof Koperski, David Lindenbaum, Guan Pang, Jing Huang, Saikat Basu, Forest Hughes, Devis Tuia, and Ramesh Raskar. Deepglobe 2018: A challenge to parse the earth through satellite images. *CoRR*, abs/1805.06561, 2018.

[DLB18] Rodrigo Caye Daudt, Bertrand Le Saux, and Alexandre Boulch. Fully convolutional siamese networks for change detection. In *2018 25th IEEE International Conference on Image Processing*, pages 4063–4067, October 2018.

[DLBG19] Rodrigo Caye Daudt, Bertrand Le Saux, Alexandre Boulch, and Yann Gousseau. Multitask learning for large-scale semantic change detection. *Computer Vision and Image Understanding*, 187:102783, 2019.

[DT05] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 886–893 vol. 1, 2005.

[EALW16] Arabi Mohammed El Amin, Qingjie Liu, and Yunhong Wang. Convolutional neural network features based change detection in satellite images. In *First International Workshop on Pattern Recognition*, page 100110W. International Society for Optics and Photonics, 2016.

[EALW17] Arabi Mohammed El Amin, Qingjie Liu, and Yunhong Wang. Zoom out CNNs features for optical remote sensing change detection. In *Image, Vision and Computing (ICIVC), 2017 2nd International Conference on*, pages 812–817. IEEE, 2017.

[EG20] Richard Evans and Jim Gao. Deepmind ai reduces google data centre cooling bill by 40%. `https://deepmind.com/blog/article/deepmind-ai-reduces-google-data-centre-cooling-bill-40`, 2020. Accessed: 2020-07-24.

[EKN+17] Andre Esteva, Brett Kuprel, Roberto A. Novoa, Justin Ko, Susan M. Swetter, Helen M. Blau, and Sebastian Thrun. Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542(7639):115–118, Feb 2017.

[ESA20a] ESA. Sentinel-2 operations. `https://www.esa.int/Enabling_Support/Operations/Sentinel-2_operations`, 2020. Accessed: 2020-07-22.

[ESA20b] ESA. Sentinel-2 user handbook. `https://sentinel.esa.int/documents/247904/685211/Sentinel-2_User_Handbook`, 2020. Accessed: 2020-08-30.

[EVGW+12] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html, 2012.

[EVZ06] Charles Elachi and Jakob J Van Zyl. *Introduction to the physics and techniques of remote sensing*, volume 28. John Wiley & Sons, 2006.

[FK+14] Benoît Frénay, Ata Kabán, et al. A comprehensive introduction to label noise. In *European Symposium on Artificial Neural Networks*, 2014.

[FM82] Kunihiko Fukushima and Sei Miyake. Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. In *Competition and cooperation in neural nets*, pages 267–285. Springer, 1982.

[FPK19] Bo Fang, Li Pan, and Rong Kou. Dual learning-based siamese framework for change detection using bi-temporal vhr optical remote sensing images. *Remote Sensing*, 11(11):1292, May 2019.

[FRR+13] David Ferstl, Christian Reinbacher, Rene Ranftl, Matthias Rüther, and Horst Bischof. Image guided depth upsampling using anisotropic total generalized variation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 993–1000, 2013.

[FSA99] Yoav Freund, Robert Schapire, and Naoki Abe. A short introduction to boosting. *Journal-Japanese Society For Artificial Intelligence*, 14(771-780):1612, 1999.

[FSI+17] A. Fujita, K. Sakurada, T. Imaizumi, R. Ito, S. Hikosaka, and R. Nakamura. Damage detection from aerial images via convolutional neural networks. In *2017 Fifteenth IAPR International Conference on Machine Vision Applications (MVA)*, pages 5–8, May 2017.

[FSR+20] André C. Ferreira, Liliana R. Silva, Francesco Renna, Hanja B. Brandl, Julien P. Renoult, Damien R. Farine, Rita Covas, and Claire Doutrelant. Deep learning-based methods for individual recognition in small birds. *Methods in Ecology and Evolution*, 2020.

[FV14] Benoît Frénay and Michel Verleysen. Classification in the presence of label noise: a survey. *IEEE transactions on neural networks and learning systems*, 25(5):845–869, 2014.

[Gao14] Jim Gao. Machine learning applications for data center optimization. *Technical Report*, 2014.

[GBC16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[GGP+19] Ritwik Gupta, Bryce Goodman, Nirav Patel, Ricky Hosfelt, Sandra Sajeev, Eric Heim, Jigar Doshi, Keane Lucas, Howie Choset, and Matthew Gaston. Creating xbd: A dataset for assessing building damage from satellite imagery. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.

[GKZ+16] Muhammad Ghifary, W Bastiaan Kleijn, Mengjie Zhang, David Balduzzi, and Wen Li. Deep reconstruction-classification networks for unsupervised domain adaptation. In *European Conference on Computer Vision*, pages 597–613. Springer, 2016.

[GL15] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1180–1189, Lille, France, 07–09 Jul 2015. PMLR.

[GMV+96] Isabelle Guyon, Nada Matic, Vladimir Vapnik, et al. Discovering informative patterns and data cleaning. In *Association for the Advancement of Artificial Intelligence*, 1996.

[Göd31] Kurt Gödel. Über formal unentscheidbare sätze der principia mathematica und verwandter systeme i. *Monatshefte für mathematik und physik*, 38(1):173–198, 1931.

[GPAM+14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014.

[GUA+16] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *J. Mach. Learn. Res.*, 17(1):2096–2030, January 2016.

[GW96] Sucharita Gopal and Curtis Woodcock. Remote sensing of forest change using artificial neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, 34(2):398–404, 1996.

[GZL+16] Maoguo Gong, Jiaojiao Zhao, Jia Liu, Qiguang Miao, and Licheng Jiao. Change detection in synthetic aperture radar images based on deep neural networks. *IEEE transactions on neural networks and learning systems*, 27(1):125–138, 2016.

[HCC+13] Masroor Hussain, Dongmei Chen, Angela Cheng, Hui Wei, and David Stanley. Change detection from remotely sensed images: From pixel-based to object-based approaches. *ISPRS Journal of Photogrammetry and Remote Sensing*, 80:91–106, 2013.

[HLvdMW17] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[HMZ19] Yuansheng Hua, Lichao Mou, and Xiao Xiang Zhu. Recurrently exploring class-wise attention in a hybrid convolutional and bidirectional lstm network for multi-label aerial image classification. *ISPRS Journal of Photogrammetry and Remote Sensing*, 149:188 – 199, 2019.

[Hor91] Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural networks*, 4(2):251–257, 1991.

[HOT06] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.

[HSK+08] Chengquan Huang, Kuan Song, Sunghee Kim, John RG Townshend, Paul Davis, Jeffrey G Masek, and Samuel N Goward. Use of a dark object concept and support vector machines to automate forest cover change analysis. *Remote Sensing of Environment*, 112(3):970–985, 2008.

[HST13] Kaiming He, Jian Sun, and Xiaoou Tang. Guided image filtering. *IEEE transactions on pattern analysis and machine intelligence*, 35(6):1397–1409, 2013.

[HTP+18] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. CyCADA: Cycle-consistent adversarial domain adaptation. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1989–1998, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR.

[Hul94] Jonathan J. Hull. A database for handwritten text recognition research. *IEEE Transactions on pattern analysis and machine intelligence*, 16(5):550–554, 1994.

[HWYD16] Judy Hoffman, Dequan Wang, Fisher Yu, and Trevor Darrell. Fcns in the wild: Pixel-level adversarial and constraint-based adaptation. *CoRR*, abs/1612.02649, 2016.

[HZRS14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 346–361, Cham, 2014. Springer International Publishing.

[HZRS15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[HZRS16a] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[HZRS16b] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[IS15] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, page 448–456. JMLR.org, 2015.

[IZZE17] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[JHL+20] Huiwei Jiang, Xiangyun Hu, Kun Li, Jinming Zhang, Jinqi Gong, and Mi Zhang. Pga-siamnet: Pyramid feature-based attention-guided siamese network for remote sensing orthoimagery building change detection. *Remote Sensing*, 12(3):484, Feb 2020.

[JM00] Jianbo Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[Joh95] George H John. Robust decision trees: Removing outliers from databases. In *KDD*, pages 174–179, 1995.

[JWF10] Piyasak Jeatrakul, Kok Wai Wong, and Chun Che Fung. Data cleaning for classification using misclassification analysis. *Journal of Advanced Computational Intelligence and Intelligent Informatics*, 14(3):297–302, 2010.

[KB14] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[KBH+17] Anna Khoreva, Rodrigo Benenson, Jan Hosang, Matthias Hein, and Bernt Schiele. Simple does it: Weakly supervised instance and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 876–885, 2017.

[KBO+16] Anna Khoreva, Rodrigo Benenson, Mohamed Omran, Matthias Hein, and Bernt Schiele. Weakly supervised object boundaries. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[KCLU07] Johannes Kopf, Michael F Cohen, Dani Lischinski, and Matt Uyttendaele. Joint bilateral upsampling. In *ACM Transactions on Graphics*, volume 26, page 96. ACM, 2007.

[KK01] J. F. Kolen and S. C. Kremer. *Gradient Flow in Recurrent Nets: The Difficulty of Learning Long Term Dependencies*, pages 237–243. Wiley-IEEE Press, 2001.

[KK11] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *Advances in neural information processing systems*, pages 109–117, 2011.

[KLA19] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[KMAB19] Maria Kolos, Anton Marin, Alexey Artemov, and Evgeny Burnaev. Procedural synthesis of remote sensing images for robust change detection with neural networks. In Huchuan Lu, Huajin Tang, and Zhanshan Wang, editors, *Advances in Neural Networks – ISNN 2019*, pages 371–387, Cham, 2019. Springer International Publishing.

[Koe84] Jan J Koenderink. The structure of images. *Biological cybernetics*, 50(5):363–370, 1984.

[KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[KWT88] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, Jan 1988.

[KZYY18] Guoliang Kang, Liang Zheng, Yan Yan, and Yi Yang. Deep adversarial attention alignment for unsupervised domain adaptation: the benefit of target expectation maximization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[Law20] Wallace Boone Law. I made bushfire maps from satellite data, and found a glaring gap in australia's preparedness. `https://theconversation.com/i-made-bushfire-maps-from-satellite-data-and-found-a-glaring-gap-in-australias-preparedness-` 2020. Accessed: 2020-07-24.

[LBBH98] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[LBH15] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436, 2015.

[LCCV16] Pauline Luc, Camille Couprie, Soumith Chintala, and Jakob Verbeek. Semantic segmentation using adversarial networks, 2016.

[LCWJ15] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I. Jordan. Learning transferable features with deep adaptation networks. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, page 97–105. JMLR.org, 2015.

[LDJ+16] Di Lin, Jifeng Dai, Jiaya Jia, Kaiming He, and Jian Sun. Scribblesup: Scribble-supervised convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[LFX+17] Zhiwu Lu, Zhenyong Fu, Tao Xiang, Peng Han, Liwei Wang, and Xin Gao. Learning from weak and noisy labels for semantic segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(3):486–500, 2017.

[LGQZ16] Jia Liu, Maoguo Gong, Kai Qin, and Puzhao Zhang. A deep convolutional coupling network for change detection based on heterogeneous optical and radar images. *IEEE transactions on neural networks and learning systems*, 2016.

[LGQZ18] J. Liu, M. Gong, K. Qin, and P. Zhang. A deep convolutional coupling network for change detection based on heterogeneous optical and radar images. *IEEE Transactions on Neural Networks and Learning Systems*, 29(3):545–559, 2018.

[LGT19a] G. Liu, Y. Gousseau, and F. Tupin. A contrario comparison of local descriptors for change detection in very high spatial resolution satellite images of urban areas. *IEEE Transactions on Geoscience and Remote Sensing*, 57(6):3904–3918, 2019.

[LGT19b] Gang Liu, Yann Gousseau, and Florence Tupin. A contrario comparison of local descriptors for change detection in very high spatial resolution satellite images of urban areas. *IEEE Transactions on Geoscience and Remote Sensing*, 2019.

[LHK+20] Luigi T. Luppino, Mads A. Hansen, Michael Kampffmeyer, Filippo M. Bianchi, Gabriele Moser, Robert Jenssen, and Stian N. Anfinsen. Code-aligned autoencoders for unsupervised change detection in multimodal remote sensing images, 2020.

[LKB+20] Luigi Tommaso Luppino, Michael Kampffmeyer, Filippo Maria Bianchi, Gabriele Moser, Sebastiano Bruno Serpico, Robert Jenssen, and Stian Normann Anfinsen. Deep image translation with an affinity-based change prior for unsupervised multimodal change detection, 2020.

[Low99] D. G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, pages 1150–1157 vol.2, 1999.

[Low04] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, Nov 2004.

[LS94] Eric F Lambin and Alan H Strahlers. Change-vector analysis in multitemporal space: a tool to detect and categorize land-cover change processes using high temporal-resolution satellite data. *Remote Sensing of Environment*, 48(2):231–244, 1994.

[LSD15a] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[LSD15b] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[LSR13] Bertrand Le Saux and Hicham Randrianarivo. Urban change detection in sar images by interactive learning. In *Geoscience and Remote Sensing Symposium (IGARSS), 2013 IEEE International*, pages 3990–3993. IEEE, 2013.

[LT16] Ming-Yu Liu and Oncel Tuzel. Coupled generative adversarial networks. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 469–477. Curran Associates, Inc., 2016.

[LWZK19] Junnan Li, Yongkang Wong, Qi Zhao, and Mohan S. Kankanhalli. Learning to learn from noisy labeled data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[LZW+19] Xia Li, Zhisheng Zhong, Jianlong Wu, Yibo Yang, Zhouchen Lin, and Hong Liu. Expectation-maximization attention networks for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[May76] Robert M. May. Simple mathematical models with very complicated dynamics. *Nature*, 261(5560):459–467, Jun 1976.

[MBP+20] Shervin Minaee, Yuri Boykov, Fatih Porikli, Antonio Plaza, Nasser Kehtarnavaz, and Demetri Terzopoulos. Image segmentation using deep learning: A survey, 2020.

[MBZ19] Lichao Mou, Lorenzo Bruzzone, and Xiao Xiang Zhu. Learning spectral-spatial-temporal features via a recurrent convolutional neural network for change detection in multispectral imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 57(2):924–935, 2019.

[McC86] Robert K McConnell. Method of and apparatus for pattern recognition, 1986. US Patent 4,567,610.

[MdM+19] D. B. Mesquita, R. F. dos Santos, D. G. Macharet, M. F. M. Campos, and E. R. Nascimento. Fully convolutional siamese autoencoder for change detection in uav aerial images. *IEEE Geoscience and Remote Sensing Letters*, pages 1–5, 2019.

[MGB+92] N Matic, I Guyon, L Bottou, J Denker, and V Vapnik. Computer aided cleaning of large databases for character recognition. In *International Conference on Pattern Recognition*, pages 330–333. IEEE, 1992.

[MH10] Volodymyr Mnih and Geoffrey E Hinton. Learning to detect roads in high-resolution aerial images. In *European Conference on Computer Vision*, pages 210–223, 2010.

[Mit97] Tom Mitchell. *Machine Learning*. McGraw Hill, 1997.

[MJ89] Henry J Moore and Bruce M Jakosky. Viking landing sites, remote-sensing observations, and physical properties of martian surface materials. *Icarus*, 81(1):164–184, 1989.

[MLX+17] Xudong Mao, Qing Li, Haoran Xie, Raymond Y.K. Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[MMMT18] Luca Maggiolo, Diego Marcos, Gabriele Moser, and Devis Tuia. Improving maps from cnns trained with sparse, scribbled ground truths using fully connected crfs. In *International Geoscience and Remote Sensing Symposium*, pages 2103–2103. IEEE, 2018.

[MO14] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets, 2014.

[MTCA17] Emmanuel Maggiori, Yuliya Tarabalka, Guillaume Charpiat, and Pierre Alliez. High-resolution image classification with convolutional networks. In *International Geoscience and Remote Sensing Symposium*, pages 5157–5160. IEEE, 2017.

[Mur12] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT Press, 2012.

[NDRT13] Nagarajan Natarajan, Inderjit S Dhillon, Pradeep K Ravikumar, and Ambuj Tewari. Learning with noisy labels. In *Advances in neural information processing systems*, pages 1196–1204, 2013.

[New19] Isaac Newton. Isaac newton letter to robert hooke 1675. *HSP Discover.* *https://discover.hsp.org/Record/dc-9792/Description*, 2019.

[Nie07] A. A. Nielsen. The regularized iteratively reweighted mad method for change detection in multi- and hyperspectral data. *IEEE Transactions on Image Processing*, 16(2):463–478, 2007.

[NWC+11] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011.

[Opp99] Alan V Oppenheim. *Discrete-time signal processing*. Pearson Education India, 1999.

[Pap66] Seymour A Papert. The summer vision project, 1966.

[PBF18] C. Paris, L. Bruzzone, and D. Fernández-Prieto. A novel method based on source domain understanding and modeling to transfer labels from land-cover vector maps to classifiers for multispectral images. In *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium*, pages 3619–3622, 2018.

[PGC+17] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017.

[PM90] Pietro Perona and Jitendra Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on pattern analysis and machine intelligence*, 12(7):629–639, 1990.

[PSA+04] Georg Petschnigg, Richard Szeliski, Maneesh Agrawala, Michael Cohen, Hugues Hoppe, and Kentaro Toyama. Digital photography with flash and no-flash image pairs. *ACM Transactions on Graphics*, 23(3):664–672, 2004.

[PUK+17] Xingchao Peng, Ben Usman, Neela Kaushik, Judy Hoffman, Dequan Wang, and Kate Saenko. Visda: The visual domain adaptation challenge, 2017.

[PVV+19] M. Papadomanolaki, S. Verma, M. Vakalopoulou, S. Gupta, and K. Karantzalos. Detecting urban changes with recurrent neural networks from multitemporal sentinel-2 data. In *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*, pages 214–217, 2019.

[R+13] Lucio Russo et al. *The forgotten revolution: how science was born in 300 BC and why it had to be reborn*. Springer Science & Business Media, 2013.

[RFB15] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[RHW86] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, Oct 1986.

[RI03] Paul L Rosin and Efstathios Ioannidis. Evaluation of global image thresholding for change detection. *Pattern Recognition Letters*, 24(14):2345–2356, 2003.

[RMH+19] Ievgen Redko, Emilie Morvant, Amaury Habrard, Marc Sebban, and Younès Bennani. *Advances in Domain Adaptation Theory*. Elsevier, 2019.

[Rob63] Lawrence Gilman Roberts. *Machine perception of three-dimensional solids*. PhD thesis, Massachusetts Institute of Technology, 1963.

[Ros58] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

[Ros60] F. Rosenblatt. Perceptron simulation experiments. *Proceedings of the IRE*, 48(3):301–309, 1960.

[RPB15] O. Ronneberger, P.Fischer, and T. Brox. U-Net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, volume 9351 of *LNCS*, pages 234–241. Springer, 2015. (available on arXiv:1505.04597 [cs.CV]).

[RRKB11] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. Orb: An efficient alternative to sift or surf. In *2011 International Conference on Computer Vision*, pages 2564–2571, 2011.

[RSSD18] S. Roy, E. Sangineto, N. Sebe, and B. Demir. Semantic-fusion gans for semi-supervised satellite image classification. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 684–688, 2018.

[Rud16] Sebastian Ruder. An overview of gradient descent optimization algorithms, 2016.

[RVBS17] David Rolnick, Andreas Veit, Serge J. Belongie, and Nir Shavit. Deep learning is robust to massive label noise. *CoRR*, abs/1705.10694, 2017.

[RVRK16] Stephan R. Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *European Conference on Computer Vision (ECCV)*, volume 9906 of *LNCS*, pages 102–118. Springer International Publishing, 2016.

[SBB18] S. Saha, F. Bovolo, and L. Brurzone. Unsupervised multiple-change detection in vhr optical images using deep features. In *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium*, pages 1902–1905, 2018.

[SBB19a] S. Saha, F. Bovolo, and L. Bruzzone. Unsupervised deep change vector analysis for multiple-change detection in vhr images. *IEEE Transactions on Geoscience and Remote Sensing*, 57(6):3677–3693, 2019.

[SBB19b] S. Saha, F. Bovolo, and L. Bruzzone. Unsupervised multiple-change detection in vhr multisensor images via deep-learning based adaptation. In *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*, pages 5033–5036, 2019.

[SBB20] S. Saha, F. Bovolo, and L. Bruzzone. Building change detection in vhr sar images via unsupervised deep transcoding. *IEEE Transactions on Geoscience and Remote Sensing*, pages 1–13, 2020.

[Sch92] J. Schmidhuber. Learning complex, extended sequences using the principle of history compression. *Neural Computation*, 4(2):234–242, 1992.

[Sch15] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.

[Sch19] Juergen Schmidhuber. Generative adversarial networks are special cases of artificial curiosity (1990) and also closely related to predictability minimization (1991), 2019.

[SDVG18] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Semantic foggy scene understanding with synthetic data. *International Journal of Computer Vision*, 126(9):973–992, Sep 2018.

[SGFT08] Steven E Sesnie, Paul E Gessler, Bryan Finegan, and Sirpa Thessler. Integrating landsat tm and srtm-dem derived variables with decision trees for habitat classification and change detection in complex neotropical environments. *Remote Sensing of Environment*, 112(5):2145–2159, 2008.

[SGSC15] Simon Stent, Riccardo Gherardi, Björn Stenger, and Roberto Cipolla. Detecting change for multi-view, long-term surface inspection. In *BMVC*, pages 127–1, 2015.

[SGZ+16] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, Xi Chen, and Xi Chen. Improved techniques for training gans. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 2234–2242. Curran Associates, Inc., 2016.

[SHM+16] David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489, Jan 2016.

[Sin89] Ashbindu Singh. Review article digital change detection techniques using remotely-sensed data. *International journal of remote sensing*, 10(6):989–1003, 1989.

[Sør48] Thorvald Sørensen. A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on danish commons. *Biol. Skr.*, 5:1–34, 1948.

[SS16] Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *ECCV 2016 Workshops*, 2016.

[SSBB19] S. Saha, Y. T. Solano-Correa, F. Bovolo, and L. Bruzzone. Unsupervised deep learning based change detection in sentinel-2 images. In *2019 10th International Workshop on the Analysis of Multitemporal Remote Sensing Images (MultiTemp)*, pages 1–4, 2019.

[SSBB20] S. Saha, Y. T. Solano-Correa, F. Bovolo, and L. Bruzzone. Unsupervised deep transfer learning-based change detection for hr multispectral images. *IEEE Geoscience and Remote Sensing Letters*, pages 1–5, 2020.

[SSM+16] Teppei Suzuki, Soma Shirakabe, Yudai Miyashita, Akio Nakamura, Yutaka Satoh, and Hirokatsu Kataoka. Semantic change detection with hypermaps, 2016.

[SSW18] Ken Sakurada, Mikiya Shibuya, and Weimin Wang. Weakly supervised silhouette-based semantic scene change detection, 2018.

[STDE19] Yu Sun, Eric Tzeng, Trevor Darrell, and Alexei A. Efros. Unsupervised domain adaptation through self-supervision, 2019.

[SWY+15] C. Szegedy, Wei Liu, Yangqing Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2015.

[SZ09] J. Sivic and A. Zisserman. Efficient visual search of videos cast as text retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(4):591–606, 2009.

[SZ15] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

[Sze10] Richard Szeliski. *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.

[TCT+15] Andrew P. Tewkesbury, Alexis J. Comber, Nicholas J. Tate, Alistair Lamb, and Peter F. Fisher. A critical synthesis of remotely sensed optical image change detection techniques. *Remote Sensing of Environment*, 160:1 – 14, 2015.

[THS+18] Yi-Hsuan Tsai, Wei-Chih Hung, Samuel Schulter, Kihyuk Sohn, Ming-Hsuan Yang, and Manmohan Chandraker. Learning to adapt structured output space for semantic segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[THSD17] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[TP91] M. A. Turk and A. P. Pentland. Face recognition using eigenfaces. In *Proceedings. 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 586–591, 1991.

[TPW17] Yaniv Taigman, Adam Polyak, and Lior Wolf. Unsupervised cross-domain image generation. In *International Conference on Learning Representations (ICLR)*, 2017.

[UVL16] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization, 2016.

[UVL17] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[vEH20] Jesper E. van Engelen and Holger H. Hoos. A survey on semi-supervised learning. *Machine Learning*, 109(2):373–440, Feb 2020.

[VJ04] Paul Viola and Michael J. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154, May 2004.

[VJB+19] Tuan-Hung Vu, Himalaya Jain, Maxime Bucher, Matthieu Cord, and Patrick Perez. Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[VKKP15] Maria Vakalopoulou, Konstantinos Karantzalos, Nikos Komodakis, and Nikos Paragios. Simultaneous registration and change detection in multitemporal, very high resolution remote sensing data. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 61–69, 2015.

[VS91] L. Vincent and P. Soille. Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6):583–598, 1991.

[VSP+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc., 2017.

[VT17] Michele Volpi and Devis Tuia. Dense semantic labeling of subdecimeter resolution images with convolutional neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, 55(2):881–893, 2017.

[VTB+13] Michele Volpi, Devis Tuia, Francesca Bovolo, Mikhail Kanevski, and Lorenzo Bruzzone. Supervised change detection in vhr images using contextual information and support vector machines. *International Journal of Applied Earth Observation and Geoinformation*, 20:77–85, 2013.

[VTK+09] Michele Volpi, D Tuia, M Kanevski, Francesca Bovolo, and L Bruzzone. Supervised change detection in vhr images: a comparative analysis. In *2009 IEEE International Workshop on Machine Learning for Signal Processing*, pages 1–6. IEEE, 2009.

[WH18] Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[WLPS18] Wahyu Wiratama, Jongseok Lee, Sang-Eun Park, and Donggyu Sim. Dual-dense convolution network for change detection of high-resolution panchromatic imagery. *Applied Sciences*, 8(10):1785, Oct 2018.

[WZL+19] M. Wu, C. Zhang, J. Liu, L. Zhou, and X. Li. Towards accurate high resolution satellite image semantic segmentation. *IEEE Access*, 7:55609–55619, 2019.

[XLL+19] Joseph Z. Xu, Wenhan Lu, Zebo Li, Pranav Khaitan, and Valeriya Zaytseva. Building damage detection in satellite imagery using convolutional neural networks, 2019.

[XXY+15] Tong Xiao, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang. Learning from massive noisy labeled data for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2691–2699, 2015.

[YCZ20] Yanan You, Jingyi Cao, and Wenli Zhou. A survey of change detection methods based on remote sensing images for multi-source and multi-objective scenarios. *Remote Sensing*, 12(15):2460, Jul 2020.

[YJL+19] M. Yang, L. Jiao, F. Liu, B. Hou, and S. Yang. Transferred deep learning-based change detection in remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 57(9):6960–6973, 2019.

[YK16] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. In *International Conference on Learning Representations (ICLR)*, May 2016.

[YKF17] Fisher Yu, Vladlen Koltun, and Thomas Funkhouser. Dilated residual networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[YPN+20] Xu Yao, Gilles Puy, Alasdair Newson, Yann Gousseau, and Pierre Hellier. High resolution face age editing, 2020.

[ZFY+17] Yang Zhan, Kun Fu, Menglong Yan, Xian Sun, Hongqi Wang, and Xiaosong Qiu. Change detection based on deep siamese convolutional network for optical aerial images. *IEEE Geoscience and Remote Sensing Letters*, 14(10):1845–1849, 2017.

[ZGLJ14] Jiaojiao Zhao, Maoguo Gong, Jia Liu, and Licheng Jiao. Deep learning to classify difference image for image change detection. In *International Joint Conference on Neural Networks*, pages 411–417. IEEE, 2014.

[ZGZ+19] P. Zhang, M. Gong, H. Zhang, J. Liu, and Y. Ban. Unsupervised difference representation learning for detecting multiple types of changes in multitemporal remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 57(4):2277–2289, 2019.

[Zhu05] Xiaojin Jerry Zhu. Semi-supervised learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2005.

[ZK15a] Sergey Zagoruyko and Nikos Komodakis. Learning to compare image patches via convolutional neural networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[ZK15b] Sergey Zagoruyko and Nikos Komodakis. Learning to compare image patches via convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4353–4361, 2015.

[ZKL+16] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[ZPIE17] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[ZSS+18] Amir R. Zamir, Alexander Sax, William Shen, Leonidas J. Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[ZTK+18] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. Residual dense network for image super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[ZTM+17] X. X. Zhu, D. Tuia, L. Mou, G. Xia, L. Zhang, F. Xu, and F. Fraundorfer. Deep learning in remote sensing: A comprehensive review and list of resources. *IEEE Geoscience and Remote Sensing Magazine*, 5(4):8–36, 2017.

[ZWZ+19] Mingmin Zhen, Jinglu Wang, Lei Zhou, Tian Fang, and Long Quan. Learning fully dense neural networks for image semantic segmentation, 2019.

[ZY17] Yu Zhang and Qiang Yang. A survey on multi-task learning, 2017.

[ZYT+18] Y. Zhuang, F. Yang, L. Tao, C. Ma, Z. Zhang, Y. Li, H. Jia, X. Xie, and W. Gao. Dense relation network: Learning consistent and context-aware representation for semantic image segmentation. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 3698–3702, 2018.

ECOLE
DOCTORALE

Titre: Réseaux de Neurones Convolutifs pour l'Analyse de Changements en Imagerie de Télédétection avec des Annotations Bruitées et des Décalages de Domaine

Mots clés: Télédétection, détection de changements, réseaux de neurones convolutifs, apprentissage multitâche, apprentissage faiblement supervisé, adaptation de domaine.

Résumé: L'analyse de l'imagerie satellitaire et aérienne d'observation de la Terre nous permet d'obtenir des informations précises sur de vastes zones. Une analyse multitemporelle de telles images est nécessaire pour comprendre l'évolution de ces zones. Dans cette thèse, les réseaux de neurones convolutifs sont utilisés pour détecter et comprendre les changements en utilisant des images de télédétection provenant de diverses sources de manière supervisée et faiblement supervisée. Des architectures siamoises sont utilisées pour comparer des paires d'images recalées et identifier les pixels correspondant à des changements. La méthode proposée est ensuite étendue à une architecture de réseau multitâche qui est utilisée pour détecter les changements et effectuer une cartographie automatique simultanément, ce qui permet une compréhension sémantique des changements détectés. Ensuite, un filtrage de classification et un nouvel algorithme de diffusion anisotrope guidée sont utilisés pour réduire l'effet du bruit d'annotation, un défaut récurrent pour les ensembles de données à grande échelle générés automatiquement. Un apprentissage faiblement supervisé est également réalisé pour effectuer une détection de changement au niveau des pixels en utilisant uniquement une supervision au niveau de l'image grâce à l'utilisation de cartes d'activation de classe et d'une nouvelle couche d'attention spatiale. Enfin, une méthode d'adaptation de domaine fondée sur un entraînement adverse est proposée. Cette méthode permet de projeter des images de différents domaines dans un espace latent commun où une tâche donnée peut être effectuée. Cette méthode est testée non seulement pour l'adaptation de domaine pour la détection de changement, mais aussi pour la classification d'images et la segmentation sémantique, ce qui prouve sa polyvalence.

Title: Convolutional Neural Networks for Change Analysis in Earth Observation Images with Noisy Labels and Domain Shifts

Keywords: Remote sensing, change detection, convolutional neural networks, multitask learning, weakly supervised learning, domain adaptation.

Abstract: The analysis of satellite and aerial Earth observation images allows us to obtain precise information over large areas. A multitemporal analysis of such images is necessary to understand the evolution of such areas. In this thesis, convolutional neural networks are used to detect and understand changes using remote sensing images from various sources in supervised and weakly supervised settings. Siamese architectures are used to compare coregistered image pairs and to identify changed pixels. The proposed method is then extended into a multitask network architecture that is used to detect changes and perform land cover mapping simultaneously, which permits a semantic understanding of the detected changes. Then, classification filtering and a novel guided anisotropic diffusion algorithm are used to reduce the effect of biased label noise, which is a concern for automatically generated large-scale datasets. Weakly supervised learning is also achieved to perform pixel-level change detection using only image-level supervision through the usage of class activation maps and a novel spatial attention layer. Finally, a domain adaptation method based on adversarial training is proposed, which succeeds in projecting images from different domains into a common latent space where a given task can be performed. This method is tested not only for domain adaptation for change detection, but also for image classification and semantic segmentation, which proves its versatility.

Institut Polytechnique de Paris
91120 Palaiseau, France